Computers, Environment and Urban Systems 48 (2014) 153–165

Contents lists available at ScienceDirect

Computers, Environment and Urban Systems

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

Review

A critical review of real-time map-matching algorithms: Current issues and future directions

Mahdi Hashemi *, Hassan A. Karimi

Geoinformatics Laboratory, School of Information Sciences, University of Pittsburgh, 135 North Bellefield Avenue, Pittsburgh, PA 15260, USA

ARTICLE INFO

Article history:
Received 30 January 2014
Received in revised form 3 May 2014
Accepted 29 July 2014
Available online 24 August 2014

Keywords:
Navigation
Map-matching algorithm
Road network
Positioning system

ABSTRACT

Current navigation systems/services allow drivers to keep track of their precise whereabouts and provide optimal routes to reach specified locations. A reliable map-matching algorithm is an indispensable and integral part of any land-based navigation system/service. This paper reviews existing map-matching algorithms with the aim of highlighting their qualities as well as unfolding their unresolved issues as a means to provide directions for future studies in this field. Existing map-matching algorithms are compared and contrasted with respect to positioning sensors, map qualities, assumptions and accuracies. The results of these comparisons provide interesting insights into the workings of existing algorithms and the issues they must address for improving their performance. Example findings are: (a) not all map-matching algorithms pay sufficient attention to topology of networks, directionality of roads or turn-restrictions; (b) most map-matching algorithms make an unbalanced trade-off between performance and accuracy; and (c) weight-based map-matching algorithms balance simplicity and accuracy and advanced map-matching algorithms provide high accuracy but with low performance. Based on the findings, suggestions are made to improve existing algorithms.

© 2014 Elsevier Ltd. All rights reserved.

Contents

1. Introduction . . . 154
2. Overview of map-matching algorithms . . . 155
   2.1. Simple map-matching algorithms . . . 155
   2.2. Weight-based map-matching algorithms . . . 158
   2.3. Advanced map-matching algorithms . . . 158
   2.4. Using multi sensors in map-matching algorithms . . . 161
3. Challenges and future directions . . . 161
   3.1. Candidate segments selection for first position . . . 161
   3.2. Candidate segments selection for next positions . . . 162
   3.3. Detecting intersection crossing . . . 162
   3.4. Best segment identification for first GPS point . . . 163
   3.5. Locating GPS position on identified segment . . . 163
   3.6. Confidence level . . . 163
4. Summary . . . 164
   References . . . 164

* Corresponding author. Tel.: +1 412 624 8858; fax: +1 412 624 2788.
E-mail address: m.hashemi1987@gmail.com (M. Hashemi).

http://dx.doi.org/10.1016/j.compenvurbsys.2014.07.009
0198-9715/© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Today, navigation systems/services are available in automobiles, cell-phones and other mobile devices. Coordinates of the device, obtained through positioning sensors, are used along with the road network database of the area to estimate user's location on a road segment (Zhang, Wang, & Wan, 2003). The process of continually estimating a user's position on a road segment is known as map matching (Greenfeld, 2002; Karimi, Conahan, & Roongpiboonsopit, 2006; Quddus, 2006). Matching position obtained through GPS, or other positioning sensors, on road segments can be performed either in real-time mode (Velaga, Quddus, & Bristow, 2009; Li, Quddus, & Zhao, 2013; Quddus, Noland, & Ochieng, 2006; Syed & Cannon, 2004) or post-processing mode (Yuan et al., 2010a; Ebendt, Sohr, Tcheumadjeu, & Wagner, 2010; Kuehne et al., 2003; Miwa, Kiuchi, Yamamoto, & Morikawa, 2012). Real-time map matching must address challenges such as finding the road segment on which the user is traveling and snapping or projecting updated GPS points on that segment in real time (White, Bernstein, & Kornhauser, 2000). In real-time mode, all GPS points up to the current one can be used to match the current GPS point. In contrast, in post-processing mode, map matching is performed after all or a large part of GPS points are collected (Bierlaire, Chen, & Newman, 2013; Lou et al., 2009; Miwa et al., 2012; Rahmani & Koutsopoulos, 2013; Yang, Kang, & Chon, 2005). This necessitates the access to all GPS points in post-processing mode in order to find the best matching path in the road network for the entire trip (Marchal, Hackney, & Axhausen, 2005). In other words, the problem in post-processing map matching is finding a path on a road network which closely matches a raw GPS trajectory (Lou et al., 2009). Some studies refer to post-processing map-matching algorithms as global algorithms (Brakatsoulas, Pfoser, Salas, & Wenk, 2005; Lou et al., 2009; Rahmani & Koutsopoulos, 2013; Yuan, Zheng, Zhang, Xie, & Sun, 2010b). Furthermore, incremental or local map-matching algorithms, which do not require all GPS points but a considerable part of them to match a path (Yuan et al., 2010b), are also categorized under post-processing algorithms. In general, real-time algorithms provide position estimates faster but post-processing algorithms are more accurate (Brakatsoulas et al., 2005). While continuity on the path during the entire trip is not guaranteed in real-time map matching, it is a condition for post-processing map matching (Brakatsoulas et al., 2005). Table 1 summarizes the main differences between real-time and post-processing map-matching algorithms.

Post-processing map matching is for applications where knowledge on the actual path for the entire trip is more important than on the instantaneous position of the user and where large data processing and intensive computation are allowed (Bierlaire & Frejinger, 2008; Rahmani & Koutsopoulos, 2013). Example applications of post-processing map matching are mining historical trajectories of a large number of experienced taxi drivers to find shortest routes between different origin–destination pairs at different times of day (Yuan et al., 2010a) and estimating travel time (Ebendt et al., 2010) or traffic (Kuehne et al., 2003; Li, Zhang, & Yu, 2011; Miwa et al., 2012) along a road segment by crowd-sourcing. Considering that post-processing map-matching algorithms potentially have to deal with large amounts of data and computationally intensive tasks, two approaches are taken to mitigate these issues. One approach is decreasing the number of points for post-processing by increasing the polling time interval (Lou et al., 2009; Miwa et al., 2012; Rahmani & Koutsopoulos, 2013; Yang et al., 2005) and another approach is recording only the position and timestamp for each point ignoring other data items such as heading and speed (Lou et al., 2009; Miwa et al., 2012; Rahmani & Koutsopoulos, 2013; Yang et al., 2005). When the time interval between two consecutive positions is long, it cannot be guaranteed that they both belong to the same road segment. If the time interval is too long, the two road segments may not even be connected, which may mean that the user has passed a few road segments between two position estimates. Due to all these possibilities, finding the correct path between consecutive GPS positions with long time intervals is a challenge in post-processing map-matching algorithms. While real-time map-matching algorithms access high frequency position data with many recorded data items such as position, speed and heading, post-processing map-matching algorithms must map match a massive amount of low frequency position data with only the latitude and longitude coordinates recorded. Consequently, real-time map-matching algorithms cannot directly be used in place of post-processing map-matching algorithms as they have to resolve different challenges (Chen et al., 2014).

Bierlaire et al. (2013) developed a post-processing map-matching algorithm for GPS-enabled smartphones only for driving. Position, speed and heading data from GPS are used to assign a probability to each candidate path based on horizontal accuracy of GPS data and the road network. Inaccurate and low-frequency positioning data are two challenges in their study. Miwa et al. (2012), Rahmani and Koutsopoulos (2013), Yang et al. (2005) and Lou et al. (2009) developed a similar algorithm specifically for vehicles with lower positioning data frequency and only position coordinates and timestamp. Sparse positioning data, with gaps up to 1.5, 3, 5 and 5 min with latitude, longitude and timestamp, were used. Chen et al. (2014) proposed a map-matching algorithm for large-scale low-frequency floating car data. They used a multi-criteria dynamic programming technique to minimize the number of candidate routes for each GPS point. Lou et al. (2009) considered two important points in their post-processing algorithm: (a) true paths tend to be straight, rather than roundabout and (b) true paths tend to follow posted speed limits on roads. Yang et al. (2005) proposed to project GPS points within 20 m from intersection nodes onto the intersection node itself, postponing resolution of the problem to the next point. Yuan et al., 2010b used Tobler's first law of geography that "everything is related to everything else, but near things are more related than distant things" to develop an interesting approach of assigning a segment to a GPS point in their post-processing map-matching algorithm. First, all segments within a constant distance from the GPS point are

Table 1
Comparing real-time and post-processing map-matching algorithms.

Criteria	Real-time	Post-processing
Problem	Assigning a road segment to current GPS point	Finding the real path through all points
Purpose	Navigation and real-time applications	Mapping large-scale floating cars data, fleet monitoring, traffic surveillance
Required data	Position, timestamp, heading, speed, accuracy	Position and timestamp
Next points	Not needed	Needed
Polling time interval	1–10 s	1–5 min
Result	Fast	Accurate
Continuous tracking	Not guaranteed	A condition

selected as candidate segments. Then, other nearby GPS points vote for candidate segments where closer points have heavier votes. Krumm, Letchner, and Horvitz (2007) scored candidate map-matched locations based on a time constraint instead of shortest path between them. Time constraint means that the sequence of matched positions must be reachable from each other in the time intervals computed from the measurements' time stamps. The rest of this paper investigates real-time map matching.

A real-time map-matching algorithm has three steps: (1) matching preliminary points until a predefined condition is satisfied, (2) matching points far from intersections on the same segment as (1), and (3) matching points around intersections on another segment (Greenfeld, 2002; Quddus, 2006; White et al., 2000). An algorithm switches between steps 2 and 3 based on whether the user is driving on the same segment or is crossing an intersection either going straight or by making a turn. However, some map-matching algorithms combine the last two steps, resulting in performance degradation and increased complexity (Kim & Kim, 2001; Syed & Cannon, 2004). Ren and Karimi (2009a, 2009b, 2012a, 2012b) categorized map-matching algorithms into geometrical, topological and advanced groups. Geometrical group includes point-to-point, point-to-curve and curve-to-curve map-matching algorithms (Karimi et al., 2006; Quddus, Ochieng, & Noland, 2007; White et al., 2000). Jagadeesh, Srikanthan, and Zhang (2004) discussed the pros and cons of geometrical algorithms which in brief reveals their excessive simplicity and failure in complex urban road networks. Topological algorithms compare the geometrical and topological characteristics of the road network with the geometrical and topological characteristics of the vehicle's trajectory (Chen, Yu, Li, & Chen, 2003; Greenfeld, 2002; Joshi, 2001; Quddus, Ochieng, Zhao, & Noland, 2003; White et al., 2000; Yu, 2006). Advanced map-matching algorithms apply a special decision-making model such as fuzzy logic (Zhihua & Wu, 2005; Jagadeesh et al., 2004; Quddus, Noland et al., 2006; Syed & Cannon, 2004). Quddus et al. (2007) introduced probabilistic map-matching algorithms as the fourth group (Ochieng, Quddus, & Noland, 2004; Zhao, 1997). This type of algorithms first derive an error region from the error variances associated with positioning data and road network. The error region is then overlaid on the road network to identify candidate road segments. Next, some geometrical and topological criteria are used to find the correct segment (Zhao, 1997). Although, probabilistic algorithms can recover from mis-matches quickly, they are computationally intensive. To improve the performance of probabilistic algorithms, Ochieng et al. (2004) proposed constructing error regions only when the vehicle approaches an intersection. An in-depth analysis of the algorithms in all four groups does not uncover clear boundaries between them, for example a map-matching algorithm could be both advanced and probabilistic.

However, this traditional categorization of map-matching algorithms is not suitable to differentiate among recent map-matching algorithms because most of them use both geometrical and topological information. Therefore, a new categorization of map-matching algorithms is used in Section 2; the new categories are "simple", "weight-based", and "advanced".

In this paper,

- "direction of GPS points" refers to direction of the line connecting two consecutive GPS points;
- "GPS point heading" refers to heading information of a GPS point;
- "direction difference" is the difference between direction of GPS points and direction of a road segment;
- "heading difference" is the difference between heading of a GPS point and direction of a road segment; and
- "GPS trajectory" refers to a sequence of time-stamped GPS points with the time interval between consecutive GPS points not exceeding a certain threshold (Lou et al., 2009).

The rest of this paper is organized as follows. Section 2 provides a critical overview and comparison of existing algorithms. Section 3 discusses different aspects of a map-matching algorithm in details. Section 4 summarizes the paper and highlights future directions.

2. Overview of map-matching algorithms

The last survey reported in literature on map-matching algorithms was in 2007 (Quddus et al., 2007). Since then, some of the issues reported in that article have been addressed. The purpose of the survey presented in this work is to fill the gap since the last survey by comparing map-matching algorithms in terms of accuracy and simplicity and to unfold some unresolved issues in this field. Table 2 summarizes map-matching algorithms with positioning technology, map scale, method and criteria used for correct segment identification and accuracy.

2.1. Simple map-matching algorithms

The earliest map-matching algorithm for car navigation reported in literature was developed by Kim, Lee, Kang, Lee, and Kim (1996). It was a simple geometrical algorithm based only on the shape of road segment. Although fast and simple, it was not accurate at intersections and parallel roads.

White et al. (2000) elaborated the map-matching problem, its geometrical foundations and explained the concepts, advantages and disadvantages of two simple geometrical map-matching algorithms (Algorithms 1 and 2) and two simple topological map-matching algorithms (Algorithms 3 and 4). Algorithm 1 finds nodes close to the GPS point and finds connected arcs to these nodes. It then projects the point onto the closest arc. Algorithm 2 is identical to Algorithm 1 except that it makes use of direction difference in addition to closeness to find the best segment. Algorithm 3 is a variant of Algorithm 2 that uses topological information to disregard unconnected arcs to previous arc. Algorithm 4 uses curve-to-curve matching and finds candidate nodes based on the same technique as in Algorithm 1. Given a candidate node, it constructs piece–wise linear curves from the set of paths that originate from that node. Then, it constructs a piece–wise linear curve using the GPS points and calculates the distance between this curve and corresponding curves in the network. Finally, it selects the closest path and projects the GPS points onto that path. All these algorithms are conventional and new algorithms use different strategies to cope with first GPS point, points around intersections and points far from intersections.

Some map-matching algorithms were developed prior to May 1, 2000 when the selective availability (S/A) of GPS was still on. Two of these algorithms are discussed by Taylor, Blewitt, Steup, Corbett, and Car (2001) and Kim and Kim (2001). The algorithm by Kim and Kim (2001) is discussed in the section on advanced algorithms. Taylor et al. (2001) developed a geometrical curve-to-curve map-matching algorithm. The novelty of this work was a least-squares model to estimate the Euclidean position of the vehicle. The least-squares estimation enables the use of height information provided by digital elevation model (DEM) which not only augments the solution but also reduces the minimum number of satellites required for a position solution. Furthermore, it enables the use of DGPS, i.e., the corrections imposed on previous GPS points during map matching are used to calculate pseudorange error of each satellite; pseudorange errors calculated at previous step are used to modify pseudoranges of those satellites at next GPS

Table 2
Comparison between real-time map-matching algorithms.

Algorithm	Map scale and place	Criteria for segment identification	Segment identification model	Positioning sensors	Network type	Correct segment identification (%)	Horizontal error (m)	Along-track error (m)
White et al. (2000)	Mercer County, New Jersey	Closeness Trajectory similarity Segment connectivity	Curve-to-curve matching	GPS	Vehicle navigation in urban roads	70.03	~32 (95%)	~29.5 (95%)
White et al. (2000)	Mercer County, New Jersey	Closeness Direction difference Segment connectivity	Uses direction difference to eliminate some candidate segments and uses only closeness to select one of them	GPS	Vehicle navigation in urban roads	72.25	~32 (95%)	~29.5 (95%)
Taylor et al. (2001)	Newcastle	Closeness Trajectory similarity	Curve-to-curve matching	GPS (S/A on) DEM	Vehicle navigation in suburban roads	99	14.18 (aver.)	-0.8 (aver.)
Srinivasan et al. (2003)	Singapore	Closeness Heading difference Turn-restriction	Topological point-to-curve matching	GPS DR	Vehicle navigation in suburban roads	80.2	21.2 (95%)	–
Greenfeld (2002)	New York	Closeness Direction difference Intersecting angle between line connecting GPS points and segment Segment connectivity	Weighted method	GPS	Bus navigation in urban roads	85.6	18.3 (95%)	15.5 (95%)
Quddus et al. (2003)	1:2500 London	Closeness Heading difference Position of point relative to segment Segment connectivity	Weighted method	GPS DR	Vehicle navigation in suburban roads Vehicle navigation in urban roads Vehicle navigation in suburban roads Vehicle navigation in urban roads	88.6 80.1 87.5 75.6	18.5 (95%) – 20 (95%) –	17.6 (95%) – 19.1 (95%) –
Velaga et al. (2009)	1:2500 London 1:2500 London and Washington D.C.	Closeness Heading difference Turn-restrictions Segment connectivity	Weighted method	GPS DR	Vehicle navigation in suburban roads Vehicle navigation in urban roads	96.71 96.36	9.81 (95%) –	7.36 (95%) –
Li et al. (2013)	Nottingham London	Closeness Heading difference Segment connectivity	Weighted method	GPS DR DEM	Vehicle navigation in suburban roads Vehicle navigation in urban roads	99.8 97.7	2.1 (95%) 11.5 (95%)	– –
Ren and Karimi (2009a)	1:2500 Pittsburgh	Closeness Direction difference Segment connectivity	Weighted method	GPS	Wheelchair navigation on sidewalk network	92.86	–	–
Pyo et al. (2001)	Taejon	Closeness Heading difference Segment connectivity	Multiple hypothesis technique	GPS DR	Vehicle navigation in urban roads	89	–	–
Yang et al.(2003)	Beijing	Closeness	Dempster-Shafer theory	GPS	Vehicle navigation in urban roads	96	–	–
Ren and Karimi (2009b)	1:2500 Pittsburgh	Closeness Segment connectivity	Hidden Markov Model	GPS	Wheelchair navigation on sidewalk network	93.70	–	–

Ochieng et al. (2004)	1:2500 London	Closeness Heading difference Segment connectivity	Not discussed	GPS DR	Vehicle navigation in suburban roads	98.1	9 (95%)	8.2 (95%)
					Vehicle navigation in urban roads	97.1	–	–
				GPS	Vehicle navigation in suburban roads	97.8	9.2 (95%)	8.1 (95%)
					Vehicle navigation in urban roads	90.2	–	–
Ren and Karimi (2009a)	1:2500 Pittsburgh	Closeness Direction difference Segment connectivity	Radial basis functional neural network	GPS	Wheelchair navigation on sidewalk network	92.23	–	–
Ren and Karimi (2012a)	1:2500 Pittsburgh	Closeness Direction difference Segment connectivity	Fuzzy logic	GPS	Wheelchair navigation on sidewalk network	92.03	–	–
Kim and Kim (2001)	Taejon	Closeness Heading difference Segment connectivity	Adaptive fuzzy logic	GPS (S/A on) DR	Vehicle navigation in urban roads	Very good	15 (max)	–
Jagadeesh et al. (2004)	Singapore city	Closeness Heading difference Segment connectivity	Fuzzy logic	Simulated	Vehicle navigation in suburban roads	95.36	–	–
					Vehicle navigation in urban roads	89.57	–	–
Syed and Cannon (2004)	Calgary	Closeness Heading difference	Fuzzy logic	GPS (high sensitivity) Gyroscope	Vehicle navigation in urban roads	92.8	16.1 (95%)	15.1 (95%)
Quddus, Noland et al. (2006)	1:2500 London	Closeness Heading difference Direction difference Segment connectivity	Fuzzy logic	GPS DR	Vehicle navigation in suburban roads	99.2	5.5 (95%)	4.2 (95%)
					Vehicle navigation in urban roads	98.5	–	–
				GPS	Vehicle navigation in suburban roads	99	6.5 (95%)	5.6 (95%)
					Vehicle navigation in urban roads	93.1	–	–

positions. In fact, they attempt to improve the positional accuracy of the GPS point before map matching, instead of using topological and directional information to ease the problem.

Srinivasan, Cheu, and Tan (2003) developed a topological point-to-curve map-matching algorithm. GPS and dead reckoning (DR) outputs are integrated using a Kalman filter. The nearest segment to the GPS point is found. The heading of GPS point must approximately match the direction of the segment and turning from previous segment to this segment must be legal. If any of these conditions are violated, the next nearest segment is chosen as the next candidate. After identifying the correct segment, the GPS point is snapped to the nearest shape-point on that segment. The most notable disadvantage of this algorithm is that it does not consider all criteria (closeness and heading difference). Therefore, if the nearest segment's direction approximately matches the GPS point's heading, it will not consider the second nearest segment which may be just a little farther but with a much more congruent direction. Second, their method of candidate segments selection could be computationally intensive due to calculation of distances between a GPS point and many segments. Other approaches calculate the distances between a GPS point and different nodes, which tends to be faster. Third, if an algorithm wants to consider turn-restrictions and connectivity constraints as unbreakable rules, it would be better to consider them in candidate segments selection process not after it (Srinivasan et al., 2003) as much time and computations can be saved.

In general, the models used to identify the correct segment in all above algorithms are too simple to be used in dense urban networks. Studies such as Taylor et al. (2001) attempted to improve GPS point's accuracy using DEM and DGPS but still their method of identifying the correct segment is straightforward.

2.2. Weight-based map-matching algorithms

Greenfeld (2002) developed a map-matching algorithm consisting of two separate sub-algorithms called "initial-mapping" and "map". The "initial-mapping" is required to locate the first GPS point somewhere on the network. Based on this initial match, the second step called "map" uses three parameters to assign a segment to the GPS point: (a) proximity of GPS point to segment, (b) similarity between the direction of GPS points and each segment and (c) intersecting angle between the line connecting two consecutive GPS points and each segment if an intersection exists. Connectivity between sequential road segments is a constraint in their algorithm. They also, proposed to consider connected segments to candidate segments as additional candidates to cope with the problem of skipping a segment during system/service outage. They do not take advantage of speed and heading information provided by GPS. The second and third criteria for best segment identification are correlated. After any large change in the direction of GPS points, the algorithm waits for the next two GPS points to decide whether it is an outlier or an actual turn, which causes delay in response.

Quddus et al. (2003) used two criteria in a simple weighted method to find the best segment among candidate segments. The two criteria are proximity of the GPS point to each segment and difference between heading of the vehicle (obtained from GPS/DR) and direction of each segment. Connectivity is used to determine candidate segments at intersections. They did not take advantage of movement direction obtained through connecting consecutive GPS points. Directionality of the road network and uncertainty of heading information at low speeds are also ignored.

Velaga et al. (2009) developed a weight-based topological map-matching algorithm. The map-matching algorithm is divided into three steps: (a) identification of the correct segment for the first GPS point, (b) matching on the same segment as long as the vehicle is not approaching an intersection and (c) matching at an intersection when the vehicle approaches an intersection. Four criteria are used in a weighted method to identify the best segment at intersections: (a) heading difference as the cosine of the angle between GPS point's heading and segment's direction, (b) proximity, (c) adding a positive score to the candidate segments to which the vehicle can legally turn and a negative score to others and (d) adding a positive score to the candidate segment which is linearly connected to the previous segment and a negative score to others. The scores of all criteria are normalized in a range of -1 to +1. Matching one GPS point takes only 1/180 s on average which is suitable for real-time applications. Therefore, considering accuracy, simplicity and performance together, this algorithm outstrips all the aforementioned algorithms. Despite its pros, there are some shortcomings. If the recorded speed for the new GPS point is zero, no segment selection is performed and the new GPS point is projected on the same previous segment. However, since the change in position may be due to low GPS accuracy in very low speeds, there is no need to update position when speed is zero. Their last criterion to identify the correct segment is somewhat subjective and not always efficient. Direction difference is not considered as a criterion. There are no precautions for situations where the vehicle has passed a segment between two consecutive GPS points (known as segment skipping) or when the driver makes illegal turns. Off-road conditions are neglected and no confidence level is defined for map-matched locations. Lastly, although two different groups of criteria's weights for urban and suburban areas are developed based on field data, they are constant. Developing some methods to calculate the criteria's weights for each GPS point based on its special circumstances may improve the output.

Li et al. (2013) developed a topological map-matching algorithm. Data from GPS, low-cost DR sensors and DEM are first integrated using an extended Kalman filter. Candidate segments are connected segments to previous segment. Proximity and heading difference are two criteria used in a weighted method to find the correct road segment. The accuracy of map-matching algorithm is enhanced by employing two network features including grade separation to determine whether segments are physically connected, and the direction of traffic flow to identify the start and end nodes of a segment. Their algorithm is among the best ones in terms of performance and accuracy. The high accuracy achieved by the algorithm comes at a cost due to the use of DR and DEM in addition to GPS. However, DEM may be unavailable or not in a suitable scale for other areas. Storing and retrieving DEM from the memory of a stand-alone navigation system or transmitting it from a server to a client may impede the map-matching process considerably. On the other hand, intersection detection is not discussed.

Most map-matching algorithms in this section divide their logic into different parts for identifying the correct segment in different situations: when the movement starts, before crossing an intersection, and after crossing an intersection. This approach is proven to be efficient because it improves the percentage of correct segment identification without making the algorithm complicated or computationally intensive. Closeness is the only common criterion used in all weight-based algorithms to find the correct segment. Each algorithm uses either heading or direction of GPS points as another criterion but not both.

2.3. Advanced map-matching algorithms

Pyo, Shin, and Sung (2001) proposed a multiple-hypothesis technique to assign a road segment to a position fix probabilistically. By overlaying the error ellipse associated with GPS/DR positioning data on the road network, candidate segments are determined, a process which is computationally intensive. Three criteria are used to assign a probability to each candidate: close-

ness, heading difference and connectivity. Connectivity is considered as a flexible parameter which means two consecutive segments do not need to be connected, but the lesser the number of segments between two consecutive segments, the more is the probability assigned to that candidate segment. The number of candidate segments increases by dilution of error ellipse in dense urban areas which increases the number of hypotheses. Although, hypotheses having a probability below a threshold are pruned to reduce the number of hypotheses, still there remain many hypotheses which dramatically increase computation time. Thus, for real-time applications, simpler models are preferred. Another drawback is that they choose candidate segments by the same method at all points irrespective of the fact that the vehicle is on the same segment or at an intersection which remarkably decreases performance of the algorithm. Finally, no methodology for measuring the confidentiality of map-matched locations is defined.

Yang, Cai, and Yuan (2003) developed a simple map-matching algorithm based on Dempster–Shafer theory of evidence. Kalman filtering is used to filter the unreasonable GPS data. Each candidate segment is weighted based on its distance to the GPS point. Their results suggest that the algorithm identifies 96% of the road segments correctly. However, because heading and direction information is completely cast off and connectivity constraint is overlooked, such a success rate seems very optimistic.

Hidden Markov model (HMM) is used as a model which involves a path through many possible states, where some state transitions are more likely than others and where the state measurements are uncertain. States are the individual road segments and transitions between road segments are governed by connectivity of the road network. State measurements are the noisy location measurements by the positioning system. Transition probabilities give the probability of a vehicle moving between candidate road matches. Ren and Karimi (2009b) and Newson and Krumm (2009) discussed the possibility of applying HMM in map-matching algorithms for wheelchair and vehicle navigation, respectively. The only criterion used by Ren and Karimi (2009b) to find the best segment is closeness besides topological constraints which means a wheelchair cannot go to an unconnected segment. Although, they claim that it is a real-time algorithm, due to the nature of HMM, their algorithm requires at least four GPS points after the current GPS point to find the sidewalk segments sequence with maximum probability which could delay the response. Because the time difference between location updates increases dramatically in Newson and Krumm (2009), they also considered the shortness of the path between consecutive GPS points as another criterion to find the best path. Although they consider their algorithm for post-processing, by increasing the location update frequency and decreasing the number of involved GPS points in HMM their algorithm can respond in real time. However, their implementation results are not included in Table 2 because it concerns a post-processing algorithm. None of the two aforementioned algorithms take advantage of heading or direction data.

Ochieng et al. (2004) used augmented GPS with DR as their positioning system. An elliptical confidence region is formed around a GPS point based on error models associated with positioning data. Road segments inside the confidence region are chosen as candidate segments. If the confidence region does not contain any segments, then it is assumed that the vehicle is off-road. If the confidence region contains more than one candidate segments, distance and heading information is used to choose one of them. Their algorithm has two modes after the first segment is identified. The first mode is applied to assign the last identified segment to current GPS point as long as no intersection has been crossed. The second mode is applied to find the best segment based on closeness and similarity in heading among connected segments to previous one when the vehicle is crossing an intersection. The general structure of this algorithm sounds reasonable but there is no explanation as to how these criteria are formulated and integrated to find the most likely segment.

Ren & Karimi (2009a, 2012a) used two criteria to select the best segment among candidate segments: distance from GPS point to each segment and direction difference between GPS trajectory (lines connecting two or more GPS points) and each segment. However, obtaining user's trajectory in their method requires three future GPS position fixes in addition to current GPS position fix which calls into question the instantaneity of real-time map matching. In addition, they do not take advantage of heading information provided by GPS because of the low movement speed of their special application case of wheelchair navigation. The difference between these two algorithms is that the former uses a weight-based method and a radial basis functional neural network and the latter uses a fuzzy inference system to identify a match segment.

Kim and Kim (2001) developed a topological curve-to-curve map-matching algorithm. Closeness, trajectory similarity and connectivity constraint are used to calculate an index called C-measure. The segment with the highest C-measure wins the GPS point. The trajectory similarity is determined by comparing the segment's direction to the vehicle's velocity vector. The extended Kalman filter is used to obtain the vehicle's position from GPS, in-car speedometer and gyroscope. If the C-measure is greater than a threshold, the next GPS points are projected on the same segment as long as the distance to the end node of the segment is greater than a threshold. Finally, the entire procedure is modeled as an adaptive fuzzy inference system. The adaptive fuzzy inference system is trained by convergent learning rule. Their method of intersection detection in GPS trajectory is faint because it only uses C-measure and distance to the end node of the segment. Additionally, their algorithm is complicated because they combine the last two steps of map matching (matching on the same segment as before and matching on another segment around intersections) into one step.

The algorithm developed by Fu, Li, and Wang (2004), first plots a circle around the current GPS point whose radius is equal to the GPS horizontal positioning error. Candidate segments are those which are connected to the last matched road and have their corresponding projection point inside the circle. If there is no candidate segment, the point is assumed off the road network. If there is more than one candidate, the best segment is identified based on closeness and heading difference in a fuzzy system. Their algorithm gives the same importance to closeness and heading difference in decision-making while directional parameters have been proven to be of higher significance (Greenfeld, 2002; Quddus et al., 2003). No statistical implementation results are revealed and the algorithm does not seem to be comprehensive because it only deals with correct segment identification around intersections. They do not even discuss how intersections are detected in GPS trajectories. This algorithm is not concerned about situations such as the first GPS point or when the vehicle is traveling on the same segment.

Jagadeesh et al. (2004) developed a simple map-matching algorithm based on a simple fuzzy inference system with only three rules. Candidate segments for the first GPS point are those in a distance of twice the GPS error from the point itself. Candidate segments around consequent intersections are connected segments to previous one. The likelihood of each road in the candidate roads list being the actual road of travel is calculated based on two factors namely direction difference and closeness. An off-road condition is detected when all roads in the candidate list have very low resemblance values. With slow moving vehicle, previous road is assigned to current GPS point. This will result in a situation wherein the vehicle has just entered a new road at an intersection

but the previously traversed road is identified by the algorithm as the current road. They used an artificial GPS trajectory generated in laboratory to test their algorithm which makes the results (see Table 2) less reliable. Generally, there is no contribution in correct road identification in comparison to the aforementioned algorithms except that they have even neglected some points such as intersection detection in GPS trajectory or confidence level in map-matched locations. However, their novelty is their method of locating GPS point on the identified segment which is explained in Section 7.

Syed and Cannon (2004) proposed a fuzzy logic-based map-matching algorithm which takes input from a high sensitivity GPS and a gyroscope. Candidate segments for the first GPS point are those in a vicinity of 50 m. The first segment is identified based on: (a) difference between the heading determined by GPS and a segment with a higher weight, (b) proximity to a segment and (c) changes in the heading of the vehicle. After the first segment is identified, the algorithm switches from initialization to tracking mode. Tracking mode also uses a fuzzy inference system to identify the correct segment with the same antecedents plus traveled distance on the last segment. They also define a confidence level for map-matched locations but no further explanation is provided about its formulation. Topological constraints such as connectivity or turn-restrictions are not considered. Combining the last two steps of map matching (matching on the same segment as before and matching on another segment around intersections) into one fuzzy inference system is the major problem associated with their work which makes the process complicated, computationally intensive and not suitable for real-time applications.

One of the most accurate map-matching algorithms is proposed by Quddus, Noland et al. (2006). They deployed their map-matching algorithm as a fuzzy inference system (Fu et al., 2004; Syed & Cannon, 2004; Zhao, 1997). The positioning system is GPS augmented, during GPS outages, with data from DR sensors. They use vehicle's speed, horizontal delusion of precision (HDOP), distance to segment, direction of the segment, direction of the vehicle, heading of the vehicle and its error as the antecedents of fuzzy rules. The consequence of fuzzy rules is the likelihood of each road segment to be selected. Fuzzy rules are organized in three sets for three situations: initial map matching, subsequent map matching on a segment and subsequent map matching at an intersection. Their algorithm is among the best in terms of structuring and taking advantage of all data provided by GPS. First, fuzzy inference systems are very complex and computationally intensive especially in this case where 23 fuzzy rules must be checked and integrated for each candidate segment to calculate its probability of being the correct one. Therefore, it does not perform better than other algorithms in terms of performance which is very important in real-time map matching. Second, there is no provision to expedite the process of candidate segments selection at first GPS position fix. This issue is discussed in Section 3 in detail. Third, for the first segment to be identified, at least four GPS points in sequence must be located on the same segment, regardless of time it takes. Fourth, at intersections, they only consider connected roads to the last segment as candidates even though the vehicle might have passed through a segment during an outage. Finally, some other details are neglected in their work such as directionality of the road network.

Most map-matching algorithms in this section use a fuzzy inference system to identify the correct segment which is complicated and computationally intensive compared with weight-based algorithms discussed in the previous section. Pyo et al. (2001), Ochieng et al. (2004) and Fu et al. (2004) used the error ellipse around the GPS point to identify candidate segments which is also computationally intensive. Closeness is the only criterion used by Yang et al. (2003), Ren and Karimi (2009b) and Newson and Krumm (2009) to identify the correct segment. Other advanced algorithms use either heading difference or direction difference as another criterion but not both. The algorithm by Quddus, Noland et al. (2006) is the only which considered both heading difference and direction difference in identifying the correct segment. The algorithms by Ochieng et al. (2004) and Quddus, Noland et al. (2006) are the only ones that use different logics for identifying the correct segment before and after crossing an intersection.

Most researchers have not provided horizontal accuracy of their algorithms in urban areas due to lack of a higher accuracy reference (true) positioning system. An important point is that the performance of a map-matching algorithm is not only a function of its logic and positioning sensors it employs, but also the quality and density of the digital road network it uses and surrounding buildings which affect GPS signal availability. For example, Quddus et al. (2003), Ochieng et al. (2004) and Quddus, Noland et al. (2006) declared that their test routes were selected carefully to have good satellite visibility whereas Ren and Karimi (2009a, 2009b, 2012a) have implemented their algorithms on sidewalks which are much more complicated and error-prone. Many researchers have not revealed any information about the scale, density and complexity of their testing environments which makes it too difficult to compare them in terms of accuracy. Considering this, there is a need to a map-matching testbed including both road network and GPS trajectories within which existing and future algorithms can be assessed and compared.

Fig. 1 shows each algorithm in Table 2 by a small circle located based on its simplicity and declared correct segment identification percentage. All algorithms are categorized in three groups of simplicity: topological, weight-based and advanced. Algorithms located at the darker part of this chart are more efficient because they are both simple and accurate. According to Table 2 and

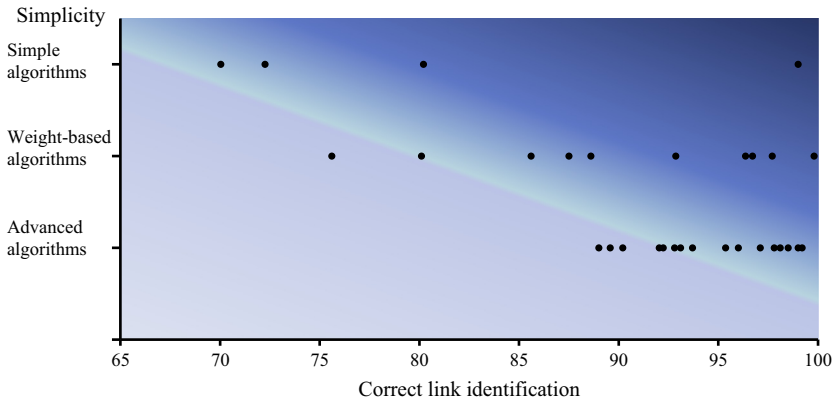

Fig. 1. Simplicity vs. accuracy of existing real-time map-matching algorithms.

Table 3
Main disadvantages of weight-based and advanced map-matching algorithms.

Issue	Wight-based algorithms	Advanced algorithms
Inefficient intersection detection	Yes	Yes
Ignoring heading or speed information provided by GPS in correct segment identification	Yes	Yes
Ignoring direction of sequential GPS points in correct segment identification	Yes	Yes
Ignoring directionality of road network or turn-restrictions	Yes	Yes
Ignoring uncertainty of heading information provided by GPS at low speeds	Yes	Yes
Not considering segment skipping under any circumstances	Yes	Yes
Not considering restricted-turns under any circumstances	Yes	Yes
Confidence level is not defined	Yes	Yes
Off-road conditions are not defined	Yes	Yes
Constant weights for criteria	Yes	Yes
Slow		Yes
Complex logic in algorithm		Yes

Fig. 1: (a) topological algorithms are simpler but less accurate, (b) advanced algorithms outperform other algorithms in terms of correct road identification and accuracy but they are inferior in terms of implementation simplicity and performance and (c) weight-based algorithms are more accurate than simple topological algorithms and less complex than advanced algorithms. However, real-time map-matching algorithms that are fast, easy to implement and do not need large memories are desired for practical navigation applications.

What makes advanced algorithms more accurate is that they generally use more inputs and more complicated approaches to identify the correct segment. One must attempt to utilize the same inputs and techniques as in advanced algorithms in a simple method. Replacing a complicated method such as fuzzy logic by a simple method such as a sophisticated weighted method may trade off a little accuracy for performance but it is worth in real-time systems and services. Studies such as Velaga et al. (2009) have proven that simple algorithms can be as accurate as advanced algorithms as long as they take account of all available data and knowledge optimally.

Table 3 summarizes the main disadvantages of weight-based and advanced algorithms reviewed in Section 2.

2.4. Using multi sensors in map-matching algorithms

The aforementioned algorithms use GPS, some augmented by DR (Hu, Khoo, Goh, & Law, 2002; Ochieng et al., 2004; Quddus et al., 2003; Retscher, 2002; Witte & Wilson, 2005; Zhao, Ochieng, Quddus, & Noland, 2003), as the only positioning sensor. Many researchers have taken advantage of other complementary sensors such as inertial measurement unit (Hasan, Samsudin, Ramli, Azmir, & Ismaeel, 2009), accelerometer (Ren & Karimi, 2012b) and camera (Ren, 2012) in addition to GPS/DR to improve positional accuracy or coverage. Multi-sensor algorithms limit their applicability as not all users or devices have those required sensors. Some researchers, for example Quddus et al. (2003) and Zhao et al. (2003), integrated GPS and DR sensor data using an extended Kalman filter in their map-matching algorithm. DR incrementally integrates the distance and direction of travel relative to a known position. Distance and direction are typically acquired from an onboard odometer and gyroscope, respectively. The main deficiency of DR is the accumulation of errors as the vehicle travels. However, DR as a relative positioning technique only augments the positioning system in areas with no GPS signal but it does not improve accuracy (Kubrak, Macabiau, Monnerat, & Bouchert, 2006; Zhao et al., 2003). Ren and Karimi (2012b) integrated orientation data from compass and knowledge of movement state (stopping, walking, running, and turning) recognized from accelerometer data with GPS positions to navigate wheelchairs and pedestrians on sidewalks. Knowledge of movement state is just used to determine the rate of location updates and is not used in the map-matching algorithm. Distance to segment, direction difference between GPS positions and segment, and heading difference between compass's heading and segment are the three criteria used to identify the correct segment. Socharoentum and Karimi (2011) proposed using some nearby car with a good GPS accuracy (low HDOP) as a dynamic base station in a DGPS method to improve the other car's positioning accuracy. A communication channel between two cars is a precondition of their proposed method.

All map-matching algorithms assume roads as single lines with no lanes. However, deployment of some applications such as collision-avoidance support systems requires the exact location of a vehicle across a road, i.e., its lane. Toledo-Moreo, Bétaille, and Peyret (2010) developed a multiple-hypothesis particle-filter-based algorithm to assign a lane to smoothed GPS/DR vehicle trajectories in multi-lane roads. However, a very precise GPS/DR positioning system and a large-scale map were indispensable to achieve a 98% correct lane allocation success rate. They just focused on developing a mathematical model for integrating positioning system's outputs with the road network to assign a lane to a point and determine its confidence level. Other aspects of map matching such as correct road identification around intersections were not explored.

3. Challenges and future directions

3.1. Candidate segments selection for first position

One notable challenge in map matching is narrowing the entire road network to a limited number of road segments called candidate segments for the first position. Among the most popular methods for finding candidate segments are: (a) finding the nearest node to the GPS point and then, finding all segments connected to that node (Fig. 2(a); Greenfeld, 2002; Quddus et al., 2003); and (b) finding all nodes inside the error ellipse around the GPS point (the error ellipse is formed from the error variance–covariance matrix associated with the positioning data) and then finding all segments connected to those nodes (Fig. 2(b); Ochieng et al., 2004). The possibility of non-existence of any nodes inside the error ellipse when the vehicle starts from the middle of a long segment (Quddus et al., 2007) results in another method: (c) finding all segments falling in (Fig. 2(c); Ochieng et al., 2004; Zewang, Yongrong, & Xing, 2002) or crossing (Fig. 2(d); Velaga et al., 2009; Quddus et al., 2007; Quddus, Noland et al., 2006) the error ellipse around the GPS point. However, due to the inherent problems associated with GPS in urban canyons and streets with dense tree cover, the error ellipse derived from quality of GPS position is normally large and contains a number of segments. This may cause difficulty in map matching to distinguish precisely on which

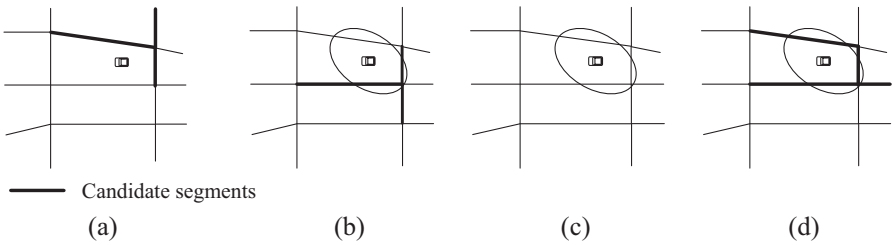

Fig. 2. Candidate segments selection: (a) segments connected to the closest node; (b) segments connected to the nodes inside the error ellipse; (c) segments falling inside the error ellipse; and (d) segments crossing the error ellipse.

particular segment the vehicle is travelling (Quddus, Ochieng, & Noland, 2006).

While most map-matching algorithms introduce a method for finding a collection of candidate segments among all segments for the first position, they rarely try to expedite the process. By utilizing a spatial indexing technique such as R-tree, a small number of primary nodes or segments need to be tested, resulting in reduced computation time, to find the nearest node to a GPS point, to find all nodes inside an ellipse, or to find segments crossing a circle. Ren (2012) developed and tested a hierarchical segment-clustering tree based on connectivity and distance to expedite the process of candidate segments selection. Segments on the lowest level of this tree, as the smallest units in the structure, are grouped together based on distance to each other. The clustering process is continued from bottom to up until the root of the tree is reached. The root of the tree represents the entire road network as one group. The minimum bounding rectangle (MBR) of the root contains the entire road network. To find candidate segments, the binary tree is traversed from top to bottom and the MBR is shrunk down until the distance between the GPS point and center of the MBR is less than a threshold. All segments crossing this MBR are candidate segments for the first point. Developing such a clustering tree is very laborious and inefficient for large road networks. Also, by adding or deleting a segment from the network, the clustering tree must be reconstructed.

Future works in this field must focus on developing more efficient metrics to expedite the process of finding: (1) closest node to a GPS point, (2) nodes inside an ellipse, and (3) segments inside an ellipse or crossing it. Also, the efficiency of different methods of candidate segments selection needs further investigation. The more frequent the true segment is included among candidate segments and the lesser the number of candidate segments, the more efficient the method will be.

3.2. Candidate segments selection for next positions

The process of candidate segments selection for next GPS points is different before and after crossing an intersection. If no intersection is crossed, there is no need for candidate segments selection. On the other hand, if the vehicle is crossing an intersection, new candidate segments must be selected. In contrast to first GPS point, now there is the additional knowledge of previous segment that helps the process of candidate segments selection. Most map-matching algorithms (White et al., 2000; Greenfeld, 2002; Ren & Karimi, 2009a, 2009b, 2012a, 2012b; Quddus et al., 2003; Quddus, Noland et al. (2006), Ochieng et al., 2004; Velaga et al., 2009) only include segments connected to the most recently matched segment to the problem space. Velaga et al. (2009) and Srinivasan et al. (2003) not only left out unconnected segments from candidates but also omitted turn-restricted roads. However, sometimes this assumption is not true. For example, there may be very short segments in the network on which the user travels very fast and does not have the opportunity to update its location. The same can happen if the GPS signal is masked or the navigation system/service is out for a few seconds and when the positioning system is resumed, the user has already passed through one or more segments; this is known as segment skipping problem (Greenfeld, 2002). In such circumstances, if the algorithm considers connectivity between two consecutive segments as a rule, it will use the segments already passed and will not be able to locate the vehicle on the correct segment. Greenfeld (2002) used connected segments to last segment plus all segments connected to them as candidate segments at intersections to cope with the problem of skipping a segment. However, it may considerably increase the number of candidate segments and still would not resolve the problem correctly in case the position update happens after two segments are passed. On the other hand, consider the situation where a car makes an illegal turn. If turn-restricted segments are already omitted from candidates (Srinivasan et al., 2003; Velaga et al., 2009), the algorithm will not be able to update the user's location correctly.

In short, connectivity and turn-restriction must properly be handled in future algorithms. We suggest while future algorithms consider connectivity and turn-restriction in the process of identifying the correct segment, they should also consider special conditions under which these rules are violated.

3.3. Detecting intersection crossing

Crossing an intersection means making a turn into a new segment or going straight into a new segment. Ren and Karimi (2012a) used two criteria to detect crossing an intersection. First, the angle between the segment and the line from a GPS point to the segment's starting node or the angle between the segment and the line from a GPS point to the segment's ending node exceeds 90°. Second, the distance between the last matched location and either of current segment's ending nodes is less than a constant threshold. However, there is no use of heading and direction information in their method which makes it weak in detecting turns.

Ochieng et al. (2004) used three criteria to detect turning maneuvers: (a) ascending or descending trend in heading for about 2–5 s, (b) an absolute difference of greater than 30° between the headings of current and last GPS position fixes and (3) an absolute difference of greater than 35° between the headings of current and second to the last GPS position fixes. They have considered one more criterion to detect crossing an intersection straightly, but it is not explained in their paper. Despite the fact that their method for detecting turning maneuvers at intersections seems effective, it has two main weaknesses: (a) it neglects that GPS heading information during turnings with low speed is very erroneous and (b) difference between headings of consecutive GPS points may rarely exceed these thresholds around intersections because of low speed and high polling frequency. One may take advantage of direction

difference between GPS points and road segment instead of heading difference between GPS points themselves (Velaga et al., 2009).

Velaga et al. (2009) introduced two criteria to detect approaching an intersection: distance from the previously map-matched location to the downstream intersection is less than 20 m or the vehicle's heading is 5° deviated from the segment. These two thresholds were derived empirically from field data containing 1800 GPS/DR position fixes. However, these threshold values depend on quality and scale of digital map, quality of positioning system and location update rate.

Quddus et al. (2003), Quddus, Noland et al., 2006) introduced three conditions to determine whether the vehicle has passed an intersection: (a) the difference between the direction of the line connecting two last GPS points $((x_{i-1}, y_{i-1})$ to $(x_i, y_i))$ and the line connecting two previous last GPS points $((x_{i-2}, y_{i-2})$ to $(x_{i-1}, y_{i-1}))$ is greater than 45°, (b) the difference between headings of last two GPS points obtained from GPS/DR is greater than 45° and (c) the angle between the segment and the line from a GPS point to the segment's starting node or the angle between the segment and the line from a GPS point to the segment's ending node exceeds 90°. Heading of GPS points at low speeds and direction of GPS points in short traveled distances are very inaccurate and this must be taken into account before using them to decide if an intersection has been passed. Second, since a vehicle turns into a new segment usually with a low speed, if the location update rate is high, the heading or direction difference between consecutive GPS points might not exceed the threshold (45°). Third, if the last matched location is extremely uncertain, matching the next GPS position on the same segment would be imprudent even if none of these criteria, proving the vehicle has passed an intersection, are confirmed.

Future studies on intersection detection must focus on methods which are not computationally intensive because cross detection must be performed with every GPS point. The criteria used to detect an intersection must be able to detect both going straight to the front segment and making a turn. Consequently, it must contain both distance and angular criteria. The distance of the vehicle to the end node of the segment is useful in detecting intersections when the vehicle goes straightly to the front segment but they cannot detect turns. Heading and direction differences are the best criteria for detecting turns; direction of GPS points is more accurate than their heading when making a turn with low speed.

3.4. Best segment identification for first GPS point

All map-matching algorithms consider the first map-matching step as a challenge (Scott, 1994; Syed & Cannon, 2004) because the available data items for identifying the correct segment are very little including heading of the GPS point and its distance to each segment. Incorrect identification of first segment may lead to a sequence of mis-matches. Therefore, confidence on result of the first step must be high before proceeding to the second step. Quddus, Noland et al. (2006) suggested to perform the initial map-matching process (using only heading of the GPS point and its distance to each segment) until there are four points located on the same segment in sequence. It requires at least four GPS points before the algorithm can produce an output but it may require additional points if the four points on the same segment are not in sequence, which means a longer processing time. The algorithm guarantees good solutions but determination of the first location may take a little time. In addition, if the assigned segment to the second or third GPS point has a high confidence level, there is no reason to wait for next points. Other researchers have not argued how they cope with the difficulty of the initialization step.

Because the correctness of the first identified segment is very important and at the same time the algorithm must not take long to update the user's location, we suggest a method that works based on a confidence level. The initialization step would continue until the confidence level is high enough to assure that the identified segment is correct. Therefore, the algorithm does not necessarily wait for a specific number of GPS points to update the user's location, instead it continuously calculates the confidence level for each identified segment and updates the user's location as soon as the required confidence is achieved.

3.5. Locating GPS position on identified segment

The most effective and widely used method for locating a GPS point on the identified segment is perpendicular projection (Velaga et al., 2009; Ren & Karimi, 2009a, 2009b, 2012a; Li et al., 2013). Ochieng et al. (2004) have proposed a much more complicated approach to locate a GPS point on the identified road segment. Their approach has also been applied by Scott and Drane (1994), Quddus et al. (2003) and Quddus, Noland et al. (2006). They first determine the location of vehicle by perpendicularly projecting the GPS position on the identified segment. They also determine the location of vehicle on the segment knowing its previous location on that segment, its speed and the time elapsed between last and current GPS position fixes. Since both methods are associated with errors, an optimal estimation procedure is applied to combine their results and determine the final location of vehicle on the road segment. They use error covariance associated with positioning data and road network in optimization process. However, this approach is much more complex than the simple method of perpendicular projection and how much the horizontal and along-track accuracies are improved is not known.

In the algorithm by Jagadeesh et al. (2004), the GPS point is perpendicularly projected on the identified segment for the first GPS point or when the identified segment is different from the previous one. The corresponding location of a GPS point on the road network, as long as the vehicle is traveling on the same segment, is determined by adding the traveled distance (distance between two consecutive GPS points) to the previous map-matched location along the segment. The assumption behind this method is that the distance between current and previous GPS positions is almost equal to the distance between corresponding true positions of the vehicle which is proven by examination of GPS signal behavior under actual field conditions (the error is less than 1 m in 95% of the cases; Jagadeesh et al., 2004). This approach is efficient in terms of simplicity and accuracy. It is even simpler than perpendicular projection method. However, the error associated with a map-matched location in this method stems from two sources: the errors associated with the first perpendicularly matched location on that segment, and the accumulative error in all measured distances between position pairs on that segment. However, it is not known whether these errors defuse or amplify each other and the results are not compared to the results of the perpendicular projection approach.

In short, the aforementioned methods have not been yet compared to the perpendicular projection method with respect to accuracy and performance. Future methods for projecting GPS points on the identified segment must compare their accuracy and performance at least with the perpendicular projection method.

3.6. Confidence level

Uncertainty in map matching can be expressed in three ways: uncertainty in identified segment, uncertainty in projected location on that segment and total uncertainty in identified segment and projected location (Karimi et al., 2006). Many researchers have developed complicated equations to combine various error sources associated with positioning data and digital roads map and dis-

cussed them in terms of confidence level. For example, Karimi et al. (2006) determined the 99% error ellipse around the true location of the vehicle (on a road) using the variance–covariance matrix associated with GPS. In fact, they expect that with a probability of 99% the positioning system will locate the vehicle somewhere inside this ellipse. They probe which part of the space inside this ellipse will be located on the true location by the map-matching algorithm. Then, they calculate the confidence level by dividing this area into the ellipse's area. To obtain the confidence level of each location, the entire process must be repeated which makes the process ineffective for real-time applications. Besides, they do not describe how the true position of the vehicle is known at every epoch. An advantage of their approach is that it takes into account density and complexity of the road network around a position. For instance, if the position is in a dense and complicated part of the network, a smaller part of the error ellipse would be projected on the true location which decreases the confidence level.

Quddus, Ochieng et al. (2006) proposed a Sugeno fuzzy inference system to formulate a confidence level for each map-matched location considering four parameters: (a) quality of position fix obtained from navigation sensors, (b) quality of digital map, (c) difference between heading of the vehicle and the identified segment and its associated error and (d) distance between the GPS point and map-matched location and its associated error. However, according to their conclusion this indicator does not reflect the confidence level on the identified segment well. Their method does not take into account the complexity and density of roads around the GPS point. The density and complexity of the road network around the GPS point extremely affect the confidence level. For example, even when the heading difference between the vehicle and the segment is small, existence of another segment near the GPS point with almost the same direction may lower the confidence level. On the other hand, the large difference between the heading of a vehicle and all other segments except the identified segment may increase the confidence level. Being computationally intensive is another disadvantage of this approach. Considering that confidence level must be calculated for each map-matched location separately, its formulation must be simple for real-time applications.

A very important point when formulating a confidence level is that it must take into account the density and complexity of roads around the GPS point. The error ellipse is not the best criterion to determine the confidence level because in dense roads the algorithm is more probable to make a mistake even if the error ellipse is small. On the other hand, in sparse roads, the algorithm is more likely to identify the segment correctly even if the error ellipse is large. Thus, confidence level models must take into account this subtle point.

4. Summary

Locating a vehicle on the road network is a crucial part of intelligent transportation systems such as route guidance and fleet management. Map-matching algorithms project raw position of the user provided by the positioning system on the road network by comparing trajectory of the vehicle with geometry and topology of the roads. A map-matching algorithm applies three strategies to deal with first GPS point, next GPS points on the same segment and next GPS points while the vehicle is switching to another segment. Mandating the algorithm to treat all GPS points in one or two steps, instead of three, has been shown to be inefficient.

Accuracy and performance are equally important in this process and none of them can immoderately be traded for the other. Many advanced map-matching algorithms require powerful computing platforms and are unsuitable for real-time applications. While some recent advanced map-matching algorithms are so complex and computationally intensive, they improve neither accuracy nor performance. On the other side, weight-based algorithms balance accuracy and simplicity well. However, they are a few steps away from producing optimal solutions. For example, the weight of each criterion in the best segment selection process remains constant through all GPS points, while it would be more efficient to determine the weight of each criterion based on special circumstances of each GPS point.

A detailed examination of state-of-the-art real-time map-matching algorithms reveal some problems requiring further research. Among the most important ones are:

- Most recent digital road networks include directionality and turn-restriction information in addition to simple topological information. However, using directionality and turn-restriction information in map-matching algorithms has not become as prevalent as topological information in state-of-the-art map-matching algorithms.
- Some map-matching algorithms include the following condition: "the next segment must be connected to the previous one". However, this condition is not met when the distance between two consecutive GPS points is very large and the vehicle has passed more than one segment since the last point. This is known as the "segment skipping" problem which needs to be addressed in future algorithms.
- Some map-matching algorithms include the following condition: "the vehicle can only turn onto legal road segments". However, this condition is not met when the driver turns, advertently or inadvertently, onto an illegal road segment. Future algorithms must address this problem.
- Although some existing algorithms have formulated a confidence level for map-matched locations, due to its importance to real-time applications, new algorithms must define a confidence level for matched locations. The same concerns off-road conditions. Besides, most formulated confidence levels ignore density and complexity of roads around the GPS point.

References

Bierlaire, M., Chen, J., & Newman, J. (2013). A probabilistic map matching method for smartphone GPS data. *Transportation Research Part C, 26*, 78–98.

Bierlaire, M., & Frejinger, E. (2008). Route choice modeling with network-free data. *Transportation Research Part C: Emerging Technologies, 16*(2), 187–198.

Brakatsoulas, S., Pfoser, D., Salas, R., & Wenk, C. (2005). On map-matching vehicle tracking data. In *Proceedings of the 31st international conference on Very large data bases* (pp. 853–864). VLDB Endowment.

Chen, W., Yu, M., Li, Z. L., & Chen, Y. Q. (2003). Integrated vehicle navigation system for urban applications. In *Proceedings of the 7th international conference on global navigation satellite systems (GNSS)* (pp. 15–22). Graz, Austria: European Space Agency.

Chen, B. Y., Yuan, H., Li, Q., Lam, W. H., Shaw, S.-L., & Yan, K. (2014). Map-matching algorithm for large-scale low-frequency floating car data. *International Journal of Geographical Information Science, 28*(1), 22–38.

Ebendt, R., Sohr, A., Tcheumadjeu, L. C., & Wagner, P. (2010). Utilizing historical and current travel times based on floating car data for management of an express truck fleet. In *5th international scientific conference: theoretical and practical issues in transport* (pp. 283–291).

Fu, M., Li, J., & Wang, M. (2004). A hybrid map matching algorithm based on fuzzy comprehensive Judgment. In *Proceedings of the 7th international IEEE conference on intelligent transportation systems* (pp. 613–617). Washington: IEEE.

Greenfeld, J. S. (2002). Matching GPS observations to locations on a digital map. In *Proceedings of the 81th annual meeting of the transportation research board*. Washington D.C.

Hasan, A. M., Samsudin, K., Ramli, A. R., Azmir, R. S., & Ismaeel, S. A. (2009). A review of navigation systems (integration and algorithms). *Australian Journal of Basic and Applied Sciences, 3*(2), 943–959.

Hu, G. R., Khoo, V. H., Goh, P. C., & Law, C. L. (2002). Internet-based GPS VRS RTK positioning with a multiple reference station network. *Journal of Global Positioning Systems, 1*(2), 113–120.

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

Joshi, R. R. (2001). A new approach to map matching for in-vehicle navigation systems: The rotational variation metric. In *Proceedings of intelligent transportation systems conference* (pp. 33–38). IEEE.

Karimi, H. A., Conahan, T., & Roongpiboonsopit, D. (2006). A methodology for predicting performances of map-matching algorithms. In *Proceedings of the 6th international symposium on web and wireless geographical information systems (W2GIS)* (pp. 202–213). Hong Kong: Springer Berlin Heidelberg.

Kim, J. S., Lee, J. H., Kang, T. H., Lee, W. Y., & Kim, Y. G. (1996). Node based map-matching algorithm for car navigation system. In *Proceedings of 29th international symposium on automotive technology and automation (ISATA)*, 10 (pp. 121–126). Florence, Italy.

Kim, S., & Kim, J.-H. (2001). Adaptive fuzzy-network-based C-measure map-matching algorithm for car navigation system. *IEEE Transactions on Industrial Electronics, 48*(2), 432–441.

Krumm, J., Letchner, J., & Horvitz, E. (2007). *Map matching with travel time constraints*. Detroit, Michigan, US: SAE World Congress.

Kubrak, D., Macabiau, C., Monnerat, M., & Bouchert, M. L. (2006). Vehicular navigation using a tight integration of aided-GPS and low-cost MEMS sensors. In *Proceedings of the institute of navigation technical meeting* (pp. 149–160).

Kuehne, R., Schäfer, R. P., Mikat, J., Thiessenhusen, K. U., Boettger, U., & Lorkowski, S. (2003). New approaches for traffic management in metropolitan areas. In *Proceedings of IFAC CTS symposium*.

Li, L., Quddus, M., & Zhao, L. (2013). High accuracy tightly-coupled integrity monitoring algorithm for map-matching. *Transportation Research Part C, 36*, 13–26.

Li, Q., Zhang, T., & Yu, Y. (2011). Using cloud computing to process intensive floating car data for urban traffic surveillance. *International Journal of Geographical Information Science, 25*(8), 1303–1322.

Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., & Huang, Y. (2009). Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems* (pp. 352–361). ACM.

Marchal, F., Hackney, J., & Axhausen, K. W. (2005). Efficient map matching of large global positioning system data sets tests on speed-monitoring experiment in zürich. *Transportation Research Record: Journal of the Transportation Research Board, 1935*(1), 93–100.

Miwa, T., Kiuchi, D., Yamamoto, T., & Morikawa, T. (2012). Development of map matching algorithm for low frequency probe data. *Transportation Research Part C, 22*, 132–145.

Newson, P., & Krumm, J. (2009). Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems* (pp. 336–343). ACM.

Ochieng, W. Y., Quddus, M., & Noland, R. B. (2004). Map-matching in complex urban road networks. *Brazilian Journal of Cartography (Revista Brasileira de zartografia), 55*(2), 1–18.

Pyo, J. -S., Shin, D. -H., & Sung, T. -K. (2001). Development of a map matching method using the multiple hypothesis technique. In *Proceedings of intelligent transportation systems conference* (pp. 23–27). Oakland (CA), USA: IEEE.

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

Quddus, M. A., Ochieng, W. Y., & Noland, R. B. (2006). Integrity of map-matching algorithms. *Transportation Research Part C: Emerging Technologies, 14*(4), 283–302.

Quddus, M. A., Ochieng, W. Y., & Noland, R. B. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies, 15*(5), 312–328.

Quddus, M. A., Ochieng, W. Y., Zhao, L., & Noland, R. B. (2003). A general map matching algorithm for transport telematics applications. *GPS Solutions, 7*(3), 157–167.

Quddus, M. A. (2006). *High integrity map matching algorithms for advanced transport telematics applications. PhD thesis.* London, United Kingdom: Department of Civil and Environmental Engineering, Imperial College.

Rahmani, M., & Koutsopoulos, H. N. (2013). Path inference from sparse floating car data for urban networks. *Transportation Research Part C, 30*, 41–54.

Ren, M., & Karimi, H. A. (2009a). A chain-code-based map matching algorithm for wheelchair navigation. *Transactions in GIS, 13*(2), 197–214.

Ren, M., & Karimi, H. A. (2009b). A hidden Markov model-based map-matching algorithm for wheelchair navigation. *Journal of Navigation, 62*(3), 383–395.

Ren, M., & Karimi, H. A. (2012a). A fuzzy logic map matching for wheelchair navigation. *GPS Solutions, 16*(3), 273–282.

Ren, M., & Karimi, H. A. (2012b). Movement pattern recognition assisted map matching for pedestrian/wheelchair navigation. *Journal of Navigation, 65*(4), 617–633.

Ren, M. (2012). *Advanced map matching technologies and techniques for pedestrian/wheelchair navigation, PhD Dissertation*. Pittsburgh, Pennsylvania, USA: School of Information Sciences, University of Pittsburgh.

Retscher, G. (2002). Accuracy performance of virtual reference station (VRS) networks. *Journal of Global Positioning Systems, 1*(1), 40–47.

Scott, C. A., & Drane, C. R. (1994). Increased accuracy of motor vehicle position estimation by utilising map data: vehicle dynamics, and other information sources. In *Proceedings of the vehicle navigation and information systems conference* (pp. 585–590). IEEE.

Scott, C. (1994). Improved GPS positioning for motor vehicles through map matching. In *Proceedings of the 7th international technical meeting of the satellite division of the institute of navigation* (pp. 1391–1400). Salt Lake City, Utah, USA.

Socharoentum, M., & Karimi, H. A. (2011). Collaborative map matching in Nav2Nav. In *Proceedings of 7th international conference on collaborative computing: networking, applications and worksharing (CollaborateCom)* (pp. 251–257). Orlando, Florida, USA: IEEE.

Srinivasan, D., Cheu, R. L., & Tan, C. W. (2003). Development of an improved ERP system using GPS and AI techniques. In *Proceedings of intelligent transportation systems conference* (Vol. 1) (pp. 554–559). IEEE.

Syed, S., & Cannon, M. E. (2004). Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons. In *Proceedings of the institute of navigation (ION) national technical meeting* (pp. 26–28). San Diego, CA.

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

Toledo-Moreo, R., Bétaille, D., & Peyret, F. (2010). Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps. *IEEE Transactions on Intelligent Transportation Systems, 11*(1), 100–112.

Velaga, N. R., Quddus, M. A., & Bristow, A. L. (2009). Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. *Transportation Research Part C: Emerging Technologies, 17*(6), 672–683.

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

Witte, T. H., & Wilson, A. M. (2005). Accuracy of WAAS-enabled GPS for the determination of position and speed over ground. *Journal of Biomechanics, 38*(8), 1717–1722.

Yang, D., Cai, B., & Yuan, Y. (2003). An improved map-matching algorithm used in vehicle navigation system. In *Proceedings of intelligent transportation systems conference* (Vol. 2) (pp. 1246–1250). IEEE.

Yang, J.-S., Kang, S. P., & Chon, K.-S. (2005). The map matching algorithm of GPS data with relatively long polling time intervals. *Journal of the Eastern Asia Society for Transportation Studies, 6*, 2561–2573.

Yu, M. (2006). *Improved positioning of land vehicle in ITS using digital map and other accessory information, PhD Dissertation*. Department of Land Surveying and Geoinformatics, Hong Kong Polytechnic University.

Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., et al. (2010a). T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems* (pp. 99–108). ACM.

Yuan, J., Zheng, Y., Zhang, C., Xie, X., & Sun, G.-Z. (2010b). An interactive-voting based map matching algorithm. In *Proceedings of 11th international conference on mobile data management (MDM)* (pp. 43–52). IEEE.

Zewang, C., Yongrong, S., & Xing, Y. (2002). Development of an algorithm for car navigation system based on Dempster-Shafer evidence reasoning. In *Submitted to IEEE international conference on intelligent transportation systems*. Singapore.

Zhang, X., Wang, Q., & Wan, D. (2003). The relationship among vehicle positioning performance, map quality, and sensitivities and feasibilities of map-matching algorithms. In *Proceedings of intelligent vehicles symposium* (pp. 468–473). Columbus, Ohio: IEEE.

Zhao, Y. (1997). *Vehicle location and navigation systems*. Boston: Artech House Publishers.

Zhao, L., Ochieng, W. Y., Quddus, M. A., & Noland, R. B. (2003). An extended Kalman filter algorithm for integrating GPS and low-cost dead reckoning system data for vehicle performance and emissions monitoring. *Journal of Navigation, 56*(2), 257–275.

Zhihua, L., & Wu, C. (2005). A new approach to map-matching and parameter correcting for vehicle navigation system in the area of shadow of GPS signal. In *Proceedings of intelligent transportation systems conference* (pp. 449–454). IEEE.