Mapping the Vehicle Trajectories onto Road Networks

A Project Report

submitted by

ANJI BABU VADAPALLI

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2013

THESIS CERTIFICATE

This is to certify that the thesis titled **Mapping the Vehicle Trajectories onto Road Networks**, submitted by **ANJI BABU VADAPALLI**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Radha Krishna Ganti
Project Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: MAY 07, 2013

ACKNOWLEDGEMENTS

First and foremost I would like to thank with immense appreciation to my guide, Dr. Radha Krishna Ganti for his guidance and continuous support throughout this project. I feel honoured and encouraged to have worked under his guidance.

I would also like to thank Dr. Krishna Jagannathan, Dr. Gaurav Raina, Dr. Sheetal Kalyani for their invaluable suggestions in this project.

I would like to thank the almighty, whose blessings have made me what I am. I would also like to take this opportunity to thank my parents for their support and encouragement.

I would like to thank my friends and labmates for their support and encouragement. I had fun filled sessions in the lab with my friends and enjoyed through and through my stay here.

ABSTRACT

KEYWORDS: GPS (Global Positioning System), map matching, ITS (Intelligent Transport Systems)

Map matching is the process of identifying the sequence of road segments traveled by the user, given a spatial road network and sequence of locations of a user moving on the network. Map matching is the key component of ITS (Intelligent Transportation System). It is the fundamental step for the many application like traffic flow analysis, route guidance systems, emergency management, advanced public transportation systems . As there are errors associated with the spatial road network and user location, pure geometric map matching algorithms can not achieve the good accuracy. We propose a topological map matching algorithm that will match the match the user's location on to the road segment. This algorithm makes use of the topological information of the road network. We tested the algorithm with GPS data of Washington state, US. The proposed algorithm is achieving good accuracy.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

GIS	Geographic Information System
UTM	Universal Transverse Mercator
GPS	Global Positioning System
DR	Dead Reckoning
HMM	Hidden Markov Model
ITS	Intelligent Transport Systems

CHAPTER 1

Introduction

1.1 Background

Vehicle telematics applications such as route guidance, fleet management, accident & emergency management, route recommendations, travel time predictions, traffic sensing, public transport operations are becoming increasingly popular. Real time sensor data such as location information collected by the vehicles such as taxis and buses in the urban areas provides vital information to these applications. All these applications make use of information provided by the mobile phone in the vehicle about it's location. Finding the road segment on which the vehicle is traveling using the positioning data is key to the applications mentioned above. Finding the road segment on which vehicle is traveling is called as map matching.

1.2 Definition of Data Inputs

Vehicle Trajectory: Vehicle trajectory T is a sequence of two-dimensional points $(p_1, p_2, ...p_n)$, where p_i represents the estimated position of the vehicle at a certain time using GPS.

Road segment: A road segment E is a directed edge that is associated with edge id, starting node, ending node and list of intermediate points that defines the it's geometry using a polyline l, Each line segment in l is a mini-edge and points connecting these mini-edges are mini-nodes. Each node or mini-node is a two-dimensional point that describes it's spatial location.

Road Network: Directed graph $G(V, E)$ with set of nodes V representing road intersections, and set of edges E representing the road segments.

1.3 Problem Statement

Given a vehicle trajectory T and a road network G, map the each point p in the trajectory T to an road segment (edge E) that the vehicle passed by.

1.4 Outline of the thesis

This thesis is organised into five chapters. Chapter one provides background to the study, problem statement. In chapter two, description of digital road network is presented. It gives the details about errors in map creation, digitization of the map, errors in digitization, digital map quality. In chapter three literature review of the map matching algorithms is presented. In chapter four, proposed topological map matching algorithm, it's implementation, and results are presented. Finally chapter five gives conclusion and future work.

CHAPTER 2

Digital Road Network Maps

2.1 Introduction

It is important to distinguish the difference between accuracy, precision, data quality, error.

Accuracy is the degree to which information on a map or in a digital database matches true or accepted values. Accuracy is an issue related to the quality of data and the number of errors contained in a dataset or map. In discussing a GIS database, it is possible to consider horizontal and vertical accuracy with respect to geographic position.

- The level of accuracy required for particular applications varies accordingly.
- Highly accurate data can be very difficult and costly to produce and compile.

Precision is the level of measurement and exactness of description in a GIS database. Precise locational data may measure position to a fraction of a unit. Precise information may specify the characteristics of features in great detail. It is important to realize, however, that precise data, no matter how carefully measured, may be inaccurate. Surveyors may make mistakes or data may be entered into the database incorrectly.

- The level of precision required for particular applications accordingly.
- Highly precise data can be very difficult and costly to collect.
- High precision does not indicate high accuracy nor does high accuracy imply high precision. But high accuracy and high precision are both expensive.

Data Quality refers to the relative accuracy and precision of a particular GIS database.

Error includes both the imprecision of data and its inaccuracies.

2.2 Errors in Map Creation

No map is perfect. Even the most accurate maps created by a GIS have some deficiencies. These deficiencies occur due to “Errors” that may have taken place at different stages of GIS implementation. These errors reduce the accuracy of the map generated. There are two types of errors in the map: Source Errors, Processing Errors.

2.2.1 Source Errors

These are the errors that are present in “Source” that is given to the GIS. They occur before the actual implementation of GIS. They are instrumental inaccuracies, human processing errors, age of data, map scale, actual changes. Instrumental inaccuracies are due to inaccuracies in the measuring instruments. Human processing errors are due to misinterpretation, typos, editing errors, incorrect use of devices error, measurement error. Age of data may be the obvious source of error, when the age of data is too old there is a possibility that some part of the data may have changed, this will cause the error if we proceed with the old data. Actual changes includes man made changes such as urban development, new roads, season and daily changes such as change of levels in the lake/river/sea, attribute changes such as forest growth, catastrophic changes or extremely harmful changes such as fires, floods, earth quakes.

Map scale can be defined as the ratio of distance on a map over the corresponding distance on the ground, represented as 1:M where M is the scale denominator. Map scale is an issue because as scale becomes larger the amount of detail that can be presented in a map is also increased. The ability to measure the length of linear features on the ground (road centreline), the position of point features (junctions and roundabouts), and the areas of polygons with a high level of accuracy are also increased. Map scales are usually divided into three categories, namely large, medium and small. A large-scale map extends over the range 1:1 to 1:24,000. Medium-scale maps range from 1:24,000 to 1:100,000. Anything smaller than 1:100,000 are considered small scale. A typical map used in the United States is at a scale of 1:50,000.

2.2.2 Processing Errors

These are the errors that will occur during the processing of the data. They are classification of data, numerical errors, map projection errors, digitization errors. The data model represents how the real world spatial entities such as roads, lakes etc. are represented in the GIS. Numerical errors are due to working with different computers each having it's own specifications, computing complexity. Numerical errors can occur due to the rounding errors, geometrical coordinate transformation. Digitization errors occur due to the conversion of analog map to the digital map.

As we know that the earth will be in circular shape, we use map projection to represent it as flat surface on computer screen, paper etc. There are many methods to do the map projection, Universal Transverse Mercator (UTM) projection is the most commonly used projection. Each state has a standard SPC (State Plane Coordinate system) based on one or more projections. Changing a map projection implies simultaneously changing the relationships of area, shape, and direction on a map. Each of these factors can introduce error into the representation of a point, line, and area on a map.

2.3 Map Digitization

The process of converting an analog map into a digital format is known as "Digitization". Although the most important technique of data input and storage in a GIS, digitization is also one of the most expensive and time consuming aspects of data input in a GIS as digitizing a large map can take hours to complete.

The spatial data stored forms the raw data for a GIS environment. Spatial data from the maps are stored in the form of points, lines and polygons. This means that features on the maps are represented digitally in three different forms i.e. points, lines and polygons. In point mode individual locations (like cities, villages etc. depending upon the scale of the map) are recorded by positioning the puck over the point and generating a single table co-ordinate pair. In a line mode, the lines are digitized by recording the co-ordinate positions of the starting point and the end point of the line segment. This is true for a straight line. However, curved lines are digitized by breaking up the curved line into a series of straight lines and recording the co-ordinates of these series of points. A polygon is a series of closed and interconnected lines. They are generally digitized in the form of a series of straight lines but the last and the first point of this series coincide with each other.

The digital capture of data from the analog sources i.e. maps, imageries, aerial photographs etc. is carried out in two different methods, manual digitization and heads up digitization.

2.3.1 Errors in Digitization

When converting the map from analog to digital we will loose some accuracy. Errors will occur during the digitization process. As mentioned earlier curved lines will be digitized as a series of straight lines. Topographical errors such as loops, knots, overshoots, undershoots, switch backs will occur frequently. The following figure shows the different topographical errors. We are representing three-dimensional earth surface as two-dimensional surface, hence we will get errors due to this. Paper maps are known to shrink with time. Any warping, stretching, folding or wrinkling of the original map will affect the digitization process as proper co-ordinates of such maps cannot be placed.

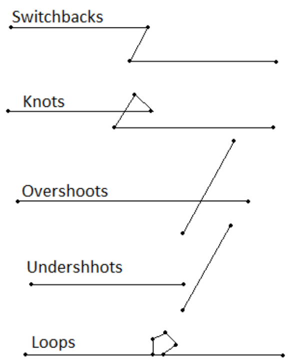

Figure 2.1: Topographical Errors during Digitization of Map

2.3.2 Digital Map Quality

Accuracy of the digitized products is the main aspect of the digitization. On-screen digitization is more accurate compared to manual digitization because the images for on-screen digitization are scanned at higher resolutions such that the operator can zoom the image to the scale of the original raster data and digitize with a higher level of accuracy. Quality of the digitization also depends on the operator who is digitizing the map, an experienced operator can digitize the map with more accuracy and speed when compared to fresh or newly appointed operator.

Digital map data is usually based on a single-line-road-network representing the centerline of the road. Road attributes such as width, number of lanes, turn restrictions at the junctions, and roadway classification (e.g., one-way or two-way road) normally do not exist in the map data. One must be aware of the following concerns regarding the quality of road network data:

- The features such as junctions, curves, roundabouts of the real world that have been omitted or simplified in the map. This is known as topological error.
- The accuracy of those features.
- How currently the map was created.
- The displacement of map feature such as specific junction, road centreline from it's actual position. This is known as geometrical error.

U.S. Bureau of the Budget set the accuracy standards for all federally produced maps these are called United States National Map Accuracy Standards[1]. The standards were established in 1941 and revised in 1947. The following were the standards.

- "On scales smaller than 1:20,000, not more than 10% of points tested should be more than 1/50 of an inch in horizontal error".
- "No more than 10% of the elevations (on an elevation map) tested will be in error by more than one half the contour interval".
- "Accuracy should be tested in comparison to actual survey data".

[1] http://nationalmap.gov/standards/nmas.html

CHAPTER 3

Overview of map-matching algorithms

3.1 Introduction

Map matching algorithms are used to find the road segment on which vehicle is traveling and it's location on the road segment. Most of the algorithms uses the navigation data obtained from GPS or GPS/DR and road network data from a digital map. One common assumption used in these algorithms is that the vehicle is constrained to finite set of road segments. Even though this assumption holds good in most of the cases we will encounter problems when the vehicle moves off-roadways such as parking or private land. In order to produce accurate results we need have digital map with high scale. Following section will briefly discuss the approaches for map matching algorithms found in the literature.

3.2 Literature Review

Approaches for map matching algorithms found in literature can be divided into four categories, they are Geometrical map-matching algorithms, Probabilistic map-matching algorithms, Topological map-matching algorithms, Advanced map-matching algorithms. In the following subsections these algorithms will be discussed briefly.

3.2.1 Geometrical map-matching algorithms

Geometric map matching algorithm uses geometric information of the road network by considering the only the shape of the edges or links or arcs (Greenfeld, 2002). It does not consider the way those edges or links or arcs are connected to each other.

Most commonly used geometric map matching algorithm is simple search algorithm (White *et al.*, 2000). It is also called as point-to-point map matching algorithm. In this algorithm each of the estimated location is mapped to the closest node or shape point in the road network. It is simple and easy to implement, all we have to do is search for the closest node or shape point in the network for which number of data structures and algorithms exist in literature. Even though it is simple and past it's performance depends on the how the nodes or shape points are connected. If edge is having more number of shape points then it is more likely to be matched properly, however for the edges with only two shape points or nodes all the points above the arc will match to the end nodes of the edge. Figure 3.1 illustrates the point-to-point map matching algorithm.

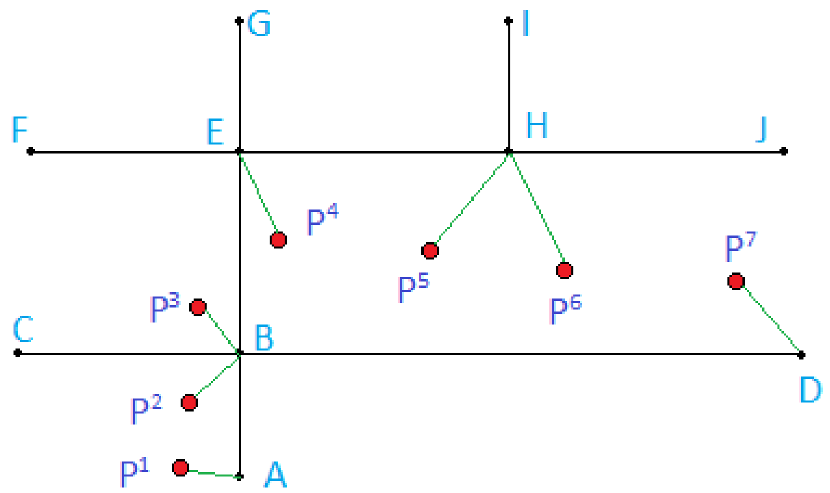

Figure 3.1: Point-to-Point map matching

In this example the points P^1 to P^7 denotes the estimated location of the vehicle on the road network with ten nodes (A to J). Actual road segments on which vehicle traveled during this period is AB and BD. But the point-to-point map matching algorithm shows that the vehicle traveled on AB, then BE and EH finally BD which is incorrect.

Another geometric map matching algorithm is point-to-curve map matching algorithm. In this algorithm the estimated vehicle location is matched onto the closest edge or arc. Edge is the straight line connecting the two nodes. For each positioning fix (estimated location) find the closest node. Compute the shortest distance from all the edges that are connected to closest and the edge with the least distance is the matched candidate for that positioning fix.

Even though this algorithm gives better results than point-to-point map matching it has many shortcomings. It will give unstable results in the dense urban road networks and closest edge always need not be the correct link and it is not using the historic information. When two edges are having the same distance from the position fix then choosing one is difficult. Figure 3.2 shows the result of the point to curve map matching algorithm for the same network that was shown in Figure 3.1. It can be easily observed that algorithm selects AB, BE and BD as correct links but the correct links are AB and BD.

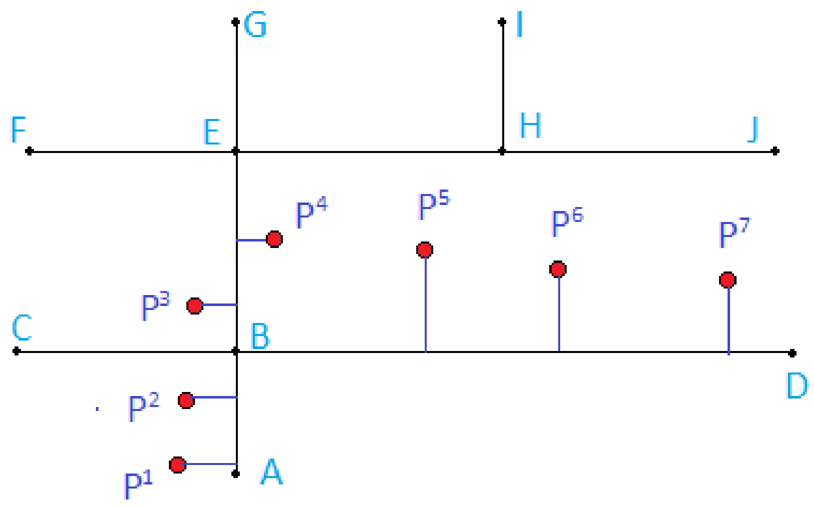

Figure 3.2: Point-to-Curve map matching

Another approach that gives better performance compared to the previously mentioned algorithms is curve-to-curve map matching algorithm. This approach firstly identifies the candidate nodes using point-to-point matching. Then, given a candidate node, it constructs piecewise linear curves from the set of paths that originates from that node.

Secondly, it constructs piecewise linear curves using the vehicle's trajectory i.e., the positioning points, and determines the distance between this curve and the curve corresponding to the road network. The road arc which is closest to the curve formed from positioning points is taken as the one on which the vehicle is apparently travelling.

This approach is quite sensitive to outliers and depends on point-to-point matching, sometimes giving unexpected results. This algorithm is more complex when compared to the previous two algorithms. Both point-to-point and point-to-curve algorithms are very unlikely to give accurate results at the intersection and (White *et al.*, 2000) suggested that more the the attention is given to the topological information, better the accuracy of algorithm.

An enhancement to the point-to-curve matching is proposed in (Srinivasan *et al.*, 2003). This advanced algorithm has two checks namely, bearing check and turn prohibition check incorporated in it. A bearing check is performed between the instantaneous vehicle heading obtained from GPS and the bearing of the road segment. A turning check is introduced to make sure that the heading difference between two consecutive position fixes from GPS agreed reasonably well with the turn calculated from the topology of the road network.

Bouju (Bouju *et al.*, 2002) proposed 3 geometric algorithms based on three factors distance, shift, direction. Distance is the shortest distance between position fix and the road (edge), shift is the difference between two position fixes not necessarily consecutive, direction is the difference between vehicle motion direction and road (edge) direction. They proposed "distance algorithm", "shift algorithm" and "shift and direction algorithm". Distance algorithm is same as point-to-curve algorithm. Shift algorithm and shift and direction algorithm are also same as distance algorithm but in shift algorithm we apply shift ΔP to the positioning fix P and in shift and direction algorithm we use the direction of vehicle, if the direction of the vehicle is too far from the road direction shift will be applied to the position fix.

Youze Tang (Tang *et al.*, 2012) proposed a off-line map matching algorithm which uses only GPS position fixes. Given a trajectory T $(p_1, p_2, .., p_n)$ algorithm will construct the index on the mini-vertices or nodes, it will retrieve the set of closest mini-vertices corresponds to each position fix p_i. Then it will determine the set of edges that are connected to those mini-vertices, based on which it computes the set of edges that position fix p_i might be mapped to. Then it will project the position fix p_i on to those candidate edges and computes the set of projected points. In the final stage it will use the dijkistra's algorithm to find the set of edges that the vehicle traveled. This algorithm is complex and it is not useful for on line purpose as it computes the edges after the entire trajectory is available.

Kuien Liu (Liu *et al.*, 2012) proposed an off-line map matching algorithm for the most simplified road networks. This algorithm does not use the geometry of the road network it uses only edge and node information. This algorithm is called as passby algorithm. Basic idea this algorithm uses is it will check whether the vehicle is passing by simplified road segment or not. It will check whether two GPS points p_i and p_l passes the starting and ending intersections of the road segments, if thy pass by then it will assign that road segment to the all GPS position fixes from p_i to p_l. Figure 3.3 illustrates what are all the factors that will considered while checking whether the vehicle pass by the simplified road segment. This algorithm uses weighted formula of $d_p, \theta_i, d_t, \theta_t$ to find whether the position fix p_i pass road intersection or not. Where d_p is the projection distance between p_i to edge e, θ_t is the intersection angle between line $\overrightarrow{p_{i-1}, p_i}$, d_t is the traversing distance projected from starting node ($e.start$) of edge to $\overrightarrow{p_{i-1}, p_i}$ and θ_t is the traversing angle between line $\overrightarrow{e.start, p_i}$ and line $\overrightarrow{p_{i-1}, p_i}$

3.2.2 Topological map-matching algorithms

Topology refers to spatial relationship between the geographical features such as points, lines, and polygons. The relationship can be between connecting or adjacent features. Topological relationships are built from simple elements into complex elements: points (simplest elements), arcs or edges (sets of connected points), areas (sets of connected arcs), and routes (sets of sections, which are arcs or portions of arcs). Topology has three basic components: Connectivity, Area Definition / Containment, Contiguity.

- Connectivity
  - In case of points the relationship can be defined as connectivity

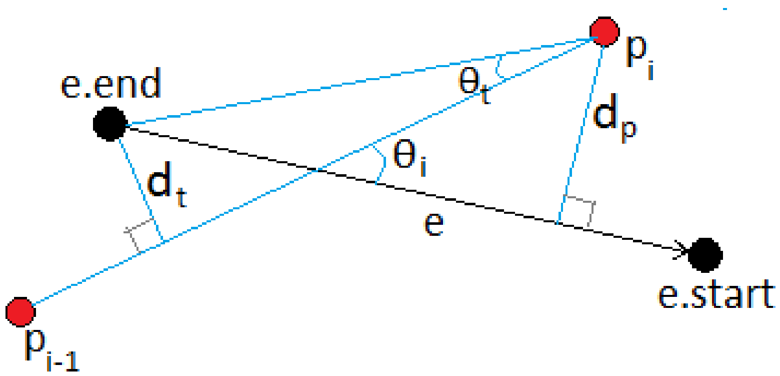

Figure 3.3: Weights used for checking the whether vehicle passby or not

- This is also called as arc - node or edge - node topology
- The points which connects the edge or arc are nodes or vertices
- The points which defines the geometry of the edge are shape points or mini-nodes or mini-vertices
- This information is used to find the how the different edges are connected to each other.

- Area Definition / Containment
  - In case of points in the polygon the relationship can be defined as Containment
  - This is also called as polygon - edge topology
  - An enclosed polygon has a measurable area.
  - Lists of arcs define boundaries and closed areas are maintained.
  - Polygons are represented as a series of (x , y) coordinates that connect to define an area.
  - This is used to find whether the point interest lies inside particular area or not
- Contiguity
  - Each edge is having the particular direction.
  - GIS maintains a list of Polygons on the left and right side of each arc.
  - We use this information to determine which features are next to one another.

The example database shown in Table 3.1 summarizes the basic components of topology.

Connectivity		Containment / Area Definition		Contiguity	
Node	Edges	Polygon	Edges	Edge	Left & Right polygons
1	a1,a2,a6	A	a1,a2,a3	a1	A/D
2	a2,a3,a5	B	a2,a5,a6	a2	A/B
3	a1,a3,a4	C	a3,a4,a5	a3	A/C
4	a4,a5,a6	D	a1,a4,a6	a4	C/D
-	-	-	-	a5	B/C
-	-	-	-	a6	B/D

Table 3.1: Topology example

Yu Meng (Meng *et al.*, 2002) proposed a simplified map matching algorithm using the topological analysis. This algorithm is based on the correlation between the trajectory of the vehicle and the topological features of the road such as road turn, road geometry, road connectivity. This algorithm is divided into 4 key operations road identification, road feature detection, road following, reliability and integrity. Road identification is the process of determining the road segment on which vehicle is traveling. Road feature detection is the process of finding the geometric and topological features of the road network such as road turn, road curvature, road connection. After the road identification the vehicle position is projected onto the proper position relative to the road identified. In reliability an d integrity number of conditional tests are applied to eliminate road segments that do not fulfil some pre-defined thresholds. The thresholds are obtained from the statistical analysis of field-test data.

Quddus (Quddus *et al.*, 2003) proposed a topological map matching algorithm. This algorithm is used together with the outputs of extended kalman filter for the integration of GPS and dead reckoning data. This algorithm assumes that the positional information from GPS,information about speed of the vehicle from and heading information are available as input. This algorithm consists of two parts, determining the road (link) on which vehicle is traveling and finding the location of the vehicle on that link. It uses three weightings for determining the road on which vehicle is traveling they are : weighting for vehicle heading and bearing of the link, weighting for proximity of a point to link, weighting score for the position relative to the link.

The basic idea is to find the set of candidate links that are closer to the position fix, for each link compute the weighting scores as explained below and sum the all weights (total score). The link (road) with the maximum total score is the link (road) on which vehicle is traveling apparently. Weighting for the vehicle heading and bearing of the link will be calculated using the heading information of the vehicle and the bearing of the link, heading of the vehicle measured relative to the northerly direction and bearing of the link will be determined from the spatial map data provided. Two types of weighting scores are used for proximity, first one is based on the perpendicular distance from the position fix to the candidate link and the second one is based on the intersection between line connecting the two consecutive position fixes and the candidate link, if intersection exists weight will be computed using the intersection angle otherwise weight will be taken as zero. Weighting score for position relative to the link is calculated using the angle between the lines connecting the starting node of the link, position fix and link.

Once we determine the link (road) on which vehicle is traveling, the physical location of the vehicle on the selected link will be carried out using the position fix, spatial data and vehicle speed from the integrated GPS unit. After assigning the vehicle position on the correct link, the algorithm will check whether the vehicle is still on the same link. For this purpose they introduced two test conditions, they are difference between the bearings or headings of the two consecutive GPS lines and the intersection angle used for the computation of the weighting score for position relative to the link. If these two values are below the certain threshold then vehicle is still on the same link, physical location of the vehicle on the link will calculated. To detect the outliers in the data this algorithm uses the angle between the two consecutive GPS lines, if the angle exceeds certain threshold then it is outlier.

3.2.3 Probabilistic map-matching algorithms

In probabilistic map matching combinations of techniques and methods based on topological information and probability theory is used. These algorithms uses the variance and covariance error values received from GPS and construct an error ellipse around the position fix to determine candidate set of edges, then use proximity and orientation to find the correct match. Some algorithms that found in the literature use hidden markov model (HMM), viterbi algorithm for map matching. Most of the algorithms which uses HMM and viterbi are the off-line map matching algorithms.

John Krumm (Newson and Krumm, 2009) proposed a off-line map matching algorithm. It is based on hidden markov model (HMM) and viterbi. Hidden states are road segments on which vehicle traveled, observation or emission probabilities are the probability that the GPS observation comes from particular state, transition probabilities are the probability that the vehicle moves from one candidate road segment to the other given two successive observations (position fixes) p_{i-1} and p_i. In this approach first the set of candidate edges will be found, candidate edges are the edges within particular threshold distance from the position fix. Emission probabilities will be computed using the great circle distance[1] between the position fix and the projected points onto the candidate edges. If p_t is the position fix and r_i is the candidate road segment then the emission probability is computed using the following formula

$$p(x_t|r_i) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-\frac{\|p_t - x_t\|^2_{greatcircle}}{2\sigma^2}} \tag{3.1}$$

where x_t is the projected of position fix p_t on to the edge r_i and σ_z^2 is the variance of the GPS measurements. Transition probabilities will be calculated using the difference of great circle distance and the route distance for given two successive observations p_{i-1} and p_i. Route distance is the "driving distance". If p_t is the position fix, r_i is the one of the candidate edge and $x_{t,i}$ is the projected point on to the r_i, similarly next position fix we have p_{t+1} , r_j, $x_{t+1,j}$ then transition probability will be calculated using the following formula:

$$p(r_j|r_i) = \frac{1}{\beta} e^{-\frac{d_t}{\beta}} \tag{3.2}$$

where

$$d_t = |\|p_t - p_{t+1}\|_{greatcircle} - \|x_{t,i} - x_{t+1,j}\|_{route}| \tag{3.3}$$

and β is describes the difference between great circle distance and driving distance,

[1]Circular distance between the two points on the earth surface

it is estimated with the help of GPS measurements and ground truth data. Once we have emission probabilities and transition probabilities we invoke viterbi algorithm to find the most likely sequence of road segments traveled by the vehicle. Authors of this paper used preprocessing of the GPS data to eliminate the input points that looks like noise, they removed the positioning fix if it is as a distance less than $2\sigma_z$ distance from it's predecessor. For removing outliers this algorithm uses the speed of the vehicle, if the calculated route requires vehicle speed to be more than 50 m/s then it is considered as a outlier.

Rudy raymond (Raymond *et al.*, 2012) also proposed a HMM map matching algorithm which is similar to the (Newson and Krumm, 2009) but it uses the sampled road network and hidden states are the nodes.

3.2.4 Advanced map-matching algorithms

Advanced map matching algorithms are referred to as those algorithms that use complex concepts such as fuzzy logic, kalman filtering, multiple hypothesis technique. Most of the algorithms are off-line. We need good processors for these algorithms.

Mengyin FU (Fu *et al.*, 2004) proposed a map matching algorithm which uses the fuzzy logic model to determine the correct edge among the candidate edges. It uses two factors minimum distance between position fix and edge, the difference between vehicle direction and edge direction. Fuzzy logic model is sensitive to the measurement noise. It does not use the historical hence the possibility of mismatch will be more.

CHAPTER 4

Topological Map-Matching Algorithm

4.1 Map Matching Process

The map matching algorithm we proposed is heavily depends on algorithms proposed in (Greenfeld, 2002) and (Quddus *et al.*, 2003). Map matching procedure was developed with the following characteristics: It is based on the topological analysis and uses only coordinate information of the observed position of the user. It doesn't use any heading and/or speed information of the user. The basic idea is to determine the candidate edges which are linked to the two closest nodes, and use three measures to find the best matched edge for the given i^{th} gps observation (p_i), they are distance, direction similarity and intersection. These measures are discussed below.

4.1.1 Identification of the Road Segment

The most difficult part of the any map matching algorithm is to find the correct edge or road segment among the candidate edges.

Weighting for Distance Proximity

This measure is shortest distance (D) between the p_i and the candidate edge. Shortest distance is not always the perpendicular distance. Shortest distance is equal to perpendicular distance between a point and edge if the projection of the point onto the edge lies between it's end segments otherwise it is the distance of the point to the closest end point of the line segment. The weighting score for distance proximity is calculated using the following formula:

$$W_d = C_d - a \cdot D^{n_d}$$

Where D is the shortest distance, C_d, a and n_d are the weighting parameters. The following are the three different cases for shortest distance (D) calculation.

Case 1: When the projected point onto the edge lies between it's end segments. Figure 4.1 illustrates this case where the point (x_3, y_3) represents p_i, (x_1, y_1) and (x_2, y_2) are

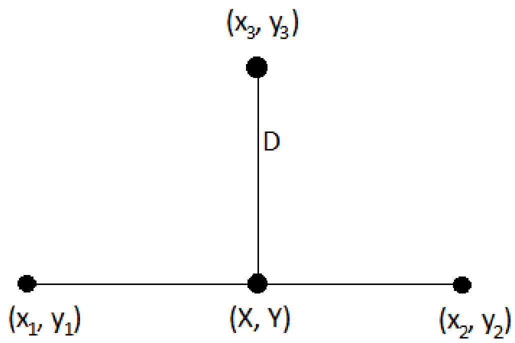

Figure 4.1: Projected point onto the edge lies between it's end segments

starting and ending nodes of the edge, (X, Y) is the projected point onto the edge. In this case shortest distance (D) is distance between (x_3, y_3) and (X, Y).

Case 2: When the projection of p_i onto the edge doesn't fall between it's end segments and p_i is close to end node of the edge. Figure 4.2 illustrates this case In this case shortest distance (D) is the distance between (x_3, y_3) and (x_2, y_2).

Case 3: When the projection of p_i onto the edge doesn't fall between it's end segments and p_i is close to start node of the edge. Figure 4.3 illustrates this case. In this case shortest distance (D) is the distance between (x_3, y_3) and (x_1, y_1)

To find whether the projected point onto the edge lies in between it's end segments or not we use dot product. Let (x_1, y_1) and (x_2, y_2) are starting and ending nodes of the edge and (x_3, y_3) is the gps point for which we need to find the road segment (edge) on which vehicle traveled. We use the following method to find the shortest distance D:

- Find the direction vector $(\bar{X}1, \bar{Y}1)$ from (x_1, y_1) to (x_2, y_2).
- Find the direction vector $(\bar{X}2, \bar{Y}2)$ from (x_1, y_1) to (x_3, y_3).

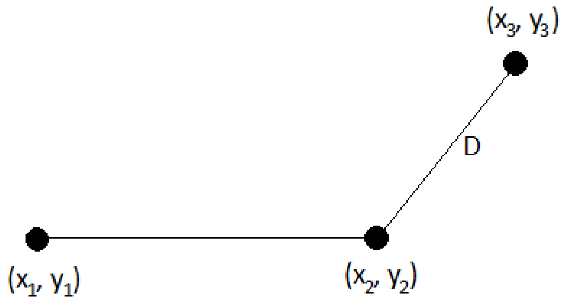

Figure 4.2: Projected point is end node of the edge

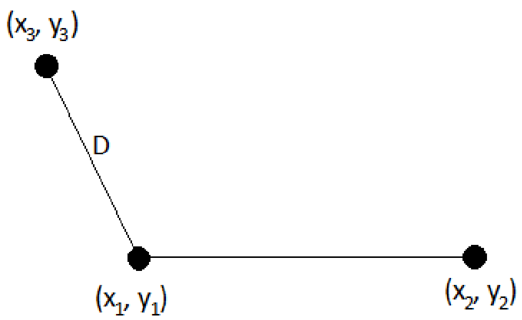

Figure 4.3: Projected point is the starting node of the edge

- Find the dot product $c1$ between $(\bar{X}1, \bar{Y}1)$ and $(\bar{X}2, \bar{Y}2)$

$$c1 = \bar{X}1 * \bar{X}2 + \bar{Y}1 * \bar{Y}2 \tag{4.1}$$

- Find the distance $c2$ between the (x_1, y_1) and (x_2, y_2)

$$c2 = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \tag{4.2}$$

- If $c1 \leq 0$ then projection of the point (x_3, y_3) doesn't fall onto the edge and shortest distance D is the distance between (x_3, y_3) and (x_1, y_1).

- If $c2 \leq c1$ then also projection of the point (x_3, y_3) doesn't fall onto the edge and shortest distance D is the distance between (x_3, y_3) and (x_2, y_2).

- If the above two conditions doesn't satisfy then projection of the point (x_3, y_3) onto the edge falls in between it's end segments and shortest distance D is the distance between (x_3, y_3) and (X, Y) where

$$X = x1 + (x2 - x1) * c1/c2 \tag{4.3}$$

$$Y = y1 + (y2 - y1) * c1/c2; \tag{4.4}$$

Weighting for Direction Similarity

This measure is the angle between the line connecting $\overrightarrow{p_{i-1}, p_i}$ and candidate edge (4.4). It gives the degree of parallelism or similarity in orientation between the gps line segment and road segment (edge). The edge with less angle will be given more weight since it is more likely the correct match. The weighting score is computed using following formula:

$$W_\theta = C_\theta \cdot cos(\theta)$$

where θ is the angle between $\overrightarrow{p_{i-1}, p_i}$ and candidate edge, C_θ is the weighting parameter.

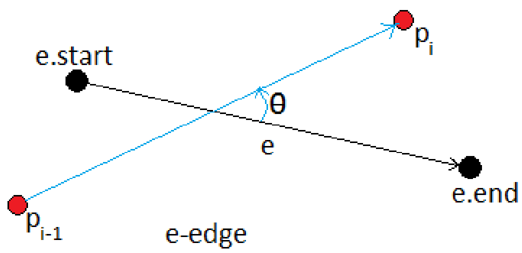

Figure 4.4: Weighting for similarity in orientation

Weighting for Intersection:

This measure is to find whether the line segment connecting $\overrightarrow{p_{i-1}, p_i}$ and candidate edge intersect. If the intersection occurs and the intersection angle is small then it is very likely the correct match. The weighting score for the intersection is calculated using following formula:

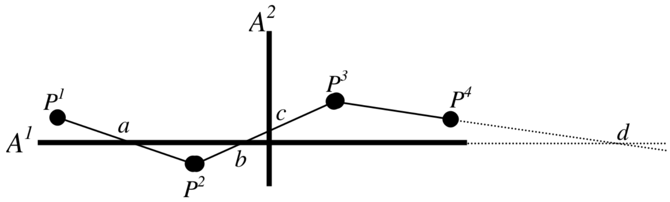

Figure 4.5: Intersections of GPS lines with road segments

$$W_i = C_i \cdot cos(\theta)^{n_i}$$

Where θ is the intersection angle, C_i and n_i are weighting parameters. We will consider intersection exists only if the intersection point lies on the both the line segments and the intersection angle is less than $\pi/4$. In the Figure 4.5, the valid intersections to be considered are a, b only. c, d are not considered as valid intersections since for d intersection point falls on the extension of the line segments adn for c intersection angle is greater than $\pi/4$.

Total Weighting Score

Total weighting W score is the sum of all the weighting scores

$$W = W_d + W_\theta + W_i \tag{4.5}$$

4.2 Outlier Identification

Outliers will present in the positioning data due the multipath effect on the signal. Due to outliers GPS signal exhibit jittery pattern. In other words, the positioning fixes following the certain road segment all of sudden spike will occur. An example for the outlier is presented in the Figure 4.6. The position fix P^4 is considered as outlier if $\Delta\delta > 45°$ i.e if there is a sudden change in the two consecutive GPS line bearings. This outlier may cause the matching error especially if $\Delta\delta$ is bigger and closer to edge DE. If we encounter a outlier simply we matched it to the previously detected road segment and it is observed in the simulation results that algorithm is achieving good accuracy with this approach.

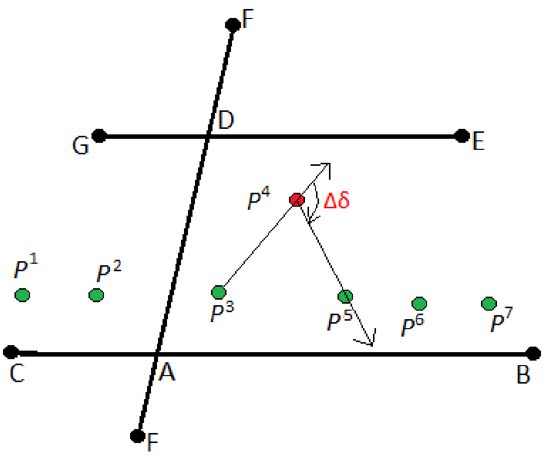

Figure 4.6: Outlier Identification

4.3 Algorithm Step by Step

For i=1 to 2

- Find the two closest nodes from the gps observation p_i.
- Determine the all edges that are connected to these two nodes.
- For each candidate edge evaluate the following measures:
  - D: Shortest distance between the p_i and the candidate edge
  - θ: Angle between the line connecting two consecutive gps points p_{i-1}, p_i and candidate edge
  - C_i: Whether the line segment connecting gps points p_{i-1}, p_i will intersect the candidate edge segment. If intersection is valid then $C_i = 10$ otherwise $C_i = 0$.

  - Compute the total score $W = W_d + W_i + W_\theta$
    where W_d is the weighting score for distance proximity, W_i is the weighting score for intersection and W_θ is weighting score for direction similarity.

- The candidate edge which is having maximum total score is the best matched edge for the given gps observation p_i

For i=3 to end

- If p_i is an outlier then match it to the previously estimated edge and increase i by 1.
- Otherwise:
  - Find the closest node from the gps observation p_i.
  - Determine the all edges that are connected to closest node, the starting and ending nodes of the two previously detected edges.
  - For each candidate edge evaluate the shortest distance (D), angle (θ) and intersection (C_i) and their corresponding weights, compute the total score.
  - If the end node of the previously estimated edge and start node of the candidate edge are same (or) previously estimated edge and candidate edge are same then double the weighting parameter C_d.
  - The candidate edge which is having highest score is the best matched edge.
  - Find whether p_{i+1} is outlier or not by computing the angle between two successive gps lines $\overrightarrow{p_i, p_{i+1}}$ and $\overrightarrow{p_{i+1}, p_{i+2}}$. If the angle is greater than the $\pi/4$ then it is an outlier.
  - Increase i by 1.

4.4 Implementation and Results

The above algorithm is implemented in C using the the data of the road network of the Washington State of US. It contains 5,35,452 nodes, 12,83,540 edges. The data we used in simulation contains 16 files: 8 trajectory files contains the gps trace route of individual trip, 8 files containing the ground truth data. Each row in the node file contains three values, each value is separated by comma.The form of node row is:
$<$ Node Id $>$, $<$ lat $>$, $<$ lon $>$.
where $<$ Node Id $>$ is the node number, $<$ lat $>$ and $<$ lon $>$ represents the location of the node within the road network in decimal degrees. Each row of the edge file contains four values separated by space. The form of edge row is:

< Edge Id >< from >< to >< cost >

Where Edge Id is the corresponding edge number, from and to are the Node Id's connecting the edge and cost is the actual cost of a vehicle to traverse from one end of the edge to the other end. Cost depends on length of the edge and the speed limit on the road segment the edge represents. Trajectory file contains three values separated by comma. The form of trajectory file row is:

< time >, < lat >, < lon >

where < time > represents time at which gps observation made, < lat > and < lon > are the latitude and longitude of the location in decimal degrees. At each time instant we need estimate the actual road segment (edge) on which vehicle is traveled

Table 4.1 shows the weighting parameters that were used in simulation. Table 4.2 gives the results of the algorithm for each trajectory file.

W_θ	C_θ=10
W_d	C_d=20 a=0.17 n_d=1.4
W_i	C_i=10 n_i=10

Table 4.1: Weighting parameters used in simulation

Route	Trajectory length (sec)	Accuracy (%)
1	1070	95.514
2	1566	95.2745
3	884	93.552
4	1017	92.527
5	2371	93.2096
6	1135	93.039
7	1543	93.3246
8	1320	96.7424

Table 4.2: Accuracy of the algorithm for different inputs

Following Figures from 4.7 to 4.18 shows the visualization plots before and after the map matching. We used array of linked list to project the each position fix p_i on to the estimated edge.

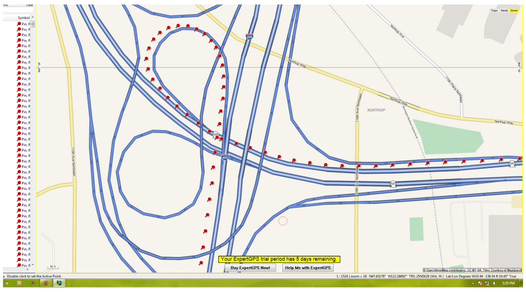

Figure 4.7: Before Matching

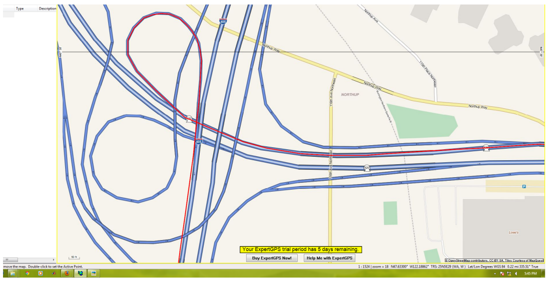

Figure 4.8: After Matching

Figure 4.9: Before Matching

Figure 4.10: After Matching

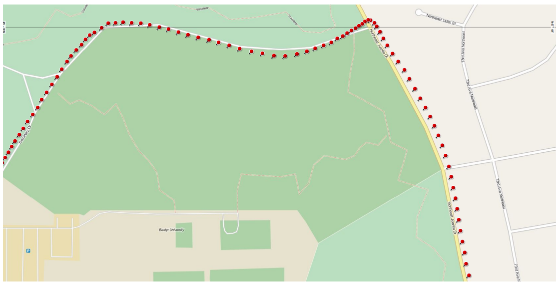

Figure 4.11: Before Matching

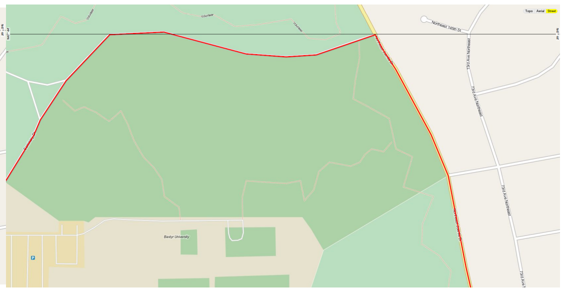

Figure 4.12: After Matching

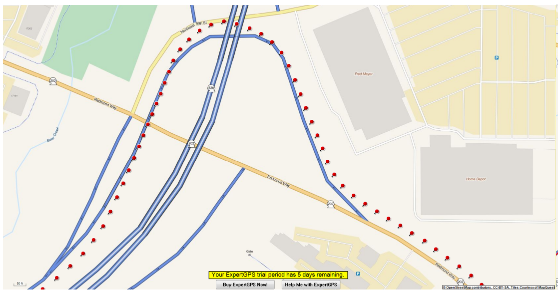

Figure 4.13: Before Matching

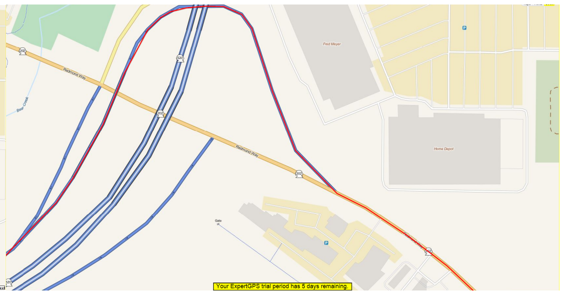

Figure 4.14: After Matching

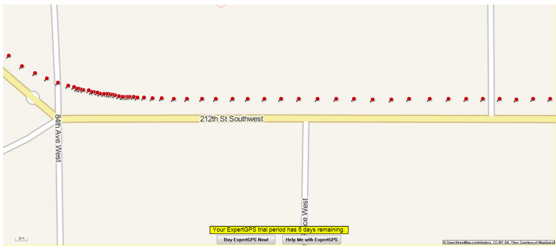

Figure 4.15: Before Matching

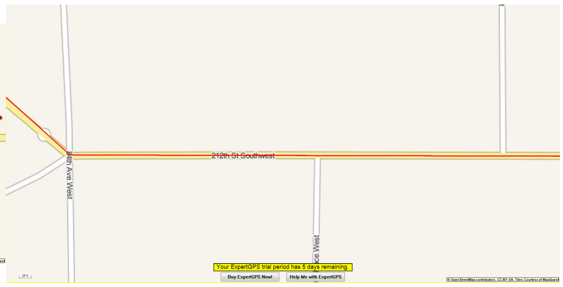

Figure 4.16: After Matching

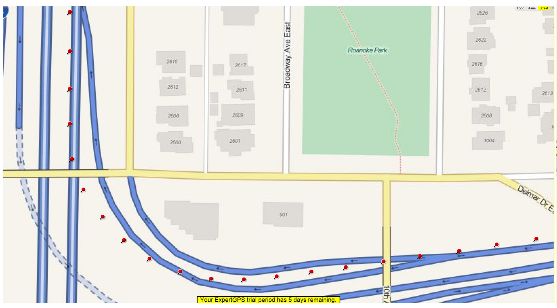

Figure 4.17: Before Matching

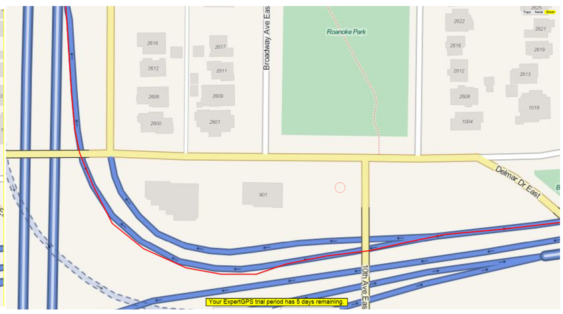

Figure 4.18: After Matching

CHAPTER 5

Conclusion

In this thesis, topological map matching algorithm is proposed. It uses only the (x, y) coordinates of the GPS position fixes. For determining the correct road segment on which vehicle was passed by it uses the previous position fix information as well, and for determining the outliers it uses the next two position fixes. From the simulation results it is observed that, the proposed algorithm is achieving good accuracy and also it's complexity is low.

Future work

This thesis considers two closest nodes while computing the set of candidate edges there might be a chance that these set of edges can't contain the actual edge on which vehicle was passed by, one can consider the set of edges within the certain threshold distance to the closest nodes as well. We can use spatial data structures such as r-trees, grid indexing to find the set of edges within certain threshold distance. Also we can use the cost's provided with the each edge.

REFERENCES

1. **Bouju, A.**, **A. Stockus**, **R. Bertrand**, and **P. Boursier**, Location-based spatial data management in navigation systems. *In Intelligent Vehicle Symposium, 2002. IEEE*, volume 1. IEEE, 2002.

2. **Fu, M.**, **J. Li**, and **M. Wang**, A hybrid map matching algorithm based on fuzzy comprehensive judgment. *In Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 2004.

3. **Greenfeld, J. S.**, Matching gps observations to locations on a digital map. *In National Research Council (US). Transportation Research Board. Meeting (81st: 2002: Washington, DC)*. 2002.

4. **Liu, K.**, **Y. Li**, **F. He**, **J. Xu**, and **Z. Ding**, Effective map-matching on the most simplified road network. *In Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 2012.

5. **Meng, Y.**, **W. Chen**, **Z. Li**, **Y. Chen**, and **J. C. Chao** (2002). A simplified map-matching algorithm for in-vehicle navigation unit. *Geographic Information Sciences*, **8**(1), 24–30.

6. **Newson, P.** and **J. Krumm**, Hidden markov map matching through noise and sparseness. *In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009.

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

8. **Raymond, R.**, **T. Morimura**, **T. Osogami**, and **N. Hirosue**, Map matching with hidden markov model on sampled road network. *In Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012.

9. **Srinivasan, D.**, **R. L. Cheu**, and **C. W. Tan**, Development of an improved erp system using gps and ai techniques. *In Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 1. IEEE, 2003.

10. **Tang, Y.**, **A. D. Zhu**, and **X. Xiao**, An efficient algorithm for mapping vehicle trajectories onto road networks. *In Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 2012.

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