

## Estimation of Mobile User's Trajectory in Mobile Wireless Network

### Framework, Formulation, Design, Simulation and Analyses

Sarfraz Khokhar

Mobile Internet Group  
Cisco Systems, Inc.  
Research Triangle Park, NC USA  
khokhar@cisco.com

Arne A. Nilsson

Department of Electrical and Computer Engineering  
North Carolina State University  
Raleigh, NC, USA  
nilsson@ncsu.edu

**Abstract**—There has been extensive research and development on Location Based Services (LBS). The focus here is on the present location of the mobile user. A new trend of mobile wireless services based on trajectory of a mobile phone and hence that of a mobile user, called Mobile Trajectory Based Services (MTBS) is emerging. We propose trajectory estimation method suitable for MTBS which uses an adaptive polling scheme such that the polling is not frequent to overburden the air resources unnecessarily, and it is not sparse that leads to an ambiguous and erroneous mobility profile. The polling method includes a new map matching scheme using GIS (Geographical Information System) map. The scheme is based on previous location, map topology and travel time from one intersection to the other. So, from erroneous and moderately polled location points with time stamps, we form a spatiotemporal trajectory for a mobile user. A set of such trajectories for all days of the week would form a mobility profile, on which plethora of MTBS could be provisioned. We present a framework, formulation, and design for mobile user's trajectory. We ran trajectory simulation for an urban area of St. Paul, Minnesota and a suburban area of North Carolina. Our design worked very well in the urban area for location error of 100 meters or less. For the rural area, we ran simulation for location error up to 400 meters. The trajectory estimation error was very minimal.

**Keywords-component; mobile location; mobile trajectory; mobile trajectory-based Services; GIS; map matching; mobility profile.**

#### I. INTRODUCTION

The location specific services came to usage with the deployment of NAVSTAR GPS (Navigation Signal Timing and Ranging Global Positioning System), commonly called, Global Positioning Systems (GPS), by US Department of Defense in the 70's. The services were exclusively for military purposes. In 1984 US government granted GPS civilian access, however with an intentional error, called selective access. In May 2000 selective access was removed and the civilian are enjoying relatively lower error positioning systems [2]. Within the last few years there is a massive surge of interest in yet another positioning methodology and applications that could use the positioning information using cellular networks and handsets. The reasons for this surge is mainly because new wireless positioning technologies have the capability of alleviating

shortcomings of GPS, such as high power consumption and slow to acquire initial position and legislation in the United States requiring cellular phone operators to provide location information to 911 emergency call centers. A large proportion of the 911 calls come from cellular phones. The FCC mandate (docket 94-102) [3] requires mobile network operators to provide positional information to emergency services accurate within 125 meters. This regulatory requirement provided a stimulus for the commercial development of wireless positioning technologies and applications built on them, hence the birth of location based services (LBS). The location in this context is either an exact mobile location or covering a large area, cell or several cells referred to as location area. Mutual advances in location determination and geo-spatial technologies have provided a rich platform to develop mobile LBS [4]. Within the context of LBS, mobile wireless communication and geo-spatial database go hand in hand. The term LBS refers to mobile services in which the user's location is used in order to add value to the service as a whole. Such services are based on user's location in x-y coordinates (or location area) on a map, which is determined by different location determination technology, such as Time of Arrival (TOA), Time Difference Of Arrival (TDOA), Angle of Arrival (AOA), Time Advance (TA), Cell-ID, and Assisted GPS (A-GPS) [4][5]. A new trend of mobile wireless services based on the trajectory of a mobile user, called Mobile Trajectory Based Services (MTBS) is in its early research [6]. MTBS capitalize on the mobility profile of a mobile user. The mobility profile of a mobile user is a set of prerecorded trajectories of a mobile user that were captured by the mobile operator over a period of time, a week for example. In this paper we propose a method of estimating a mobile user's trajectory.

With GPS one has the luxury of numerous data points, since the location estimation takes place in the GPS receiver, so no extra air resources are used. Secondly GPS has incredible accuracy [7]. In case of mobile wireless, anomalies such as minor interference and reflection because of non line of sight or multipath result in erroneous time of travel estimation and hence a wrong distance. This leads to an erroneous location point determined by the underlying technology. For AOA, non line of sight causes wrong angle

measurement which results in erroneous location. Mobile wireless location estimation systems are prone to anomalies such as hearability of remote base stations, and geometric dilution of precision. Hence, the location coordinates estimated in mobile wireless system are erroneous; therefore we cannot trace the actual trajectory with simple interpolation methods. In mobile wireless the location error could be as large as 300 meters. Another important factor is the location polling frequency. A very frequent polling would potentially yield to a dense spatial distribution, obviously offset with error and could potentially yield a set of candidate trajectories closer in shape to the actual trajectory, however it is overburdening of the air channels and the location servers, which would affect the general operation of a wireless network. Higher polling frequency would also consume more energy on the mobile devices. On the other hand a very low frequency would lead to very ambiguous trajectory. Such trajectory would not resemble in shape with any of the paths taken from a GIS (Geographical Information System) map. If it does resemble to one, it would very likely be a wrong one. So, we need to poll enough for location estimation to yield us the trajectory that conserves underlying feature of the shape of the curve. Our assumption is that a mobile user travels on roads. Map matching techniques map the estimated raw location point to a point on the road network [8][9]. It can also map potentially the whole curve consisting of erroneous estimated location points to a curve consisting of a set of roads on GIS map which corresponds to the mobile user's true trajectory. It is called curve-to-curve matching [10]. Each road in GIS database is associated with direction and maximum speed limit [11]. Since there is an error,  $\epsilon$ , associated with estimated location, the mobile user could be on any road segment contained in a circle of radius  $\epsilon$  with center at the estimated location coordinates.

To analyze and formulate a mobile user's trajectory we define a framework of a metric space, we called it Mobile User space. Each estimated location corresponds to a closed ball; we refer to it as an error ball. A set of such embedded error balls forms a basis for the estimation of the desired trajectory. The trajectory so formed is used to map match curve-to-curve with the GIS database

The core of our map matching technique is based on adaptive location polling which is governed by the map topology and its constraints; we called it Intersection Polling Method (IPM). The idea behind IPM is to poll the mobile user's location, when the user is estimated to be at an intersection. Using the intersection points the mobile user's trajectory would be estimated with the least number of points. However, due to variations in the speed of a mobile user for different reasons and location error, we cannot determine exactly, when the mobile user is at an intersection every time. Therefore we may need to poll the mobile user's location more than one time from one intersection to another. Before the mobile user is polled for location, the previous erroneous location needs to be corrected and

candidate road segments need to be identified. The process of determination of candidate road segments referred to as geodesic candidates must be fast.

The roads are represented in GIS as polylines. So, essentially GIS database for a road network is a set of coordinate points. The estimated location is a point with some error. In our study we assume a uniform circular error distribution. We introduce conditions for geodesic candidacy in terms of the estimated location and the road segment vertices, i.e., the road network GIS database. Using these conditions the geodesic candidacy is established with very little processing, since it is merely a comparison. [12] provides candidacy conditions, however they are not in terms of data points that could be processed to establish candidacy before another location polling may be required specially when there are frequent turns. After we have a set of all the geodesic candidates, the trajectory is estimated by connecting the segments. In some cases there may be multiple geodesic candidates for a road segment. The most likely candidate is picked by applying directional and path constraints. We introduce a measure of directional negativity of a geodesic with respect to the direction of travel. The direction of travel is determined from the two consecutive estimated locations. A candidate geodesic with an angle difference of greater than  $\frac{\pi}{2}$  or less than  $-\frac{\pi}{2}$  with the direction of travel, is defined to have a directional negativity and therefore is eliminated. Path constraints from the previous corrected location to the recent corrected location on the candidate geodesic further eliminates the unrealistic geodesics. In case there is at least one instance where we still have multiple geodesic candidates for the final trajectory, a curve-to-curve matching is performed on each candidate path with the path of the estimated location points. The candidate path that resembles the most with that of the estimated location points is selected as the path of the mobile travel for that particular segment of the mobile trajectory.

To run simulation on our proposed method we used GIS data [13] of metropolitan area of St. Paul, Minnesota and that of a suburban area of North Carolina. We developed a software tool, Digitizer, using C Sharp in Microsoft Windows to convert images of the paper maps of the areas into GIS database. The database schema of Digitizer is similar to what is widely used for road network [14]. The proposed mobile trajectory estimation method was implemented in Microsoft Windows as well. The proposed method performed very well with the exception of a scenario of very large location error in dense metropolitan area. In rural area it performed flawlessly even with larger errors.

This paper is an extended version of a conference paper referenced in [1]. It is organized as follows. Section II outlines how road networks have been represented in the literature and describes our road network model and its representation. Section III lays out mobile trajectory

estimation framework. It covers in details our metric space framework; we called it Mobile User space. Erroneous mobile location correction, location polling scheme and topological constraints are discussed in this section. Section IV covers the detailed steps to estimate the trajectory, including a flow chart for the algorithm. In Section V we describe the implantation of the proposed method, including details of Digitizer and our mobile trajectory estimation application, Mobility Profiler. Section VI is dedicated for simulation results. The paper concludes in Section VII.

## II. ROAD NETWORK REPRESENTATION

We are interested in the mobility profiling of a mobile user who traverses on roads. This leaves us with the road network as the natural container of a mobile user. A theoretical framework for the mobility profiling is closely tied to the road network's representation. A variety of road network representation schemes have been proposed, each discussed in a different framework. A brief description of some popular schemes is given below:

1) Steering direction summarizes the segmentation data by a steering direction, independent of the actual road geometry. This is the approach taken in the ALVINN (Autonomous Land Vehicle in a Neural Network), a neural net road follower [15]. In principle, ALVINN could learn appropriate steering commands for roads which change slope, bank, etc. In practice, images are backprojected onto a flat ground plane and re-projected from different points of view to expand the range of training images. This may prove to be a limiting factor on hilly roads.

2) Linear on a locally flat ground plane road model has three parameters, the road width and two parameters describing the orientation and offset of the vehicle with respect to the centerline of the road. LaneLok and Scarf algorithms take this approach [16]. The main limit of this type of scheme is the need to move a sufficiently small distance between road parameter estimates so that the straight path being driven along does not diverge too much from the actual road.

3) Modeling the road as a cross-section swept along a circular arc explicitly models road curvature but retains the flat earth assumption used in linear models. VaMoRs (Universität der Bunderwehr München's autonomous road vehicle) [17], and YARF (Yet Another Road Follower) [18] use this approach. The equations describing feature locations can be linearized to allow closed form least squares solutions for the road heading offset, and curvature, as well as the relative feature offsets.

4) A more general model of road geometry retains the flat earth assumption, but requires only that road edges to be locally parallel, allowing the road to bend arbitrarily. This can be done by projection onto the ground plane, VITS (Vision system for autonomous land vehicle navigation) system [19], or in the image plane. The lack of higher order constraint on the road shape can lead to serious errors in the

recovered road shape when there are errors in the results of the underlying image segmentation techniques.

Several algorithms have been developed to recover three dimensional variations in road shape under the assumption that the road does not bank. These current algorithms use information from a left and right road edge, which precludes integrating information from multiple road markings. Such algorithms may be very sensitive to errors in feature location by the segmentation processes. This is due to the assumption of constant road width, which leads to errors in road edge location being interpreted as the result of changes in the terrain shape. Circular arc models would appear to be the technique of choices in the absence of algorithms for the recovery of three dimensional road structures which are robust in the presence of noise in the segmentation data. They have a small number of parameters, they impose reasonable constraints on the overall road shape, and statistical methods can be used for estimating the shape parameters, with all the statistical theory and tools that use of such methods allows the system to apply to the problem.

We are most interested in road network representation as described in GIS database. This road network representation scheme is similar to 2), mentioned above. Our road representation, also referred as road network element in our work, is described in the following subsection.

### A. Road Network Element

We represent a road as piece-wise linear curve also known as polyline. The curve is intersected by other road segments to realize intersections of different shapes: cross, Y, T etc. The other parameters associated with roads, we are interested in our work, are: direction and posted speed limits. Our road representation is depicted in Figure 1.

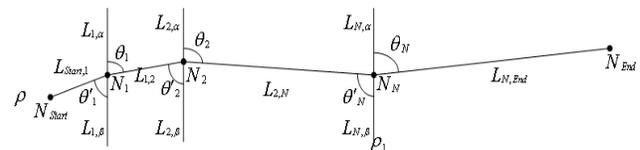


Figure 1. Road network element model.

The above model is a generic representation of roads of all shapes. A road,  $\rho$ , starts from point  $N_{Start}$  and terminates at  $N_{End}$ . These points are referred to as start node and end node respectively. A road may be intersected by other roads,  $L_{j,\alpha}, L_{j,\beta}, j = 1, 2, 3, \dots, N$ . at  $N_i, i = 1, 2, 3, \dots, N$ . In most of the cases  $L_{j,\alpha} \equiv L_{j,\beta}$ , i.e., both the segments are part of the same intersecting road network element for which  $\rho$  is an intersecting road element. The intersecting roads intersect with  $\rho$  at  $N_i, i = 1, 2, 3, \dots, N$ , making angles  $\theta_i, \theta'_i$  with  $L_{j,\alpha}$  and  $L_{j,\beta}$  respectively. In most of the cases, for which

$L_{j,\alpha} \equiv L_{j,\beta}$ , implies  $\theta_i \equiv \theta'_i$ . The number of intermediate nodes  $N \geq 0$ , i.e., intersections, is arbitrary.

**B. Road Network Database**

As mentioned in the section above, our road network representation has the elements as described in Tables I and II. We store the roads in a database (GIS database) in the following format. Road ID is a distinct number assigned to different roads. The whole segment of the polyline representing that particular road has a starting point and end point. The order of the starting and end points codes the direction of traffic on the road. The two way roads are flagged in the database as bi-directional.

TABLE I. PARAMETERS OF A ROAD ELEMENT

ROAD ID	START POINT	END POINT	INTERSECTIONS	POSTED SPEED
$N$	$N_{Start}(x_0, y_0)$	$N_{End}(x_p, y_p)$	$(x_1, y_1) \dots (x_N, y_N)$	S

TABLE II. END TO END DEFINITION OF A ROAD ELEMENT

ROAD ID	POINTS COORDINATES	IS ROAD ELEMENT DIRECTIONAL
$N$	$(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)$	False/True

The intersections are represented as Cartesian coordinates of a road segment where it intersects or meets another road element. For mobility profiling technique we propose, these points are pivotal in determining rather calibrating the exact shape of the mobile user's trajectory. The posted speed which is a very important weight of the road network also plays a very important role in determining the mobile trajectory. This is the speed associated with the road segment as posted by department of transportation. All digital representation of a road carries this weight as one of the most important parameter. We shall discuss the usage of both the speed and intersection in details in mobility profiling section.

The detailed geographical definition of a road is simply represented as ordered pairs of shape determining points. For example a straight road may be represented just by two points whereas a curved road might take numerous points to represent it reasonably. The precision is chosen at the time of digitization of the map. The biggest cost of precision is the database memory. We shall discuss digitization in details in Section V.

**III. MOBILE TRAJECTORY ESTIMATION FRAMEWORK**

Essentially, a road network is stored as Cartesian coordinate ordered pairs of points. These points are finite and have a notion of distance, Euclidean or otherwise, as a metric between them. Mobile location is represented by a point or a collection of potential location points (in an erroneous environment). These location points are associated to a mobile user traveling on a road network which is also represented by a collection of ordered points. All these points, both for location representation and road network representation, have a notion of distance, ruling over the points, therefore a metric space is a natural framework for

the theory and analyses of mobile trajectory formation. It has been established in literature that the language in which a large body of ideas and results of functional analyses are expressed is that of the metric spaces [20].

In road networks the distance between any two points  $P_i(x_i, y_i), P_j(x_j, y_j)$  is not necessarily Euclidian. The distance is Euclidian only if the points are connected on the same polyline. The metric (distance) for road network is the length of the polyline line from  $P_i(x_i, y_i)$  to  $P_j(x_j, y_j)$  as shown in Figure 2.

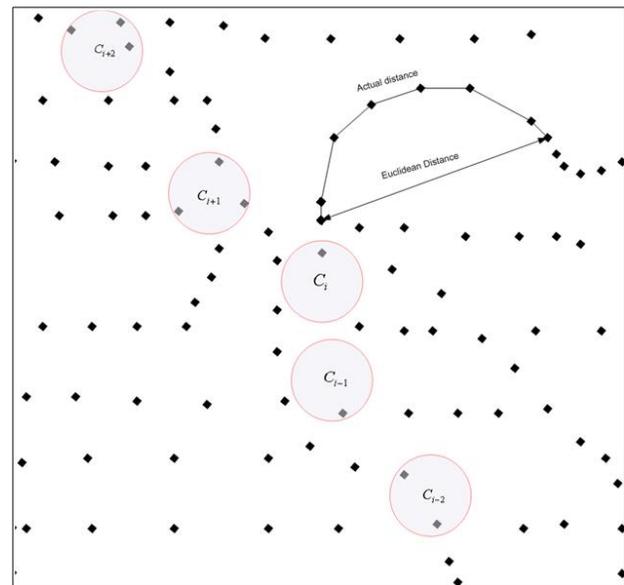


Figure 2. A logical representation of topology of metric space. The distance is through the connected points.

Definition 1: For a metric space  $(\mathbf{X}, d)$ , a geodesic path joining  $x \in \mathbf{X}$  to  $y \in \mathbf{X}$  (or more briefly, a geodesic from  $x$  to  $y$ ) is a map  $c$  from a closed interval  $[0, l] \subset \mathbb{R}$  to  $\mathbf{X}$  such that  $c(0) = x$ ,  $c(l) = y$  and  $d(c(t), c(t')) = |t - t'|$   $\forall t, t' \in [0, l]$  (in particular,  $l = d(x, y)$ ). The image  $\alpha$  of  $c$  is called a geodesic segment with end points  $x$  and  $y$ .

Definition 2: A geodesic space is a metric space wherein any two points are joined by a geodesic.

Definition 3: A metric space  $(\mathbf{X}, d)$  is a length space if for every  $x, y \in \mathbf{X}$ ,  $d(x, y) = \inf_{\gamma} L(\lambda)$ , where  $\lambda$  is a rectifiable curve between  $x$  and  $y$ .

Let  $P_L \equiv v_0(x_0, y_0), v_1(x_1, y_1), \dots, v_n(x_n, y_n)$  defines a polyline with  $n+1$  vertex. The polyline has  $n$  edges.

$$P_L = \begin{cases} [(y_1 - y_0)x + (y_0x_1 - x_0y_1)] / (x_1 - x_0) : & x_0 \leq x \leq x_1 \\ [(y_2 - y_1)x + (y_1x_2 - x_1y_2)] / (x_2 - x_1) : & x_1 \leq x \leq x_2 \\ \vdots & \\ [(y_n - y_{n-1})x + (y_{n-1}x_n - x_{n-1}y_n)] / (x_n - x_{n-1}) : & x_{n-1} \leq x \leq x_n; \quad x_{i+1} \neq x_i \end{cases} \quad (1)$$

Please note  $x_{i+1} = x_i$  is a special case where the segment is,  $\{(x_k, y_k) \mid y_i \leq y_k \leq y_{i+1}; x_k = x_i = x_{i+1}\}$ ,

It is customary to use parametric representation of polylines: The parametric representation of (1) is:

$$P_L = \begin{cases} \frac{(t_1 - t)v_0 + (t - t_0)v_1}{t_1 - t_0} : & t_0 \leq t \leq t_1 \\ \frac{(t_2 - t)v_1 + (t - t_1)v_2}{t_2 - t_1} : & t_1 \leq t \leq t_2 \\ \vdots \\ \frac{(t_n - t)v_{n-1} + (t - t_{n-1})v_n}{t_n - t_{n-1}} : & t_{n-1} \leq t \leq t_n \end{cases} \quad (2)$$

where,  $|t_i - t_{i-1}| \neq 0$ , corresponds to  $|v_i - v_{i-1}|$ ,  $\forall i$  as shown in Figure 3. Let  $c: [a, b] \rightarrow V$  represents the polyline in interval  $[a, b]$  and  $t_0 = a < t_1 < t_2 < \dots < t_n = b$ , the polygonal length of the polyline curve or the distance covered in  $|t_n - t_0|$  is:

$$P_L(c, \{t_0, t_1, \dots, t_n\}) = \sum_{i=0}^n d(c(t_i), c(t_{i-1})). \quad (3)$$

Please note the polyline representation of a road is an approximation of a smooth curve which does not necessarily have these edges. The edges are referred as geodesics of the road representation. A road network, e.g., shown in Figure 3 is a set,  $\mathbf{M} = \{P_{Lj} : j = 1, 2, \dots, N\}$  that represents a road network of  $N$  polylines. Since our road element is a 2-dimensional model the information of intersections of the roads is not necessarily inclusive. The road may pass over another road without any intersection. Therefore connectivity matrix must accompany the set  $\mathbf{M}$  for complete representation of a road network. Figure 4 is a representation of the actual road network. It is a graphical view of actual GIS database of an urban map.

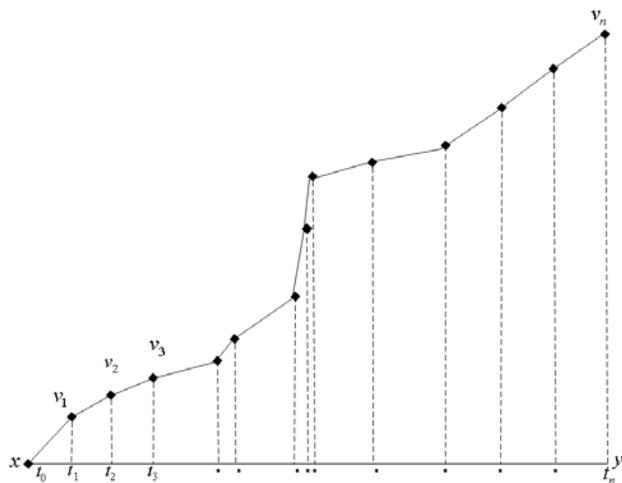


Figure 3. A parametric representation of a road element in topological space

Let  $v_i(x_i, y_i)$  and  $v_{i+1}(x_{i+1}, y_{i+1})$  be the end points on a geodesic and,

$$\mathbf{G}_i = \left\{ p(x, y) \mid y = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} x + \frac{y_i(x_{i+1} - x_i) + x_i(y_i - y_{i+1})}{x_{i+1} - x_i}, \right. \\ \left. \frac{x_i}{x_{i+1}} \leq \frac{x}{x_{i+1}} \leq 1; \frac{y_i}{y_{i+1}} \leq \frac{y}{y_{i+1}} \leq 1; x_{i+1} \neq x \right\} \quad (4)$$

In other words  $\mathbf{G}_i$  contains all the points of a geodesic between two arbitrary vertices of a digital map, i.e., GIS database of the road network.  $x_{i+1} = x_i$  is a special case, as explained before. From this point on this special case shall be implied whenever we state  $\mathbf{G}_i$ .

Let  $N$  be the total number of geodesics in a digital map,  $\mathbf{G} = \bigcup_{i=1}^N \mathbf{G}_i \subset \mathbf{R}^2$  is set of all permissible points on a road network map. Let  $d$  be a metric of distance between two locations  $P_1$  and  $P_2$  on the road network map. For all  $x, y \in \mathbf{G}$ ,  $d(x, y) = \inf_{\gamma} L(\lambda)$ , where  $\lambda$  is rectifiable curve between  $x$  and  $y$ .

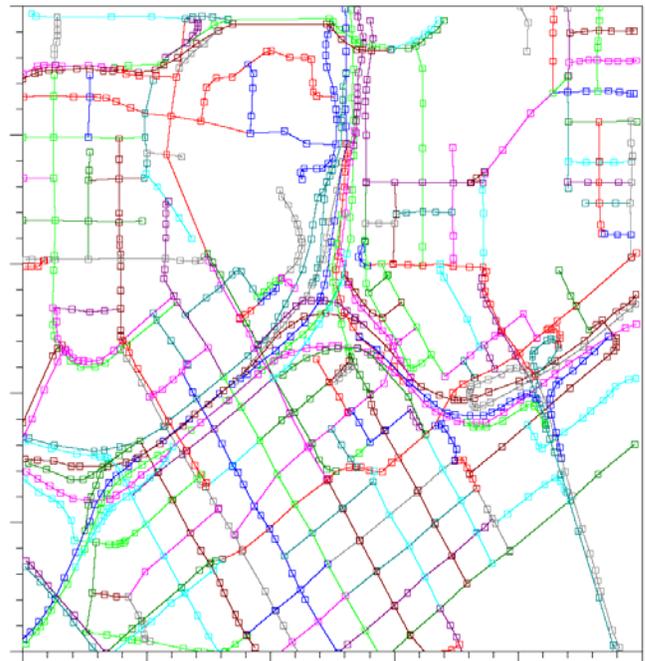


Figure 4. Topological representation of road map of an urban area of St. Paul, Minnesota. Roads are represented by connected and intersecting polylines. Polylines are expressed in vertices shown as squares.

We are interested in the analysis of  $(\mathbf{G}, d)$  from metric space point of view. We call  $(\mathbf{G}, d)$  Mobile User space. A Mobile User space,  $(\mathbf{G}, d)$ , is a non-symmetric metric space because  $d(x, y) \neq d(y, x)$  always.

**Lemma 1:** A Mobile User space  $(\mathbf{G}, d)$ , is not a geodesic space.

**Proof:** Let  $x, y \in \mathbf{G}$ ,  $d(x, y) = \|x, y\|$ , where  $\|\cdot\|$  is Euclidian distance, iff  $x$  and  $y$  belong to the same geodesic. But  $x, y$  are arbitrary points,

$\therefore d(x, y) \neq \|x, y\|$  for any arbitrary points  $x, y \in \mathbf{G}$ . Hence  $(\mathbf{G}, d)$  is not a geodesic space.

*A. Closed Ball Embedded in Mobile User Space*

In all mobile location technologies, the estimated location is not an exact point. If exact location coordinates are claimed there is always an error associated with it. There are many factors contributing the errors. No single technology can provide a pin point location of the mobile user using the existing infrastructure of mobile systems. Instead of an exact location of a mobile user we know the feasible location area of the mobile user. One cannot come up with the exact shape of the feasible area; however there is always an upper bound on the exact area of the region where a mobile user is located. In our research we model this feasible area with a circle. The location probability density is constant throughout the circle. In some research work, a model with Gaussian distribution around the center of the circle has been used [21]. In real world scenario such model would lead to erroneous results when dealing with identifying the several candidate roads a mobile user is potentially on, which pass through the feasible location area. Different location estimation technologies have different accuracy associated with it. Assisted-GPS technique is the most accurate among all, TOA and Time Advance being among the least. We shall represent them with circles of corresponding error radius. When a circular feasible area is considered on a road map (mobile user space), this leads to a topology where a ball is embedded in a mobile user space. We shall discuss and analyze this topological space in details and derive some results, to be used in the grand scheme of mobility profiling. In this subsection we define the key terms used in our framework and formulate geodesic candidacy. This will cover how candidate road segments are selected.

Let  $(\mathbf{G}, d)$  be a mobile user space and a ball  $\mathbf{B}_\mu(x_0) = \{\mathbf{x} \in X : d(\mathbf{x}, x_0) \leq \mu\}$  being embedded in this metric space. Here  $X$  is a universal set. The ball,  $\mathbf{B}_\mu(x_0)$ , is essentially the feasible circular location area with center at  $x_0$  and radius equal to  $\mu$ , estimated by a mobile wireless system. When a request for location estimation is made to the cellular system it shall return the location as,  $x_0$ .

**Location Error Proposition:** The embedding ball,  $\mathbf{B}_\mu(x_0)$  intersects or contains at least one geodesic of  $\mathbf{G}$ , i.e.,  $\mathbf{B}_\mu(x_0) \cap \mathbf{G} \neq \emptyset$ .

**Proof:** Let  $\mathbf{G}$  contain  $N$  geodesics  $\Rightarrow \mathbf{G} = \bigcup_{i=1}^N \mathbf{G}_i$ . Where,

$$\mathbf{G}_i = \{p(x_i, y_i) \mid y_i = mx_i + b; u_k \leq x_i \leq u_{k+1}; v_k \leq y_i \leq v_{k+1}\}$$

Let  $\mathbf{G}_l = \bigcup_{i=1}^l \mathbf{G}_i$  be the geodesics surrounding the ball,

$\mathbf{B}_\mu(x_0) = \{\mathbf{x} \in X : d(\mathbf{x}, x_0) \leq \mu\}$ . If  $x_l \in \mathbf{G}_l$ , by definition of the location error,  $d(x_l, x_0) \leq \mu$ ,  $\Rightarrow x_l \in \mathbf{G}_l$

$$\therefore \mathbf{G}_l \subseteq \mathbf{G}, x_l \in \mathbf{B}_\mu(x_0) \cap \mathbf{G}$$

Hence  $\mathbf{B}_\mu(x_0) \cap \mathbf{G} \neq \emptyset$ .

We are interested in conditions imposed on geodesics  $\mathbf{G}_i$  for passing through the ball. The reason of the investigation is to map the erroneous location to the nearby road. Map matching corrects the location of a mobile user by projecting the estimated location to a road which user is most likely on.

Let  $\mathbf{V} = \{v_i : i = 1, 2, \dots, N + 1\}$  be the vertices of  $N$  polylines in  $\mathbf{G}$  and  $\overset{\circ}{E}$  be the error in location estimation, the associated ball, let us call it error ball from this point on, is  $\mathbf{B}_E^\circ(x_0) = \{\mathbf{x} \in X : d(\mathbf{x}, x_0) \leq \overset{\circ}{E}\}$ . We shall define two important categories of geodesics:

**Definition 1:** A geodesic  $\mathbf{G}_i$ , as in (4), of any two vertices,  $v_i(x_i, y_i), v_{i+1}(x_{i+1}, y_{i+1})$ , of a road, is said to be a candidate geodesic if it touches any point of the error ball, i.e.,  $\mathbf{G}_i \cap \mathbf{B}_E^\circ(x_0) \neq \emptyset$ .

**Definition 2:** A geodesic,  $\mathbf{G}_i$ , of any two vertices,  $v_i(x_i, y_i), v_{i+1}(x_{i+1}, y_{i+1})$ , of a road, whose straight line extension,

$$\mathbf{L}_i = \left\{ p(x, y) \mid y = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} x + \frac{y_i(x_{i+1} - x_i) + x_i(y_i - y_{i+1})}{x_{i+1} - x_i}; x_i \neq x_{i+1} \right\}, \quad (5)$$

touches any point of the error ball, is said to be pseudo candidate geodesic, i.e.,  $\mathbf{L}_i \cap \mathbf{B}_E^\circ(x_0) \neq \emptyset$ .

**First Condition of Geodesic Candidacy:** A simple and trivial geodesic candidacy condition is:  $\mathbf{W} \cup \mathbf{B}_E^\circ(x_0) = \mathbf{B}_E^\circ(x_0)$ , where  $\mathbf{W} = \{v_j\}$  for some  $j$ .

Please note  $\mathbf{W} = \{\emptyset\}$  is not a valid scenario. If a vertex lies within the error ball the corresponding geodesic shall be a

candidate geodesic, however a geodesic may pass through the error ball but the vertices could lie outside the error ball. For the study of geodesic candidacy when the vertices are outside of the error ball we would need to study pseudo geodesic candidacy.

**Pseudo Geodesic Candidacy Theorem:** In a mobile user metric space,  $(\mathbf{G}, d)$ , the necessary and sufficient condition for pseudo geodesic candidacy of any arbitrary vertices  $v_i(x_i, y_i), v_{i+1}(x_{i+1}, y_{i+1}) \in \mathbf{B}'_E(\mathbf{x}_0)$ , for an error

ball,  $\mathbf{B}_E(x_0) = \{\mathbf{x} \in X : d(\mathbf{x}, x_0) \leq \overset{\circ}{E}\}$  is

$$\left| \frac{y_r(x_{i+1} - x_i) - x_r(y_{i+1} - y_i) + x_i y_{i+1} - x_{i+1} y_i}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \right| \leq \overset{\circ}{E}, \quad (6)$$

where  $x_0(x_r, y_r)$  is the center of the circle, i.e., the estimated location of a mobile user and  $\bar{\mathbf{W}} = \mathbf{B}'_E(x_0)$ .

**Proof:** Let  $v_i \equiv (x_i, y_i)$  and  $v_{i+1} \equiv (x_{i+1}, y_{i+1})$  be the vertices of a polyline such that  $v_i, v_{i+1} \in \mathbf{W}$  make a pseudo geodesic candidate.  $\overline{KL}$  in Figure 5 is such an example.

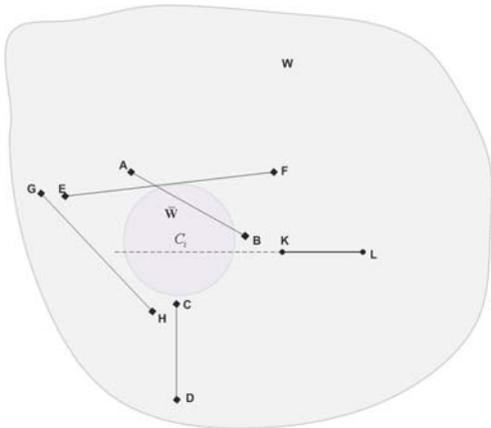


Figure 5. Straight lines through some vertices outside the error ball may pass through it

It follows from Definition 2 that a geodesic is a pseudo candidate if and only if its straight line extension,  $\mathbf{L}_i$ , given in (5) and  $C_E(\mathbf{x}_0) = \{\mathbf{x} \in X : d(\mathbf{x}, \mathbf{x}_0) = \overset{\circ}{E}\}$  have a solution

$$\{p_1, p_2\} \equiv \{A \pm Q, B \pm mQ\}, \text{ where } A \text{ and } B \text{ are constants and } Q = \frac{\sqrt{\overset{\circ}{E}^2 - [(y_r - mx_r - b) / \sqrt{1+m^2}]^2}}{\sqrt{1+m^2}}. \quad (7)$$

Here, and throughout this paper,  $m$  represents the slope of a geodesic and  $b = \frac{y_i(x_i - x_{i+1}) + x_i(y_{i+1} - y_i)}{x_i - x_{i+1}}$ .

For the solution,  $\{p_1, p_2\}$ , to be real, the discriminant  $\overset{\circ}{E}^2 - [(y_r - mx_r - b) / \sqrt{1+m^2}]^2$ , in (7) must be non-negative.

$$\begin{aligned} \therefore \overset{\circ}{E}^2 - \left[ \frac{y_r(x_{i+1} - x_i) - x_r(y_{i+1} - y_i) + x_i y_{i+1} - x_{i+1} y_i}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \right]^2 &\geq 0 \\ \Rightarrow \left| \frac{y_r(x_{i+1} - x_i) - x_r(y_{i+1} - y_i) + x_i y_{i+1} - x_{i+1} y_i}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \right| &\leq \overset{\circ}{E} \end{aligned}$$

**Corollary 1:** For a vertical geodesic the pseudo candidacy requirement is  $|x_i - x_r| \leq \overset{\circ}{E}$ . The corollary easily follows by putting the vertical condition:  $x_i = x_{i+1}$  in the above condition.

**Corollary 2:** For a horizontal geodesic the pseudo candidacy requirement is  $|y_r - y_i| \leq \overset{\circ}{E}$ . The corollary easily follows by putting the vertical condition:  $y_i = y_{i+1}$  in the above condition.

*B. Estimated Location Correction within an Error Ball*

Once the candidate geodesics are identified, the estimated erroneous location can be corrected by mapping it to the candidate geodesics. The estimated location is presented as an error ball,  $\mathbf{B}_E(x_0) = \{\mathbf{x} \in X : d(\mathbf{x}, \mathbf{x}_0) \leq \overset{\circ}{E}\}$ , with

center  $x_0(x_r, y_r)$  and feasible area of  $\pi \overset{\circ}{E}^2$ . The location coordinates are the center of the error circle,  $x_0$ . Since a mobile user traverses on road, the feasible areas transform to feasible locations on the candidate geodesic contained in the error ball.

The feasible location now is bounded by the lines  $\mathbf{L}_i$  within the error circle. The feasible location is given by,  $\bigcup_{i=1}^l \mathbf{L}_i = \mathbf{B}_E(\mathbf{x}_0) \cap \left( \bigcup_{i=1}^l \mathbf{L}_i \right)$ , where  $i = 1, 2, 3, \dots, l$  and  $l$  is the number of candidate geodesics of the error ball.

If  $v_i(x_i, y_i), v_{i+1}(x_{i+1}, y_{i+1}) \notin \mathbf{B}_E(\mathbf{x}_0)$ , we shall use the center of the line segment,  $\mathbf{L}_i$  as the corrected location on  $\mathbf{L}_i$  since that is the average of all the possible location points on the line segment,  $\mathbf{L}_i$ . The corrected location is

essentially projection of the center of the error ball on the line segment,  $\mathbf{L}_i$ .

To find the corrected location we solve the line,  $(y_{i+1} - y_i)x + (x_i - x_{i+1})y + x_{i+1}y_i - x_iy_{i+1} = 0$ , passing through the vertices,  $v_i(x_i, y_i)$ ,  $v_{i+1}(x_{i+1}, y_{i+1})$  and the normal,  $(x_i - x_{i+1})x + (y_i - y_{i+1})y + x_r(x_{i+1} - x_i) + y_r(y_{i+1} - y_i) = 0$ , passing through the center,  $x_0(x_r, y_r)$ , of the error ball.

The corrected location is:

$$P_c(x_c, y_c) = \left( \frac{x_r + m(mx_i - y_i + y_r)}{m^2 + 1}, \frac{y_i + m(my_r - x_i + x_r)}{m^2 + 1} \right). \quad (8)$$

In case the vertices of the polyline lie within the error ball, i.e.,  $v_i(x_i, y_i) \in \mathbf{B}'_E(\mathbf{x}_0)$  for  $i = 1, 2, 3, \dots, k$ , then the corrected location is the vertex  $v_i$  for which,  $\|x_0 - v_i\|$  is minimum.

**Lemma 2:** In a mobile user metric space,  $(\mathbf{G}, d)$ , if a geodesic of vertices  $v_i(x_i, y_i), v_{i+1}(x_{i+1}, y_{i+1}) \in \mathbf{B}'_E(\mathbf{x}_0)$ ; and their corresponding geodesic satisfies pseudo candidacy as given in (6) and  $0 \leq (x_{i+1} - x_i)(x_r - x_i) + (y_{i+1} - y_i)(y_r - y_i) \leq D^2$ , where  $D$  is the Euclidean distance between the vertices, the geodesic is a candidate geodesic.

**Proof:** Let  $v_i(x_i, y_i)$  and  $v_{i+1}(x_{i+1}, y_{i+1})$  be the vertices of a polyline such that  $v_i, v_{i+1} \in \mathbf{W}$  make a geodesic candidate.  $\overline{AB}$ , and  $\overline{EF}$  in Figure 5 are such examples. Let  $x_0(x_r, y_r)$  be the center of the error ball.

The corrected location,  $P_c(x_c, y_c)$ , of the estimated location, i.e., the projection of the center of the error ball onto the geodesic is as shown in (8) above.  $P_c(x_c, y_c)$ ,  $v_i(x_i, y_i)$ , and  $v_{i+1}(x_{i+1}, y_{i+1})$  are collinear points. Let  $P_c(x_c, y_c)$  divides the geodesic with ratio  $\frac{Q}{D} : 1$  as shown in Figure 6.

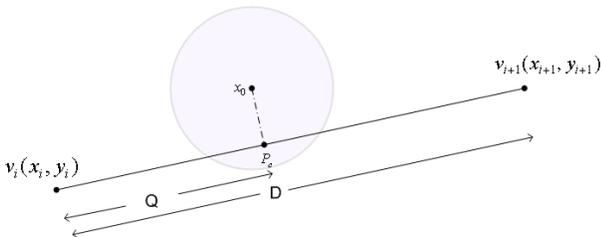


Figure 6. The corrected location must lie between the two vertices for the geodesic to be candidate geodesic

$$\therefore Q = \frac{x_c D - D x_i}{x_{i+1} - x_i} = \frac{y_c D - D y_i}{y_{i+1} - y_i}. \text{ For the geodesic to be the}$$

candidate geodesic,  $Q$  must be positive and less than  $D$ . For  $Q = D$  or  $Q = 0$ , the geodesic is tangent to the error circle. For negative value of  $Q$ , the geodesic shall be a pseudo candidate or may not intersect the error circle at all. So, for the geodesic to be a candidate geodesic the following condition must meet:

$$0 \leq Q \leq D \text{ where } D = |v_{i+1} - v_i|$$

$$\begin{aligned} \frac{x_c D - x_i D}{x_{i+1} - x_i} &= \frac{x_r + m(mx_i - y_i + y_r)}{\sqrt{m^2 + 1}} - x_i \sqrt{m^2 + 1} \\ &= \frac{(x_r - x_i) + m(y_r - y_i)}{\sqrt{m^2 + 1}} \end{aligned}$$

From the candidacy condition,

$$\begin{aligned} 0 \leq \frac{(x_r - x_i) + m(y_r - y_i)}{\sqrt{m^2 + 1}} \leq D \\ \Rightarrow 0 \leq \frac{(x_r - x_i) + \left(\frac{y_{i+1} - y_i}{x_r - x_i}\right)(y_r - y_i)}{\sqrt{\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)^2 + 1}} \leq D \\ \Rightarrow 0 \leq (x_{i+1} - x_i)(x_r - x_i) + (y_{i+1} - y_i)(y_r - y_i) \leq D^2 \quad (9) \end{aligned}$$

Here, the candidacy condition is proved using

$$\begin{aligned} Q &= \frac{x_c D - D x_i}{x_{i+1} - x_i}. \text{ We would get the same result for} \\ Q &= \frac{y_c D - D y_i}{y_{i+1} - y_i}, \text{ as well.} \end{aligned}$$

**Corollary 3:** For a vertical geodesic the candidacy requirement is,  $0 \leq (y_{i+1} - y_i)(y_r - y_i) \leq D^2$ . The corollary easily follows by putting the vertical condition:  $x_i = x_{i+1}$  in the above condition.

**Corollary 4:** For a horizontal geodesic the candidacy requirement simplifies to,  $0 \leq (x_{i+1} - x_i)(x_r - x_i) \leq D^2$ . The corollary easily follows by putting the horizontal condition:  $y_i = y_{i+1}$  in the above condition.

There are several different approaches proposed in literature to identify the candidate road segments from the digital map against an estimated location. Most of them deal with GPS or GPS integrated with dead reckoning and hence are suitable only under those environments. One of the benefits of GPS from identifying a candidate road segment point of view is its known feasible area. Several methods are available to determine the feasible area. Variance-covariance associated with GPS receiver output has been used to define the feasible area as ellipse with known distribution function [22][23]. Secondly, the error is small and symmetrical;

therefore inherently it is less complex to identify the candidate road segments as compared to mobile network. Segment based methods where the heading of the vehicle from GPS points is compared with the road segments in the digital map for maximum parallelism. The road segments that are along the same direction as the line joining the GPS points are marked as the candidate road segments. This method has been used along with other measures [24][25] and as such [26]. Because of the small error and error symmetry this measure is considered quite reliable. However this is not suitable for mobile network, where the error is relatively large and not symmetrical. For example consider Figure 7. A vehicle moves from estimated location  $L_1$  to  $L_2$  in a mobile network. Clearly segment  $s_2$  is the candidate road segment because  $s_2$  passes through both the location feasible areas, however using direction similarity, road segment  $s_1$  is picked wrongfully. Other disadvantage of this method in mobile system is that we need to have two locations polled very close to one another to identify the candidate road segments.

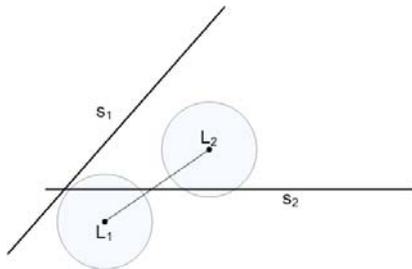


Figure 7. Vehicle heading from  $L_1$  to  $L_2$  wrongfully matches the road segment  $s_1$

[27] identifies the candidate segment from projection of the estimated location on to the road segments in the digital map. Let  $S$  and  $E$  be the two vertices (also referred to as shaping points) of a geodesic and  $G$  be the estimated location point. Define

$$K = \frac{\overline{SG} \cdot \cos \theta}{|\overline{SE}|} = \frac{\overline{SG} \cdot \overline{SE}}{|\overline{SE}|^2} \quad (10)$$

If  $0 \leq K \leq 1$ , road segment  $\overline{SE}$  (Figure 8) is selected, if  $K < 0$ , the road segment preceding the node  $S$  is selected, if  $K > 1$  the road segment following the node  $E$  is selected. The drawback with this method is that it does not take into account the error region. Let us consider a comparison in Figure 8.

It presents two scenarios. In scenario 1, the selected road segment is a valid one, whereas in scenario 2, the selected road segment is not a valid candidate segment contrary to the proposed condition. The candidate selected in scenario 2 is definitely related to some other error circle, since  $\overline{SE}$  does not even touch the error circle. Using this method erroneous

geodesic shall be identified as candidates. This method is only valid for identifying a candidate road segment among the road segments that are fully or partially contained within the GPS feasible area.

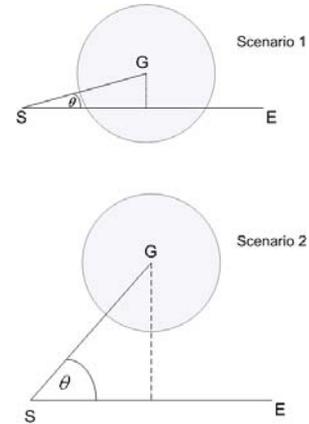


Figure 8. In both the projection scenario  $\overline{SE}$  is selected as the candidate segment

[28] identifies the above mentioned flaw in general in finding the candidate geodesic when the vertices of geodesic (or arc in GPS literature) are outside the feasible area. It proposes this problem as future research to address such issue. [12] proposes a method that is along the line of proposed research.

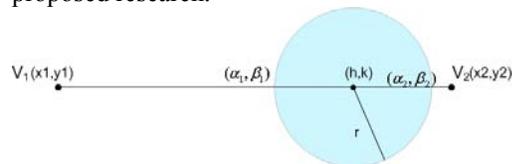


Figure 9.  $\overline{V_1V_2}$  is a candidate geodesic in an error circle of radius  $r$  centered at  $(h, k)$

The proposed conditions to identify a candidate geodesic are as follow:

First find the solution  $(\alpha, \beta)$  for the equation of the error circle,  $(x - h)^2 + (y - k)^2 = r^2$  and the straight line  $\overline{V_1V_2}$ , Figure 9. Here,

$$\alpha(\alpha_1, \alpha_2) = \frac{m^2k + h - mc}{1 + m^2} \pm \frac{\sqrt{r^2 - g^2}}{\sqrt{1 + m^2}}, \quad (11)$$

$$\beta(\beta_1, \beta_2) = \frac{m^2k + mh + c}{1 + m^2} \pm \frac{m\sqrt{r^2 - g^2}}{\sqrt{1 + m^2}}, \quad (12)$$

where,  $m = \frac{\Delta y}{\Delta x}$ ,  $c = \frac{x_1 y_2 - y_1 x_2}{\Delta x}$  and  $g = \left( \frac{k - mh - c}{\sqrt{1 + m^2}} \right)$

The conditions for candidacy are run online between polls. The calculations must be fast with as little run time as possible. Secondly, depending upon the size of the digital map and the search method, there could be very large

number of comparisons and processing required before the next poll. The author proposed for further research to identify candidate segments without solving (11) and (12). Our method identifies candidate geodesics without going through the process of finding  $(\alpha, \beta)$  therefore cutting significant processing. Secondly, the conditions do not specify which particular solution point,  $(\alpha_1, \beta_1)$  or  $(\alpha_2, \beta_2)$  to be used for verification. The results would be different for different solution points. This is extremely important from implementation point of view.

### C. Discrete Error Balls

The simplest way to form a trajectory is by interpolating the location points collected between the start point and the destination. If we have frequent locations with very low error, hence a large number of points to interpolate, we may potentially come up with a trajectory close to the actual path. However this paper is on forming a trajectory with optimal number of points. As a matter of fact, our trajectory stems from sequence of discrete error balls. How far apart and frequent the error balls are, corresponds to the polling frequency. This is the frequency with which the mobile user's location is estimated in terms of error ball. The center of the error ball is the estimated location. The polling frequency is directly related to the topology and geometry of the road network the mobile user is traveling on. If a road network is dense and there are numerous intersections, we would need more embedding error balls to estimate the trajectory of a mobile user. On the other hand if roads are sparse with less intersection, for example rural areas, we need fewer error balls to form a trajectory.

Let an error ball be located at  $C_i$  at time  $\tau_i$ . The next location of the error ball,  $C_{i+1}$ , will be estimated at time,  $\tau_{i+1}$ , using a method, we called it Intersection Polling Method (IPM). Let  $\mathbf{B}_E(C_i) = \{\mathbf{x} \in X : d(\mathbf{x}, C_i) \leq E\}$  be the error ball located at  $C_i(x_r, y_r)$ , at  $\tau_i$ . Let  $v_i(x_i, y_i)$  and  $v_{i+1}(x_{i+1}, y_{i+1})$  be two points on a candidate geodesic of the error ball. The corrected location on the geodesic as mentioned above, in (8) is

$$P_c(x_c, y_c) = \left( \frac{x_r + m(mx_i - y_i + y_r)}{m^2 + 1}, \frac{y_i + m(my_r - x_i + x_r)}{m^2 + 1} \right).$$

For example in Figure 10, the corrected locations on the candidate geodesics are  $P_{c1}$  and  $P_{c2}$ . Let  $\{I_i, I_{i+1}\}$  be the nearest intersections of the road which the geodesic is part of, and  $s_i$  be posted speed of the road segment. The next position of the error ball is estimated at

$$\tau_{i+1} = \min_{k \in \{i, i+1\}} \frac{1}{s_k} d(P_c, I_k), \quad (13)$$

where  $d$  is the metric distance of the Mobile space.

In scenarios where we have multiple candidate geodesics, as shown in Figure 10, the next position of the error ball is estimated at

$$\tau_{i+1} = \min \left\{ \min_{k \in \{i, i+1\}} \frac{1}{s_i} d(P_{c1}, I_k), \min_{k \in \{j, j+1\}} \frac{1}{s_j} d(P_{c2}, I_k) \right\}, \quad (14)$$

where  $s_i$  and  $s_j$  are posted speeds on road segments,  $L_i$  and

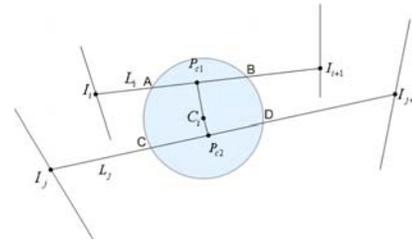


Figure 10. The next position of the error ball is estimated at the nearest intersection from the corrected locations.

$L_j$ , respectively. If the roads segments are directional (one-way) then the candidate intersections are limited to the ones that are towards the directions of the road. In Figure 10 if the road segments are,  $\overrightarrow{I_i I_{i+1}}$ , and  $\overrightarrow{I_{j+1} I_j}$ , the candidate intersection are  $\{I_{i+1}, I_j\}$ . In case there is only one geodesic passing through the error circle, the direction of motion and hence the candidate intersection assumed is towards the opposite end of the geodesic from the corrected points. It will help in eliminating the intersection that has been a candidate in previous estimation.

Let us consider the segment,  $L_i$ . The probability of the mobile user is the same on all points of  $\overline{AB}$ . For the estimation of  $\tau_{i+1}$  towards  $I_i$ , if  $P_{c1}$  is taken as the start point for the estimation of  $\tau_{i+1}$  but the mobile user is closer to point A, the time,  $\tau_{i+1}$ , would be longer than the required one. To fix this problem we need to further refine our approach of intersection polling method. In the refined approach instead of assuming the present location of the mobile user, i.e., the location of mobile user at  $\tau_i$ , we assume that the present location is the intersection of the geodesic and the error circle. We call this intersection point shifted projection point. Since there are two such shifted projection points, for example A,B for geodesic  $L_i$  and C,D for geodesic  $L_j$  in the figure above, we choose the shifted projection point that is closer to the road intersection. The shifted projection points of the geodesic,  $G_i$  with the error circle  $\mathbf{C}_E(C_i) = \{\mathbf{x} \in X : d(\mathbf{x}, C_i) = E\}$  are

$$\mathbf{C}_E(C_i) \cap G_i = \{w_1, w_2\}. \quad (14)$$

Where,  $w_1 = \left( \frac{my_r + x_r - mb}{1+m^2} + \frac{\sqrt{E^2 - [(y_r - mx_r - b) / \sqrt{1+m^2}]^2}}{\sqrt{1+m^2}}, \right.$  (15)

$$\left. \frac{m^2 y_r + mx_r + b}{1+m^2} + \frac{m\sqrt{E^2 - [(y_r - mx_r - b) / \sqrt{1+m^2}]^2}}{\sqrt{1+m^2}} \right)$$

$w_2 = \left( \frac{my_r + x_r - mb}{1+m^2} - \frac{\sqrt{E^2 - [(y_r - mx_r - b) / \sqrt{1+m^2}]^2}}{\sqrt{1+m^2}}, \right.$  (16)

$$\left. \frac{m^2 y_r + mx_r + b}{1+m^2} - \frac{m\sqrt{E^2 - [(y_r - mx_r - b) / \sqrt{1+m^2}]^2}}{\sqrt{1+m^2}} \right)$$

The refined IPM dictates that, the location is updated at

$\tau_{i+1} = \min_{k \in \{i, i+1\}} \frac{1}{S_i} d(w_i, I_i)$ . We pick  $w_i$  for which  $d(w_i, I_i)$  is

minimum.

Let us briefly analyze the performance of Intersection Polling method in terms of number of polls. The performance of the polling method is directly tied to the topology of the map. Topological factors that directly affect the performance of this method are: the number of intersections during the course of a path and direction of the roads. This polling method would perform the best on long directional (one-way) roads or the on the grid-like roads, directional or non-directional. Grid-like roads are directional in general anyway. Assuming the start and destination points are arbitrarily anywhere on the roads but not on the intersection, the number of polls required for a path containing N intersections for a grid-like road network are  $N + 1$ . The advantage comes from the fact that after only one polling the next intersection is identified because the next three intersections are approximately same distance away. Figure 11 depicts part of two-way grid-like road network. The next intersection from L is either A, B or C. Since A, B and C are almost equidistance from L, the polling time for each of them is the same. At the next polling time, any of the intersections, A, B or C, which is contained in the error ball, is map-matched as the next intersection of the path. Therefore for each intersection in the trajectory, only one location polling is enough.

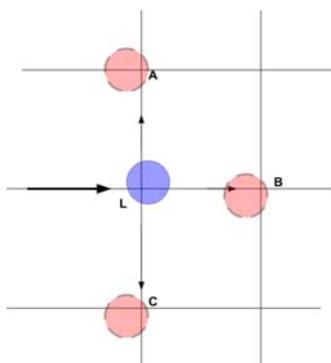


Figure 11. Selection of next intersection in a grid-like road network.

On directional roads of arbitrary layout, the highest number of polls for a path is  $2N + 1$ . In this scenario going from one intersection to the other, would require two polls, one for the nearer intersection, the second one for the remaining intersection, i.e., the farther one. Please see Figure 12. Going from the present intersection L, to B, we need to poll once at  $A^*$  after time  $\tau_1$ , required to reach intersection A and second poll at B. The number of polls could be reduced to  $N + 1$  with a minor modification. The modification is; after polling for the nearer intersection, A, after time  $\tau_1$ , pick the other intersection after time  $\tau_2 - \tau_1$  without polling, where  $\tau_2$  is the travel time from L to B. The drawback of this approach is that the destination would always be an intersection. To continue with performance analyses, we stick with our original approach. The lowest number of polls in this scenario is  $N + 1$ . Here, at each polling the nearer intersection is map matched, for example intersection A, in Figure 12. So, the average number of polls on directional roads with arbitrary layout

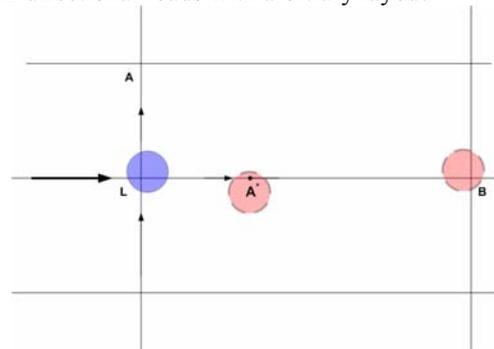


Figure 12. Selection of next intersection in a directional road network.

is  $\frac{3}{2}N + 1$ . For non-directional with arbitrary layout road network, going from one intersection to the next one, we would require maximum three polls. One poll for the nearest intersection, the second one for the nearer of the remaining two intersections and the third one for the last intersection. This scenario is depicted in Figure 13. Here,  $LA < LB < LC$ . Selection of intersection C is the worst case scenario. Intersection A will be polled first, then B and C respectively. So, we have to poll three times between each intersection in this scenario. So, total number of highest possible polls for a non-directional arbitrary layout of road network is  $3N + 1$ . The lowest number of polls on a path on this network would be  $N + 1$ , as already explained in previous scenarios. Therefore the average number of polls for a path on such road network would be  $2N + 1$ .

Since GPS location estimation does not cost any additional air resources, estimating the course of trajectory with least number of GPS location points has not been an area of interest. Furthermore most of the vehicles equipped

with GPS have DR (Dead Reckoning) integrated also. DR keeps tracks of the length and absolute heading of the displacement vector from previous known position. So, the present estimated location of the vehicle is known at almost all the times. Estimating the trajectory of a vehicle with numerous closely packed location points which have low symmetrical error is far easier than with those which are sparse with relatively larger error especially in urban areas.

Let us define an absolute ideal polling method that requires the least number of polling yet does not miss any turns. Most of the proposed methods in GPS world are arc-based (segment based). Such an ideal method would need at least one location point for a directional arc and two location points for a non-directional arc. We assume that this polling method is robust and can handle all the possible topologies of the digital map. An arc is defined as the road segments between two intersections. So, we can estimate the

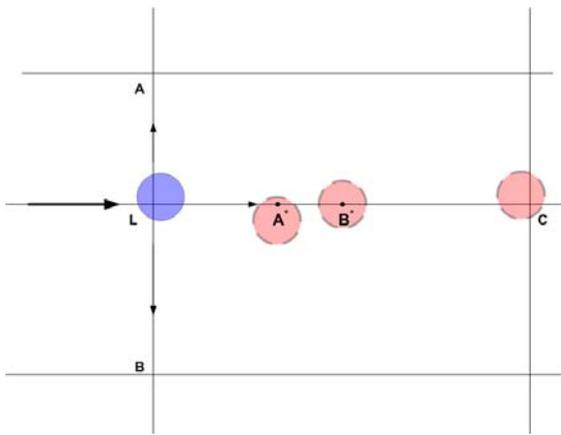


Figure 13. Selection of next intersection in a two-way road network.

performance of such an ideal polling method in terms of number of polls. This ideal poller would require  $\frac{3}{2}N + 1$

polls for an arbitrary path. Let us compare the average performance of Intersection Polling method with this ideal poller. The mean of number of polls required for an arbitrary path in all the three scenarios mentioned above is:

$$\frac{(N + 1) + \left(\frac{3}{2}N + 1\right) + (2N + 1)}{3} = \frac{3}{2}N + 1, \text{ which is equal to}$$

the number of polls of an ideal poller using arc-based map matching methods as defined above.

**D. Error Ball's Directional Oreintation**

In this section we introduce the notion of error ball's image. An image of an error ball is, simply, an instance of its previous embedding. The trajectory of a mobile user is essentially estimated from the interpolation of several images of the embedded error ball. So, at a given point during the trajectory formation we have an embedded error ball and several of its images as shown in Figure 14. In the

beginning, obviously, we just have the embedded ball, images are yet to be imprinted. Angle is one of the most important metric for comparing direction. We are interested in comparing the directional orientation of the transition of the error ball with respect to the actual path segments. We call this angle, direction difference. This direction difference is inscribed between the line joining the center of the embedded error circle and its image and a geodesic. In case there are multiple geodesics in the polyline and we are interested in finding the direction difference of this particular path segment and the transition of error ball, we connect the start point (vertex) to the end point of the path segment. The angle it makes with the reference, x-axis we call it the actual direction of motion during that interval.

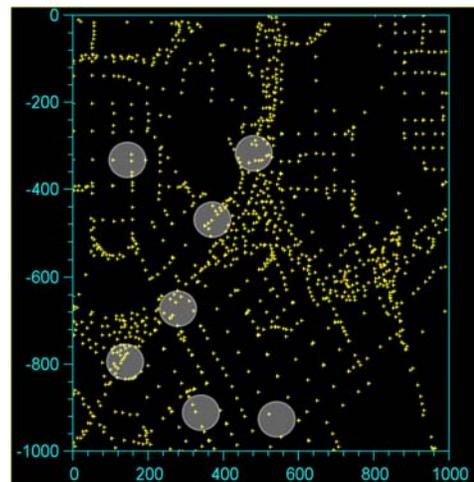


Figure 14. A graphical representation of our GIS database, error ball and its images.

Since the center of error ball,  $C_1$ , and that of its image,  $C_0$  (Figure 15), are not the actual locations of a mobile user at time  $\tau_{i+1}$  and  $\tau_i$ , respectively, the direction difference will not be accurate. There is an offset in the direction difference.

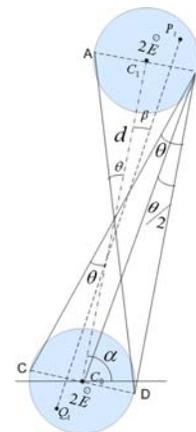


Figure 15. Illustration of direction of transition of an error ball,  $\alpha$ , the direction difference offset,  $\theta$ , and difference error,  $\beta$ .

In the figure above  $\mathbf{B}_E(C_1)$  is an error ball centered at  $C_1$  and  $\mathbf{B}_E(C_0)$  is its previous image with center at  $C_0$ . Let  $P_i \in \mathbf{B}_E(C_1)$  and  $Q_i \in \mathbf{B}_E(C_0)$  be the two actual locations of the mobile user,  $\angle(\overline{C_0C_1}, \overline{Q_iP_i}) = \beta$  is the direction error. We define the maximum offset error of an error ball as,

$$\theta = \max_{j \in \mathbf{B}_E(C_1)} \left[ \max_{i \in \mathbf{B}_E(C_0)} \angle(\overline{C_0C_1}, \overline{Q_iP_j}) \right]. \quad (17)$$

As illustrated in the figure above the maximum offset error is transcribed by the extreme points on the diameters of the two circles, which are diagonally opposite. For example C,B and A,D are such extreme points. Therefore the maximum offset,

$$\theta = \tan^{-1} \left( \frac{2E}{d} \right). \quad (18)$$

Please note  $\theta$  is symmetrical, i.e.,

$$\angle(\overline{C_0C_1}, \overline{BC}) = \angle(\overline{C_0C_1}, \overline{AD}) = \theta, \text{ and it can be easily seen}$$

$$\text{that } -\frac{\pi}{2} < \theta < \frac{\pi}{2}.$$

#### E. Direction Negativity

Let  $\gamma_1 : p_1, q_1$  be a path segment with start point  $p_1$  and end point  $q_1$  and  $\gamma_2 : p_2, q_2$  be another path segment with start and end points  $p_2$  and  $q_2$ , the angle between the two path

$\angle(\gamma_1, \gamma_2) = \angle(\overline{p_1q_1}, \overline{p_2q_2})$  is called the direction difference between paths  $\gamma_1$  and  $\gamma_2$ . A direction negativity

exists between two paths  $\gamma_1, \gamma_2$  if  $-\frac{\pi}{2} < \angle(\gamma_1, \gamma_2) < \frac{\pi}{2}$ .

Please note the range of  $\angle(\gamma_1, \gamma_2)$  is  $[0, \pi]$  or  $[0, -\pi]$ .

The maximum negativity is reached when;

$$\angle(\gamma_1, \gamma_2) = \begin{cases} \pi \\ \text{or} \\ -\pi \end{cases}$$

This can be clearly visualized if you think one of the paths along the y-axis, i.e., a vertical path, the other path along the positive or the negative side of the x-axis. On either sides of the second path does not contain any y-component in their path. If the direction remains within first quadrant for the positive side of the path and remains within second quadrant for the negative side of the second path, they both have y-components that are positive. If the second path that is along the positive side of the x-axis moves towards fourth quadrant and the one along the negative side of the x-axis

move towards the third quadrant, we start getting the negative y-components in the paths, i.e., they move into direction negativity. If we need to find the angle between a

path,  $\gamma$ , and the transition of the error ball,  $\angle(\gamma, \overline{\mathbf{B}_E(C_1)})$ ,

we need to add the offset to  $\angle(\gamma, \overline{C_0C_1})$ . In a worst case

$$\text{scenario the offset is, } \theta = \tan^{-1} \left( \frac{2E}{d} \right).$$

$$\Rightarrow \max \angle(\gamma, \overline{\mathbf{B}_E(C_1)}) = \angle(\gamma, \overline{C_0C_1}) + \tan^{-1} \left( \frac{2E}{d} \right). \quad (19)$$

where,  $d$  is the displacement of the transition of the error ball.

The condition for direction negativity between a path,  $\gamma$ , and transition of the error ball can be easily formulized using inner product. The condition is

$$\frac{\pi}{2} < \cos^{-1} \left( \frac{(x_j - x_i)(x_1 - x_0) + (y_j - y_i)(y_1 - y_0)}{\sqrt{[(x_j - x_i)^2 + (y_j - y_i)^2][ (x_1 - x_0)^2 + (y_1 - y_0)^2]}} \right) < \frac{-\pi}{2}, \quad (20)$$

where  $(x_i, y_i)$  and  $(x_j, y_j)$  are the end points of the path (or geodesic),  $\gamma$ , and  $(x_1, y_1)$ ,  $(x_0, y_0)$  are the coordinates of the center of the error ball and the error ball image, respectively. Here  $\frac{-\pi}{2}$  is taken clockwise which is  $\frac{3\pi}{2}$

counter clockwise. Please note a proper offset needs to be applied for accurate direction difference.

#### F. Path Constraint

The different paths between the corrected points in error ball and its previous image are extremely important entity in determining the trajectory of a mobile user. This set of paths is essentially the domain of the possible paths. There are numerous possible paths to connect a projected location from the error ball image to the error ball. We need to assert a constraint to limit the number of paths between the corrected locations. The simplest, yet very pivotal constraint is the travel time. If the travel time of a path between the farthest corrected points is very large than the transition time of the ball, that path is very unlikely. This is very much related to the topology and geometry of the map. In most cases if the travel time of a path is more than twice the transition time of the error ball that path could be eliminated. On directional road, Manhattan style for example, the longest path is five times the direct distance assuming roads around the blocks make a square. The exact time multiplier very much depends on the map topology.

#### IV. PROPOSED METHOD FOR TRAJECTORY ESTIMATION

Location Points Collection: Suppose a mobile user starts from home at location **A**, travels to location **B**, **C**, and **D**,

and then comes home at location A. Figure 16 depicts logical parts of this travel trajectory. The points B, C, and D are the rest points. These rest points partition the trajectory into trajectory legs during the whole day of mobility. In this example there are four trajectory legs.

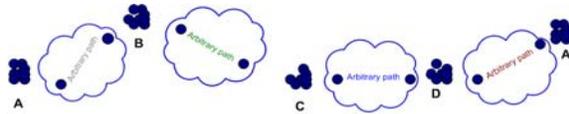


Figure 16. Logical partition of trajectory

We shall describe the mobility profiling method for one leg. For rest of the legs the steps are the same.

Step 1: Poll the mobile user for his location. Find the candidate geodesics as described in Section III B.

Step 2: Repeat Step 1 after determining polling time as described in Sections III C. Continue polling until destination of the leg is reached.

Step 3: Apply the condition for directional negativity to eliminate multiple geodesics in the error ball as explained in Section III E, please see Figure 17.

Step 4: Apply the path constrains using appropriate time multiplier as explained in Section III F.

Step 5: Map match the so obtained curve or set of curves with the digital map to obtain the closest to shape trajectory of the travel leg. Please see Figure 19.

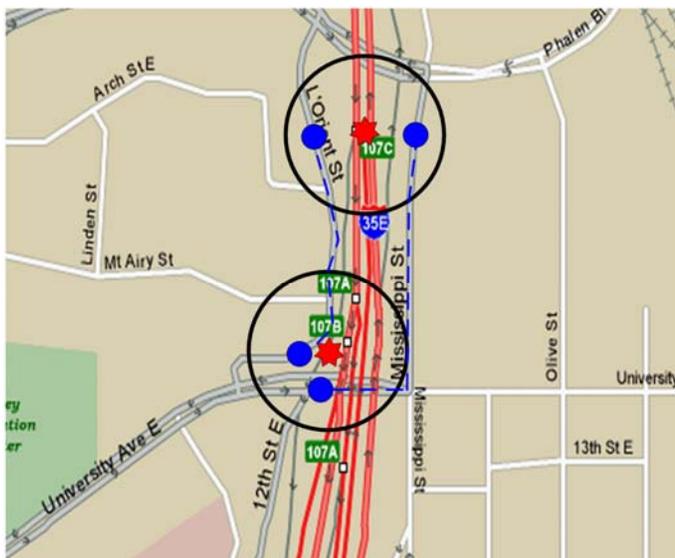


Figure 17. Direction negativity and path constrains on error ball and its images. If the user was on University Ave E prior to the recent last two polls shown in this picture, after applying the constrains, we are left with only two candidates roads which are shown in broken blue lines.

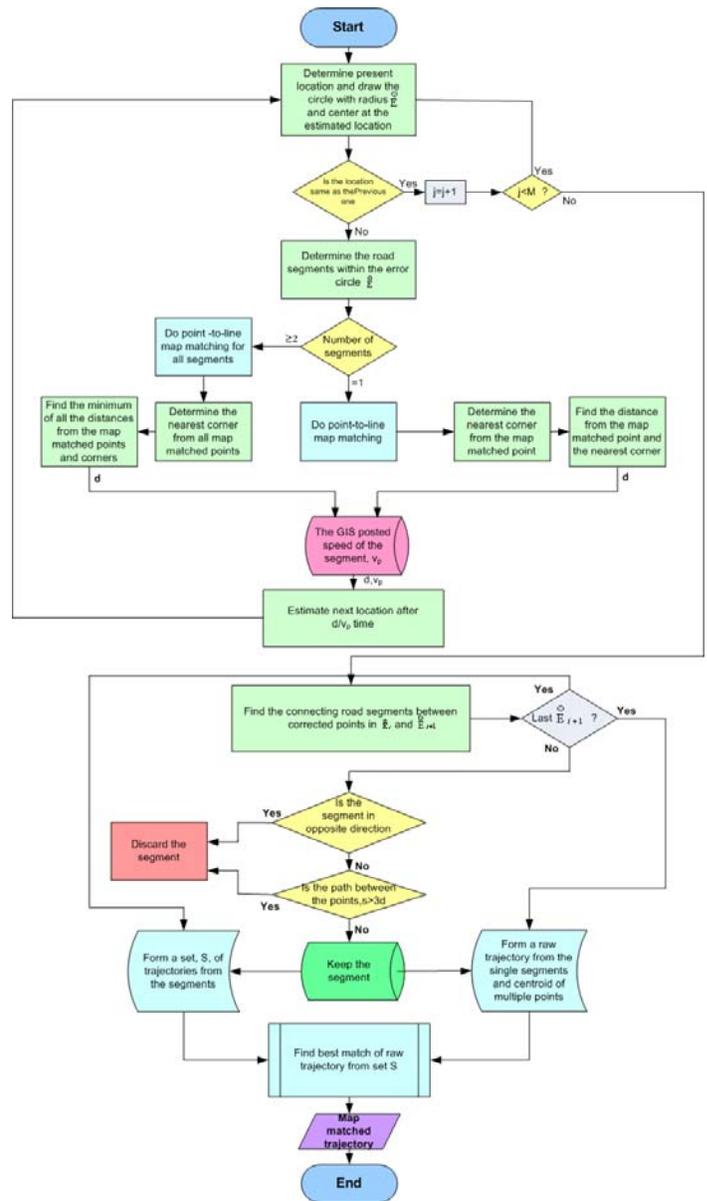


Figure 18. Trajectory estimation flow chart.

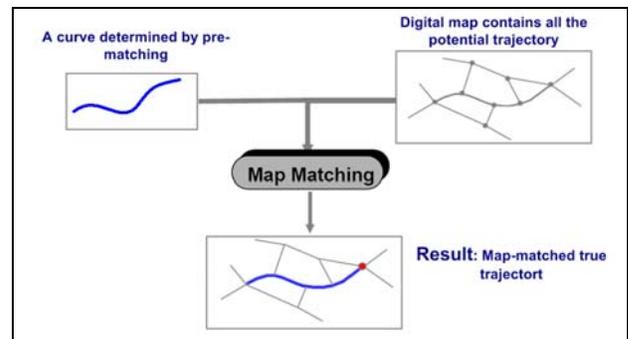


Figure 19. Map matching of pre-matched candidate curve with the set from GIS database.

A flow chart of the detailed algorithm is shown in Figure 18. The algorithm was implemented using C# (C sharp) in Microsoft windows. C# fulfilled the graphics requirements we needed for our simulation easily.

## V. IMPLEMENTATION OF PROPOSED METHOD

The implementation has two parts, as described in the following subsections.

### A. GIS Database of Road Networks

Our mobility profiling system is essentially a GIS embedded with a cellular system. The cellular part of the system estimates the mobile location; the GIS part corrects the estimated location and provides all necessary references for trajectory estimation. Database is the backbone of a GIS. For trajectory estimation we use a digital database of a road network. We have paper road maps, Figures 21 (scale 1:10,000) and 22 (scale 1:50,000), which need to be digitized to make a digital database. We use vector format data of a road network. Manual digitization is one of the widely used methods to digitize a paper map to vector data. It uses a digitizing table. A digitizing table is essentially an input device that interfaces with a computer, to store the coded data in the form of coordinate points. A digitizing table is an expensive tool. We developed a Microsoft Windows based application that emulates a digitizing table. The digitizing application, let us call it Digitizer from this point on, is not hard coded to exclusively digitize the maps shown in Figures 21, and 22 but it can load any map that needs to be digitized using the user interfaces depicted in Figure 20. The whole digitization process is a little labor intensive. We used point mode digitization.

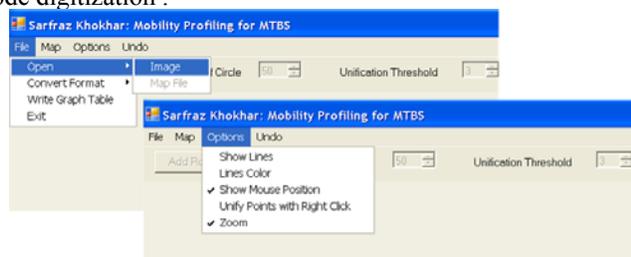


Figure 20. Digitizer's GUI.

The map image could be in JIFF, BMP, or JPEG format. Once the image is loaded the digitization starts by naming the road segment, i.e., polyline, and selecting the posted speed, and direction attribute of the road segment. The segments are digitized by mouse clicks at important points using point mode method. The most effective feature in the Digitizer is zooming, Figure. 21. This is where the Digitizer adds value in accuracy. The main source in digitization error is introduced by the operator who is doing the digitization [29][30]. One of the main reasons of operator's error in digitizing roads is that some roads are not thick enough on the map and it is difficult to stay on the center line of the road. The operator is supposed to stay within a band centered on the road center line. In cartography it is referred as epsilon band [31]. Zoom feature will help the operator to

stay within the epsilon band and closer to the center line. The zoom window features a very fine pointer for further

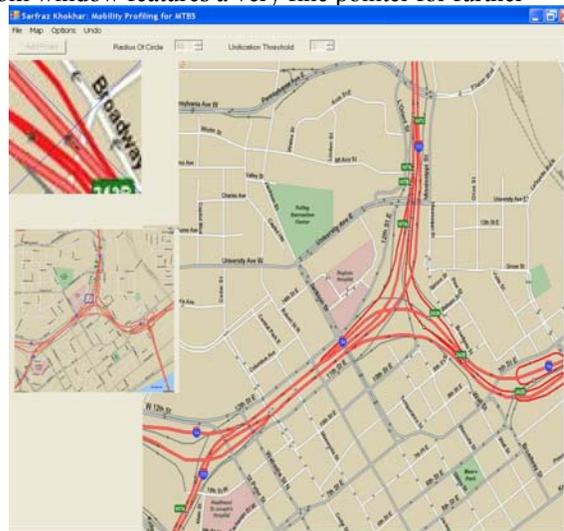


Figure 21. Zoom feature of the Digitizer helps in digitization by hand. The map being digitized is of an urban area of Saint Paul Minnesota.

accuracy. On thin or closely packed roads as shown in zoom window, the accuracy is about four times that that of a digitizing table. Since an operator concentrates on only one screen as compared to the digitizing table where the operator's attention is divided into two places, the monitor screen and the map, digitizing with software version of the digitizing table is more efficient. Once a map is digitized using a digitizing table, its digitized version is verified with zoom feature on the computer. So, zoom is a reliable feature in cartography world already. The process of manual digitization is a psychophysical process that depends on a harmony between human perception, adjustment of mental impression and physical balance of arm muscles [32], so, the accuracy of manual digitations varies from operator to operator. Manual digitization error on a 1:24,000 map ranges from 20-55 feet. Using a feature like zoom would reduce the error significantly.

The intersecting roads have a common point which must be selected as one of the points of the road segments during their digitization. One issue while marking this common point is that, one may be few pixels off while marking this point during the definition (digitization) of each road segment. This issue is remedied with one of the features of Digitizer; we called it "Unification Threshold". To combine points on different road segments into one intersecting point, we right click the mouse around the points, a unifying routine combines all the points in the threshold area. The threshold area is configurable as shown in Figure 20. After all roads are digitized and stored in the database, the ASCII database was converted to binary for faster access. We can convert ASCII data to binary and binary to ASCII. To verify the accuracy of the so obtained digitized database, we reconstructed the road network from the data accurately.

The mobility trajectory estimation simulation was done in two different topological scenarios: an urban area of St. Paul, Minnesota and suburban area of North Carolina. We digitized two maps, one shown in Figure 21 (urban) and the other one depicted in Figure 22 (suburban).

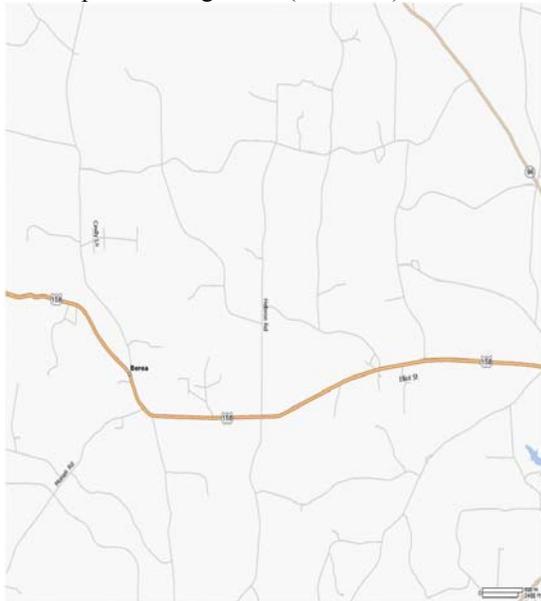


Figure 22. Map of suburban area digitized for GIS database.

The digitized version of the maps mentioned above are given below, Figures 23 and 24. Please note these maps are erected from the GIS data base as mentioned in Section II B.

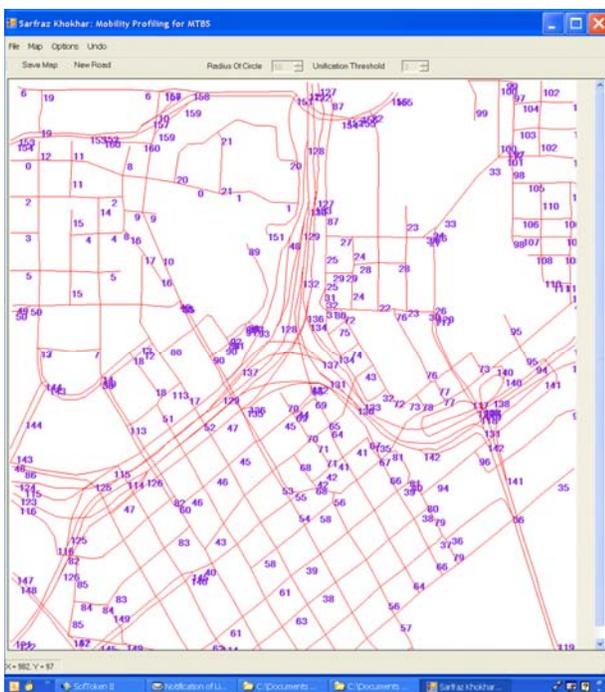


Figure 23. Reconstruction of map of Figure 21 from GIS database.

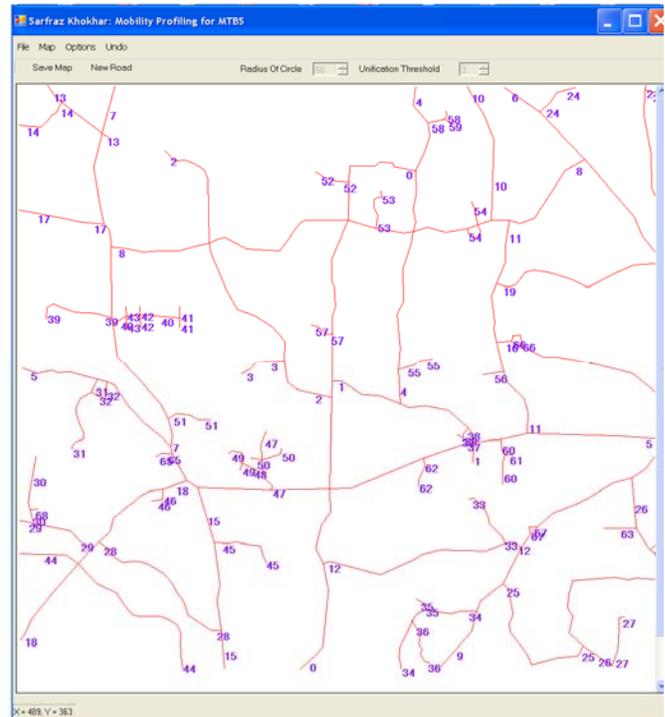


Figure 24. Reconstruction of map of Figure 22 from GIS database.

### B. Mobility Profile Simulation

To simulate, the mobile user's mobility, we developed another Windows based application, Mobility Profiler Simulator. The application has two major parts: one simulates mobility of the mobile user via traversing of a dinky car on the road network of a map; the other is an implementation of the mobility profiling method described in section IV. Figure 25 depicts the mobility simulation window of the application.

An optional step after invoking the application is to load the reference map image for viewing the simulation with respect to the map. Then digital map (GIS database) is loaded into the application. This is the actual road network which is referenced by the mobility profiler for location polling and map matching. The dinky car traverses on the path provided by the so called "path file". The path file is a text file containing the points and line segment information that defines a particular path for the mobile user. After path is loaded, the simulation is started from another window. The dinky car, shown in the circle in Figure 25, travels along the paths dictated in the path profile. This simulates the actual mobility of the mobile user. The Mobility Profiler Simulator estimates the mobile trajectory independently by polling mobile user's locations, correcting them by identifying the road segments and them combining the road segments.

The animation speed of the dinky car can be increased or decreased for demonstration purposes using the control option called Animation Speed. The dinky car travels on the



Figure 25. Mobility Profiling simulator's GUI, the mobility simulation window.

road segments according to the posted speed on them. A variation on the speed can be adjusted. The polling agent in mobility profiler polls for location of the car according to an implementation of Intersection Polling method, described in details in Section III C. The location is estimated with a random error. The process of location polling has been animated in the application to depict how it would happen in the actual deployment in the field. In this process the polling agent inquires the exact location of the dinky car, the location is returned to the profiler with a radiating signal (animation) from the car, adding a random error. The error has been modeled with a circle of radius equal to the error associated with the underlying location estimation technology and centered at the returned estimated location.

The second window of the application, the mobility profiler, is invoked from a menu from the first application. This part of the simulator provides the interface for input variables; the location estimation error of the underlying location estimation technology and the posted speed variations. We can define lower and upper bounds for the speed variation in the posted speed of GIS database. These input variables are entered in the counters shown in red rectangles in the Figure 26. This window plots the trajectory per algorithm given in flow chart in Figure 18. During the first parsing of the simulation candidate road segments and the corresponding location points are determined. In the second parsing, direction and the distance constraints are applied. After these constraints are applied, the estimated trajectory is plotted as show in Figure 26.

If this trajectory does not match exactly with a trajectory from the digital map, we map match it with our digital map. In most of the cases the constraints rectify the shape of the trajectory. The actual path (from path file) and the estimated trajectory, both, are plotted in the simulator window. The reference map, which merely acts as background image makes the simulation result more meaningful for viewing, Figure 27. This gives a better understanding of the error in



Figure 26. Mobility profiling simulator's GUI, the trajectory estimation window.

the estimated trajectory. It shows the exact locations on the map where the estimated trajectory deviates from the actual one. The option Offset XY translates the reference trajectory right and down according to the set values in these counters for easy viewing and comparing of the two trajectories. Please refer to the first rectangle in the menu bar in Figure 26.

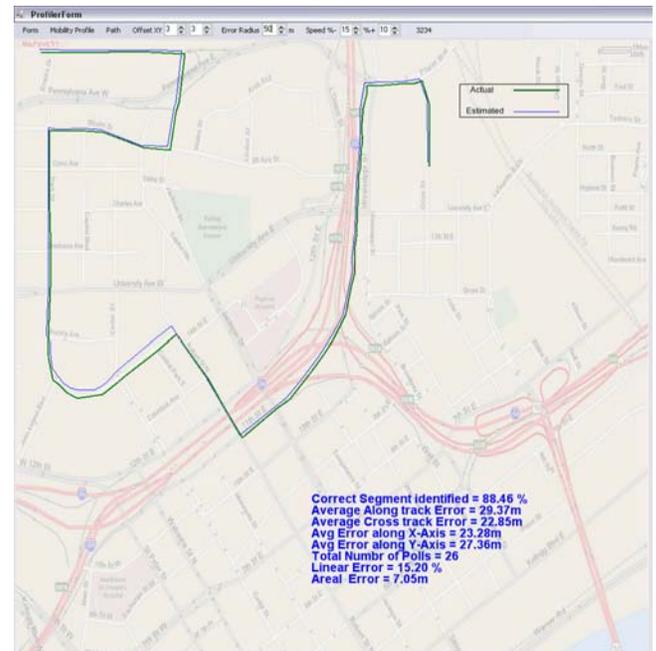


Figure 27. Mobility profiling simulator window with reference map.

All the simulation results are presented with reference path and reference map as shown in Figure 27. The green curve (darker if black and white) depicts the actual path whereas the blue curve (lighter if black and white) is the estimated trajectory. We ran the simulation, both in urban and suburban areas. The main variables in the simulation are the location error and the speed variation, which we could vary during the simulation also, if needed. After a simulation finishes, the mobility profiler prints important statistics pertaining to the simulation. It prints the following statistics:

- Percentage of segments identified correctly
- Average Along Track Error
- Average Cross Track Error
- Average Error along X-axis
- Average Error along Y-axis
- Total number of location polls
- Linear Error ( defined in next section)
- Areal Error (defined in next section)

### VI. TRAJECTORY ESTIMATION SIMULATION

Location points are building blocks of a trajectory. The estimated location of a mobile user does not necessarily map on a road because of the error in location estimation methods. Among the vertices defining roads, the points corresponding to roads intersections play a pivotal role. In urban area the roads are densely populated hence a large number of road intersections. This results in potentially high number of turns that leads to high polling frequency. Secondly, because of high road density, an error area (estimated location) encloses a large number of roads. On the other hand in rural or suburban areas the road density is smaller as compared to urban areas; therefore estimated location corresponds to fewer roads as candidates of the mobile user trajectory. So, the rural areas should provide more accurate user's trajectory. In this chapter we investigate both scenarios and provide the simulation results. We shall demonstrate how estimated location error affects the accuracy of the mobile user's trajectory. We shall correlate the error with the underlying location estimation technologies and present suitability of the location technology under different road topologies.

Our estimated trajectory is represented as a sequence of points, so essentially as a set. The error associated with the estimated trajectory is a measure of dissimilarity between the actual and estimated trajectories. The actual trajectory is also a sequence of points, so Hausdorff distance [33] would be natural choice as an error index. However the problem with Hausdorff distance is that, it gives the shortest distance between two points in the sets and it does not cover the course of the distance traveled. Therefore using Hausdorff distance to measure the dissimilarity would lead to wrong findings. The issue with Hausdorff distance is covered in Fréchet distance [34]; it takes the course of distance into account. It is explained by man-dog example in literature [35][36]. Fréchet distance is widely used to find distance

(dissimilarity) between two curves. The curves are approximated as polygonal figures as done in the case of roads. However Fréchet distance shows an anomaly when comparing the two trajectories as depicted in Figure 28. The dissimilarity between two paths A and B and that of between A and C, according to Fréchet distance is the same, which is misleading.

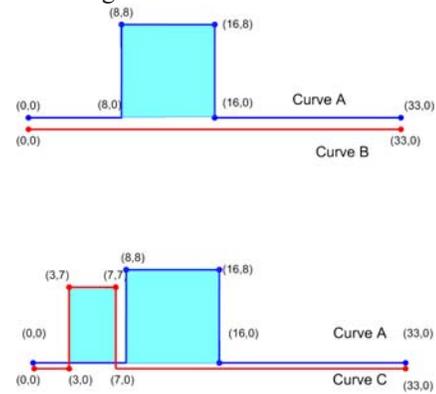


Figure 28. Anomaly in Fréchet distance.

The Fréchet distance between A and B is 8, which is also the case with A and C. Clearly A and C are more dissimilar than A and B. Therefore we need to define a meaningful index of error for an estimated trajectory. We called this error Trajectory Estimation Error (TEE). TEE can be represented as linear error or areal error as defined below. Both, linear and areal errors together, give a good understanding of the trajectory estimation error.

#### A. Linear Error

Linear error demonstrates the length of the actual trajectory that is missing in the estimated trajectory. It is ratio of the path length that is erroneous,  $\lambda_e$ , to the total path length  $\lambda_t$ .

$$E_L = \frac{\lambda_e}{\lambda_t} \tag{21}$$

#### B. Areal Error

Areal error demonstrates the dissimilarity between the two trajectories (curve), the actual and the estimated one by the area bounded by them as shown in Figure 28. According to Fréchet distance the two set of curves, A,B and A,C are equally dissimilar, however their areal error portrays a better error index. For better comparison of areal error, bounded area per unit length of trajectory would be a better index of error. The areal error per unit length is:

$$E_A = \frac{\alpha}{L} \tag{22}$$

Where,  $\alpha$  is the error area, and  $L$  is the total length of the actual trajectory. We have used linear error along with areal error to get a better degree of comparison between actual and the estimated trajectories.

### C. Trajectory Estimation Simulation Results

Trajectory estimation simulation was performed using two digital maps discussed above, one for urban area of Saint Paul Minnesota and the other for suburban area of North Carolina. A range of error corresponding to different location estimation technologies was used for the simulation. Figures 29 through 38 show the simulation results for an urban area for different location error. Green trajectory is the actual trajectory whereas the blue is the estimated one.

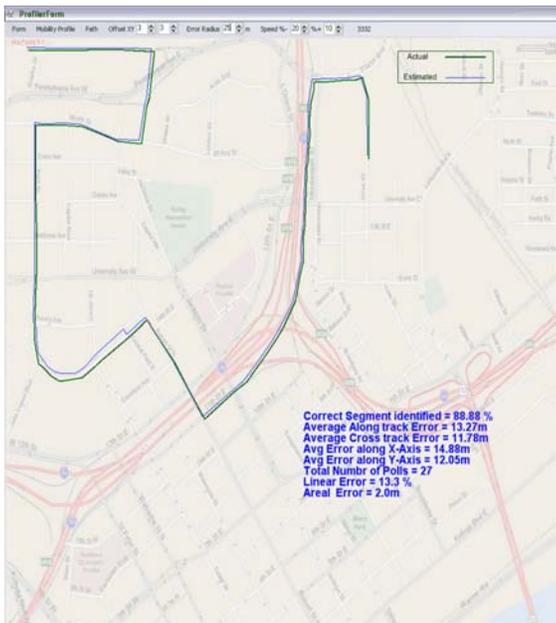


Figure 29. Result with location error of 25m.

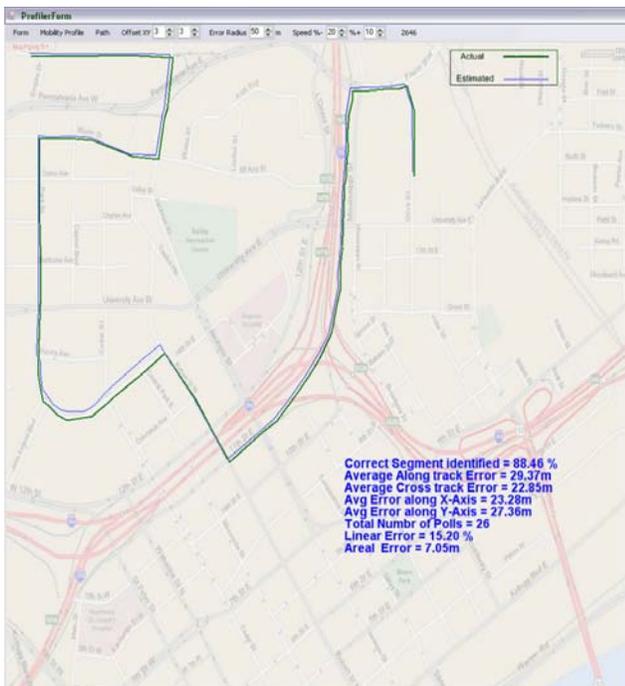


Figure 30. Result with location error of 50m.

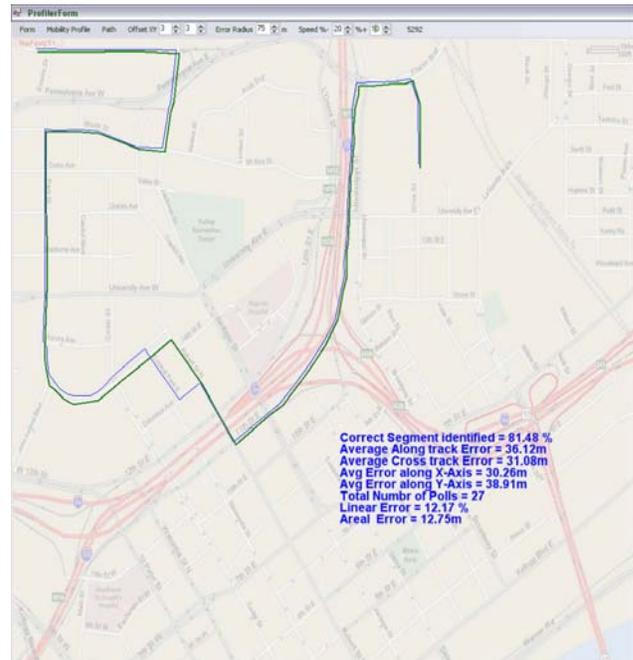


Figure 31. Result with location error of 75m.

In urban area for location error of 50m or less, the trajectory error was on the closely parallel roads only (Figures 29 and 30). However, for larger location error, 100m and more, the trajectory error expanded farther than the closely parallel roads (Figures 32 through 37). Figure 38 shows a graph for location error versus trajectory error for the urban area.

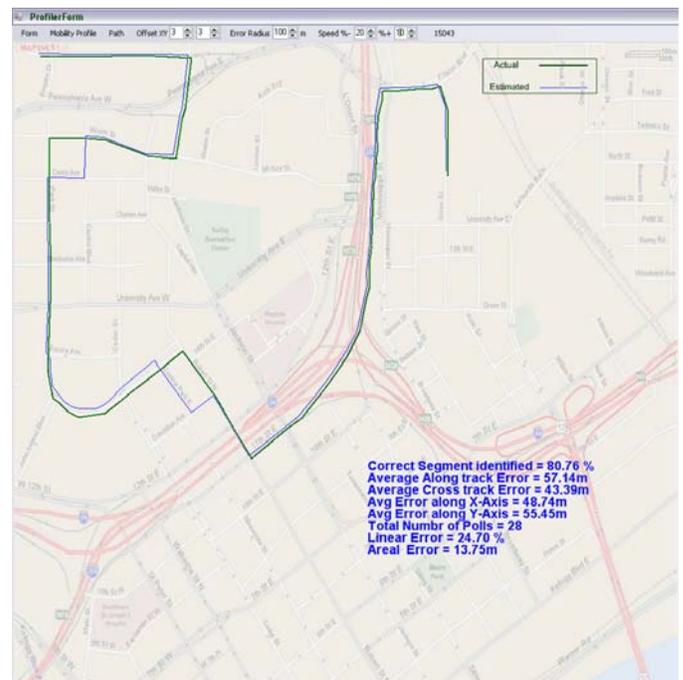


Figure 32. Result with location error of 100m.

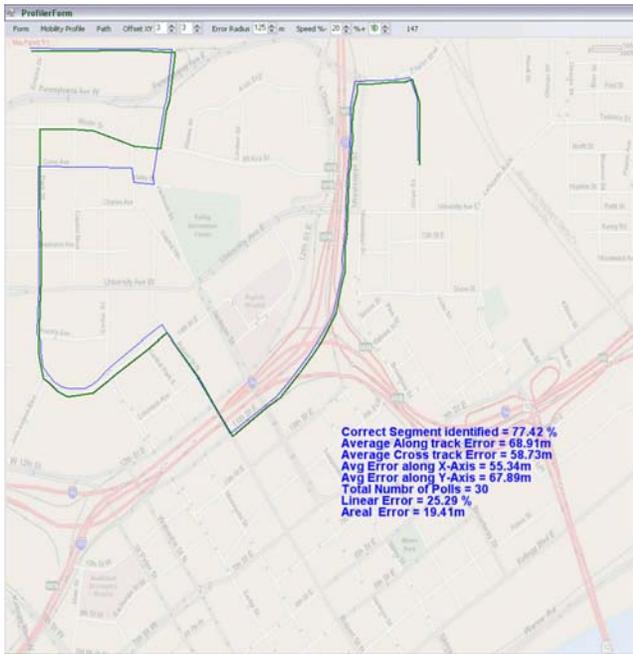


Figure 33. Result with location error of 125m.

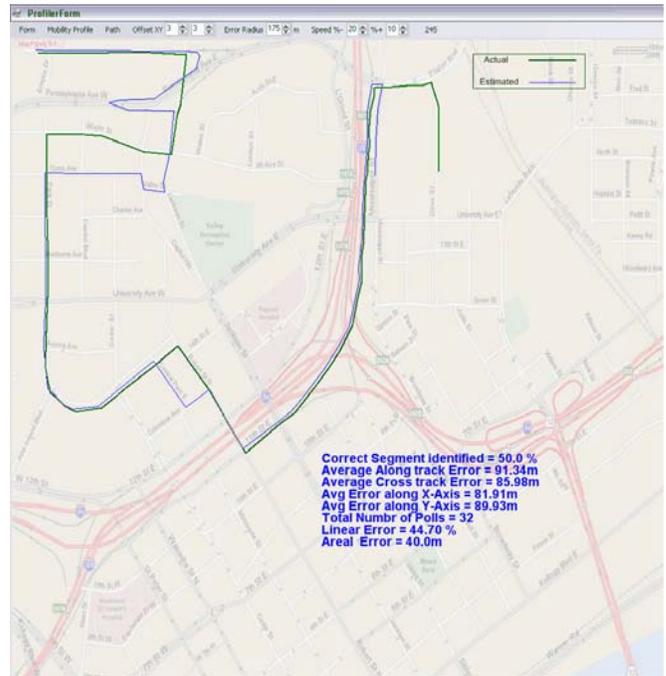


Figure 35. Result with location error of 175m.

For the suburban area the algorithm missed only a short segment of trajectory (Figures 40 through 42), even for large location error of 400m. We get this performance because of low number of intersections per unit area in suburban areas.

Both areal and linear errors in trajectory estimation not only stem from location estimation error, but are intrinsic to the map topology and path. For example, if we choose a different path the same parameters used in Figure 36 would yield a different result, please see Figure 37.

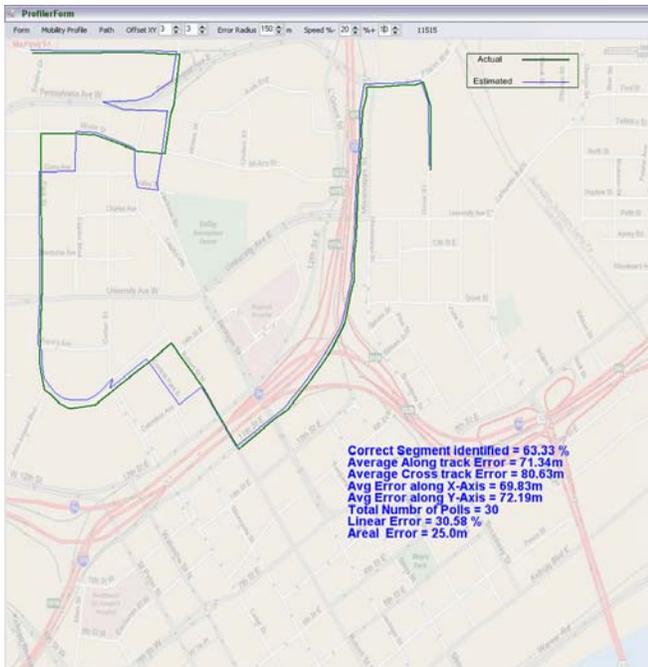


Figure 34. Result with location error of 150m.

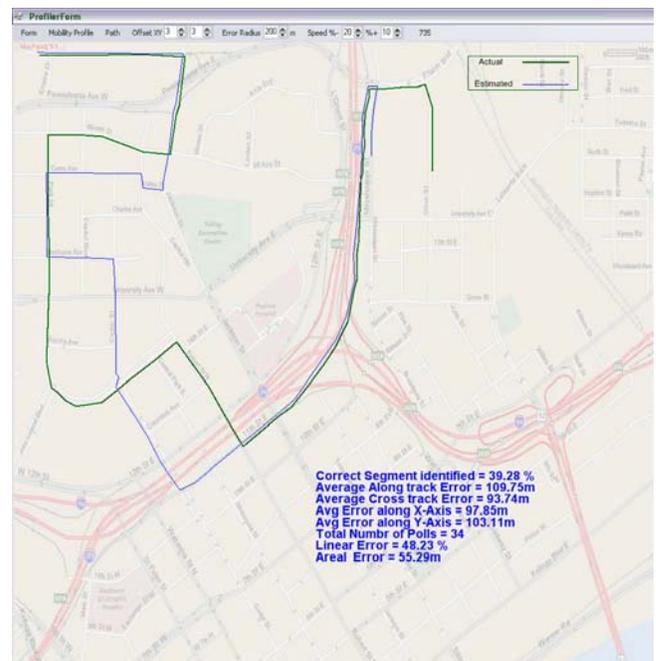


Figure 36. Result with location error of 200m.

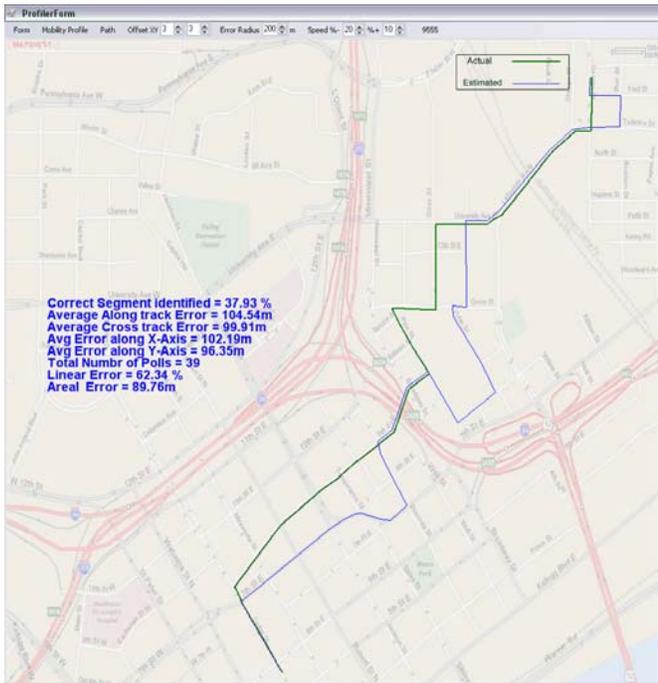


Figure 37. En error of 200 m yields higher trajectory error using dense part of the map.

The relationship of Trajectory Estimation Error (TEE) to the location estimation error (LEE) can be easily interpreted from the results above. Figure 38 depicts the LEE-TEE relationship for LEE of 0-200m with a step of 25m. For urban area, after 200m of LEE, the trajectories are not reliable for profiling. TEEs depicted in the graph are the typical values for ten simulations for each LEE. For smaller location estimation errors, 25m and 50m, each simulation run for that particular LEE gave almost similar trajectory. The error mainly comes from the two parallel roads about thirty meters apart. For LEE of 75m and 100m, there was some difference in the shape of trajectory for each run. After around 125m of LEE the trajectory curve (shape) varies significantly for each run with the same LEE, because the error circle was large enough to start inscribing multiple candidate road segments and road intersections. The three rectangles in Figure 38 provide a reference of known accuracy of different location estimation technologies, namely Assisted GPS, TDOA technologies such as E-OTD and U-TDOA (Uplink-TDOA), and AOA. It is evident from the figure which location estimation technologies suit for mobile trajectory estimation. It is important to note that dependence of TEE on LEE is very much tied to the map topology and the trajectory of the mobile user. Within the same map two trajectories of the same length under same LEE would lead to different set of linear and areal error as observed in comparing Figures 36 and 37. Since our location error is random with the upper bound of  $\overset{\circ}{E}$ , with each run of LEE for specific  $\overset{\circ}{E}$ , we get different values of

TEE. This is true for both, linear and areal errors. With large number of simulation runs we get a range of TEE related to LEE. As LEE increases the range widens as expected. Please see Figure 39.

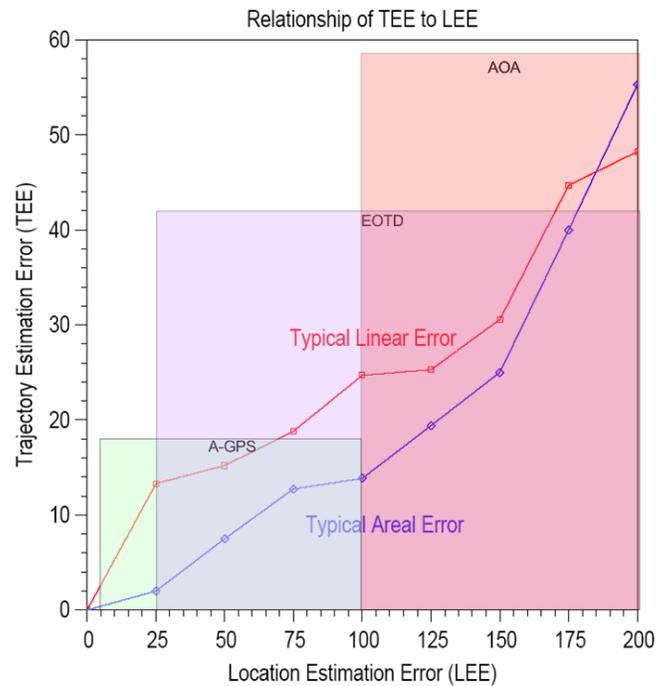


Figure 38. Trajectory estimation error vs location estimation error related to different location estimation techniques.

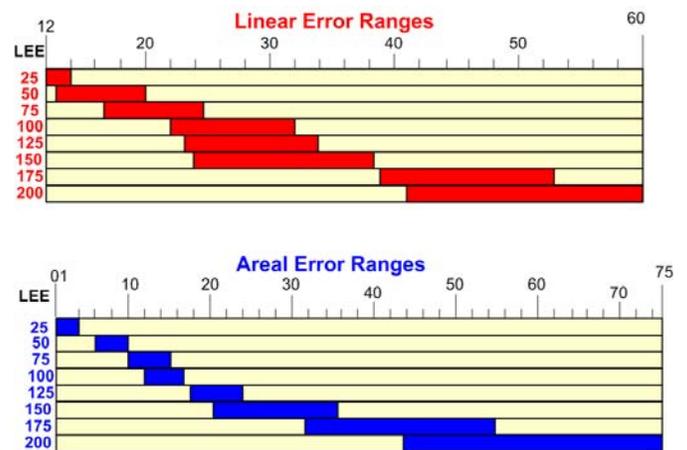


Figure 39. Linear and Areal Error Ranges.

Figures 40 through 42 show simulation results for rural area shown in map in Figure 22 for error of 100m, 200m and 400m respectively. The results are the same for these location error. The mobility profiler missed a small corner near the main highway. For smaller location error around 50 m, this minor error in trajectory estimation diminishes. This

behavior is manifested because there are two parallel path that are about 100 m apart and the both of them are candidate road segment. The segment that is picked in estimation is more closer in shape of the curve prescribed by the estimation location points.

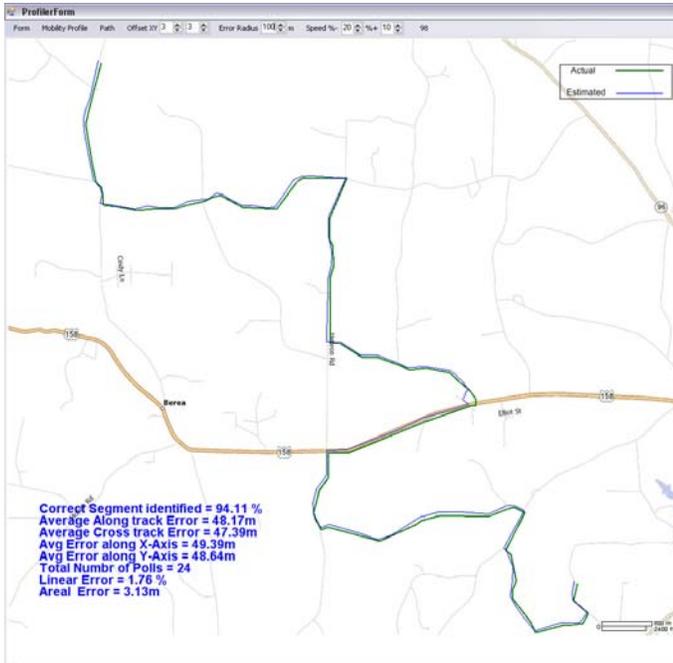


Figure 40. Suburban Trajectory estimation with 100m error.

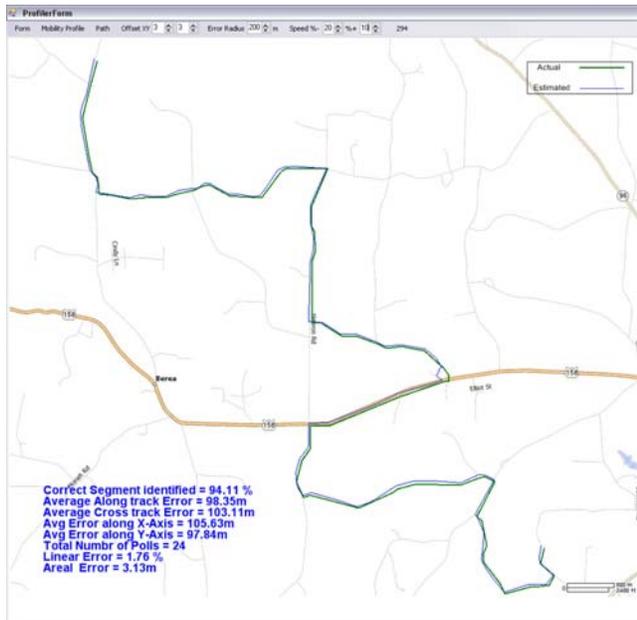


Figure 41. Result with location error of 200m.

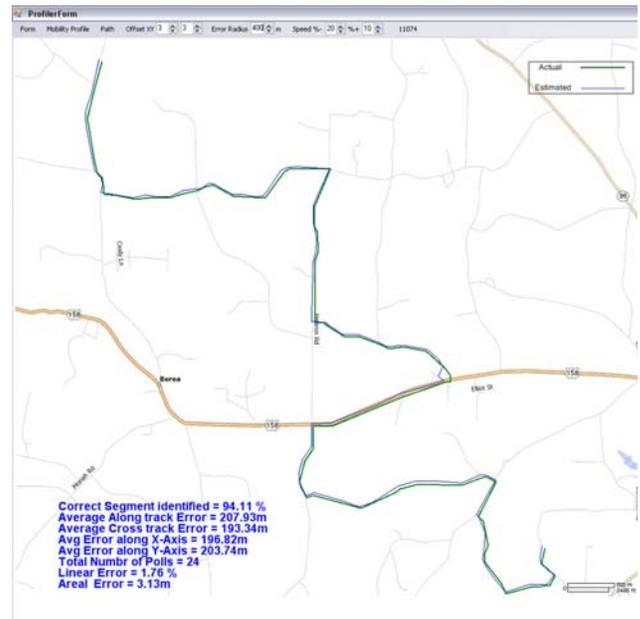


Figure 42. Suburban Trajectory estimation with 400 m error.

## VII. CONCLUSION

We applied the proposed trajectory estimation algorithm in two field scenarios; urban area and rural area. The objective was to determine how existing location estimation technologies would perform with the algorithm in urban and suburban areas. In urban area, where road network is dense and consequently there are a large number of potential turns for a mobile user, the trajectory error increases sharply with the error of underlying location estimation technology. Even within the urban area, the topological and geographical differences between different parts of the same road network bring about substantial difference in the accuracy of the estimated trajectory as seen in comparing Figures 36 and 37. The right half of the road network in Figures 36 has fewer turns. There are fewer inlets and outlet points per unit area on that side of the network. That's why; increasing the location estimation error did not affect that part of the road network as much as it did on the other side. When same location error was applied to a different path going through higher number of potential turns (Figure 37) the error increased to almost double. With the increase of LEE not only TEE increases, but also the number of location polls required estimating the mobile trajectory. Even under topological and geographical constraints, with larger error, it requires a larger number of polls to get out of the cluster of roads. These roads are enclosed within the error circle. Larger the error is, more candidate road segments for trajectory estimation are enclosed in the error circle. So, in urban area relatively precise location estimation methods like A-GPS or accurate time based methods such as EOTD are best suited for trajectory estimation. Whereas in suburban or

rural area, almost all technologies that fulfill E-911 mandate [3] qualify to estimate trajectory with good accuracy.

## REFERENCES

- [1] Sarfraz Khokhar, and Arne A. Nilsson, "Estimation of mobile trajectory in a wireless network: A basis for user's mobility profiling for mobile trajectory based services," 3<sup>rd</sup> International Conference on Sensor Technologies and Applications, Athens, 2009.
- [2] Elliott D. Kaplan, and Christopher Hegarty, "Understanding GPS: Principles and applications," Artech House Publishers, 2005, Second Edition.
- [3] Allison Kealy, Stephan Winter, and Günther Retscher, "Intelligent location models for next generation location-based services," Journal of Location Based Services, vol. 1, Issue 4, December 2007, pp. 237–255.
- [4] Lin Ding-Bing, Juan G Rong-Terng, and Lin Hsin-Piao, "Mobile location estimation and tracking for GSM systems," Fifteenth IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications, Barcelona, 2004, pp. 2835–2839.
- [5] Sarfraz Khokhar, Arne A. Nilsson, and Mo-Yuen Chow, "Mobile location estimation methods," GIS 2001 Conference & Exposition, Vancouver, BC, February, 2001.
- [6] Sarfraz khokhar, Arne A. Nilsson, "Introduction to mobile trajectory based services: A new direction in mobile location based services," International Conference on Wireless Algorithms, Systems and Applications, Boston 2009, Springer-Verlag Berlin Heidelberg, pp. 398–407.
- [7] Stanley Sokolowski, Raymond L. Filler, Nhut Vo, and Paul Olson, "Enhanced C2 functionality in urban and other complex terrain," IEEE Military Communications Conference, vol. 1, 17-20 October, 2005, pp. 258–264.
- [8] Goran M. Djuknic, and Robert E. Richton, "Geolocation and assisted GPS," Computer, Volume 34, Issue 2, February, 2001, pp. 123–125.
- [9] Wenjie Liao, Weifeng Lv, Tongyu Zhu, and Dong Dong Wu, "A map matching algorithm for intersections based on Floating Car Data," 10th International Conference on Advanced Communication Technology, vol. 1, 17-20 February, 2008, pp. 311–316.
- [10] David Bernstein, and Alain Kornhauser, "An introduction to map matching for personal navigation assistants," New Jersey TIDE Center 1996.
- [11] Chenhao Wang, Zhencheng Hu, and Keiichi Uchimura, "A Precise road network modeling and map matching for vehicle navigation," 11th International IEEE Conference on Intelligent Transportation Systems, 12-15 October, 2008, pp. 1084–1089.
- [12] Mohammed A. Quddus, "High integrity map matching algorithms for advanced transport telematics applications," Centre for Transport Studies, Imperial College London, United Kingdom, January, 2006.
- [13] Hong-mei Yin, and Shan-wu Su, "Modeling for geospatial database of national fundamental geographic information," IEEE International Conference on Geoscience and Remote Sensing, 2006, pp. 865–868.
- [14] Stephen Wise, "GIS Basics," Taylor & Francis, April, 2007.
- [15] Dean Pomerleau, "A neural network based autonomous navigation," Vision and Navigation, Kluwer Academic Publishers, 1990.
- [16] Surender K. Kenue, "LAMELOK: Detection of lane boundaries and vehicle tracking using image-processing techniques," Parts I and II, SPIE Mobile Robots IV, 1989.
- [17] E.D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen, "The seeing passenger car 'VaMoRs-P'," Proceedings of the Intelligent Vehicle Symposium, 1994, pp. 68–73.
- [18] Karl Kluge, and Charles Thorpe, "Representation and recovery of road geometry in YARF," Proceedings of the Intelligent Vehicles Symposium, 1994, pp. 1141–19.
- [19] Mathew A. Turk, David G. Morgenthaler, Keith D. Gremban, and Martin Marra, "VITS: A vision system for autonomous land vehicle navigation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, Issue 3, May, 1988, pp. 342–361.
- [20] Satish Shirali, and Harkrishan L. Vasudeva, "Metric spaces," Springer-Verlag London Limited, 2006.
- [21] Sulata Mitra, and Sipra DasBit, "On location tracking and load balancing in cellular mobile environment- A probabilistic approach," 5th International Conference on Electrical and Computer Engineering, December, 2008, Dhaka, Bangladesh.
- [22] Washington. Y. Ochieng, Mohammad A. Quddus, and R.B. Noland, "Map-matching in complex urban road networks," RBC - Revista Brasileira de Cartografia, N<sup>o</sup> 55/2, dezembro, 2003.
- [23] Khalid Touil, Mourad Zribi, and Mohammed Benjelloun, "Application of transferable belief model to navigation system, Integrated Computer-Aided Engineering," IOS Press, 2007, pp. 93–105.
- [24] Mohammad A. Quddus, Washington. Y. Ochieng, L. Zhao, and Robert .B. Noland, "A general map matching algorithm for transport telematics applications," GPS Solutions, 2003, pp.157–167.
- [25] Yu Meng, Zhilin Li, Wu Chen, and Yongqi Chen, "Reliability and integrity issues for vehicle positioning systems," The International Symposium on GNSS/GPS, Sydney, Australia, December, 2004.
- [26] Sudarshan S. Chawathe, "Segment-based map matching," Intelligent Vehicles Symposium, June, 2007, pp. 1190–1197.
- [27] Pen -Shan Hung, Tsui -Chuan Su, "Map-matching algorithm of gps vehicle navigation system," Geographic Information System Research Center, Reng Chia University, Taichung, Taiwan, 1998.
- [28] Mohammed A. Quddus, Washington Y. Ochieng, and Robert B. Noland, "Current map-matching algorithms for transportation applications: State-of-the art and future research directions," Transportation Research Part C, vol. 15, 2007, pp. 312–328.
- [29] Paul V. Bolstad, and James L. Smith, "Errors in GIS assessing spatial data accuracy," Journal of Forestry, November, 1992, pp. 21–29.
- [30] Paul V. Bolstand, Paul Gessler, and Thomas M. Lillesand, "Positional uncertainty in manually digitized map data," International Journal of Geographical Information Systems, vol. 4, 1990, pp.399–412.
- [31] R. Dunn, A. R. Harrison, and J. C. White, "Positional accuracy and measurement in digital database of land use: an empirical study," International Journal of Geographical Information Systems, vol. 4, 1990, pp. 385–398.
- [32] M. M. Yagoub, "The effect of psychophysics on the I of GIS/LIS," Geoinformatics and Socioinformatics, The Proceedings of Geoinformatics Conference, Ann Arbor, 19-21 June, 1999, pp. 1–9.
- [33] Helmut Alt, and Ludmila Scharf, "Computing the Hausdorff distance between curved objects," The 20th European Workshop on Computational Geometry, 2004, pp. 233–236.

- [34] Günter Rote, "Computing the Fréchet distance between piecewise smooth curves," The 20th European Workshop on Computational Geometry, 2004, pp. 147–150.
- [35] R. Sriraghavendra, K. Karthik, and Chiranjib Bhattacharyya, "Fréchet distance based approach for searching online handwritten documents", Ninth International Conference on Document Analysis and Recognition , vol. 1, 23-26 September, 2007, pp. 461–465.
- [36] Hui Ding, G. Trajcevski, and P. Scheuermann, "Efficient similarity join of large sets of moving object trajectories," 15th International Symposium on Temporal Representation and Reasoning, 16-18 June, 2008, pp. 79–87.