

Map Matching on Sparse GPS Data: A Perspective of a Developing City

Nova AHMED^a, Md Mahfuzur RAHMAN SIDDIQUEE^a, Refaya KARIM^a,
Mohsina ZAMAN^a, Rifat MONZUR^a, Moinul HOSSAIN^b

^a *ECE, North South University, Dhaka, Bangladesh*

^b *Dept. of Civil and Environmental Engineering, IUT, Dhaka, Bangladesh*

^a *E-mail: nova@northsouth.edu*

^b *E-mail: moinul048i@yahoo.com*

Abstract: It is interesting to observe unique challenges and opportunities faced in developing countries to solve a problem that has been extensively studied for years and acceptably resolved in the developed world. Map-matching is one of such areas of research. Existing well established map-matching algorithms are based on data mostly from the developed world where road networks are planned, GPS coverage is excellent and the data collection units used are consistent and/or of high quality. We have worked on Dhaka, the capital city of Bangladesh and collected trip data through crowd sourcing using our custom built smart phone based location data collection application to find a reasonable map matching algorithm. Our work on Geometry based, Fuzzy and a Naïve map-matching algorithm reveal that a simpler version of the existing algorithms is sufficient for a sparse road network and limited GPS coverage.

Keywords: Map matching, naïve algorithm, sparse road network, limited GPS coverage, crowd sourcing.

1. INTRODUCTION

Global Positioning System (GPS) and digital map, in the form of satellite image or Geographical Information Systems (GIS) have opened up various avenues in modern days by enabling planning, design and multi-dimensional applications. In transportation engineering, the design, planning and development can take into account parameters such as travel time, delay, flow, density, etc. and feed these into macro-meso-microscopic models for decision making. On the other hand, interesting applications can be developed that take into account such real-time parameters, e.g., real time emergency vehicle dissemination, alternate route detection, optimal route planning, logistics management, public transport dispatching, etc. However, to successfully achieve that, we need a methodology to connect the GPS location to a spatial structure on the map (e.g, roads, intersections, etc.). There has been a wide range of research work concerning map matching algorithms each with its unique opportunities and limitations. However, most of these algorithms (Quddus *et al.*, 2006; Pink and Hummel, 2008; Ioannis and Poulicos, 2007; Zhang and Gao, 2008; Ochieng *et al.*, 2003; Taghipour *et al.*, 2008) are designed considering developed world constraints. In general, the geometric patterns of roads in the developed countries are well defined, encroachment of right of way by unauthorized land-use is uncommon and the balance between accessibility and mobility are well maintained through traffic management or enforcement. Also, in most cases, existing map-matching algorithms are developed either based on data collected through experimental setup or from various corporate data sources, such as, taxi dispatching systems where the quality of data have high acceptability and uniformity (i.e., devices are similar in nature). The

scenario is quite different in the case of developing or under developed countries. The road networks are not well planned and often road hierarchy is not properly maintained. Here, GPS coverage is poor and often collecting data through the use of GPS enabled probe vehicle is expensive. Moreover, availability of limited number of companies dealing with vehicle dispatching through the use of GPS technology as well as lack of culture of sharing such data for research purpose make such research even more complex. Hence, it is important to investigate how the well-established map-matching algorithms which perform acceptably in the developed countries score in context of developing countries. We have focused our study in Dhaka, the capital city of Bangladesh which is well known as one of the most congested cities in the world. Regarding the issue of GPS data, we have employed crowdsourcing as GPS enabled smart-phones are common and the cell phone network with internet support is quite extensive – though the accuracy of the GPS of locally available smart-phones may not be comparable with those found in commercial fleet tracking systems which have often been used in developing the existing algorithms. From the point of view of the application of map-matching algorithms, often it is not important to match every GPS point with the base map. Rather, identifying which link the GPS location belongs to is sufficient. Hence, in this study we put to test the available map-matching algorithms to evaluate how well they can identify which link a GPS point belong to. Finally, we have also proposed a Naïve algorithm to see if is sufficient for our unique requirements.

The rest of the paper is organized in the following way: we discuss the related work in the next section, followed by opportunities, challenges and data collection methods in section 3, we discuss various map matching algorithms in section 4 followed by evaluation and lastly, we conclude our work in the final section.

2. LITERATURE REVIEW

It is interesting to see a large array of research ongoing in this area. A lot of research has been carried out to develop map matching algorithms. Map matching process normally comprises of 3 main components: GPS location, digital road map and an algorithm. Different map matching algorithm processes are mentioned below.

A probabilistic approach that uses Hidden Markov Model to carry out map matching is proposed by Pink and Hummel (2008). Robustness of this approach is improved by use of extended Kalman filter to introduce constraints for vehicular motion and of cubic spline polynomial to reconstruct the original road network topology. The method works well for even severely sparse GPS data. However, as no specific road segment is identified where the GPS points may lie, we are unable to employ this approach. Another probabilistic approach is proposed by Ioannis and Poulicos (2007), where first a rough possible neighbourhood is identified, then a transition probability is calculated to find out the possibility of the vehicle travelling to any of the neighbourhood roads. However, this method was not relevant to our work as no specific road segments are identified where the vehicle may be positioned. Thirdly, a topological map matching algorithm and its improvement is proposed by Li, Li, Shut & Wu (2007). Another approach to a map matching algorithm process can be an advanced technique, such as fuzzy logic. Quddus *et al.* (2006) proposed a map matching algorithm that uses the GPS location (longitude and latitude), heading error: the absolute difference between the direction of the vehicle and the direction of the link, the perpendicular distance from the position to the link, and the horizontal dilution of precision to initially identify the correct link road, using a Sugeno fuzzy algorithm. Then a second Fuzzy Inference System (FIS) is proposed to fix the link identified by the first FIS with the subsequent GPS position. Finally a

third FIS is used to identify the link the vehicle may be in after it crosses a junction or intersection. A similar approach was used by Zhang and Gao (2008) as mentioned above, where rather than using the third FIS for the link identification at junctions, they proposed computing a formulae using the GPS positions and the links found in the elliptic region around the GPS positions that will assign a weight to each route produced. The route with smallest weight is considered the best route. However, we have considered two major classes of algorithm, a simpler version of Fuzzy logic based map matching and simple version of Geometry based map matching algorithm. We have also considered a naïve algorithm that uses very simple methodologies in the calculation mechanism. We have considered the simpler versions considering the straight forward road network along with sparse GPS data availability.

3. CURRENT SITUATION, CHALLENGES AND DATA COLLECTION

It is interesting to note that Dhaka city, Bangladesh does not contain a fully connected GPS based map. There have been opportunities to create open source maps like Open Street Map but they are not matured and connected through the road network.

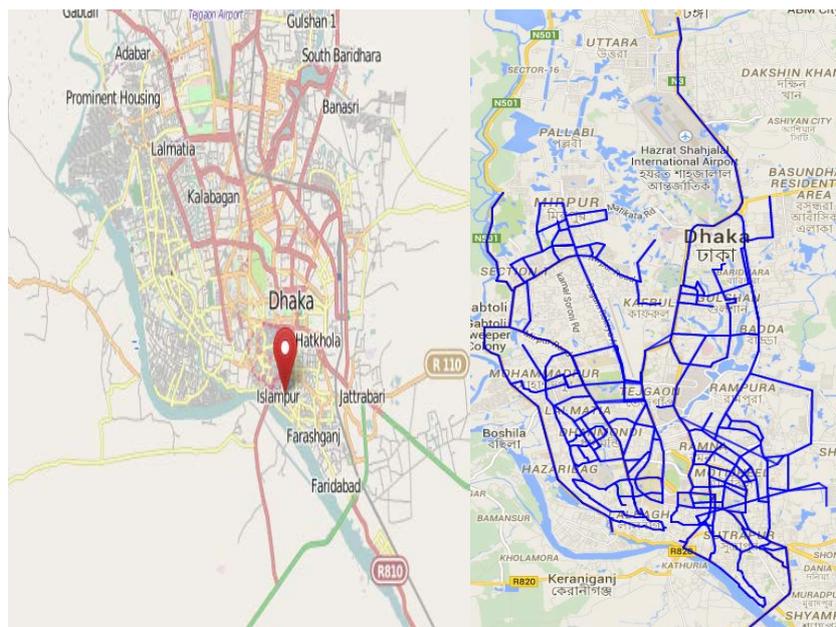
3.1 Open Street Map (OSM)

Open street map is an open source manually editable map, built by using as input, the GPS traces from any GPS device by users and satellite images. It creates and distributes free geographic data to the world.

In the Open Street Map community, users upload their GPS traces from casual trips to the website. There are specialized members of the community who gather all the traces for a particular area, analyze them manually, and create or edit the maps by hand.

3.2 Traditional Methods over Open Street Map

Open street map (OSM) was first used for showing the road network of Dhaka. The city of Dhaka was represented by a graph consisting of around 10^5 nodes and edges. This information was downloaded as an Extensible Markup Language (XML) file. However when a Depth First Search algorithm was run over the graph, we observed that the graph was disjoint. Certain roads that overlapped and should be joined were not actually joined. This problem was mainly because individual users independently gave their GPS device traces as inputs. OSM uses out-of-copyright maps and manual editing to create road maps freely available on the Web. As thousands of volunteers have contributed and edited data, the connectivity and geometry is dependent on all those independent volunteers and hence lacks consistency. This has led to the problem of a disjoint graph that we obtained when we analyzed the data of Dhaka.



(a) (b)
 Figure 1. (a) Open Street Map (b) Map we have developed

Due to the problem mentioned above, we manually created nodes and edges for the frequently travelled roads of Dhaka city. We identified the major roads and highways within Dhaka city by conducting some surveys and only plotted those roads and related nodes and edges on our map. The nodes represent the GPS positions which were collected using the mobile application. The edges are the connection between any 2 nodes, where the connections were derived from the trips which obtained by travelling the roads using vehicles equipped with the sensors.

We chose the traditional method because there will be no inconsistency in data as the map is not built on data that have been obtained by thousands of independent users. We plotted the graph for Dhaka city on Google map using 550 nodes and 1265 edges as can be seen in Figure 1-(a) showing Open street map and 1-(b) illustrating the map we have developed. When building the map if there is a bidirectional road from node A to node B then 2 edges are saved for such a road, i.e edge 1 – node A to node B, and edge 2 – node B to node A.

3.3 Data Collection

We have used crowdsourcing to collect GPS information as we do not have a very dense collection of data points in many of the major areas in Dhaka, Bangladesh. We have asked undergraduate and graduate students to collect GPS data using a very simple mobile application developed on Android Platform as seen in Figure 2. We have collected the GPS data over a full semester of four months' time frame.

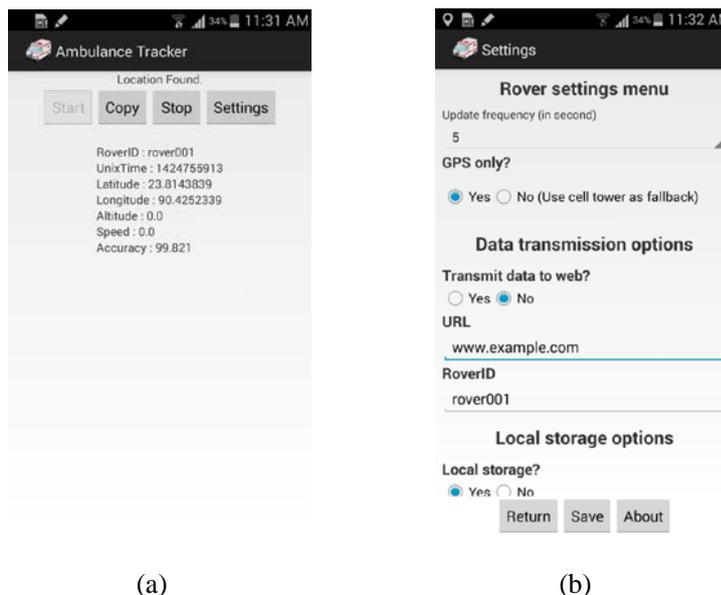


Figure 2. (a) Data Collection Interface (b) Parameter Setup Interface

We have worked hard to take into the limitations and find solution that are specific the particular region we are working on.

4. MAP MATCHING ALGORITHM

We have used two major map matching algorithm – one based on geometry, one based on fuzzy logic and finally we have developed a naïve algorithm. We intended to incorporate an algorithm using probabilistic modeling but due to the resource scarcity in modeling, we have not approached that method. We have simplified the standard algorithms to accommodate our unique challenges namely sparse GPS data and traffic data.

4.1 Algorithms using Geometry

Quddus *et. al.* (2006) describes a geometric approach of map matching algorithm as seen in Figure 3. They used weighting factors for determining correct path. To match the very first GPS point with map, they used closest pair technique. After matching the very first point, they determined the four unique weighting factors and matched with the path which has the maximum sum of weighting factors. They also determined the physical position between a path using the bearing of the link and vehicle speed from GPS. Sometimes, due to errors in GPS data, the GPS points are positioned in a wrong path. To resolve this issue, they used outlier to determine the correct path.

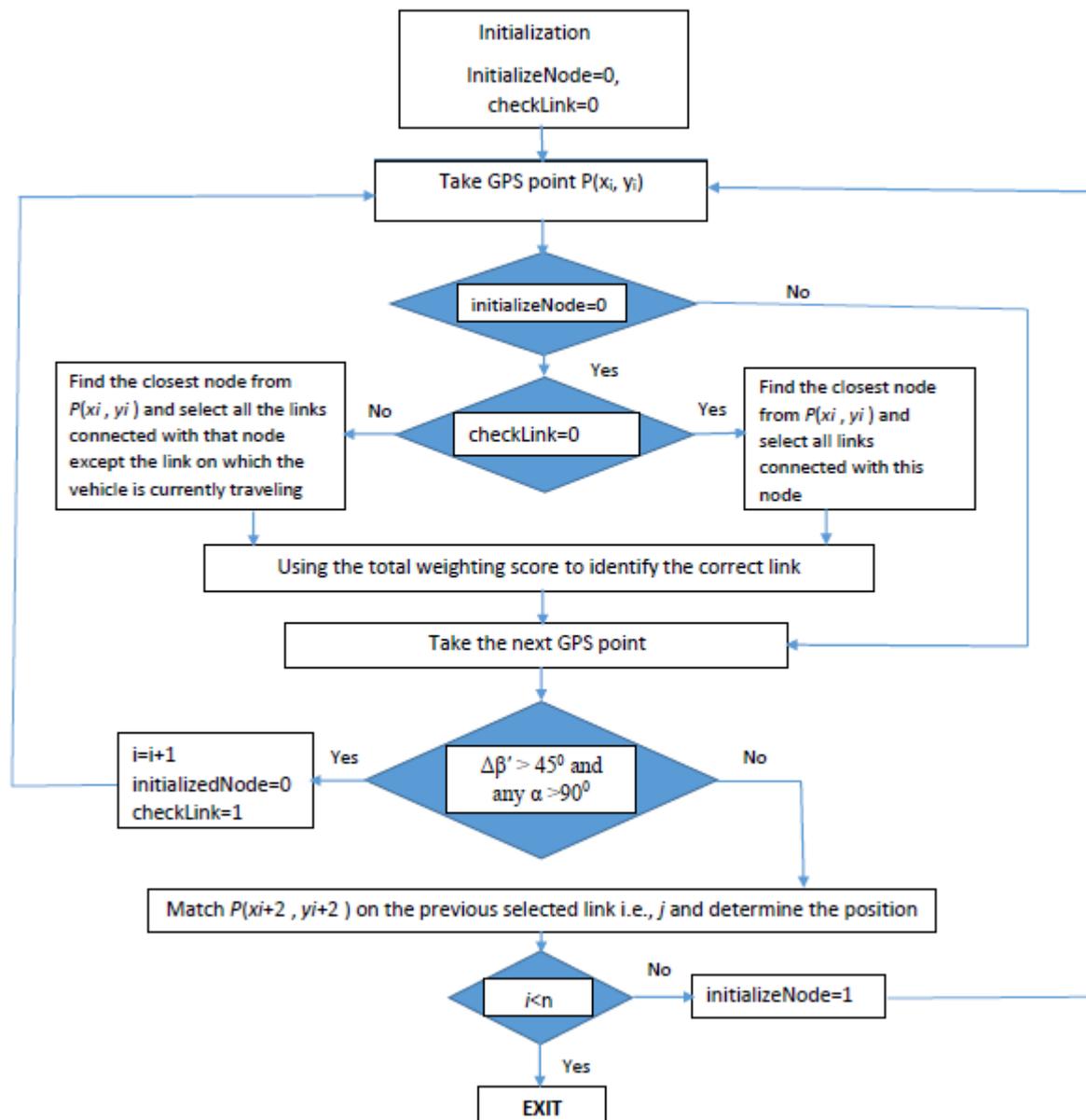


Figure 3. Flow chart of Quddus Algorithm

Taghipour *et al.* (2008) describes some improvements in determining the correct path. They also consider the length of the correct path and the width of the next intersection. They only consider changing the current path when it covers at least the half of the width of next intersection and length of the current path. However, we did not consider outlier and improvement of this algorithm because we only consider the main roads and so our graph is very sparse. As we did not need to determine the position between paths, we did not consider the speed and bearing of the link.

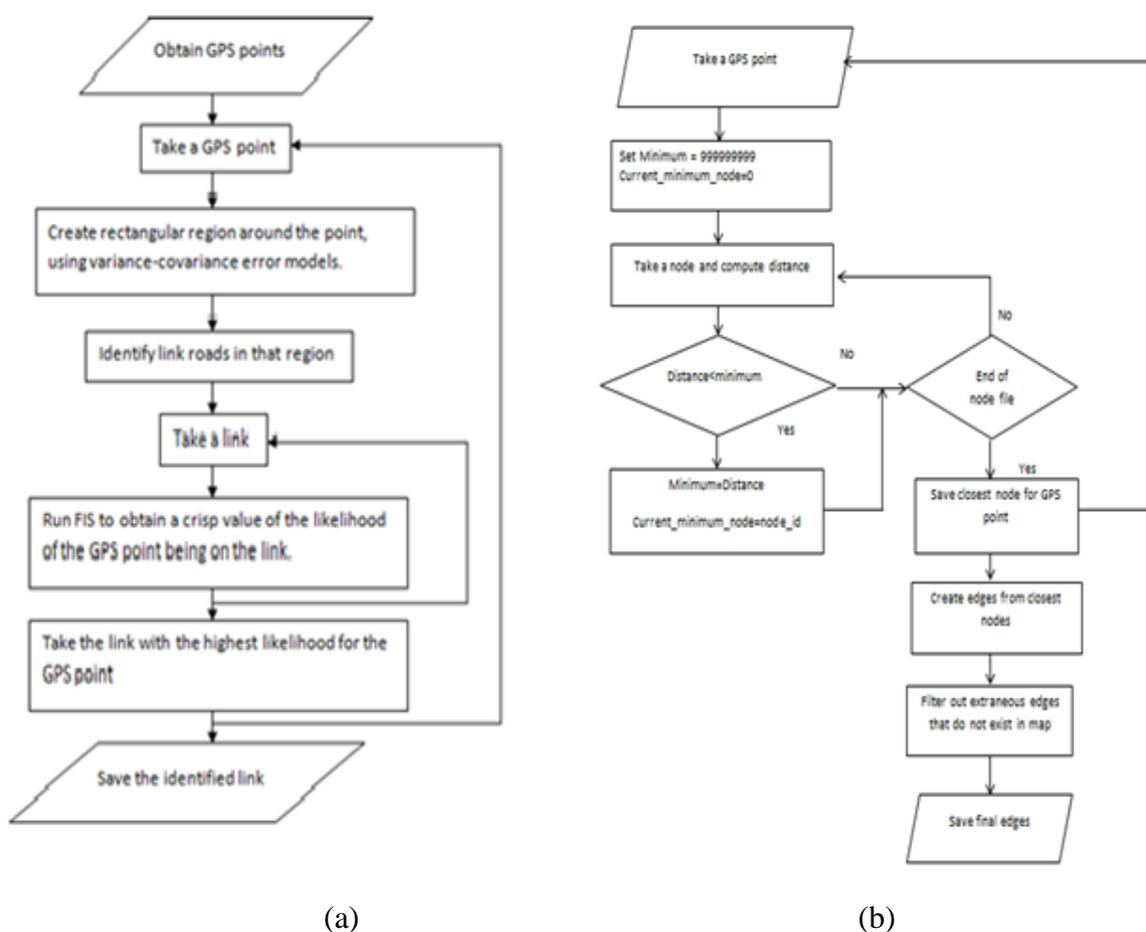


Figure 4 (a) Flow Chart of Fuzzy Algorithm (b) Flow Chart of Naïve Algorithm

4.2 Algorithms using Fuzzy Logic

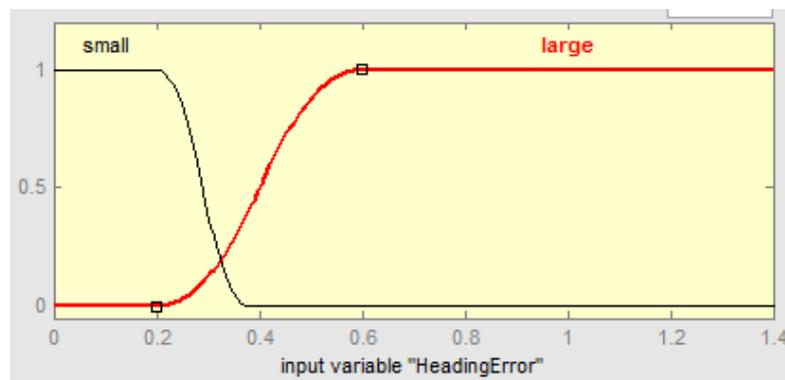
The fuzzy approach that has been considered in this paper implements the first FIS system of the map matching algorithm (MMA) that was proposed by Quddus *et al.* (2006) as it identifies the link a given GPS point is at, which is the objective. Given a GPS point, the inputs used by the FIS system were heading error, speed of the vehicle, and the shortest Euclidean distance of the GPS point to the link which is shown in Figure 4-(a).

According to Quddus *et al.* (2006), the most important variables for the initial FIS system are the heading error and the shortest Euclidean distance. Also, it is mentioned, variables such as speed could be used as the quality of the vehicle’s direction depends on the speed of the vehicle, and HDOP could be used as a quality indicator of the GPS point received.

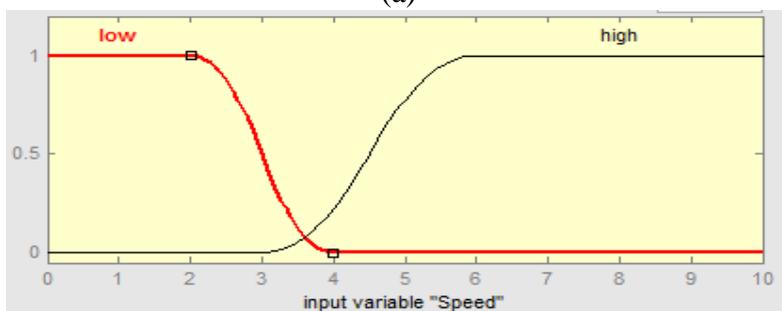
Unlike the MMA that was proposed by Quddus *et al.* (2006), we did not use the HDOP value as an input. This was because our collected data did not have that value due to the simplicity of the sensors that were used for data collection. A zero order Sugeno fuzzy model has been used. The input variable of the FIS are:

- Heading error: the absolute difference between the direction of the link and the GPS point.
- Euclidean distance between GPS point and link
- Speed of the vehicle.

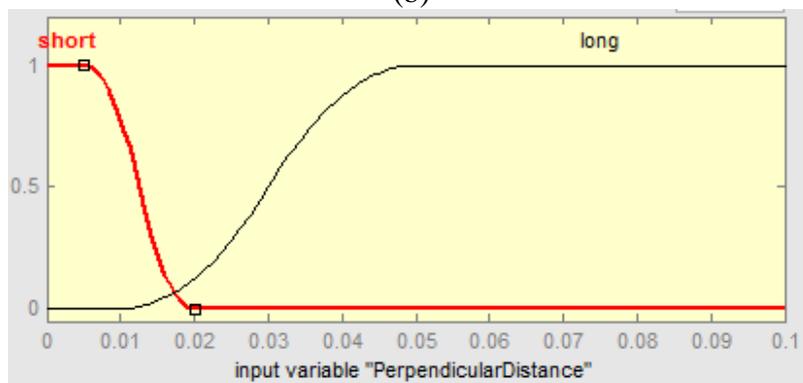
The fuzzified inputs to the FIS system are shown in Figure 5 (a, b and c).



(a)



(b)



(c)

Figure 5. Fuzzy System Parameters

Three constants for the output of the Sugeno FIS system have been considered. They are as follows: $low = 10$, $average = 50$, $high = 100$.

The following rules were applied to the FIS as shown in Figure 6:

1. If (HeadingError is small) and (Speed is high) then (Likelihood is average) (0.3)
2. If (HeadingError is large) and (Speed is high) then (Likelihood is low) (0.3)
3. If (HeadingError is large) and (PerpendicularDistance is long) then (Likelihood is low) (0.3)
4. If (HeadingError is small) and (PerpendicularDistance is short) then (Likelihood is high) (1)
5. If (PerpendicularDistance is short) then (Likelihood is average) (1)

Figure 6. Applied Rules for FIS

4.3 Naïve Algorithm

As our graph is sparse, we have attempted a naïve algorithm to carry out map matching from the collected GPS point. First, for each GPS point, we have identified for each GPS point a node on the map, which is closest to that point by computing minimum distance from all nodes in the map. Then edges were created by connecting the sequential closest nodes. Lastly, extraneous edges, which do not exist on our built map, were filtered out to obtain the final set of edges that the vehicle traversed. The algorithm is explained in Figure 4-(b).

5. RESULTS AND ANALYSIS

We have considered our results section to evaluate various map matching algorithms under the current situation of GPS location data in the city of Dhaka, Bangladesh.

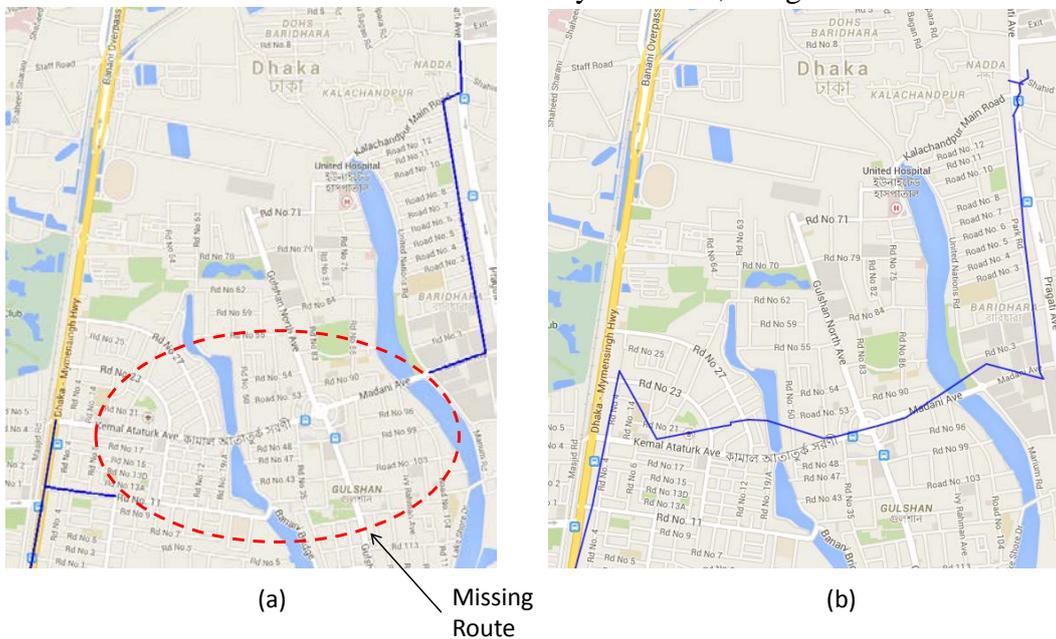


Figure 7. (a) Route Detected by Qudus Algorithm (b) Actual Route Travelled

We can see the Qudus Algorithm implemented in the given screenshots that shows the actual route along with the route Qudus algorithm has calculated as shown in Figure 7- (a) and Figure 7-(b). It is evident that some of the data points are missing which is mainly from the poor GPS data collected by our collection method.

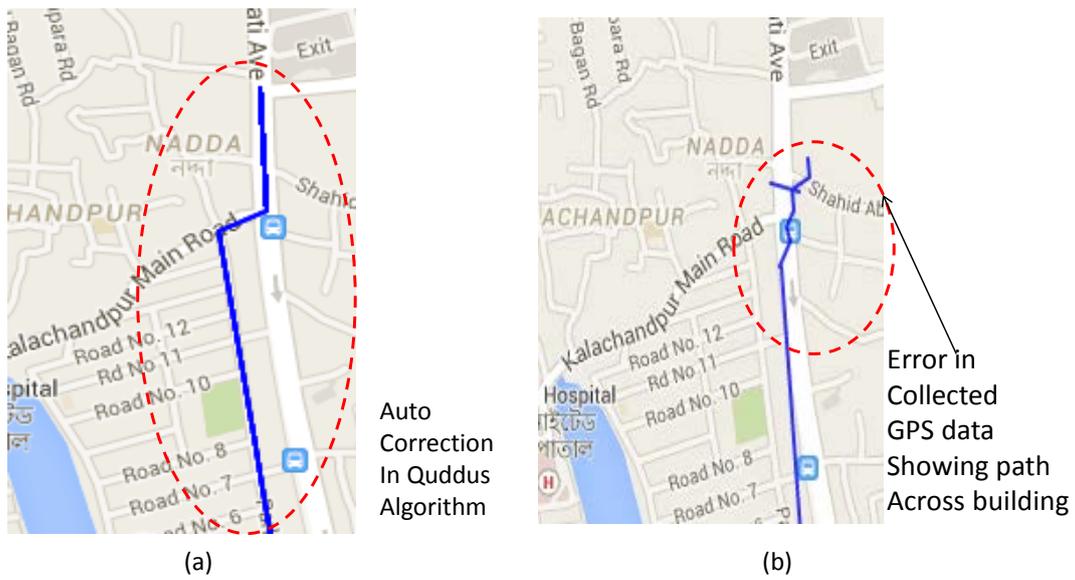


Figure 8. (a) Auto Correction of Route in Quddus Algorithm (b) Error Prone GPS Route

It is interesting to note how the auto correction mechanism in Quddus algorithm has been able to show an alternative route in the algorithm where the GPS location was error prone and showed a path through a building. This algorithm has calculated an alternative path beside the actual route as can be seen in Figure 8-(a) indicating the corrected result and Figure 8-(b) showing the result with error in GPS data. In summary we can mention that the major problem of Quddus Algorithm is missing links but on the other hand it has very few extra links. Complexity of the algorithm is $O(m*n*\log_2n)$; where m represents GPS points, n indicates nodes in map. For 300 GPS points time taken is 0.379 seconds.

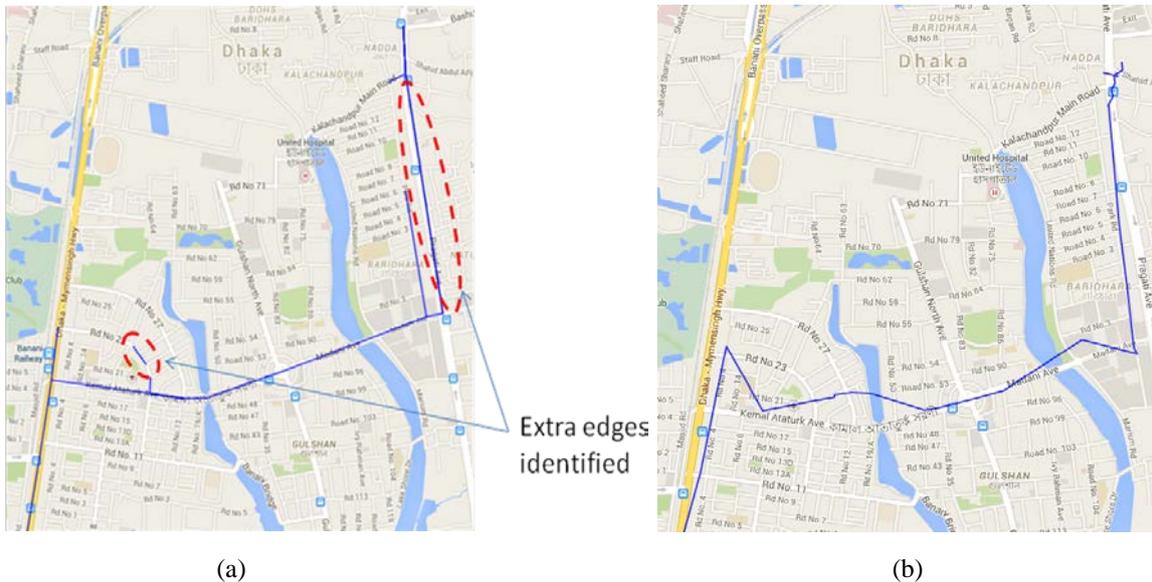


Figure 9. (a) Output of Fuzzy Algorithm (b) Actual route travelled

Unlike the Quddus algorithm, the Fuzzy Algorithm could identify the some of the missing links despite the poor GPS data collection. Fuzzy algorithm as shown on Figure 9-(a) and compared against the actual route on Figure 9-(b), shows very few missing links. However the problem with the fuzzy algorithm is that, when compared to the Quddus algorithm, it has more unwanted extra links being identified. It has a complexity of $O(m \cdot l)$; where m represents GPS points, l indicates selected links in the confidence region. The selected links, l , is a subset is the edges of the map that has been built, and the edges are a function of the nodes that make the map. So worst case complexity can be $O(m \cdot n^2)$, if there are n^2 edges and all the edges were found in the confidence region. However, in our map the edges are not n^2 in number but rather $2 \cdot n$, and also all the edges of the map cannot be found in the confidence region due to the variance-covariance error calculations. For 300 GPS points time taken is 0.36 seconds.

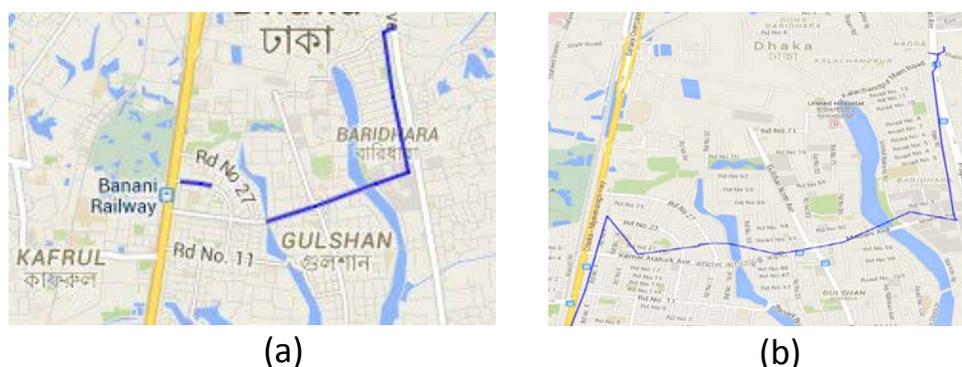


Figure 10. (a) Output of Naïve Algorithm (b) Actual route travelled

Analyzing Naïve algorithm illustrated in Figure 10-(a) we have many missing links and no extra edges compared to the other two algorithms which is one positive side of this algorithm. It has complexity of $O(m \cdot n^2)$; where m is the number of GPS points, n is number of nodes in map. For 300 GPS points time taken is 0.438 seconds.

It is evident that Quddus algorithm performs best among the three compared algorithms in terms of run time complexity and erroneous output. A modification of Naïve algorithm may be developed in future to reduce the run time complexity to improve real time performance.

5. CONCLUSION AND FUTURE WORK

We have worked on various map matching algorithms which take into account various limitations faced in a developing or under developed country. We have used real GPS data and road network data from the city of Dhaka, Bangladesh and have shared our journey through how we have overcome various challenges and compared contrasted versions of various map matching algorithms in this manuscript. Our contribution lies in the following areas:

- Developing our own map of Dhaka using crowd sourced GPS data
- Developing and analyzing two existing map matching algorithm with real data of Dhaka
- Proposing a simple naïve algorithm that works well without much complexity for our unique requirements

The incomplete road network and limited GPS coverage are two major challenges we have

faced and the existing algorithms have been changed based on our limited data availability. We are hopeful to accommodate our limitations in system development and implement systems that would open up new opportunities for the developing regions in coming days.

REFERENCES

- Quddus, M.A., Noland, A., Ochieng, W.Y., (2006) A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport, *Journal of Intelligent Transportation Systems: Technology, Planning and Operations*, 10(3), 103-115.
- Pink, O., Hummel, B., (2008), A statistical approach to map matching using road network geometry, topology and vehicular motion constraints, *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference*, Beijing, China, pp.862-867, 12-15 October.
- Ioannis, K., Poulicos, P., (2007), A Map Matching Algorithm for Car Navigation Systems with GPS Input. Paper presented at the International Conference on Geographic Information System, Aalborg University, Denmark, 8-11 May.
- Zhang, Y., Gao, Y., (2008), A Fuzzy Logic Map Matching Algorithm, *5th International Conference on Fuzzy Systems and Knowledge Discovery*, vol.3, 132-136, 18-20 October.
- Ochieng, W. Y., Quddus, M., Noland, R. B., (2003), Map Matching in Complex Urban Road Networks, *Brazilian Journal of Cartography*, 55 (2), 1-14.
- Taghipour, S., Meybodi, M.R.; Taghipour, A., (2008), An Algorithm for Map Matching For Car Navigation System, *3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA)*, 1-5, 7-11 April.
- Li, X., Li, M., Shu, W., Wu, M.(2007) A Practical Map Matching Algorithm for GPS-based Vehicular Networks in Shanghai Urban Area, *IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07)*, Shanghai, China, 454-457,12-14 December.