THE JOURNAL OF NAVIGATION (2009), **62**, 383–395. © The Royal Institute of Navigation
doi:10.1017/S0373463309005347 Printed in the United Kingdom

A Hidden Markov Model-Based Map-Matching Algorithm for Wheelchair Navigation

Ming Ren, Hassan A. Karimi

(*University of Pittsburgh*)
(Email: mren@sis.pitt.edu)

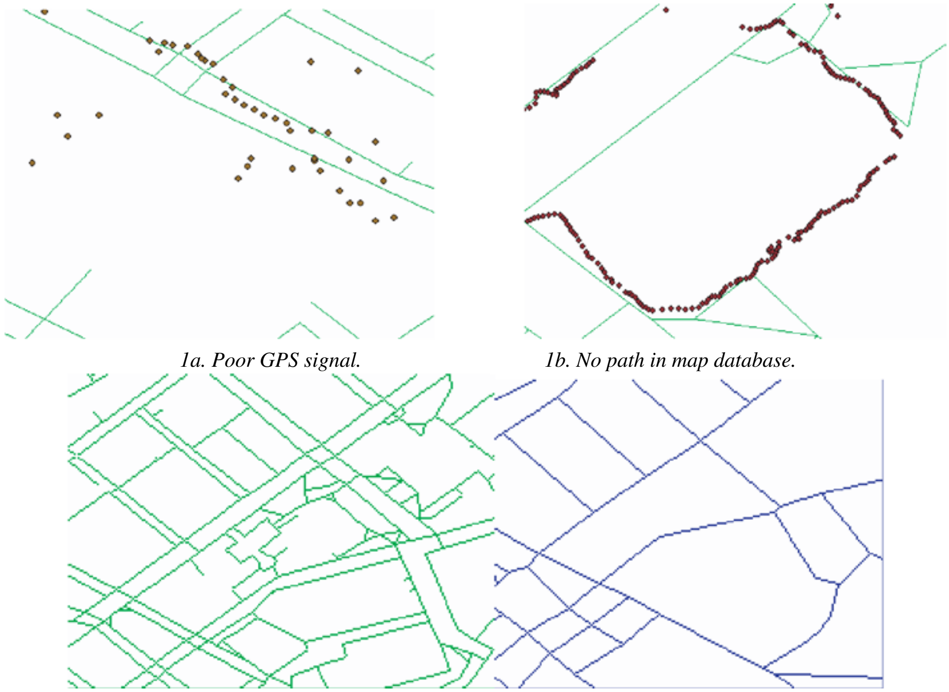
Application of map-matching techniques to GPS positions can provide accurate vehicle location information in challenging situations. The Hidden Markov Model (HMM) is a statistical model that is well known for providing solutions to temporal recognition applications such as text and speech recognition. This paper introduces a novel map-matching algorithm based on HMM for GPS-based wheelchair navigation. Given GPS positions, a hidden Markov chain model is established by using both geometric data and the topology of sidewalk segments. The map-matching algorithm employs the Viterbi algorithm to estimate correct sidewalk segments as hidden states in a HMM in order to match GPS trajectory on the corresponding segment sequence. The HMM-based map-matching algorithm was validated on a campus sidewalk network for wheelchair navigation. The results show an improvement in tracking a wheelchair in dense urban conditions both in accuracy and in computational time.

KEY WORDS

1. Hidden Markov Model. 2. Map-matching. 3. Wheelchair navigation.

1. INTRODUCTION. GPS-based wheelchair navigation systems use geopositioning and map-matching to compute the location of a wheelchair on a sidewalk (Ming and Karimi 2008). Wheelchair navigation systems deal with more challenging problems than do those for car navigation. Since wheelchair users outdoors usually take sidewalks that are close to buildings, their navigation systems are more susceptible to GPS signal loss or degradation than navigation systems used in cars. A further characteristic of wheelchair navigation, unlike car navigation, is that typically there are no other sensors to help replace or augment GPS data.

Two main error sources make this work challenging. First, the accuracy of GPS is always an issue in dense urban areas, where high buildings, among other obstacles, block satellite signals. Second, wheelchair navigation takes place on a sidewalk network instead of the road network used for car navigation, and this brings more difficulties in map-matching. The quality of a sidewalk network map database is affected by data collection techniques and the operating skill of a map maker. In addition, since most roads have sidewalks on both sides, a sidewalk network is much denser than its corresponding road network, so a challenge in wheelchair navigation



1a. Poor GPS signal.

1b. No path in map database.

1c. Sidewalk network (left) versus road network (right) in the same area.

Figure 1. Sources of inaccuracy in wheelchair navigation.

using GPS-based geo-positioning technique is determining on which side of the road the wheelchair is moving. This also makes a simple map-matching algorithm, like nearest road matching, unlikely to succeed. Additionally, wheelchair users sometimes move on a random path rather than follow the sidewalk. This further compounds the map-matching algorithm for wheelchair navigation. Examples of these cases are illustrated in Figure 1. Figure 1a shows how inaccurate the GPS signal is in some places where there are high buildings. Figure 1b shows the trajectory of a user who travelled along a route for which there was no corresponding sidewalk in the area's map database. Figure 1c provides a comparison of the density of a sidewalk network with that of its corresponding road network.

In general, map-matching algorithms integrate estimated locations from any kind of positioning sensors, not only GPS but also other positioning and orientation data, with spatial network data on a digital map to identify the correct link on which a vehicle is travelling and to determine the location of a vehicle on that link (Karimi et al 2006; Quddus 2006; Ochieng et al 2004). In this paper, we present an HMM-based map-matching algorithm for wheelchair navigation that will determine the sidewalk segment on which a wheelchair is located based on available GPS data projected onto an identified sidewalk segment. Although we concentrate only on GPS data in this paper, our algorithm is applicable to other positioning sensors, including odometer, compass readings, triangulation from Wi-Fi base stations or cell towers, and all such combinations.

The paper is organized as follows. Section 2 briefly overviews current methods for map-matching. Section 3 describes the proposed HMM-based map-matching algorithm. Section 4 presents the results of evaluating the HMM-based map-matching algorithm using the University of Pittsburgh campus sidewalk network. Conclusions and plans for future research are given in Section 5.

2. BACKGROUND.

Main approaches to map-matching include point-to-point, point-to-curve, curve-to-curve and topological map-matching (Karimi et al., 2006; Quddus 2006; Ochieng et al., 2004). The simplest way to match GPS data is to find the nearest point on the map based on point-to-point distances. Another technique to match GPS data to sidewalk segments is simply to select the nearest segment by computing perpendicular distance from each GPS point to all candidate segments on a map. Curve-to-curve map-matching algorithms use the similarity between the vehicle's trajectory and the curves as matching criteria. The most similar curve is selected as the correct segment. Topological map-matching (Meng et al., 2006; Quddus et al., 2003) considers both geometrical data and topological relationships of GPS positioning trajectory and candidate segments as the decision factors for map-matching. The candidate which earns the highest score from topological and geometric based calculations is considered as the vehicle's true location. In order to further enhance the matching accuracy, many advanced mathematical methods have been introduced in this field, such as a fuzzy logic approach and Kalman filters. Results show that these advanced approaches perform better than basic map-matching algorithms (Quddus et al 2007). However, all these map-matching techniques are developed and tested for car navigation where centreline road networks are used. The only map-matching algorithm for wheelchair navigation is the chain-code-based map-matching algorithm developed by Ming and Karimi (2008).

3. HIDDEN MARKOV MODEL MAP-MATCHING.

The Hidden Markov Model is a statistical model in which the system being modelled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden (unknown) parameters from the observable parameters (Wikipedia, Ephraim and Merhav 2002). The HMM has been used in temporal recognition applications such as text and speech recognition. We argue that map-matching is also a temporal recognition application susceptible to a Markov process where the aim is to find actual paths and actual locations, i.e., the hidden information, using GPS data as observed measurements. Map-matching, as a time-series problem, resembles temporal pattern recognition applications, such as speech, handwriting, gesture recognition and bioinformatics, where the hidden Markov model is applied. With these characteristics, a map-matching algorithm based on HMM, where finding the correct sidewalk segment amongst all candidate sidewalk segments given a GPS trajectory, is set forth in this research.

The Viterbi Algorithm (Forney 1973) is a recursive optimal solution to the problem of estimating the state sequence of a discrete-time finite-state Markov process. Many problems can be cast in this form. We applied the Viterbi algorithm to estimate the sidewalk segments based on observed GPS positions. The key innovation using HMM in this algorithm for wheelchair navigation is matching sidewalk segments

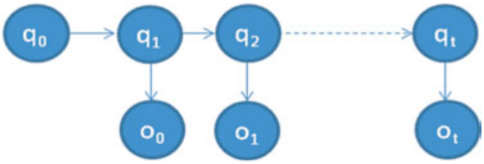


Figure 2. Architecture of an HMM.

based not only on the geometry of the location readings, but additionally on the topology of the segments.

3.1. *Hidden Markov Model.* The HMM is represented by a finite set of states, each of which is associated with a probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state an outcome or observation can be generated according to the associated probability distribution. It is only the outcome, not the state, that is visible to an external observer and therefore states are 'hidden' to the outside; hence the name Hidden Markov Model (Rabiner 1989). The general architecture of a hidden Markov model is shown as Figure 2.

The architecture has two layers: $\{o_t\}$ represents the observable layer and o_t corresponds to an observation value at time t. $\{q_t\}$ represents the hidden layer, and q_t, at time t, comes from one state in a state space. In order to model a hidden Markov process, the following elements are needed:

- The number of states in the model, n.
- The number of observations, m. If the observations are continuous then m is infinite.
- A set of state transition probabilities. $A=\{a_{ij}\}$

$$a_{ij}=p_r\{q_{t+1}=j|q_t=i\}, 1\leqslant i,j\leqslant n \tag{1}$$

where q_t denotes the state at time t.

- An observation probability distribution in each of the states, $B=\{b_j(k)\}$.

$$b_j(k)=p_r\{o_t=o_k|q_t=j\},\ 1\leqslant j\leqslant n,\ 1\leqslant k\leqslant m \tag{2}$$

where o_t is the observation at time t and o_k denotes the k_{th} observation.

- The initial state distribution, $\pi=\{\pi_\mathrm{i}\}$, where,

$$\pi=p_r\{q_1=i\}, 1\leqslant i\leqslant n \tag{3}$$

With these, $\lambda=(\pi, A, B)$ can be used to denote an HMM with probability distributions.

3.2. *A Hidden Markov Model for Map-Matching.* In a hidden Markov model, the state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution over the possible observations. Therefore, the sequence of observations provides some information about the sequence of states by means of a HMM (Cappé et al., 2005). Given the parameters of the model, the Viterbi

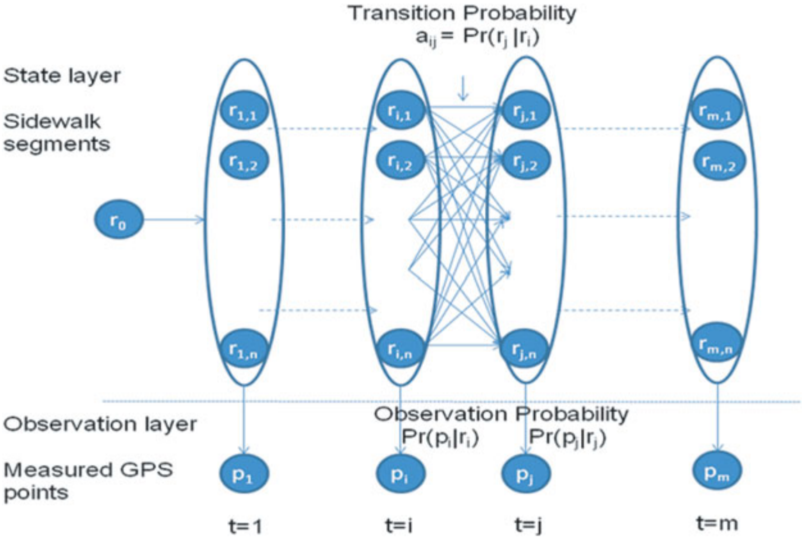


Figure 3. The hidden Markov model for map-matching.

algorithm can solve the problem of how to find the most likely sequence of hidden states that could have been generated by using a given observed sequence. The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states, called the Viterbi path. In map-matching for wheelchair navigation, observed GPS points are the visible observation layer and correct sidewalk segments are the invisible state layer.

Let $Pt \in \{p_1, p_2, \ldots, p_m\}$ denote the observation (i.e., a GPS data point) obtained every second t for $1 \leqslant t \leqslant m$.

Let $Rt \in \{r_1, r_2, \ldots, r_n\}$ denote the actual location (i.e., the correct sidewalk segment) at time t.

Suppose we obtain a series of GPS observations within the time period m, so we could obtain m GPS points as an observation sequence from time t_1 to time t_m. In the state space, there are n states, which represent n candidate segments. The transition probability from any time i to the next time j represents the probability of a user moving from one segment to another segment. The model could be structured as shown in Figure 3. The goal is to find the sidewalk segment sequence that has maximum probability given the observations. That is, finding a sequence of actual locations, $R_1 \ldots R_t$, such that $Pr(R_1R_2 \ldots Rt|P_1P_2 \ldots P_t)$ is maximized.

Based on conditional probabilities from basic probability theory, for any sequence $R_1 \ldots R_t$ of actual locations we have:

$$\Pr(R_1R_2 \ldots R_t|P_1P_2 \ldots P_t) = \frac{\Pr(R_1R_2\_\mathrm{R_tP_1P_2\_P_t})}{\Pr(P_1P_2\_\mathrm{P_t})} \tag{4}$$

Given the observations, the denominator of this expression is determined (the exact value is unknown, but that value only depends on the observations, not on the path $R_1 \ldots R_t$.). So the problem is equivalent to finding $R_1 \ldots R_t$ such that $Pr(R_1R_2 \ldots R_tP_1P_2 \ldots P_t)$ is maximized.

From the basic identities of probability theory, for any events A, B, C we have, $Pr(ABC)=Pr(A)Pr(B|A)Pr(C|AB)$. We use this to decompose the complicated event:

$R_1R_2 \ldots R_tP_1P_2 \ldots P_t$ as a product ABC. We define $A=R_1 \ldots R_{t-1}P_1 \ldots P_{t-1}$, $B=R_t, C=P_t$,

By applying the above formula, we obtain:

$$\begin{aligned} Pr(R_1R_2 \ldots R_tP_1P_2 \ldots P_t) = & Pr(R_1R_2 \ldots R_{t-1}P_1P_2 \ldots P_{t-1}) \\ & \times Pr(R_t|R_1 \ldots R_{t-1}P_1 \ldots P_{t-1}) \\ & \times Pr(P_t|R_1 \ldots R_{t-1}P_1 \ldots P_{t-1}R_t). \end{aligned}$$

Furthermore, we obtain:

$$\begin{aligned} Pr(R_1 \ldots R_tP_1 \ldots P_t) &= Pr(R_1 \ldots R_{t-1}P_1 \ldots P_{t-1})Pr(R_{t-1} \rightarrow R_t)\,\mathrm{Pr}\,(R_t \rightarrow P_t) \\ &= Pr(R_0)\prod\nolimits_{t=0}^{T} \mathrm{Pr}\,(\mathrm{R_t}|\mathrm{R_{t-1}}) \prod\nolimits_{\mathrm{t}=0}^{\mathrm{T}} \mathrm{Pr}\,(\mathrm{Pt}|\mathrm{Rt}). \end{aligned} \tag{5}$$

We assume each probability in the state transition matrix and in the observation probability matrix in the HMM is time independent. Therefore, given prior probability $Pr(R_0)$, observation probability $Pr(P_t|R_t)$ and state transition probability $Pr(R_t|R_{t-1})$, we can use the Viterbi algorithm to find the path through the states that maximizes the probability of a sequence of sidewalk segments. The Viterbi algorithm uses dynamic programming methods to efficiently accomplish this, so that the actual path consisting of a sequence of sidewalk segments can be identified.

In Equation (5), in order to apply the Viterbi algorithm, we need to know prior probability, observation probability and state transition probability in the HMM. Prior probability $Pr(R_0)$ is $Pr(r_j)$, when $j=1, \ldots, n$, which is simply computed by $1/n$ as a uniform distribution reflecting the fact that we have no known bias about which is the correct sidewalk segment. Hence, how to compute observation probability and state transition probability becomes the key point.

First, we compute the observation probability, which is the probability of the measured location p_i given r_j. We can compute this with the Bayesian rule:

$$\mathrm{P_r(p_i|r_j)} = \frac{\mathrm{P_r(r_j|p_i)\,Pr\,(p_i)}}{\sum_{\mathrm{k}=1}^{\mathrm{n}} \mathrm{Pr(r_k|p_i)\,Pr\,(p_i)}} \qquad i=1,2\ldots,m; \quad j=1,2\ldots n \tag{6}$$

We presume that Pr(pi), a prior probability in Equation (6), follows a uniform distribution. Therefore, Equation (6) could be further simplified as:

$$\mathrm{P_r(p_i|r_j)} = \frac{\mathrm{P_r(r_j|p_i)}}{\sum_{\mathrm{k}=1}^{\mathrm{n}} \mathrm{p_r(r_k|p_i)}} \qquad i=1,2\ldots,m; \quad j=1,2\ldots,n \tag{7}$$

$\mathrm{Pr(r_j|p_i)}$ is the probability that r_j is the correct sidewalk segment out of the candidate sidewalk segments given that measured location is P_i. We computed this by assuming that, for most of the GPS points, the closer a sidewalk segment is to the observed point, the higher the probabilities that it is the correct segment. This is borne out by our informal observations of nearest segment matching. Considering the relationship of distance and observation probability as an inverse proportion, we first compute the probability of the perpendicular distance from GPS point p_i to the segment $\mathrm{r_j}$ over the summation of the distances from p_i to all the candidate segments, and then use

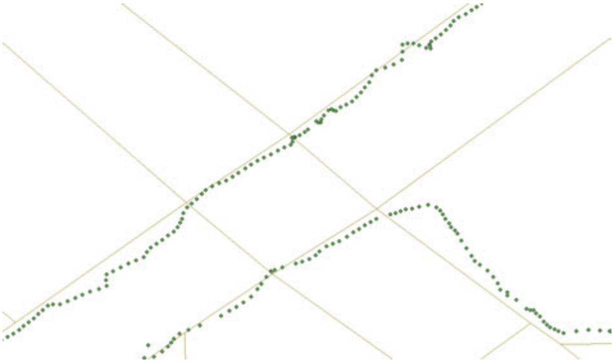


Figure 4. An example of GPS points overlaid on sidewalks on campus.

reciprocal relation of the probability based on distances to approximate observation probability. This leads to:

$$\begin{aligned}\mathrm{P_r}(\mathrm{r_j}|\mathrm{p_i}) &= \frac{\textit{Expected } \{\textit{number of time in segment } \mathrm{r_j}|\textit{GPS measured location } \mathrm{p_i}\}}{\textit{Expected } \{\textit{all the times in all possible segments } \mathrm{r_1} \ldots \mathrm{r_n}|\textit{GPS measured location } \mathrm{p_i}\}} \\ &= 1/\frac{\textit{Distance from } \mathrm{p_i} \textit{ to } \mathrm{r_j}}{\sum_{k=1}^{n} \textit{Distance from } \mathrm{p_i} \textit{ to } \mathrm{r_k}} \qquad (8)\end{aligned}$$

In wheelchair navigation, wheelchair users either move on the same segment, or they make a turn at a junction such as an intersection, exit, or entrance. Therefore, we need to compute the transition probability a_{ij}, which represents the probability of the wheelchair user moving from one sidewalk segment corresponding to a measured point to another sidewalk segment corresponding to another measured point. For this, in this algorithm we use topological relationship to compute the transition probability. We only consider three topological relationships: same segment, connected segment and unconnected segment. It is impossible for a wheelchair user to move from a segment to an unconnected segment in consecutive time windows. Therefore, the transition probability from time i to the next time $j=i+1$, a_{ij} would be zero where the two segments are not connected. If two sidewalk segments are connected, this transition probability should be higher than if two sidewalk segments are unconnected, since wheelchair users would travel on the same segment most of the time except at an intersection or junction. Thus, the transition probability of moving on the same segment has the highest value. By setting $a_{ij}=e^{-rij}$, we create an exponential curve for this probability distribution, where r_{ij} corresponds to the topological relationship between two segments. By normalization, a_{ij} changes between 0 and 1. The next important step is to build a transition matrix $\{r_{ij}\}$ and set the value for each element in this matrix. The following set of rules must be followed

(1) If two segments are connected, r_{ij} is set to 1;
(2) If they are unconnected, then r_{ij} is set to ∞.
(3) Otherwise, r_{ij} is 0, when two segments are the same, that is $i=j$.

Take our measured GPS points on campus as an example, shown in Figure 4. First, we model sidewalk segments as a set $\{r_1,r_2, \ldots, r_{12}\}$ as Figure 5 shows. Next, we build a matrix $\{r_{ij}\}$, based on the topology of the segments in Figure 6. Figure 7 shows the

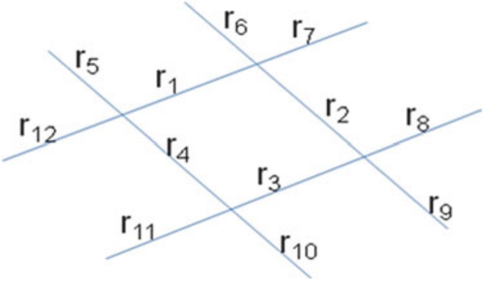


Figure 5. An abstracted sidewalk network model.

$$\begin{array}{lcccccccccccc}
 & r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 & r_9 & r_{10} & r_{11} & r_{12} \\
r_1 & 0 & 1 & \infty & 1 & 1 & 1 & 1 & \infty & \infty & \infty & \infty & 1 \\
r_2 & 1 & 0 & 1 & \infty & \infty & 1 & 1 & 1 & 1 & \infty & \infty & \infty \\
r_3 & \infty & 1 & 0 & 1 & \infty & \infty & \infty & 1 & 1 & 1 & 1 & \infty \\
r_4 & 1 & \infty & 1 & 0 & 1 & \infty & \infty & \infty & \infty & 1 & 1 & 1 \\
r_5 & 1 & \infty & \infty & 1 & 0 & \infty & \infty & \infty & \infty & \infty & \infty & 1 \\
r_6 & 1 & 1 & \infty & \infty & \infty & 0 & 1 & \infty & \infty & \infty & \infty & \infty \\
r_7 & 1 & 1 & \infty & \infty & \infty & 1 & 0 & \infty & \infty & \infty & \infty & \infty \\
r_8 & \infty & 1 & 1 & \infty & \infty & \infty & \infty & 0 & 1 & \infty & \infty & \infty \\
r_9 & \infty & 1 & 1 & \infty & \infty & \infty & \infty & 1 & 0 & \infty & \infty & \infty \\
r_{10} & \infty & \infty & 1 & 1 & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty \\
r_{11} & \infty & \infty & 1 & 1 & \infty & \infty & \infty & \infty & \infty & 1 & 0 & \infty \\
r_{12} & 1 & \infty & \infty & 1 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & 0
\end{array}$$

Figure 6. State transition matrix.

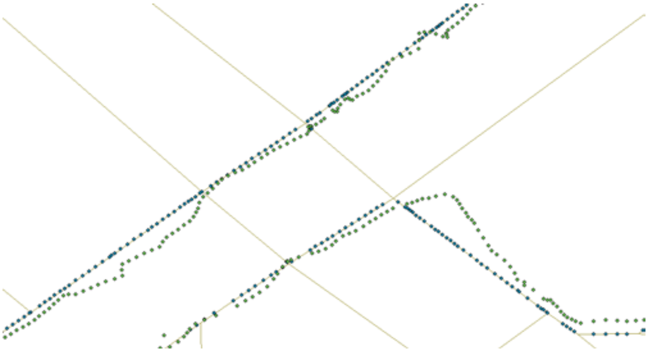


Figure 7. Map-matching locations versus GPS positions.

map matching results by applying the Viterbi algorithm to the entire sequence of location measurements shown in Figure 4.

3.3. *HMM-based Map-matching Process.* For a hidden Markov model, two parameters, n and m, have to be initialized, where m is the size of an observation sequence and n is the state number in a state space. For map-matching, the size of the observation sequence is the number of measured GPS points and the state number is

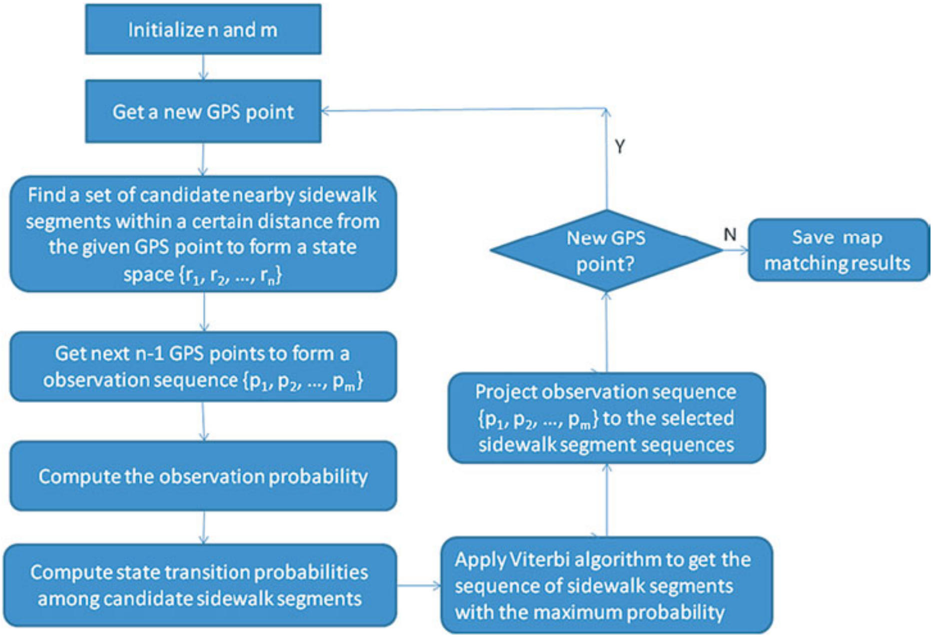

Initialize n and m
Get a new GPS point
Find a set of candidate nearby sidewalk segments within a certain distance from the given GPS point to form a state space {r_1, r_2, ..., r_n}
Get next n-1 GPS points to form a observation sequence {p_1, p_2, ..., p_m}
Compute the observation probability
Compute state transition probabilities among candidate sidewalk segments
Apply Viterbi algorithm to get the sequence of sidewalk segments with the maximum probability
Project observation sequence {p_1, p_2, ..., p_m} to the selected sidewalk segment sequences
New GPS point?
Y
N
Save map matching results

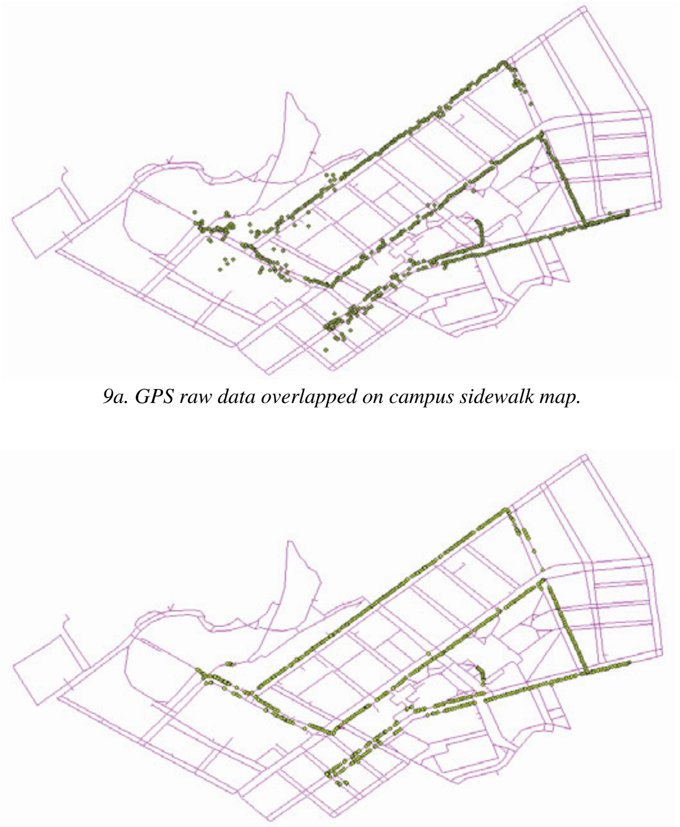
Figure 8. Flowchart of HMM-based map-matching process.

the number of candidate sidewalk segments close to the observed GPS points. In this algorithm, after setting these two values, we take several steps to complete the matching process.

First, a set of nearby candidate sidewalk segments is chosen based on the first GPS data observed in each sequence. Second, the transition matrix on the selected set of nearby candidate sidewalk segments is built (see Figure 6). This matrix not only shows the topology of segments but implies two moving modes, which are changing mode and continuing on same segment mode. In the case of continuing on same segment mode, where r_{ij} is equal to 0, current and previous positions should be matched on the same segment. Conversely, if r_{ij} is 1, then the wheelchair is moving in a changing mode, where current and previous positions are on two connected segments. Consequently, we could compute transition probabilities based on the transition matrix. Third, the perpendicular distance from each GPS point to each segment in the set of candidate sidewalk segments is computed, so that observation probabilities for each measured location are calculated. Last, the Viterbi algorithm to the observation probabilities and transition probabilities to compute the maximum probability sequence of sidewalk segments are applied. Once the most likely sidewalk segment is obtained, GPS points are projected to the segments and the map-matching result is shown on the map. The flowchart of the process is shown in Figure 8.

4. VALIDATION.

To validate the HMM-based map-matching algorithm developed in this work for wheelchair navigation, the sidewalk data along with associated parameters on the University of Pittsburgh campus area were digitized and utilized. The sidewalk database, consisting of the sidewalk network, buildings,



9a. GPS raw data overlapped on campus sidewalk map.

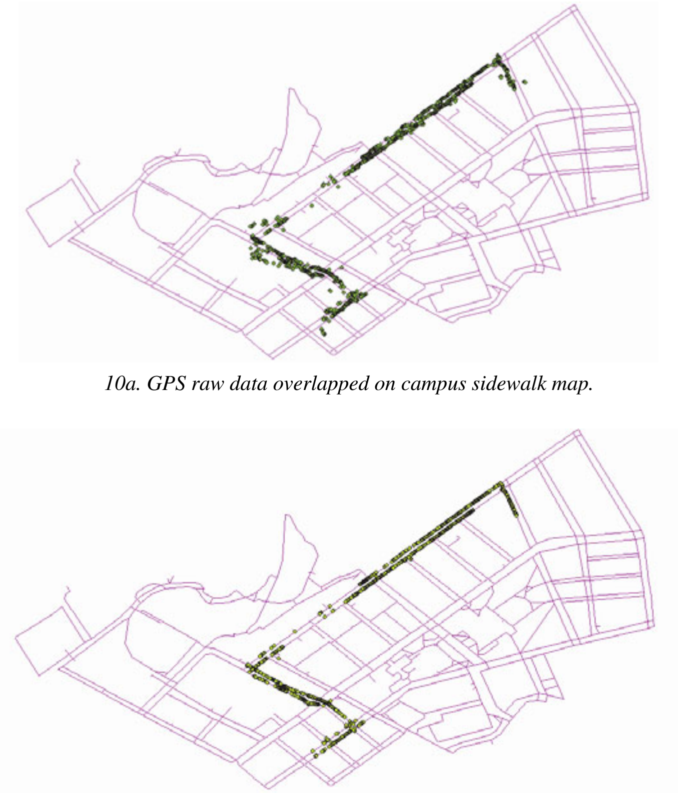
9b. Projected result data to the matched sidewalk segments on campus sidewalk map.

Figure 9. Route 1 comparing map-matching result with GPS raw data on campus sidewalk map.

landmarks, and accessibility information, are built for wheelchair navigation in order to assist wheelchair users travelling outdoors (Kasemsuppakorn and Karimi 2008). For the testing, GPS points were collected by walking and using a stand-alone GPS receiver, and map-matched to the established sidewalk network. The computing platform used was a PC machine with Intel Core 2 1.4G HZ CPU. The software for the HMM-based map-matching algorithm was written in JAVA in an open source GIS tool called Geotools (Geotools 2008).

4.1. *Analysis of Results*. Three groups of data sets, collected on main campus of University of Pittsburgh by GPS receiver, were processed to validate the presented algorithm. Route 1 covered the most main sidewalks on campus; Route 2 included the sidewalks around high buildings; Route 3 included a loop and some small paths. Fully considering the various types of sidewalks on the different areas, we used the three selected routes to test the HMM-based map-matching algorithm. In comparison, three-route GPS raw data with map-matching results were overlapped on campus sidewalk map, shown in Figures 9, 10, and 11.

Model parameters n and m were specified through experiments. Based on the topology extracted from the campus sidewalk data, the size of the state space, i.e., the



10a. GPS raw data overlapped on campus sidewalk map.

10b. Projected result data to the matched sidewalk segments on campus sidewalk map.

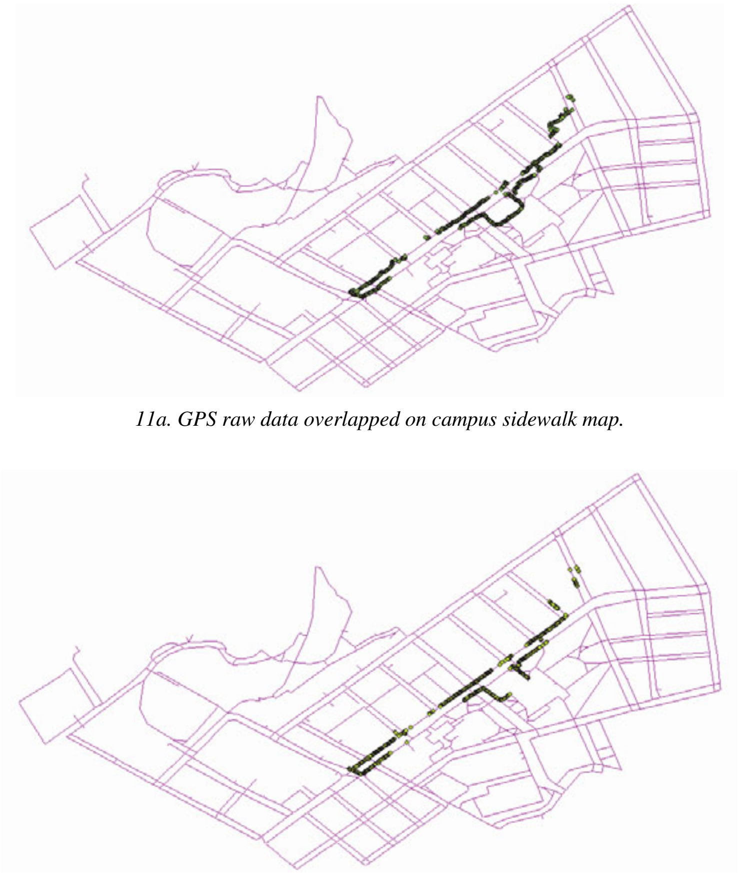
Figure 10. Route 2 comparing map-matching result with GPS raw data on campus sidewalk map.

number of segment candidates in one map-matching process, notated by n, was set as twelve. Experiments using 3- to 8-point sequence were conducted to determine the suitable number of points in one sequence. It was realized that for the real-time requirement of map-matching a 4-point sequence is appropriate for this HMM-based map-matching algorithm. The map-matching performances are presented in Table 1. The average time per four points represents the average time taken for one sequence matching computation. In the offline matching, the total computation time shows the total time to complete the matches of all GPS points in one route. Since correct link identification after applying a map-matching algorithm and average computation time are the most important performance parameters for evaluation, we use statistical data to show that this algorithm performs well and satisfies the requirements of real-time map matching in wheelchair navigation.

As with all map-matching algorithms, there are mismatched points due to errors in geo-positioning systems and the digital map quality, and both affect the performance of the map-matching algorithm. We observe that most mismatched points in Route 3 occur when the data collector moved on paths with no corresponding segments on the digital map. Meanwhile, in the case of Routes 1 and 2, we realize that many mismatched points occur on sidewalks of narrow roads due to GPS errors.

Table 1. Performance results.

Route	Total Number of GPS Points	Number of Mismatched Points	Correct Link Identification After HMM Correction(%)	Total Computation Time (s)	Average Time/4 Points (ms)
1	682	52	92·4%	0·625	3·666
2	1516	70	96%	1·406	3·709
3	933	68	92·7%	0·859	3·682



11a. GPS raw data overlapped on campus sidewalk map.

11b. Projected result data to the matched sidewalk segments on campus sidewalk map.

Figure 11. Route 3 comparing map-matching result with GPS raw data on campus sidewalk map.

5. CONCLUSIONS AND FUTURE RESEARCH.

In this paper, we presented a novel map-matching algorithm to estimate wheelchair location in sidewalk networks. The HMM-based map-matching algorithm matches with high accuracy GPS data to segments based on finding an optimal compromise between GPS data and topological structure, implicitly accounting for the topology of the sidewalk network. This trade-off is accomplished with a hidden Markov model. The algorithm was tested in many situations where it successfully ignored complex distractions to find the correct path. However, some GPS points were still mismatched due to failure in differentiating between the two sides of narrow roads. Future work on this algorithm should include a more careful characterization of topological data. Map-matching results may be more accurate if we could account for the fact that transition probability between two candidate sidewalk segments changes with a wheelchair moving. In addition, applying this algorithm to other navigation environments, such as car navigation, will further validate its appropriateness for all navigation applications.

REFERENCES

Cappé, O., Moulines, E., Rydén, T. (2005). Inference in Hidden Markov Models, *Published by Springer*.

Ephraim, Y. and Merhav, N. (2002). Hidden Markov processes, *IEEE Trans. Inform. Theory*, vol. 48, pp. 1518–1569.

Forney, G. D. (1973). The Viterbi algorithm. *Proceedings of the IEEE* 61(3): 268–278.

Geotools (2008). The java GIS toolkit, http://sourceforge.net/projects/geotools/

Hidden Markov model. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Hidden_Markov_model

Jagadeesh, G. R., Srikanthan, T. and Zhang, X. D. (2004). A Map Matching Method for GPS Based Real-Time Vehicle Location, *Journal Of Navigation*, **57**, 429–440.

Karimi, H A, Conahan, T. and Roongpiboonsopit, D. (2006). A Methodology for Predicting Performances of Map-Matching Algorithms, *W2GIS* 202–213.

Kasemsuppakorn, P. and Karimi, H. A. (2008). Data requirements and spatial database for personalized wheelchair navigation, *2nd International Convention on Rehabilitation Engineering & Assistive Technology*.

Krumm, J., Letchner, J. and Horvitz, E. (2007). Map Matching with Travel Time Constraints, *SAE 2007 World Congress*, April 16–19.

Meng, Y. (2006). Improved Positioning of Land Vehicle in ITS Using Digital Map and Other Accessory Information, *PhD Thesis*, Department of Land Surveying and Geoinformatics, Hong Kong Polytechnic University.

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

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

Quddus, M. A., Noland, R. B., Ochieng, W. Y. (2004). Validation of map-matching algorithm using high precision positioning with GPS, *Journal of Navigation* **58**, 257–271.

Quddus, M. A. (2006). High Integrity Map Matching Algorithms for Advanced Transport Telematics Applications *A thesis of the University of London*.

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

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77 (2), p. 257–286.

Ren, M., Karimi, H. A. (under review, 2008). A Chain-Code-Based Map Matching Algorithm for Wheelchair Navigation. *Transactions in GIS*.

Taylor, G., Blewitt, G., Steup, D., Corbett, S., Car, A. (2001). Road reduction filtering for GPS-GIS navigation, *Transactions in GIS*, ISSN 1361–1682, 5(3), 193–207.

Taylor, G., Brunsdon C., Li J., Olden A., Steup D., Winter M. (2006). GPS accuracy estimation using map-matching techniques: Applied to vehicle positioning and odometer calibration, *Computers, Environments, and Urban Systems* 30, 757–772.