Transactions in GIS, 2009, 13(2): 197–214

Research Article

A Chain-Code-Based Map Matching Algorithm for Wheelchair Navigation

Ming Ren
Geoinformatics Laboratory, School of Information Sciences
University of Pittsburgh

Hassan A Karimi
Geoinformatics Laboratory, School of Information Sciences
University of Pittsburgh

Abstract
Accurate vehicle tracking is essential for navigation systems to function correctly. Unfortunately, GPS data is still plagued with errors that frequently produce inaccurate trajectories. Research in map matching algorithms focuses on how to efficiently match GPS tracking data to the underlying road network. This article presents an innovative map matching algorithm that considers the trajectory of the data rather than merely the current position as in the typical map matching case. Instead of computing the precise angle which is traditionally used, a discrete eight-direction chain code, to represent a trend of movement, is used. Coupled with distance information, map matching decisions are made by comparing the differences between trajectories representing the road segments and GPS tracking data chain-codes. Moreover, to contrast the performance of the chain-code algorithm, two evaluation strategies, linear and non-linear, are analyzed. The presented chain-code map matching algorithm was evaluated for wheelchair navigation using university campus sidewalk data. The evaluation results indicate that the algorithm is efficient in terms of accuracy and computational time.

1 Introduction

In this article, we focus on wheelchair navigation that is an emerging navigation application. One reason for choosing wheelchair navigation is that map matching in the wheelchair environment is more complex than other navigation environments, such as car navigation, due to the imposition of several unique constraints. Therefore, a map matching algorithm suited for wheelchair navigation would be easily applicable to other

Address for correspondence: Ming Ren, Geoinformatics Laboratory, School of Information Sciences, University of Pittsburgh, 135 N. Bellefield Ave, Pittsburgh PA, 15213, USA. Email: hkarimi@mail.sis.pitt.edu

doi: 10.1111/j.1467-9671.2009.01147.x

navigation environments. Tolerico et al. (2007) have identified the research challenges along with the unique requirements and constraints in wheelchair navigation and have highlighted the distinctions between wheelchair navigation and car and pedestrian navigation. Many wheelchair users hesitate to visit an unfamiliar place because they have no information about the new environment and its accessibility issues. These problems can be anticipated and avoided if wheelchair users are provided with accurate and reliable information about the environment and mobility options. A map matching algorithm, which integrates the positional data with the digital map data, is a means to obtain epoch-by-epoch wheelchair locations on the sidewalk network, and constitutes the basis for all subsequent navigation activities.

Map matching algorithms integrate estimated locations, from a positioning sensor like Global Positioning System (GPS), with spatial network data on a digital map to identify the correct link on which a vehicle is traveling and to determine the location of a vehicle on that link (Ochieng et al. 2004, Karimi et al. 2006, Quddus 2006). A variety of map matching algorithms have been developed for car navigation taking different approaches, including geometric techniques, topological analysis of spatial data, probability theory, Kalman filters, and fuzzy logic (Quddus et al. 2007). However, map matching algorithms for wheelchair navigation must take into account unique characteristics different from those algorithms for car navigation. Firstly, GPS position fixes are less reliable at speeds of less than 3.0 m/s (Taylor et al. 2001, Ochieng et al. 2004), which is the case for wheelchairs. At such a low speed, the uncertainty in the vehicle position could contaminate the derivation of heading based on displacement (Taylor et al. 2006). Secondly, when a wheelchair passes through an intersection its trajectory typically will not follow a link on the corresponding digital map because of the absence of such features as roundabouts, junctions, medians, and curves. Thirdly, typical GPS accuracy of 10 m or even 5 m is not high enough to distinguish between the two sidewalks on either side of a narrow road, in which case additional data from other resources is needed to make that distinction. This article presents a chain-code-based map matching algorithm suitable for real-time wheelchair navigation. At the time of writing this article, there is a void in the literature on map matching for wheelchair navigation. Therefore, the major contribution of the article is development of a map matching algorithm that is not only suitable for wheelchair navigation but also is applicable to other navigation applications including car navigation and pedestrian navigation. The article is organized as follows. Section 2 briefly overviews the current methods for map matching. Section 3 describes the proposed chain-code-based map matching algorithm. Section 4 presents the results of evaluating the map matching algorithm using the University of Pittsburgh campus sidewalk network. Conclusions and future research are given in Section 5.

2 Background: Map Matching

Map matching is an essential function of any land-based navigation application. In this article, we focus on the research question of how to accurately map GPS-captured positional data onto linear data (i.e. a road or sidewalk network). A straightforward solution to the map matching problem, as offered by White et al. (2000), is to snap the GPS recorded location to the nearest road link node or road link. Approaches for map matching algorithms can be categorized into three groups: geometric map matching, topological map matching, and advanced map matching. Geometric map matching includes: point-to-point map matching, point-to-curve map matching, and curve-to-curve map matching (Karimi et al. 2006, Quddus et al. 2007). In point-to-point map matching each newly obtained position is matched to the closest "node" or "shape point" of a road segment. While point-to-point map matching is both easy to implement and computationally very fast, it is very sensitive to the geometry of the road network. In point-to-curve map matching, each newly obtained position is matched to the closest "line segment" (curve) in the road network which is selected as the segment on which the vehicle is traveling. Although point-to-curve map matching can identify road segments more accurately than the point-to-point map matching, in dense networks, such as those for urban areas, it may not be able to produce good solutions. In curve-to-curve map matching, a vehicle's trajectory (i.e. a curve) is matched to road segments (i.e. curves). Curve-to-curve map matching finds a matched road segment in three steps. In the first step it constructs piecewise linear curves using candidate nodes through point-to-point map matching. In the second step, it constructs piecewise linear curves using the vehicle's trajectory. In the third step, it calculates the distance between the vehicle's trajectory (step 2) and the curves corresponding to road segments (step 1). The road segment closest to the vehicle's trajectory is selected as the solution. Curve-to-curve map matching may not always produce a good solution as it is very sensitive to outliers and relies on point-to-point map matching (Ochieng et al. 2004).

Topological map matching (Quddus et al. 2003, Meng et al. 2006) takes into account not only geometrical information but also topological information of the road network and the previous GPS data collected. In topological map matching, the vehicle's trajectory and the topological features of the road (road turn, road curvature, and road connection) are matched. However, in some cases, topological map matching resorts to a post-processing mode to identify the correct road segment and in some cases it even relies on a global matching strategy, neither is suitable for real-time applications. Modifying the weighting scheme by introducing additional criteria and other parameters, such as a vehicle's speed, a vehicle's position relative to candidate road segments, heading information directly from GPS data, or position data obtained by integrating GPS and Dead Reckoning (DR) systems, will improve the performance of topological map matching.

These map matching approaches, geometric and topological, are used as the basis for developing other advanced map matching algorithms where they use additional techniques to improve performance, such as a Kalman Filter or an Extended Kalman Filter, a flexible state-space model and a particle filter, and a fuzzy logic model (Jagadeesh et al. 2004, Quddus et al. 2007).

3 Chain-Code-Based Map Matching Algorithm

While research in map matching is mostly focused on finding enhanced and new map matching algorithms for car navigation, in this work a map matching algorithm for wheelchair navigation is described. The common solution of map matching depends on two major steps: (1) finding the correct segment; and (2) finding the best estimate for the location of the user on that segment. How to find the correct segment is the key issue. Typically two main parameters, direction and distance, are coupled to solve this problem. Distance is the length of the projected line from the GPS position to a sidewalk link, and direction is a measure of the angle between two or more successive positions. Different techniques, such as a compass, can be used to measure direction; however, we

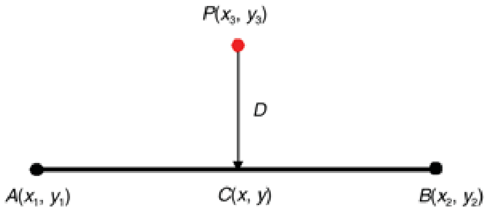

Figure 1 Perpendicular distance

present a new chain-code technique that utilizes the available information rather than inclusion of a new device such as a compass.

3.1 Definition of Distance

Let C be the projection of P on a sidewalk segment AB. The distance is defined in Figure 1.

$$D(P, AB) = \begin{cases} D(P, C) & \text{if } C \in [AB] \\ \text{Min}\,\{D(P, A), D(P, B)\} & \text{elsewhere} \end{cases} \tag{1}$$

3.2 Eight-Direction Chain Code

Chain codes (also known as Freeman's codes) are a common technique to represent a contour of an object in image processing (Freeman and Saghri 1974, Haron et al. 2005). The directions of contour boundaries are coded with integer values $k = 0, 1, \ldots, K - 1$ in the counterclockwise direction starting from the direction of the positive x-axis. A curve or contour is thus a chain of directions. The number of directions K takes is $2^{(M+1)}$ where M is a positive integer, such as 4, 8. The chain codes where $K > 8$ are called generalized chain codes, like 16 (Freeman 1978).

The presented algorithm for map matching uses chain codes to represent the direction of sidewalk segments and trajectory of moving wheelchairs. A directed straight-line segment connecting two adjacent points is called a link, and a chain is defined as an ordered sequence of links with possible interspersed codes. Chain encoding in this work is based on resolution of the direction to the adjacent links at the intersection. Since the angle between any two adjacent links on most intersections is usually greater than 45°, we use an eight-direction Chain Code 0, 1, 2, 3, 4, 5, 6, 7 to represent eight-direction interval on the counterclockwise direction as shown in Figure 2.

With this definition, 0 corresponds to an angle between –22.5° to 22.5°, 1 corresponds to an angle between 22.5° to 67.5°, and 7 corresponds to angles between –67.5° to –22.5°.

Figure 3 shows an example of GPS data and the sidewalk segment on a digital map. In this example, the GPS tracking route has a chain code of 0, 0, 1, 7, 0, 0, and the chain code of the sidewalk link C-A-B is 0,0 and F-A-D is 2,2.

Moreover, in order to find the closet segment to GPS tracking points, we use the difference between sidewalk link chain codes and the trajectory chain code of a wheelchair (Dcc) to show the extent of consistency of direction among them. With this, Dcc is defined as follows:

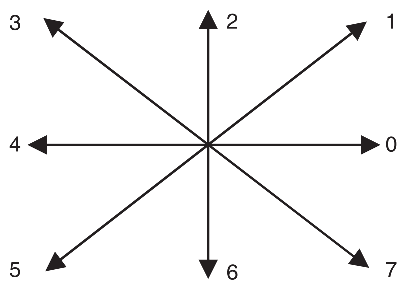

Figure 2 8-Direction chain code

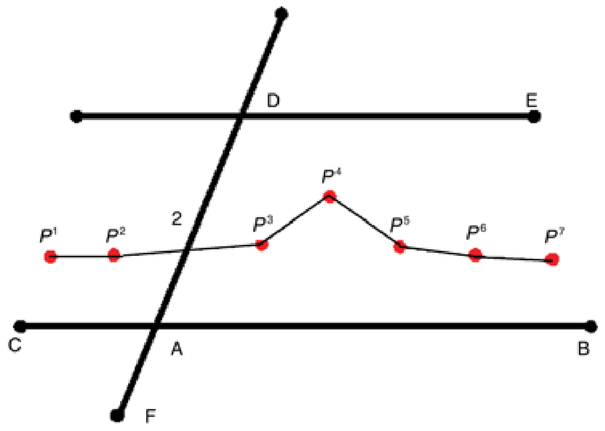

Figure 3 Digital map with GPS data

$$\Delta = |\text{Chain-Code (movement of wheelchair)} - \text{Chain-Code (sidewalk links)}|$$

$$\text{Dcc} = \begin{cases} \Delta & \text{if } \Delta < 4 \\ (|\Delta - 8|) \bmod 4 & \text{otherwise} \end{cases} \quad (2)$$

With this definition, discrete chain codes take the place of precise angle values; discrete Dcc takes the place of angle differences. This representation not only eliminates noise within short-distance moving, but also is computationally fast for real-time navigation.

Furthermore, as mentioned earlier, GPS position fixes are less reliable at a speed of less than 3.0 m/s. In such cases, in order to reduce the uncertainty of the direction under wheelchair users' control, the algorithm invokes a three-step Dcc between wheelchair trajectory and sidewalk segments rather than only taking a one-step Dcc.

In Figure 4, $P^1, P^2, \ldots, P^7$ show the same GPS trajectory as the one in Figure 3. With GPS data such as P^2, D_i (i = 1, 2, 3) could be calculated as the perpendicular distance to segments. When the first step is complete, the chain code from P^2 to P^3 is calculated, which is 0. Since heading directions are more meaningful than each-step directions, after two steps, the chain code from P^2 to P^4 is calculated, which is 1. Similarly, after three steps, the chain code from P^2 to P^5 is obtained as 0. Therefore, Dcc between P^2P^3 and CA is $|0 - 0|$; Dcc between P^2P^4 and CA is $|1 - 0|$, and Dcc between P^2P^5 and CA is $|0 - 0|$.

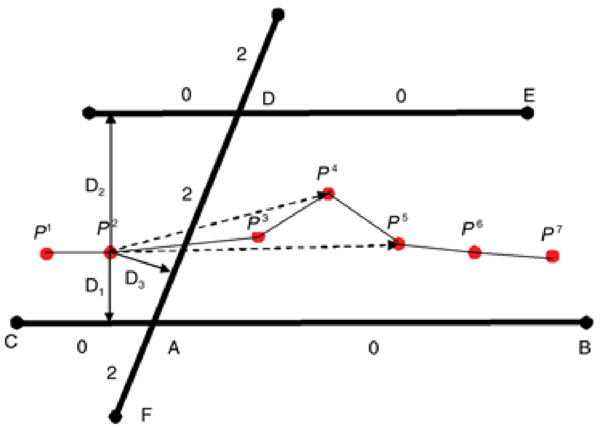

Figure 4 Example of chain-code-based map matching

3.3 Chain-Code-Based Map Matching Technique

In this algorithm, distance and direction are coupled to find the best location of the wheelchair on the sidewalk. First, the algorithm uses perpendicular distances from GPS data to each sidewalk segment candidate and the direction difference between the wheelchair trajectory and each sidewalk segment to select a sidewalk segment. As shown in Figure 4, in order to identify which segment a GPS point, such as P^2 in Figure 4, is most possibly mapped onto, both distance and direction movement are calculated to be weighted scores. All the segment candidates, close to P^2, are a link set {"CA", "DE", "AD"}; the distances from P^2 to these links are {D_1, D_2, D_3} and three-step Dcc calculations between trajectory and these segment candidates are taken as described in the previous section. Next, among all the segment candidates, the sidewalk segment with the highest matching evaluation will be chosen. In this case, segment "CA" would be considered as the best selection based on the map matching evaluation, and as a consequence, P^2 is determined to be projected to "CA". We propose two approaches to make the map matching decision: a linear and a non-linear model.

3.3.1 Linear model

Distance and difference in direction are two determining factors to calculate the matching result (i.e. identify the correct segment). The linear model is built on the linear relationship between the matching result (M) and evaluation parameters including Dcc and Distance. Figure 5 depicts the linear model.

The linear evaluation equation used in this work is as follows:

$$V_{ij} = W_{ij} * D_{ij} + \sum_{m=1}^{3} W_{ijm} * \text{Dcc}(\text{Step}[i+m], \text{sidewalk segment}[j]) \tag{3}$$

$$M_{ij} = 1/V_{ij} \tag{4}$$

where i is the indexing number of the GPS points and j is the indexing number of the segment candidates; D_{ij} is distance from the ith GPS point to the jth segment;

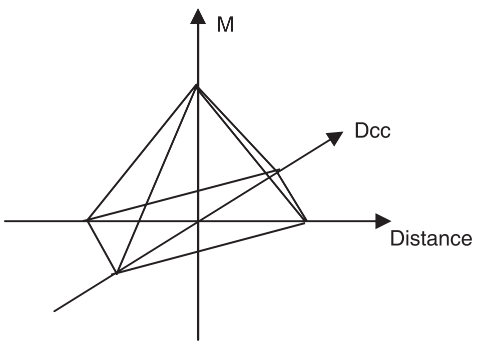

Figure 5 Linear model

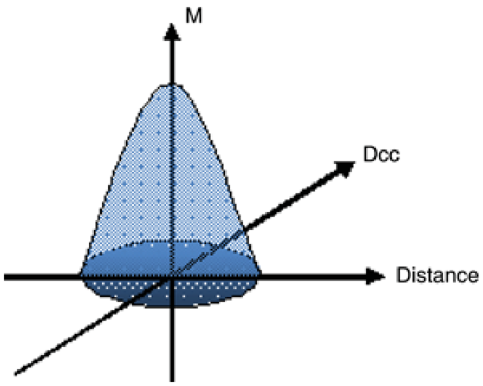

Figure 6 Non-linear model

Dcc(Step[$i + m$], sidewalk segment[j]) is three-step Dcc with m going from 1 to 3; W_{ij} is the weight of D_{ij} and W_{ijm} is the weight of Dcc. For the ith GPS point, M_{ij} is used to calculate the total weight assigned to the jth candidate link. The link with the highest M_{ij} is selected as the correct link for GPS point i. Therefore, the larger M_{ij}, the smaller is V_{ij}. In Equation (3), the total weighting score can then be obtained by summing up the individual scores, including weighting distance and three-step weighting Dcc as shown in Figure 5.

3.3.2 Non-linear model

Based on its definition in Section 3.2, Dcc is a discrete value ranging from 0 to 4; whereas, the absolute value of distance is a continuous real number from 0 to a large number. A non-linear model provides an alternative means to fit the non-linear relationship between the combination of these two input parameters and matching result to make map matching decision, which is depicted in Figure 6.

In this non-linear model, evaluation estimation is a fitting curve procedure between the response variable (matching result) and a list of input parameters (three-step Dcc and Distance). Among the most common non-linear models, neural networks constitute a widely used approach. This approach attempts to find a relationship, i.e. a function, between the inputs and the provided output(s), in order for the network to find a correct answer for the new inputs when the network is provided with unseen inputs. As one of various structures in the neural network family, radial basis function (RBF) networks (Howlett et al. 2001) incorporate a static Gaussian function as the non-linearity for the hidden layer processing elements, so the network could provide a good non-linear transformation in this map matching algorithm for each input vector, distance and three-step Dcc, to obtain the non-linear map matching result (M). The Gaussian function responds only to a small region of the input space where the Gaussian is centered. Therefore, in this map matching algorithm, the link which provides the highest output of the Gaussian function, i.e. M_{ij}, is chosen as the correct link for that positioning fix.

3.3.2.1 Radial basis functional neural network

The structure of RBF networks usually has three layers. Each hidden unit in the network has two parameters: a center u_j and a width σ_j associated with it. The output of each hidden unit depends only on the radial distance between the input vector and the center parameter for that hidden unit. The response of each hidden unit is scaled by its connecting weights W_{kj} to the output units and then summed to produce the overall network output:

$$y_k(x) = \sum_{j=1}^{M} W_{kj}\theta_j(x) + W_{ko} \tag{5}$$

where the Gaussian activation function for RBF networks is given by:

$$\theta_j(x) = \exp\left(-\frac{||x - u_j||^2}{2\sigma_j^2}\right) \tag{6}$$

where x is the d dimensional input vector with elements x_i, u_j is the vector determining the center of the basis function, and $y_k(x)$ is the output of RBF neural network.

There are two steps to train a RBF neural network: (1) determine the parameters of the basis functions through unsupervised training using only the input data set; and (2) determine weights W_{kj} using both input and output data (hidden units are activated using an input pattern and the weights to the output layer are then modified to produce the desired output for the given input). Once all the parameters are produced by training in a RBF network, the neural network model could be used to compute the output based on new input data.

3.3.2.2 Design of RBF neural network

In this research, the RBF Neural Network is designed as a supervised network. The input data has four features: distance and the three-step Dcc, so the input layer has four neurons, and the output layer has one neuron which is used to evaluate the extent to which a GPS point is close to candidate segments. The first step is to train the neural network. The training data consists of many pairs of input and output. Many pairs of Distance and three-step Dcc are calculated as input vectors, and output values are designated as 0 or 1, depending on whether the calculated candidate segment is the correct link or not. Next, the trained neural network is used to perform the non-linear map matching

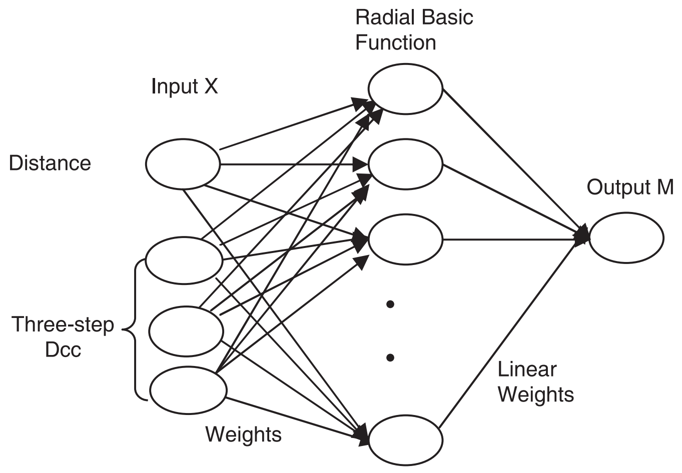

Figure 7 BF neural network for map matching evaluation

evaluation. Given a 4-d input vector with distance and three-step Dcc, the segment with the smallest output value is the selected segment, where the output of evaluation, M_{ij}, obtains the largest value. Based on the size of sample data in experiments, the hidden layer is designed as a 132-neuron layer. The structure of the RBF neural network for this map matching algorithm is shown in Figure 7.

By definition in Equations (5) and (6), the corresponding non-linear evaluation equation used in this work is defined as follows:

$$M_{ij} = \sum_{k=0}^{n} \alpha_k * \exp\left(-\frac{\beta \,||\, x_{ij} - u_k \,||^2}{2\sigma_k^2}\right) \qquad V_{ij} = 1/M_{ij} \tag{7}$$

$$x_{ij} = \begin{bmatrix} D_{ij} \\ \text{Dcc(Step}[i+1]\text{, sidewalk segment}[j]) \\ \text{Dcc(Step}[i+2]\text{, sidewalk segment}[j]) \\ \text{Dcc(Step}[i+3]\text{, sidewalk segment}[j]) \end{bmatrix} \tag{8}$$

where i is the indexing number of GPS points, and j is the indexing number of segment candidates. X_{ij} is input, in which D_{ij} represents distance from the ith GPS point to the jth segment, and Dcc(Step[i + m], sidewalk segment[j]) represents the three-step Dcc with m going from 1 to 3. α_k is the weight of the kth output value in the hidden layer. β is the weight matrix of the input vector in the hidden layer. As the output, M_{ij} is used to calculate the total weight assigned to the jth candidate link for the ith GPS point. The link with the smallest V_{ij} is viewed as the correct link.

3.4 Map Matching Process

Figure 8 shows the two pre-processing steps performed on the sidewalk network database prior to the map matching technique: (1) dividing the sidewalk network into

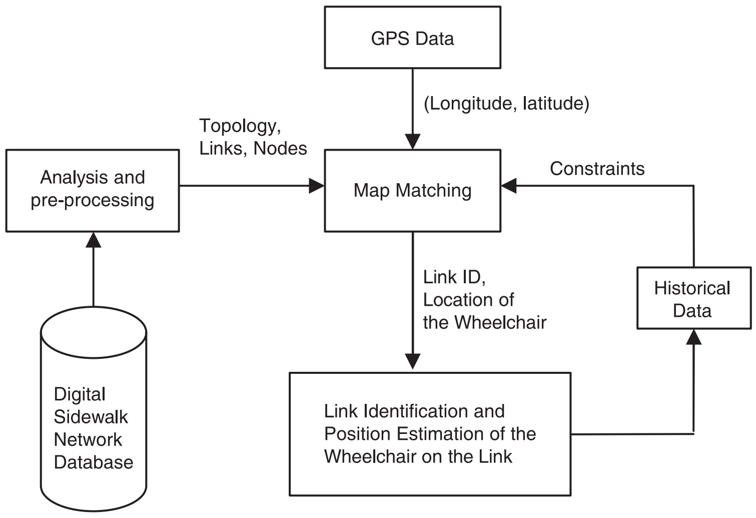
GPS Data
(Longitude, latitude)
Topology, Links, Nodes
Analysis and pre-processing
Map Matching
Constraints
Link ID, Location of the Wheelchair
Historical Data
Digital Sidewalk Network Database
Link Identification and Position Estimation of the Wheelchair on the Link

Figure 8 Map matching process

sidewalk segments as sidewalks in the network database consist of several segments; and (2) creating topological data structures with attributes, distance and chain codes, for map matching.

Second, two input data sources: GPS data and pre-processed spatial data are drawn into the map matching process. Given calculated parameters, distance and three-step Dcc, the map matching process in Figure 8 tries to find the correct segments based on either a linear model or a non-linear model.

Third, once a correct sidewalk segment is identified, the GPS point is projected onto the segment to obtain position estimation of the wheelchair. Moreover, after these three steps, the historical trajectory of the wheelchair and topological information for the sidewalk network could be used as a constraint to speed up the map matching process.

In this matching process, the point-to-curve approach is used to select the correct location on the sidewalk. There are two main modes in the algorithm: (1) turning mode; and (2) normal moving mode. In turning mode, the algorithm performs map matching for each GPS tracking point around intersections in order to identify a new segment. After passing an intersection, depending on the speed of the wheelchair, it can be estimated whether a wheelchair user is still traveling on the selected sidewalk segment or not. In the case of the normal moving mode, current and previous positions can be used as constraints for the next step in map matching based on the topology of sidewalk networks. Such constraints accelerate the matching process.

Once the correct sidewalk segment is determined, i.e. the first step of map matching, finding an estimate of the location of the wheelchair user on that segment is straightforward.

The flowchart shown in Figure 9 describes the process of chain-code-based map matching, including pre-processing to initialize data structures, evaluation based on three-step Dcc and distance between sidewalk segments and the trajectory made of GPS points, and a constraint on a selected sidewalk segment.

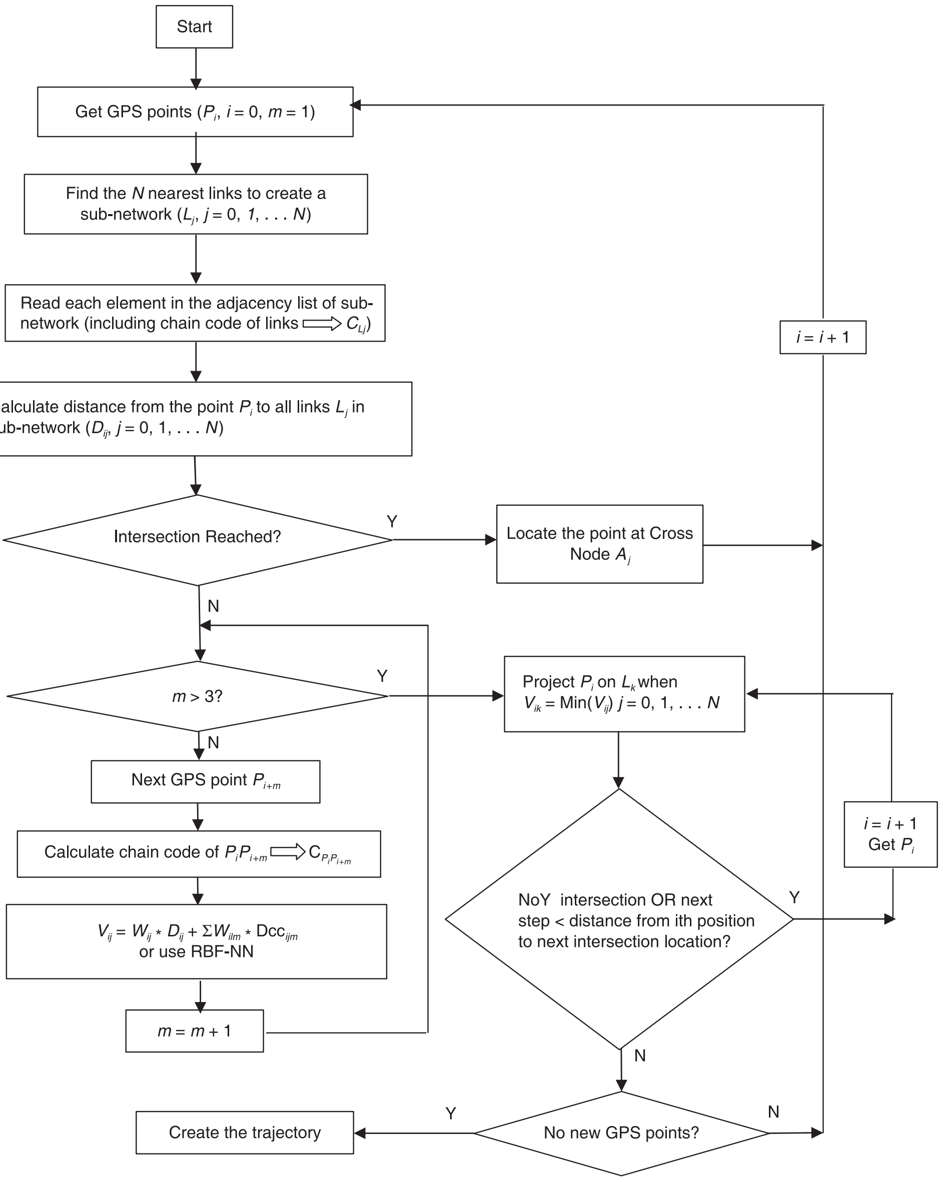
Start
Get GPS points (P_i, $i = 0$, $m = 1$)
Find the N nearest links to create a sub-network (L_j, $j = 0, 1, \ldots N$)
Read each element in the adjacency list of sub-network (including chain code of links $\Rightarrow C_{Lj}$)
Calculate distance from the point P_i to all links L_j in sub-network (D_{ij}, $j = 0, 1, \ldots N$)
Intersection Reached?
Y
Locate the point at Cross Node A_j
N
$m > 3$?
Y
N
Next GPS point P_{i+m}
Calculate chain code of $P_iP_{i+m} \Rightarrow C_{P_iP_{i+m}}$
$V_{ij} = W_{ij} * D_{ij} + \Sigma W_{ilm} * Dcc_{ijm}$ or use RBF-NN
$m = m + 1$
Project P_i on L_k when $V_{ik} = \mathrm{Min}(V_{ij})$ $j = 0, 1, \ldots N$
NoY intersection OR next step < distance from ith position to next intersection location?
Y
N
$i = i + 1$
$i = i + 1$ Get P_i
Create the trajectory
Y
No new GPS points?
N

Figure 9 Flowchart of chain-code-based map-matching algorithm

4 Validation

To validate the chain-code-based map matching algorithm developed in this work, the sidewalk data along with associated parameters on the University of Pittsburgh campus area were digitized and utilized (Kasemsuppakorn and Karimi 2008) and GPS points in three routes were collected by walking and using a stand-alone GPS receiver. The computing platform used was a PC machine with a "Intel Core 2 2.13G HZ" CPU. The software for the chain-code-based map matching algorithm was written in JAVA in an open source GIS tool called Geotools (see http://sourceforge.net/projects/geotools/ for additional details).

4.1 Evaluation of Linear Map Matching Models

For evaluation purposes, we employed different forms of linear and non-linear models in the matching evaluation.

As discussed in Section 3.3.1, the distance, three Dcc after step one, two, and three are the four influencing factors in matching evaluation equation. Their linear relationship was formalized as Equation (3). After normalization for these four variables, four estimated weight parameters meet the condition as follows:

$$W_{ij} + \sum_{m=1}^{3} W_{ilm} = 1 \tag{9}$$

After testing different combinations, the four weight parameters are given as:

$$\begin{cases} W_{ij} = 0.5 \\ W_{ij1} = 0.1 \\ W_{ij2} = 0.2 \\ W_{ij3} = 0.2 \end{cases} \tag{10}$$

4.2 Evaluation of Non-Linear Map Matching Models

We used a RBF neural network to build a non-linear evaluation model. As mentioned in Section 3.3.2, the RBF neural network considered in this work has a three-layer structure, including 4-d input layer, 132-d hidden layer and 1-d output layer. In the training stage, 132 pairs of parameters, with input vectors and output values, are calculated and normalized as a sample data set for training the RBF neural network. Figure 10 shows the training performance.

The curve in Figure 10 shows that training is completed after several iterations, when error gets close to 0. This means that we could use this trained model to perform map matching evaluation.

Table 1 shows the trained network structure. The input vector is defined as a 4-d vector including distance, Dcc (after step1), Dcc (step2), and Dcc (step3). The output of the trained neural network is the evaluation result. The candidate segment with the minimum output value is viewed as the matched segment. Table 2 shows different output values according to different input vectors for a GPS point and all relevant segment candidates. It could be drawn that the first pair of input parameter results in the smallest output value in the list, which means that the corresponding segment provides the closest match to the GPS data, and thus will be identified as a selected link for the GPS data.

4.3 Performance

Two groups of tests were conducted to evaluate the performance of the chain-code-based map matching algorithm. First, three data sets, collected on the main campus of the University of Pittsburgh by GPS receiver, were processed to validate the map matching algorithm. Linear map matching approaches were performed to compare results with

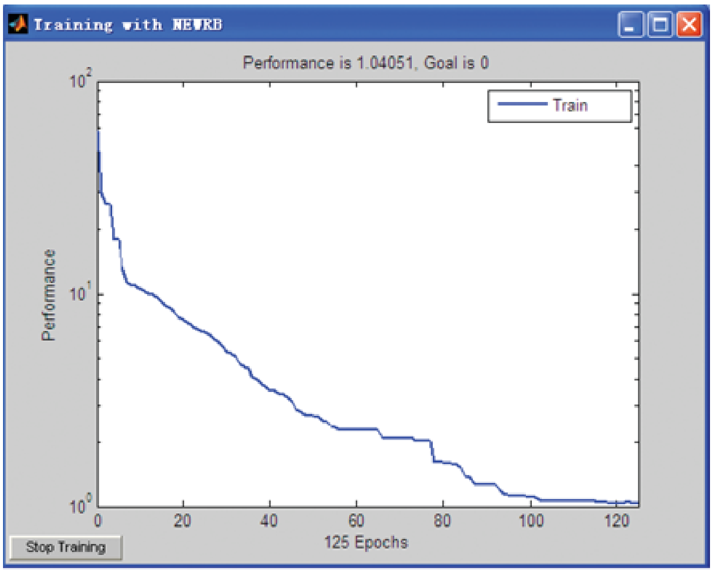

Figure 10 Training with RBF neural network

Table 1 RBF neural network structure

Inputs	Hidden layer	Output layer	Weights from input layer to hidden layer	Weights from hidden layer to output layer
Size: 4	Size: 132	Size: 1	4 ∗ 132	132 ∗ 1

Table 2 Map matching evaluation using RBF neural network

Input[0] Distance	Input[1] Dcc[step1]	Input[2] Dcc[step2]	Input[3] Dcc[step3]	Output
0.0116309625	0	0	0	0.0068
0.1534199589	0	0	0	0.3967
0.0482950975	2	2	2	1.3588
0.2902582818	1	2	2	13.221
0.4500096318	3	4	4	542.71
1.0987441382	1	1	1	874.33

constraints and without constraints. Second, linear models and non-linear models are separately applied to analyze their performances in making map matching decisions.

In the first group, we used the linear matching approach to test three routes on the campus sidewalk map. Table 3 shows the performances of map matching by using the linear model. The results of map matching with constraints, compared with GPS raw data, are shown in Figures 11, 12, and 13.

Table 3 Linear-model map matching results

		Map Matching Without Constraints		Map Matching With Constraints	
Testing Environment	Total Number of GPS Points	Correct Link Identification (%)	Average Time/Point (sec)	Correct Link Identification (%)	Average Time/Point (sec)
Route 1 in Urban Area	682	93.2%	0.025	93.4%	0.0044
Route 2 in Urban Area	1,517	94.6%	0.026	95.4%	0.0043
Route 3 in Urban Area	933	89.6%	0.026	89.8%	0.0046

Table 4 Comparing the linear model and the non-linear model

	The Linear Model With Constraints		The Non-Linear Model With Constraints	
Scenarios	Correct Link Identification (%)	Average Time/Point (sec)	Correct Link Identification (%)	Average Time/Point (sec)
Route 1	93.4%	0.0044	93.3%	0.017
Route 2	95.4%	0.0043	94.9%	0.02
Route 3	89.8%	0.0046	88.5%	0.019

The map matching performances are presented in Table 3 where five statistics are listed on three chosen routes in this group.

Compared with those implemented by other map matching algorithms for car navigation (see Quddus (2007) for example), these results indicate that the chain-code-based map matching algorithm is able to achieve a better performance in terms of correct identifications using a combination of distance and direction.

Testing results show that both positioning systems and the digital map quality 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 Route 2, we realize that many mismatched points occur on sidewalks of narrow roads due to GPS errors, because two sides of some narrow roads cannot be distinguished if their distances are below the GPS error range which is generally assumed to be a 10 m radius or more. Moreover, with constraints, both correct identifications and time performance are better than those in map matching without constraints.

In the second group, we applied both a linear model and a non-linear model to map matching on all routes. As Table 4 shows, the result leads to the conclusion that linear

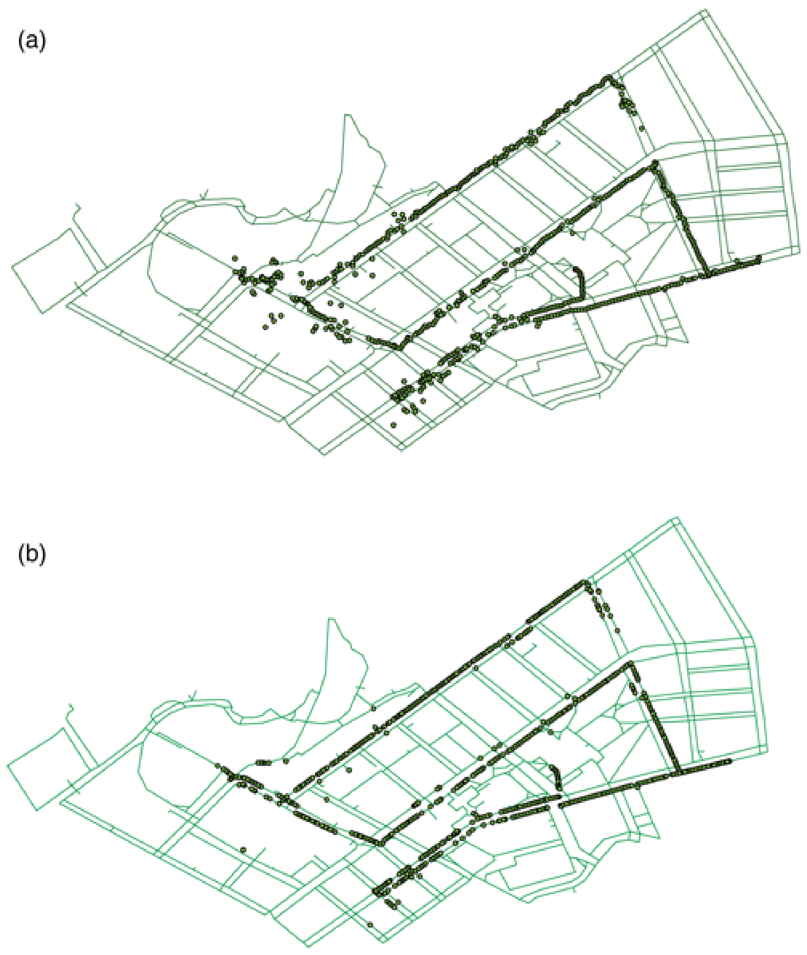

Figure 11 Route 1 comparing map-matching result with GPS raw data on campus sidewalk map: (a) GPS data before map matching; and (b) the result of map matching with constraints using the linear model

evaluation is more advantageous to implement than non-linear evaluation in terms of time performance. Compared to the linear model, map matching results in the non-linear model show slightly lower correct link identification. Therefore, in order to meet the need of real-time navigation, the linear model is preferred as the model for the map matching decision.

5 Conclusions and Future Research

In this article, we presented a novel map matching algorithm to estimate wheelchair location in sidewalk networks. The Chain-Code-Based Map Matching Algorithm is based on the analysis of distance and direction between GPS trajectory and the topology

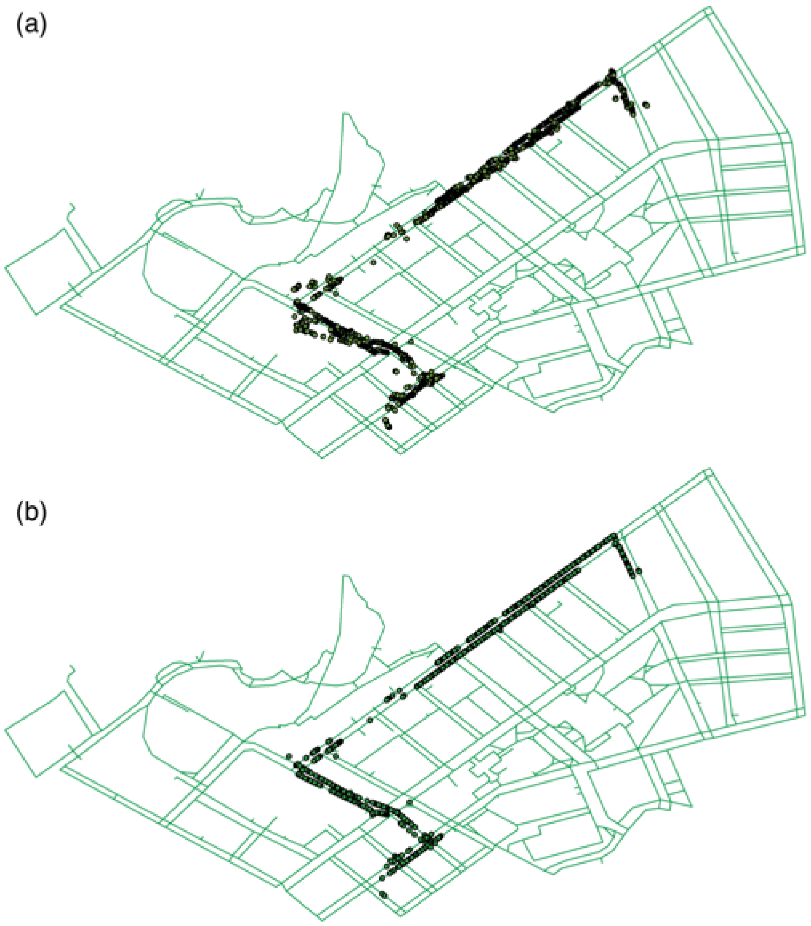

Figure 12 Route 2 comparing map-matching result with GPS raw data on campus sidewalk map: (a) GPS data before map matching; and (b) the result of the map matching with constraints using the linear model

of sidewalk segments. Rather than using precise angles to compute the three-step Dcc, the algorithm uses a discrete value to describe direction. This technique eliminates high-level directional errors created by small wheelchair heading changes, as the wheelchair continues on the same path. For each GPS point, a decision is made based not only on the three-step Dcc and distance, but also by considering topological information constraints.

The results of the tests show that our chain-code-based map matching algorithm obtains satisfactory results. However, some GPS points were mismatched due to failure in differentiating between the two sides of narrow roads. Both linear and non-linear models were tested for map matching evaluation, and the evaluation results indicate that the linear model performs better in terms of computational time and therefore is suitable for real-time processing.

Our future research includes testing our hypothesis that this map matching algorithm is easily applicable to car navigation and pedestrian navigation. Another area

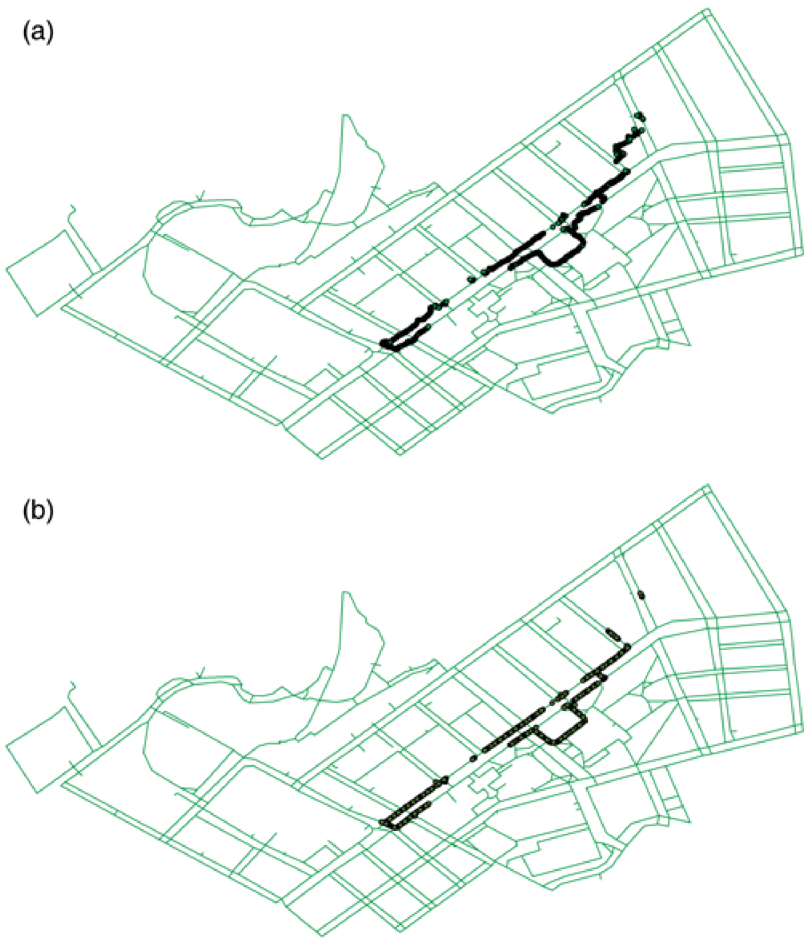

Figure 13 Route 3 comparing map-matching result with GPS raw data on campus sidewalk map: (a) GPS data before map making; and (b) the result of map matching with constraints using the linear model

for our future research is improving the map matching accuracy of the algorithm in areas with narrow roads.

References

Freeman H and Saghri A 1978 Generalized chain codes for planar curves. In *Proceedings of the Fourth International Joint Conference on Pattern Recognition*, Kyoto, Japan

Freeman H 1974 Computer processing of line-drawing images. *Computing Surveys* 6: 57–97

Haron H, Shamsuddin S M, and Mohamed D 2005 A new corner detection algorithm for chain code representation. *International Journal of Computer Mathematics* 82: 941–50

Howlett R J and Jain L C (eds) 2001 *Radial Basis Function Networks*. New York, Springer

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: 429–40

Karimi H A, Conahan T, and Roongpiboonsopit D 2006 A methodology for predicting performances of map-matching algorithms. In *Proceedings of Sixth International Conference on Web and Wireless Geographic Information Systems (W2GIS)*, Hong Kong, China: 202–13

Kasemsuppakorn P and Karimi H A 2008 Data requirements and spatial database for personalized wheelchair navigation. In *Proceedings of the Second International Convention on Rehabilitation Engineering and Assistive Technology*, Bangkok, Thailand

Meng Y 2006 Improved Positioning of Land Vehicle in ITS Using Digital Map and Other Accessory Information. Unpublished Ph.D. Dissertation, 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* 55(2): 1–18

Quddus M A 2006 High Integrity Map Matching Algorithms for Advanced Transport Telematics Applications. Unpublished Ph.D. Dissertation, University of London

Quddus M A, Noland R B, and Ochieng W Y 2004 Validation of map-matching algorithm using high precision positioning with GPS. *Journal of Navigation* 58: 257–71

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: 312–28

Quddus M A, Ochieng W Y, Zhao L, and Noland R B 2003 A general map-matching algorithm for transport telematics applications. *GPS Solutions* 7(3): 157–67

Taylor G, Blewitt G, Steup D, Corbett S, and Car A 2001 Road reduction filtering for GPS-GIS navigation. *Transactions in GIS* 5: 193–207

Taylor G, Brunsdon C, Li J, Olden A, Steup D, and Winter M 2006 GPS accuracy estimation using map-matching techniques: Applied to vehicle positioning and odometer calibration. *Computers, Environments, and Urban Systems* 30: 757–72

Tolerico M L, Ding D, Cooper R A, Spaeth D M, Fitzgerald S G, Cooper R, Kelleher A, and Boninger M L 2007 Assessing mobility characteristics and activity levels of manual wheelchair users. *Journal of Rehabilitation Research and Development* 44: 561–72

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

Yang J S and Kang S P 2005 The map matching algorithm of GPS data with relatively long polling time intervals. *Journal of the Eastern Asia Society for Transportation Studies* 6: 2561–73