THE JOURNAL OF NAVIGATION (2012), **65**, 617–633.
doi:10.1017/S0373463312000252

Movement Pattern Recognition Assisted Map Matching for Pedestrian/Wheelchair Navigation

Ming Ren and Hassan A. Karimi

(*Geoinformatics Laboratory, School of Information Sciences, University of Pittsburgh, USA*)
(E-mail: mng_ren@yahoo.com)

Today's mobile technology features several sensors that when integrated can provide ubiquitous navigation assistance to pedestrians including wheelchair users. Common sensors found in most smartphones are Global Positioning System (GPS), accelerometer, and compass. In this paper, a user's movement pattern recognition algorithm to improve map matching efficiency and accuracy in pedestrian/wheelchair navigation systems/services is discussed. The algorithm integrates GPS positions, orientation data from compass, and movement states recognized from accelerometer data in a client/server architecture. The algorithm is tested in an Android mobile phone, and the results show that the proposed map matching algorithm is efficient and provides good accuracy.

KEY WORDS

1. Map matching. 2. Movement pattern recognition. 3. Multi-sensor devices. 4. Pedestrian/wheelchair navigation.

Submitted: 2 April 2012. Accepted: 6 May 2012. First published online: 15 June 2012.

1. INTRODUCTION. Pedestrian tracking and navigation are now being widely researched and applied (Retscher et al., 2006; Sun et al., 2009; Ren and Karimi, 2009a, 2009b, 2011). Wheelchair users, as a special subset of pedestrians, also need navigation systems/services in their daily life. As distinct from car driving, pedestrians walk or wheelchair users operate at relatively lower speeds and their outdoor activities usually are closer to buildings, where the Global Navigation Satellite System (GNSS) cannot provide accurate and reliable position information due to the noise interruption and multipath effect by the environment. Furthermore, Global Positioning System (GPS) receivers (e.g., those in smartphones) have difficulties in distinguishing the two sides of a street if the street is less than 10 m in width. In addition, pedestrians and wheelchair users are free to advance, stop and make turns at will on sidewalks outdoors. This makes tracking people on sidewalk networks more challenging in pedestrian/wheelchair navigation systems/services than tracking people on road networks in car navigation systems/services. Considering the challenges in continually

finding user's location in real time on sidewalk networks, utilizing GPS as the only positioning sensor is not feasible and reliable to track pedestrians and wheelchair users outdoors.

Integration of GPS with other complementary sensors, such as Inertial Measurement Unit (IMU), has been introduced in recent years. Two different approaches for integrating GPS with IMU have been explored (Ahmed et al., 2009). The first approach uses conventional GPS and Inertial Navigation System (INS) integration algorithms as used in car navigation systems. The IMU is tightly coupled with GPS at the measurement level using an Extended Kalman Filter (EKF) in an effort to achieve best positioning results (Venkatraman et al., 2009; Ahmed et al., 2009). The second approach makes use of the fact that the user takes one step at a time and counts steps to obtain user's movement distance. For pedestrians, by counting steps and knowing the step length in the direction of motion, the navigation system can obtain walking distance. This method of obtaining navigation solution is often known as 'Pedometry'. For wheelchair users, one step is equal to one wheel circle. Given the circumference of the wheel, the moving distance in the motion can also be easily calculated. However, both approaches need specially designed hardware. The first approach requires chip-level integration of GPS and IMU, and the second approach requires a wearable device tied on the body for step counting or on a wheel for circle counting.

Due to recent advances in computing and mobile device technologies, smartphones, like iPhone and Android phones, are growing in popularity. Navigation services on smartphones can be based on common technologies such as GPS, cameras, accelerometers, compasses, and even gyroscopes. Given the popularity of smartphones and the availability of technologies for navigation services, this paper discusses a map matching algorithm based on multi-sensor data on a smartphone as the platform for pedestrian/wheelchair navigation.

iPhone and Android platforms are different in terms of GPS location management. The iPhone development platform only provides distance-based user-location updating (Arfe et al., 2011). On an Android development platform, the user-location updating function has two modes: distance-based location updating and time-based location updating (Arfe et al., 2011). The distance-based location updating mode updates user's location only when the user travels for a distance that is greater than a pre-determined distance. The time-based location updating mode updates user's location each time a given time interval is reached. Both location updating modes have their own advantages and disadvantages. In the time-based location updating mode, if the time interval is small, frequent user-location updates lead to more awareness of the mobile user's location. However, transmission of too many updates in short time intervals may overload the network. On the other hand, in the distance-based location updating mode, infrequent location updates may cause a lack of awareness about the actual user location but cost less in terms of data transmission compared to the frequent user-location update mode.

Neither of these location updating modes is suitable for pedestrian/wheelchair navigation services. Current GPS technology is unable, due to its accuracy range, to detect movement of pedestrians or wheelchair users who typically move at low speeds. Pedestrians or wheelchair users may move, stop, or make turns at will, which makes pre-setting a time interval for location updates difficult. As a result, updating a pedestrian's or wheelchair user's location based on time is impractical. Figure 1 shows

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

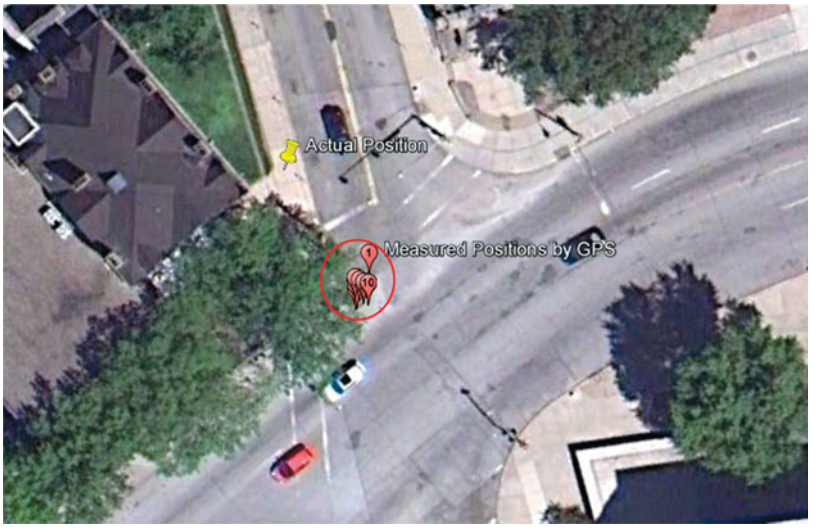


Figure 1. An example of GPS error in the scenario in which a user is stopped on a sidewalk.

an example of GPS error that can result when a user is stopped at an intersection by a red traffic light.

When there is no movement, GPS keeps updating the same location resulting in multiple positions, circles in Figure 1. In this example, with no knowledge about user's movement, the map matching algorithm will treat all the received positions as individual locations and match them onto the sidewalk. Map matching GPS data when there is no user's movement will not represent user's actual location, since every distinct GPS position within the error circle will be located on a different point of the sidewalk segment.

In order to address the problem of location updating in pedestrian/wheelchair navigation services, unlike traditional methods used in INS for geo-positioning, a map matching algorithm is proposed in this paper to take user's behaviours into account and then use movement pattern to assist map matching. For the algorithm, user's movement modes are no movement, walk, run, and turn.

The rest of the paper is structured as follows. Section 2 discusses background and related works. Section 3 describes an approach to user's movement pattern recognition on mobile phone. Section 4 introduces map matching algorithm assisted by user's movement recognition. Section 5 discusses testing platform and experimental evaluation. Conclusions and future research are discussed in Section 6.

2. BACKGROUND.

Map matching is an essential function of any land-based navigation application (Karimi et al., 2006; Karimi, 2011). Approaches for map matching algorithms can be categorized into three groups: geometric map matching, topological map matching, and advanced map matching (Quddus et al., 2007). Topological map matching (Meng, 2006; Quddus et al., 2003) utilizes geometrical and topological information of the road network and the previous GPS data collected. In topological map matching, vehicle's trajectory and topological features of the road (road turn, road curvature, and road connection) are matched.

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

Introducing additional sensors into the system, e.g., Dead Reckoning (DR) and gyroscope in vehicles, can improve the performance of topological map matching. Moreover, some other advanced map matching algorithms use additional techniques, such as a Kalman Filter or an Extended Kalman Filter, a flexible state-space model and a particle filter, and a fuzzy logic model (Quddus et al., 2007; Jagadeesh et al., 2004), to improve performance.

iPhone and Android platforms currently represent the cutting edge of mobile technology and have been widely adopted by people around the world. These two smartphone platforms come with built-in GPS receivers and integrated motion sensors, such as an accelerometer, a compass and even a gyroscope, which can be used for orientation detection, gesture recognition, and image stabilization, among other things.

Activity recognition from accelerometer data has been a research topic for many years and is usually formulated as a signal processing and classification problem (Mathie et al., 2004; Ravi et al., 2005; Sun et al., 2009). Research in activity recognition has focused on identification of physical activities, such as walking, jogging, resting, standing, climbing, or running. Accelerometers have been used as motion detectors (DeVaul and Dunn, 2001) as well as for body-position and posture sensing (Foerster et al., 1999). Inspired by the accelerometer-related research on activity recognition, this paper applies signal processing and pattern recognition techniques to process accelerometer data to recognize user's movement behaviour. The following section will present a new map matching algorithm that is assisted by the recognition of user's movement pattern. The algorithm has two major steps. The first step involves using accelerometer data to recognize user's movement behaviour and the second step involves performing map matching by using positioning data from GPS, orientation data from a compass, and knowledge of user's movement pattern.

3. MOVEMENT PATTERN RECOGNITION OF PEDESTRIAN/WHEELCHAIR USERS OUTDOORS.

Pedestrian/wheelchair navigation activities that occur outdoors can be grouped into four movement modes: no movement, walking, running, and turning. To detect these four modes of movement, four classes corresponding to these modes are defined in a decision tree classifier. A decision tree classification and recognition was developed and is described below.

Figure 2 shows the process of movement pattern classification and recognition. The process has two stages: a training stage and a testing stage. Both training stage and testing stage consist of four steps, three of which are the same in each stage. In the first step, data are collected from multi-sensors. In the second step, the raw data are pre-processed. In the third step, features are extracted from pre-processed data and raw data to create feature vectors. In the last step of the training stage, a decision-tree classifier is generated which will be used for recognition in the testing stage.

3.1. *Signal Pre-processing*. Accelerometers are sensitive to shaking and vibration, while digital compasses are susceptible to noise disturbances in the Earth's magnetic field. The magnetic distortion may vary significantly with time and location due to environmental changes. Because of this, before accelerations and orientations are measured by an accelerometer and a compass, they must be calibrated in order to reduce noise disruption of the environment.

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

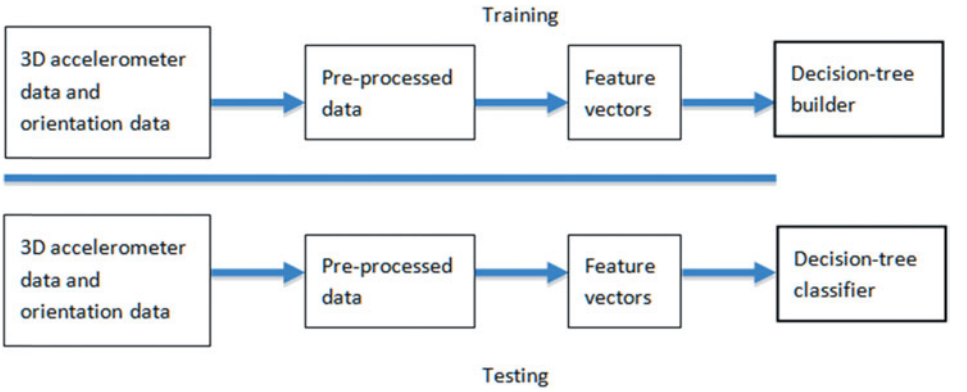


Figure 2. Overview of movement pattern recognition.

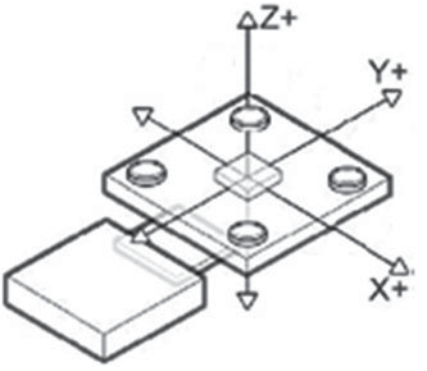


Figure 3. 3D accelerometer.

Once calibrated, Discrete Fourier Transform (DFT) is used to convert the acceleration data from time-domain values to frequency-domain features. In practice, a Fast Fourier Transform (FFT) algorithm is used to speed up DFT computations. For fast computation of FFT, a window size of 128 is used; this size was chosen as it can provide sufficient data for feature extraction in the next step and can meet the demand of computation in real-time navigation services.

3.2. *Feature Extraction*. With noises, if the raw accelerometer data were used directly as inputs to the decision tree classifier, the activity classification would produce poor results. It is possible to extract appropriate features by applying pre-processed data to enhance the quality of classification. In this paper, features are extracted from raw accelerometer signals through a sliding window of 128 samples, 64 of which overlap with their predecessors. The reason for utilizing sliding windows with 50% overlap to extract features is explained in the literature (e.g., see Bao and Intille, 2004).

Since a 3D accelerometer, used in most smartphones, can measure acceleration more accurately than a 2D accelerometer can, this paper uses typical 3D accelerometers available in smartphones. Figure 3 shows a sketch map of a 3D accelerometer, indicating three-axis directions. These three-axis accelerations are measured as a_x, a_y and a_z.

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

It should be noted that it is unnecessary to recognize all types of activities with high accuracy; it is sufficient to distinguish between the different modes (i.e., no movement, walking, running, and turning) for the purpose of navigation. To distinguish between the four modes, four features are extracted from each of the three axes in the accelerometer, giving a total of twelve attributes. The extracted features are mean, standard deviation, energy, and correlation.

Possible range of acceleration data varies with different activities. The energy feature is widely considered in activity measurement and recognition, while correlation is especially useful for differentiating among activities that involve translation of dimensions. No translation in dimensions is produced by the sensor when there is no movement, while walking and running usually involve translation in one dimension. Finally, turning involves translation in more than one dimension. Turning could be making a left turn, making a right turn, or making a U turn. It can be identified by orientation changes measured with a compass.

Taking the x-axis of the accelerometer as an example, equations to represent each of the four features are as follows (features in axis y and axis z are computed in the same way).

$$m_x = \frac{1}{N}\sum_{j=1}^{N} a_{x_j} \tag{1}$$

where:

a_x is x-axis acceleration.
m_x is the mean of all x-axis accelerations values in sample size N.

$$s_x = \sqrt{\frac{\sum (a_x - \overline{a_x})^3}{N-1}} \tag{2}$$

where s_x is the standard deviation of x-axis accelerations in sample size N.

$$E_x = \frac{\sum_{j=1}^{N} |fx_j|}{N} \tag{3}$$

where:

E_x is energy.
f_x is the component produced by FFT.
N represents the length of the sliding window.

$$corr_{xy} = \frac{cov(a_x, a_y)}{\sigma_{a_x}\sigma_{a_y}} \tag{4}$$

where $corr_{xy}$ is correlation between each pair of axes as the ratio of the covariance and the product of the standard deviations.

3.3. *Feature Selection and Classification.* After the feature vector is generated, the next step is feature selection and classification. Of the twelve features computed, only eight are considered to be useful to recognize user's movement. For example, when a smartphone is held face-up, its embedded accelerometer is faced up as shown in Figure 3. The x direction indicated in the figure is perpendicular to the direction of movement and direction of up-and-down vibration in the movement. Therefore, the

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

Table 1. Selected features.

Symbol	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
Feature	m_y	m_z	s_y	s_z	E_y	E_z	$corr_{xy}(1,2)$	$corr_{xz}(1,2)$

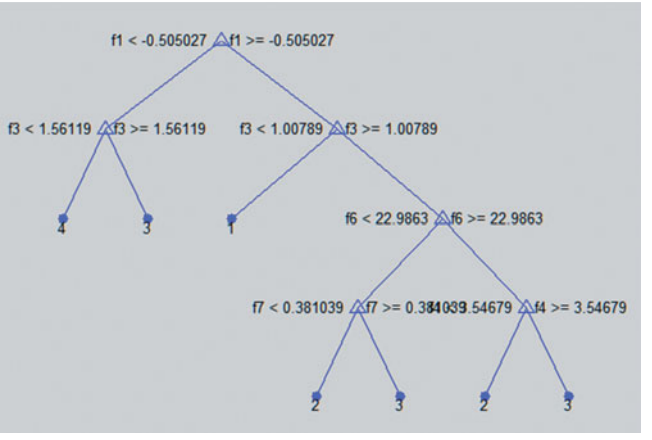


Figure 4. Movement recognition decision tree.

features related to the x-axis movement are not useful for distinguishing between the four movement modes. In this case, the features computed based on the x-axis acceleration can be removed from the feature list. Table 1 shows the eight features after eliminating the x-axis – related features.

This eight-feature vector is further compressed by employing a decision tree to eliminate redundant features. In the training stage, a decision tree is constructed based on a training data set. Figure 4 shows the movement pattern recognition decision tree after feature selection. Eventually the selected features are f_1, f_3, f_4, f_6 and f_7, which correspond to m_y, s_y, s_z, E_z and $corr_{xy}(1,2)$, respectively.

In Figure 4, the leaf nodes 1, 2, 3, and 4 represent the four movement modes, i.e., no movement, walking, running, and turning, respectively. Given the decision tree built in the training stage, to identify user's movement pattern in the testing stage, consecutively collected accelerometer data are processed in real time to compute the five selected features as feature vectors. Through the decision tree, the extracted feature vectors are used to determine the mode to which user's movement belongs. The identified mode will be used for map matching, as described in the next section.

4. MOVEMENT PATTERN RECOGNITION ASSISTED MAP MATCHING.

There are generally two types of navigation platforms. One works as a standalone system, while the other works on a network with clients and servers. Standalone platforms are referred to as 'navigation systems' and network-based navigation platforms are referred to as 'navigation services' (Karimi, 2011). Since smartphones have relatively limited memory and computing capabilities, it is difficult to build standalone pedestrian/wheelchair navigation systems on them.

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

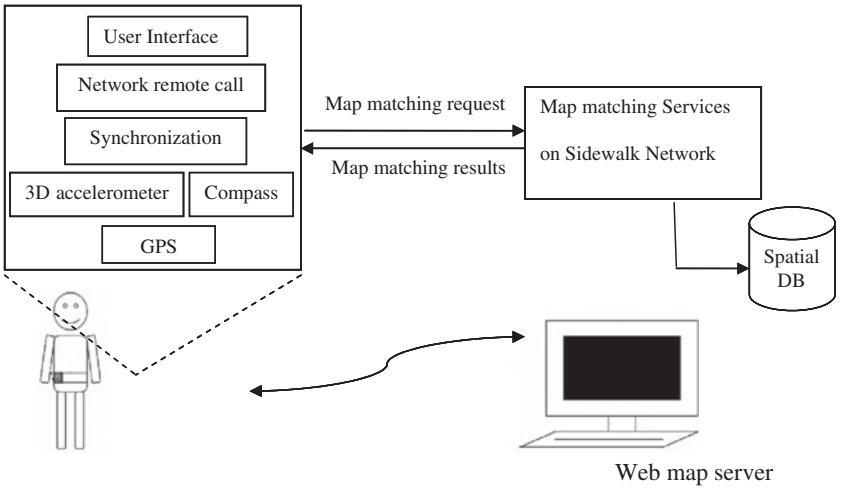


Figure 5. Client/Server architecture for map matching.

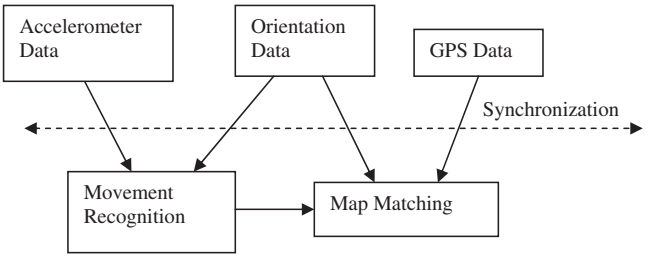


Figure 6. Multi-sensor data integrated map matching.

Therefore, in this paper, a client/server architecture is discussed to provide multi-sensor map matching for pedestrian/wheelchair navigation services.

A client/server architecture generally involves multiple clients connecting to a central server. In our client/server architecture, the smartphone is responsible for collecting real-time data (positioning data as well as other types of data such as heading data), synchronizing multi-sensor data, requesting map-matched results from the server, and updating the map that is presented to the user. The map data is stored in the server to perform map matching, among other functions. Figure 5 illustrates the architecture for map matching.

In the client/server architecture, user's movement pattern is recognized in the client by using accelerometer data, where user's location updates are sent to the server in pre-set time intervals that vary with the user's movement modes. In the turning mode, with the help of a compass in the smartphone, different turning types, such as left turns, right turns, and U turns can be further distinguished from one another. This orientation data can enhance the accuracy of GPS-based map matching. Given the differences between the three sensors (i.e., accelerometer, compass, GPS), they need to be synchronized in order to ensure that they work effectively in tandem. Figure 6 shows the relationship between the three sensors' data, and how they are fused and synchronized for map matching.

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

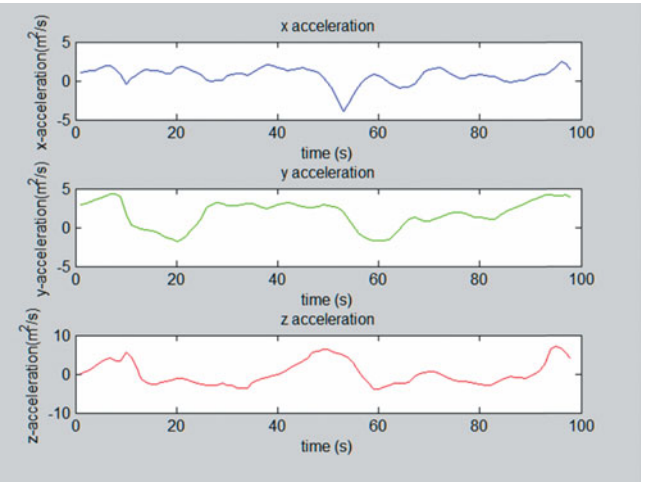


Figure 7. Accelerometer Data (acceleration in m^2/s).

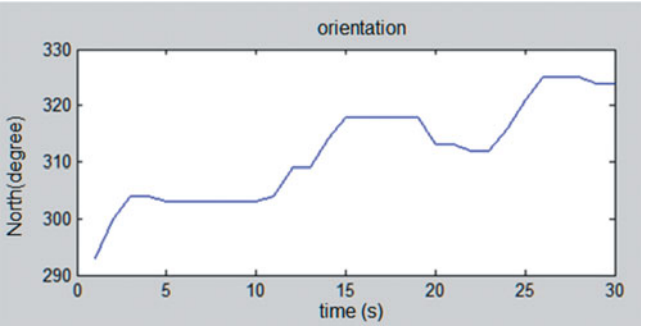


Figure 8. Orientation Data (angles in degrees).

In the multi-sensor integration map matching, GPS data are used for absolute positioning in recognizing user's movement. Accelerometer data are used for recognizing four modes of movement, as described earlier. When referring to the North direction, for example, orientation data, obtained from the compass, indicate the orientation of user's movement. This helps in distinguishing between different turning modes. To recognize movement pattern as accurately as possible, accelerometer data are sampled in the highest frequency in order to obtain sufficient samples for FFT processing and feature extraction. Figure 7 shows a snapshot of the 3D accelerometer data. Figure 8 shows an example of orientation data relative to the North direction.

In the multi-sensor data integration map matching, synchronization is essential to keep all sensor data working in tandem. Data collected from different sensors have different sampling frequencies. Knowing the user's current movement mode, the sampling time is determined by the synchronization function. Figure 9 shows the user in a walking mode at time t_0 and in a running mode between t_1 and t_2. Once the change in movement pattern is detected, the sampling frequency changes to a suitable interval for sampling data in the running mode. As the user stops between t_4 and t_5, the sampling frequency changes again, since no movement is detected.

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

Figure 9. Timing diagram for synchronization.

For map matching, using user's position based only on GPS data, the synchronization timeline starts from the moment when the smartphone begins receiving GPS data. While the map matching algorithm waits for the GPS receiver to provide its first position, [this is known as the Time-To-First-Fix (TTFF) problem (Lehtinen et al., 2008)], accelerometer and compass data can be obtained and used to detect user's movement behaviour. As user's movement mode changes, the time interval of sending GPS data and updating user's location by map matching services will also change. In Figure 9, each time point marked on the timeline indicates when all the sensor data are synchronized, given user's movement mode changes.

The flowchart of the movement pattern-recognition-assisted map matching algorithm is shown in Figure 10. First, as GPS data and user's heading information provided by the compass are updated in real time, a set of nearest sidewalk segments is chosen as candidates. By comparing heading values of the user in consecutive time, e.g., heading in time t_{c-1} and heading in time t_c, heading changes above or below a threshold are used to judge whether the user is making a turn or not. By comparing the orientation of a currently map-matched segment with the heading of the user and knowing the current map-matched location of the user, segment candidates can be further limited under different circumstances during the movement. Next, map matching decisions are made by evaluating a weighted combination of distance for positioning data to the candidate segments, and heading differences of the positioning trajectory and segment orientation. Once a position data is map matched, the map matching algorithm will wait for the next set of GPS and orientation data from the client.

5. EXPERIMENTS.

To validate the proposed movement-pattern-recognition-assisted map matching algorithm, the sidewalk network of the University of Pittsburgh's main campus was used and GPS points for three routes were collected

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

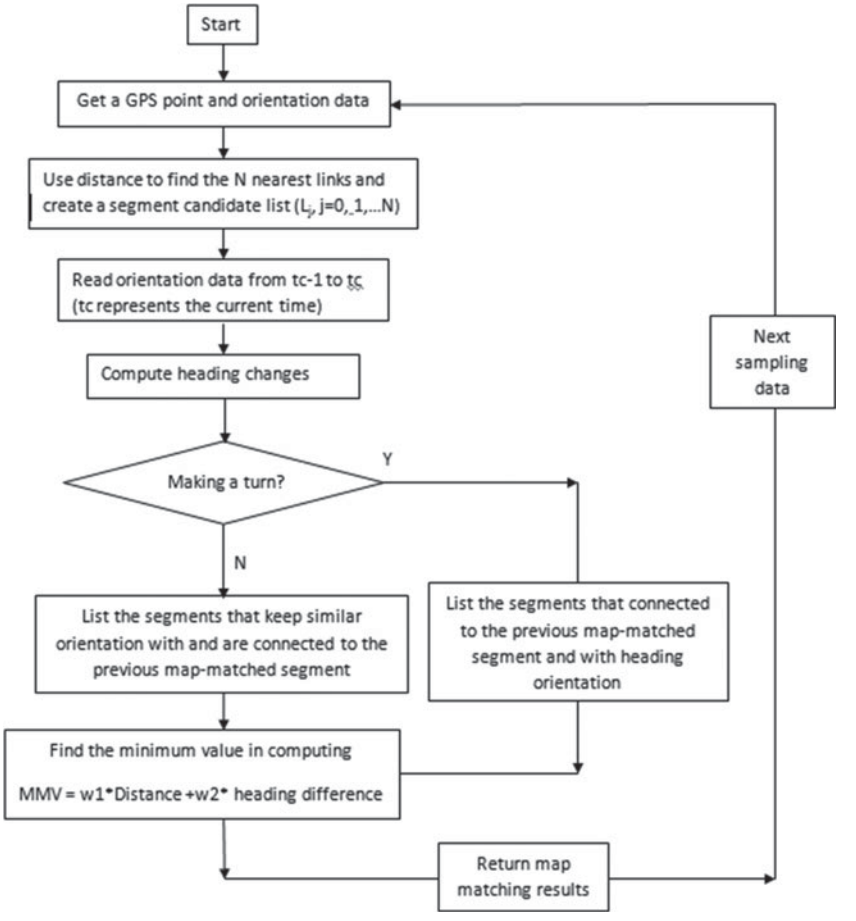


Figure 10. Flowchart of the movement pattern-recognition-assisted map matching algorithm.

by walking and using an Android phone (Motorola BackFlip). The server was a PC machine with an 'Intel Core 2 2.13 GHz' CPU.

The experiments were performed in two parts. The first part aimed to validate the movement pattern recognition approach. The experimental data contained both training data, which were collected for movement pattern classification, and testing data, which were used to recognize user's movement on real routes. The second part of the experiment aimed to evaluate the map matching performance, as assisted by user's movement pattern recognition.

5.1. *Data Collection and Data Sampling*. This section describes the experimental setup to collect sensor data for user's movement pattern recognition and location estimation. All the sensors (GPS, accelerometer, and compass) used in the experiments are available on the Android smartphone. An image of this phone and the direction of its 3D accelerometer are shown in Figure 11.

Figure 12 shows a sample of the recorded data in a log file. The log file includes collected GPS, accelerometer data, and compass data with time stamps. GPS data are tagged by GPS in the log file, which contain longitude, latitude, altitude, accuracy,

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

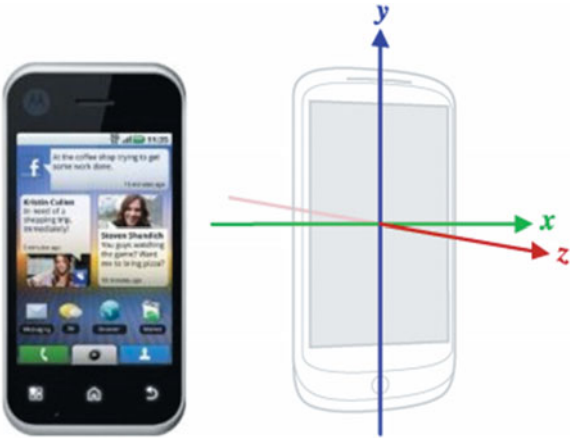


Figure 11. Motorola Backflip smartphone and the direction of its 3D accelerometer.

```
15:20:36:367 ORIENTATION 61.0,-22.0,-23.0
15:20:36:398 ACCELEROMETER -3.7732618,2.8921957,5.707777
15:20:36:405 ACCELEROMETER -2.087744,1.3599066,8.025364
15:20:36:407 ACCELEROMETER -0.21068975,-0.30645782,10.515334
15:20:36:411 ACCELEROMETER 1.340753,-1.4939818,12.737153
15:20:36:439 ACCELEROMETER 2.183512,-1.9536686,14.211981
15:20:36:442 ACCELEROMETER 2.5474305,-2.183512,14.614207
15:20:36:445 ACCELEROMETER 2.5091233,-1.9919758,13.809755
15:20:36:472 ORIENTATION 76.0,4.0,5.0
15:20:36:476 ACCELEROMETER -0.47884035,0.0,10.496181
15:20:36:479 ACCELEROMETER -1.4173675,0.6512229,9.423578
15:20:36:513 ACCELEROMETER -1.5131354,0.5746084,9.308656
15:20:36:516 ACCELEROMETER -1.1683705,0.1340753,9.883265
15:20:36:519 ACCELEROMETER -0.7086837,-0.32561144,10.668563
15:20:36:524 ACCELEROMETER -0.15322891,-0.7278373,11.473015
15:20:36:538 GPS -
79.95231617,40.44751463,26.832815170288086,0.0,283.70001220703
125,0.0
15:20:36:544 ACCELEROMETER 0.34476504,-1.0726024,11.894394
15:20:36:573 ACCELEROMETER 0.6703765,-1.1492168,11.971008
15:20:36:576 ORIENTATION 77.0,4.0,1.0
```

Figure 12. A sample of a log file recording GPS, accelerometer, and orientation data.

bearing and speed in order. Accelerometer data are tagged by the accelerometer, and are 3-axis acceleration data, i.e., acceleration in x-direction, y-direction, and z-direction. Compass data consist of 3-axis orientation data, orientation in x-direction, y-direction, and z-direction.

On Android phones, the GPS update frequency is controlled by either setting a Minimum Time Interval (minTime) or setting a Minimum Distance Interval (minDistance). If the value of minTime is greater than 0, the Location Manager in smartphones could stop working for a minTime of milliseconds between location updates to conserve power. If the value of minDistance is greater than 0, locations will only be updated when the device (and thus the user) moves by a distance of minDistance metres. Since GPS receivers on smartphones do not provide accurate distance measurement in low-speed movements, distance-based location updates are not appropriate for pedestrian/wheelchair navigation. For this reason, minDistance is set to 0.

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

Table 2. Classifier accuracy in identifying four different movement behaviours.

	Correct Recognition Accuracy Rate (%)
Walking	92·8%
No movement	97·8%
Running	93·4%
Turning	90·6%

In order to save energy and minimize computation time (map matching is potentially a complex task and its response is needed in real time), the following strategy, based on user's movement pattern recognition, is executed to update user's location.

- Update GPS position every 3 seconds if the user's movement mode is recognized as walking; set minTime to 3 s.
- Update GPS position every 2 seconds if the user is running; set minTime to 2 s.
- Stop updating GPS position if the user is not moving.

The Motorola BackFlip (the smartphone used in these experiments) can provide sampling frequency of, at most, 110 Hz on its accelerometer. A sliding window is set, including 128 sampling data which is the same amount of data collected within a time interval of 1·16 s, in 110 Hz. With 50% overlap between two continuous sliding windows, a three-second interval covers at least four sets of sampling values and a two-second interval covers at least two sets of sampling values. Assuming that the user does not change movement mode very often, continuous movement in a single mode can provide sufficient sampling data for recognizing pedestrian/wheelchair movement pattern.

5.2. *Training and Testing*. To analyse movement pattern, we collected a set of training data by labelling user's behaviour, such as walking, no movement, running, and turning. The training data set was used to build a decision tree as the classifier, and is shown in Figure 4. We then tested movement recognition on real routes within the study area. Along the testing routes, user's movement pattern in different places is recorded manually as ground truth. By comparing the ground truth data with results of the movement pattern recognition algorithm, the accuracy of recognizing different movement behaviours is shown in Table 2.

Of the user's total walking movements, 92·8% were recognized correctly; 1·4% were recognized incorrectly as no movement; 2·6% were recognized incorrectly as running; 3·2% were recognized incorrectly as turning. 97·8% of no movement were recognized correctly, but 2·2% were recognized incorrectly as walking. Similarly, 93·4% of running movements were recognized correctly, but 6·6% were recognized incorrectly as walking. Turning movement was recognized 90·6% correctly, but 9·4% were recognized incorrectly as running. The confusion matrix of cross-validation on the feature classification of movement behaviours is shown in Table 3.

Given that over 90% of all movement modes can be correctly recognized, it is feasible to use the movement behaviour recognition algorithm to determine actual movement behaviour. Based on the movement pattern recognition, map matching is expected to perform more efficiently, as illustrated in the next section.

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

Table 3. Confusion matrix of cross-validation on feature classification of movement behaviour.

Movement Mode \ Recognition	walking	no movement	running	turning
walking	92·8%	1·4%	2·6%	3·2%
no movement	2·2%	97·8%	0	0
running	6·6%	0	93·4%	0
turning	9·4%	0	0	90·6%

5.3. *Map Matching Validation*. To evaluate the performance of the movement-pattern-recognition-assisted map matching algorithm as outlined in Figure 9, we tested the algorithm on a set of routes on the main campus of the University of Pittsburgh. Three routes were chosen in the experiment. Route 1 was selected to represent a short route, where the map matching algorithm was validated in a scenario where the user moved close to buildings. The user started walking along a wide street and then turned into a narrow street. Route 2, as a medium length route, was selected to validate the map matching algorithm in the area with narrow streets and dense buildings. Route 3, the longest of the three routes, was chosen to validate the map matching algorithm in an area where GPS data are influenced by multipath reflection due to buildings, grasslands, main streets, and small paths. The user's movements on Route 2 and Route 3 include all the four movement modes discussed in the earlier section.

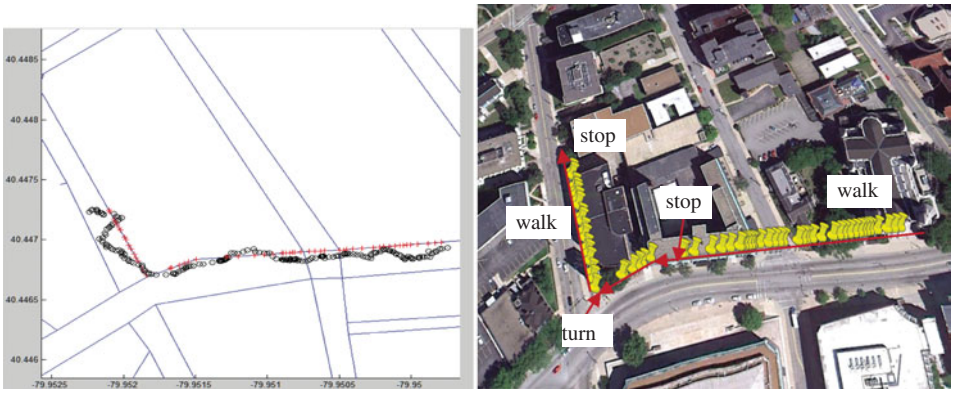
Figures 13–15 show the comparison of raw GPS positions and map-matched locations in the three routes. In Figures 13–15(a), black points indicate raw GPS positions and red points indicate map-matched locations overlapped on the sidewalk network. Figures 13–15(b) show map matching results with ground truth labelled by movement modes overlaid on a Google satellite map.

Table 4 shows the result of analysing the map matching performance in efficiency and accuracy. Taking Route 1 as an example, due to correctly recognizing user's different movement modes, computation for map matching is reduced; 41 out of 133 GPS datasets were sent for map matching to the server. The visualized results in Figure 13 also show that the user's locations still can be continuously and clearly tracked without any influences by the reduction of map matching results. By knowing user's turning behaviour, the map matching algorithm was only performed on the sidewalk connected to the previously walked-on sidewalk when the user approached an intersection. This improved the accuracy of the map matching algorithm.

Compared with the high-quality data collected on the campus using a professional-grade GPS receiver as ground truth data, the segment identification accuracy in our experiments is influenced largely by the poor quality of GPS data collected by a consumer-grade GPS receiver embedded in the smartphone.

Figures 13–15 show raw GPS data received from the smartphone. It is clear that the GPS data received from the smartphone can be noisy and inaccurate, especially when users are on narrow streets and when users move close to tall buildings. Low GPS data accuracy caused most of the mismatched points in the results. For example, in Route 2, mismatched points occurred during the time the user was turning to a narrow street at the intersection. Even though the turning behaviour of the user was recognized, the

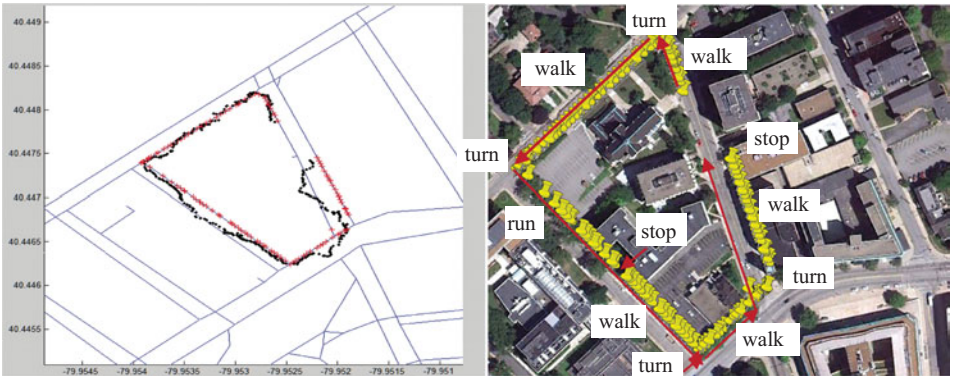
https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press



a. Raw GPS data and map matching locations overlaid on campus sidewalk map

b. Map matching locations compared with ground truth overlaid on Google satellite

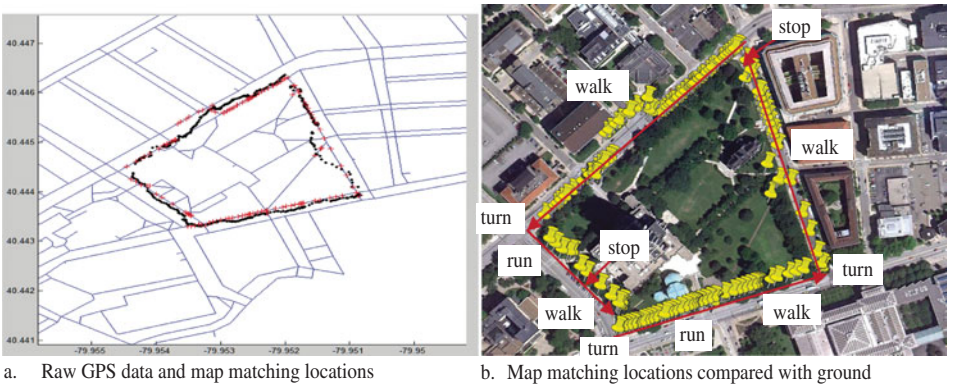
Figure 13. Route 1 comparing map matching result with GPS raw data.



a. Raw GPS data and map matching locations overlaid on campus sidewalk map

b. Map matching locations compared with ground truth overlaid on Google satellite map

Figure 14. Route 2 comparing map matching result with GPS raw data.



a. Raw GPS data and map matching locations overlaid on campus sidewalk map

b. Map matching locations compared with ground truth overlaid on Google satellite map

Figure 15. Route 3 comparing map matching result with GPS raw data.

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

Table 4. Map matching performance (efficiency and accuracy).

Testing Environment	Total Number of GPS Points per second	Total Number of GPS Points Sent to Server	Correct Link Identification (%)	User movements
Route 1	133	41	100%	walk->stop->walk->turn->walk->stop
Route 2	320	100	82%	walk->turn->walk->turn->walk->run->stop->walk->turn->walk->stop
Route 3	589	195	87·2%	walk->turn->walk->run->stop->walk->turn->run->stop

GPS accuracy is not high enough to distinguish between the two sides of the narrow street. The mismatched points at the intersection led to mismatched projections on the connected segment, as shown in Figure 14. This is the reason why the map matching accuracy shown in Table 4 is only 82%, mainly due to the smartphone GPS' inability to distinguish between the two sides of the narrow street. By recognizing user's movement pattern in Route 2, out of 320 GPS datasets, 100 were sent to the server for map matching.

In Route 3, the map-matched results also show problems with finding the side of the street on which the user was actually walking. To distinguish between the two sides of a street, high positional accuracy data will be needed in future works. Except for the problem of identifying the side of a street, the experimental results show that the map matching algorithm can correctly estimate user's location in the majority of the routes as compared with ground truth shown in Figure 15(b). Furthermore, the algorithm has low cost of communication and computation. By recognizing user's movement on each route, shown in the last column of Table 4, instead of sending user's location, either based on changes of time or changes of distance, movement pattern recognition based on location updates can significantly reduce communication costs between the server and the clients and reduce the calculation costs of map matching. The results shown in Figures 13–15 also demonstrated that the movement pattern recognition based on location updates can provide location estimation continuously without redundancy. It can be seen that the number of actual GPS data sent to the server, as shown in the second column, is less than one-third that of the number of GPS datasets received per second, as shown in the first column.

6. CONCLUSIONS AND FUTURE RESEARCH.

In this paper, we demonstrate that efficient outdoor localization can be achieved on existing mobile platforms by augmenting GPS sensors with the phone's built-in compass and accelerometer. Using methods that recognize user's movement pattern provides efficient updates of user's locations. Furthermore, combining the geometric and topological information of sidewalk networks with heading information obtained from compass further improves the accuracy of positioning in map matching. The experimental results show that the movement pattern recognition assisted map matching algorithm can make the map matching process in pedestrian/wheelchair navigation

https://doi.org/10.1017/S0373463312000252 Published online by Cambridge University Press

systems/services efficient. Furthermore, the algorithm can be used as the basis for many useful mobile location-based applications beyond pedestrian/wheelchair navigation systems/services.

Our future research includes testing user's movement pattern recognition in different outdoor location-based activities and testing the map matching algorithm in real-time pedestrian/wheelchair navigation systems/services.

REFERENCES

Ahmed, H. M., Samsudin, K., Ramli, A. R., Azmir, R. S. and Salam, A. I. (2009). A Review of Navigation Systems (Integration and Algorithms), *Australian Journal of Basic and Applied Sciences*, **3**(2), 943–959.

Arfe, A., Deguy, P., Guillot, L., Guilly, T. L. and Louge, R. (2011). Android Application for Aalborg University. *Project report*.

Bao, L. and Intille, S. (2004). Activity recognition from user annotated acceleration data. *In Proceedings of the 2nd International Conference on Pervasive Computing*, pages 1–17.

DeVaul, R. and Dunn, S. (2001). Real-time motion classification for wearable computing applications. *Technical report*, MIT Media Laboratory.

Foerster, F., Smeja, M. and Fahrenberg, J. (1999). Detection of posture and motion by accelerometry: a validation in ambulatory monitoring. *Computers in Human Behavior*, 571–583.

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. A. (2006). Methodology for predicting performances of map-matching algorithms. *W2GIS* 202–213.

Karimi, H. A. (2011). *Universal Navigation on Smartphones*, Spring.

Lehtinen, M., Happonen, A. and Ikonen, J. (2008). Accuracy and time to first fix using consumer-grade GPS receivers. *Software, Telecommunications and Computer Networks.*

Mathie, M. J., Celler, B. G., Lovell, N. H. and Coster, A. C. F. (2004). Classification of basic daily movements using a triaxial accelerometer. *Medical & Biological Engineering & Computing*, Vol. 42.

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.

Ravi, N., Dandekar, N., Mysore, P. and Littman, M. L. (2005). Activity recognition from accelerometer data. *Proceedings of the Seventeenth Innovative Applications of Artificial Intelligence Conference*, 11–18.

Ren, M. and Karimi, H. A. (2009a). A hidden Markov model map matching algorithm for wheelchair navigation. *Journal of Navigation*, Vol. 62, No. 3, pp. 383–395.

Ren, M., Karimi, H. A. (2009b). A Chain-code-based map matching algorithm for wheelchair navigation. *Transactions in GIS*.

Ren, M. and Karimi, H. A. (2011). A fuzzy logic map matching for wheelchair navigation, *GPS Solutions*, June.

Retscher, Günther and Thienelt, Michael. NAVIO – A navigation service for pedestrians. *Journal of Space Communication*, Issue No. 9, 2006.

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), 157–167.

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.

Venkatraman, K., Karthick, N., Naren, J. and Amutha, B. (2009). Sensor-Based Dead Reckoning for land vehicle navigation system. *International Journal of Recent Trends in Engineering*, Vol 2, No. 4.

Sun, Z., Mao, X., Tian, W. and Zhang, X. (2009). Activity classification and dead reckoning for pedestrian navigation with wearable sensors, *Measurement science and technology*.