Deriving Network-Constrained Trajectories from Sporadic Tracking Points Collected in Location-Based Services

LI Xiang  ZHANG Xihui  LIN Hui

Abstract The paper proposes an economical and fast algorithm for deriving trajectories from sporadic tracking points collected in location-based services (LBS). Although many traffic studies or applications can benefit from the derived trajectories, the sporadic tracking points are always implicitly overlooked by most of existing map-matching algorithms. The algorithm proposed in this paper finds network paths or trajectories traveled by vehicles through augmenting GPS data with odometer data. An odometer can provide data of traveled distance which are compared with the lengths of candidate network paths in order to find the most approximate network path approaching the trajectory of a vehicle. Tracking points are classified into anchor points and non-anchor points. The former are used to divide trajectories, and the latter screen candidate network paths. An elliptic selection zone and a reduction process are applied to the selection of possible road segments composing candidate network paths. A brute-force searching algorithm is developed to find candidate network paths and calculate their lengths. A two-step screening process is designed to select the final result from candidate network paths. Finally, a series of experiments are conducted to validate the proposed algorithm.

Keywords LBS; GPS; trajectory; odometer; tracking point; map-matching algorithm

CLC number P208;P228.4

Introduction

The advances of positioning and wireless communication technologies are facilitating the growth of location-based services (LBS)[1]. A typical scenario of LBS is as follows: Vehicles are equipped with GPS receivers, dead reckoning, or other positioning devices, and motorists as subscribers of LBS send their requests as well as current location information to a remote service center via the short message system (SMS) of a wireless network. The service center processes these requests in reference to the accompanying location information, and returns the results to the motorists via SMS. During the above procedure, the location information is collected, transferred, applied, and then discarded or archived once the motorists’ requests are met. Beyond the user-centric appli-
cation scheme, we may further make use of the collected location information. One potential application is to derive trajectories of individual vehicles from location information, and use the trajectories as a supplement to conventional traffic data which are useful and critical to diverse traffic studies and applications. For example, some departments of transportation (DOTs) in the U.S. are considering to purchase trajectory data from private sectors that provide LBS\(^2\). Compared to conventional traffic data, vehicle trajectories can provide a ‘natural’ and ‘direct’ measure of traffic, such as route choice information of individual vehicles which is difficult to collect with existing fixed-location traffic sensors (e.g. CCTV, loop detectors). Furthermore, the aforementioned location information is a byproduct of LBS, and therefore no extra investment is needed in order to collect trajectory data. Compared to the maintenance of fixed-location traffic sensors, the application and maintenance of trajectory data from LBS is quite economical.

The location information of each tracking point physically refers to a data packet which generally consists of geographical coordinates, instantaneous velocity and direction, time, etc. Deriving trajectories from tracking points is a process of reconciling the vehicle’s location with the underlying road network, namely map matching\(^3\). As a result, these trajectories are network-constrained. To date, a number of map-matching algorithms have been proposed\(^4\)-\(^11\), most of which are designed for real-time traffic applications and derive trajectories through improving the accuracy of each tracking point with mathematical algorithms or data fusion. However, because these algorithms usually target frequently-collected tracking points (e.g. at least one point per road segment), they may not be applicable to the tracking points collected in LBS in view of the following reason. According to the context of this paper, deriving trajectories from tracking points will be conducted in the aforementioned service center. From the service center’s perspective, it is stochastic to receive tracking points because motorists send their requests from time to time, which means that the vehicles may have traveled through a couple of road segments between two contingent tracking points. Therefore, even if each tracking point can be precisely matched to a position in the underlying road network, it is still difficult to figure out a continuous trajectory.

To overcome the above problems, this paper proposes an economical and fast algorithm to rebuild trajectories from tracking points collected in LBS through augmenting GPS data with odometer data. The algorithm is supposed to be executed in an LBS center or other traffic management centers. Instead of real-time traffic applications, this algorithm can provide trajectory data to some long-term or offline applications, such as traffic planning, route choice analysis, OD matrix, and road pricing\(^1\),\(^12\).

The reminder of the paper is organized as follows: Section 1 introduces the main principles. Section 2 presents the approach in details; Section 3 implements and validates the approach in a series of experiments; Section 4 concludes the paper.

1 Principles

Suppose that \(P_1\) and \(P_2\) denote two tracking points of a vehicle. \(R_1, R_2, \cdots, R_n\) represent road segments traveled through by this vehicle from \(P_1\) to \(P_2\). Our objective is to find a network path consisting of \(R_1, R_2, \cdots, R_n\) sequentially. This network path is used to represent the trajectory part between \(P_1\) and \(P_2\). One measure to find the traveled network path is to compare the traveled distance (\(\Delta D\)) of this vehicle between \(P_1\) and \(P_2\) to the length (\(\sum L\)) of each candidate network path, and select the most approximate \(\sum L\) and its corresponding road segments.

\(\Delta D\) can be known through an odometer installed in a vehicle, which is usually a cheap distance sensor providing the traveled distance of a vehicle relative to an initial point. An odometer can also be integrated into an inertial navigation system (INS) or dead reckoning system with other sensors, such as gyroscopes and compasses. Combining GPS receiver with an INS can usually locate a vehicle accurately at a high frequency. However, an INS is often designed for real-time navigation which is too complicated and expensive to equip a large number of vehicles. By contrast, our purpose is to find an economical map-matching approach in order to apply stochastic
and sporadic tracking points to offline traffic applications. Therefore, we only choose odometers in this research. Applying GPS data and odometer data to a map-matching process can also be found in the literature\[13\], but they still assume that GPS data are collected at a high frequency and try to provide a vehicle’s position every second.

With an odometer, two traveled distances, namely \(d_1\) and \(d_2\), on points \(P_1\) and \(P_2\) can be collected, \(\Delta D = d_2 - d_1\). However, it is not straightforward to compare the \(\Delta D\) to each \(\sum L\) of candidate network paths because of the following issues:

1. Considering the first and the last road segments, i.e. \(R_1\) and \(R_n\), in a candidate network path, \(P_1\) or \(P_2\) may correspond to a middle point on \(R_1\) or \(R_n\), and thus the \(\sum L\) of the candidate network path should cover only part of \(R_1\) or \(R_n\) and entire \(R_2\), …, \(R_{n-1}\).

2. A large road network may include a number of road segments and intersections. It will be a time-consuming task to calculate the lengths of all network paths between \(P_1\) and \(P_2\).

3. There might be more than one network path whose \(\sum L\) approximates to \(\Delta D\), and the most approximate path may not be necessarily the correct one due to inevitable positioning errors.

In the proposed approach, we address the above issues by classifying tracking points into anchor points and non-anchor points and reducing the number of possible road segments as possible as we can. An anchor point is defined as a tracking point that has been matched to a road segment, i.e. it is known which road segment the vehicle at this anchor point is moving on. A non-anchor point may be so close to several road segments concurrently that thus it is difficult to identify which road segment the vehicle at this non-anchor point is moving on. With a pair of anchor points, the first and the last road segments in each candidate network path have been determined, and \(\Delta D\) between the two anchor points can be used to build a selection zone which only covers possible road segments and thus accelerates the calculation of \(\sum L\). Non-anchor point can be used to screen resulting candidate network paths for the most possible one. The above principles are implemented in the next section.

## 2 The proposed approach

### 2.1 Classify tracking points

We use buffer and overlap analysis to classify the tracking points into anchor points and non-anchor points. For each tracking point, we create a buffer circle whose center is this point and radius the sum of its maximum error range of GPS data (MER) and a lane offset (OFF). MER is the error from GPS receivers, while OFF is the offset between a road segment’s centerline and actually-traveled lanes. The MER of a tracking point is given as follows:

\[
MER = \delta \times PDOP
\]

where \(\delta\) is the precision of GPS pseudo-range (the typical value is 2–6 meters), PDOP (position dilution of precision) is a measure of the geometrical strength of the GPS satellite configuration. Most GPS receivers output NEMA GGA message, and PDOP is one of the elements of GGA. The OFF is as large as a half of a road segment’s width. For example, the OFF of a two-lane road segment may be 3.5 meters or wider (a lane’s width is typically 3.5 meters). As road segments around a tracking point may have different widths, we employ the maximum OFF to fit all road segments.

As shown in Fig.1, according to the definition of MER, the smaller circle with a radius MER covers all possible actual positions, i.e. actual points of a tracking point. An actual point is matched to a point, namely matched point, on the centerline of a road segment. Centerlines are used in the calculation and selection of candidate network paths. An actual point may not be

**Fig.1 The buffer circle of a tracking point**
identical to its matched point unless vehicles can only move along the centerlines. Therefore, OFF is used to extend the search area from the smaller circle to a bigger one. The bigger circle, which is the final buffer circle of the tracking point, covers all possible locations of actual points and matched points, and it intersects with the centerlines of all concerned road segments. A road segment is regarded as a concerned road segment of a tracking point if and only if the segment’s centerline intersects with the point’s buffer circle. MER and OFF jointly define an error range of a tracking point in a relatively simple way. It is remarkable that this range must be larger than those calculated through some complicated methods proposed to precisely locate matched points [11, 14, 15]. However, given the sporadic tracking points and their potential traffic applications, precise positions of actual or matched points are often unnecessary, because determining the concerned road segments of a tracking point is adequate for the proposed approach.

If a tracking point has only one concerned road segment, this point is an anchor point. Otherwise, it is a non-anchor point. The buffer circle of an anchor point usually intersects with a road segment at two points, e.g. points A and B in Fig. 1. Let \( d_{A \rightarrow B} \) denote the direction from point A to point B, and \( -\) denote the tracking point’s instantaneous direction which is extracted from the point’s location information. If \( |d_{A \rightarrow B}| < 90^\circ \), that means the vehicle visits A earlier than B. Otherwise, B is visited earlier than A. Very rarely, if \( |d_{A \rightarrow B}| = 90^\circ \), it means that the relationship between the vehicle and the road segment is uncertain, and thus this anchor point will be treated as a non-anchor point. A more general example can be found in Fig.2, where there are five tracking points, and three of them are anchor points. According to their instantaneous directions, we can ascertain that the vehicle travels through concerned road segments. For example, around tracking point \( P_3 \), the vehicle travels from points C to D, i.e. the vehicle enters road segment \( R_y \) at intersection \( I_3 \) and leaves at intersection \( I_4 \).

2.2 Build selection zones

A selection zone is used to relieve the calculation of candidate network paths. Considering \( P_1 \) and \( P_3 \) in Fig.2, which are a pair of contingent anchor points, it has been known that the vehicle travels from \( I_1 \) to \( I_2 \) via \( R_x \), and from \( I_3 \) to \( I_4 \) via \( R_y \). Therefore, the objective is to find the unknown network path via which the vehicle travels from \( I_2 \) to \( I_3 \).

Points \( A/C \) and \( B/D \) are two intersection points between the buffer circle of \( P_1/P_3 \) and \( P_2/P_3 \’s \) concerned road segment is \( R_x/R_y \). Point \( A/C \) is visited earlier than point \( B/D \). Without loss of generality, we name \( I_2 \) ‘head intersection’ denoted by \( I_h \) and \( I_5 \) ‘tail intersection’ denoted by \( I_t \), respectively. Let \( \sum L_x \) denote the length of the unknown network path, \( \Delta D \) denote the traveled distance between the pair of contingent anchor points, and \( H_1, H_2, T_1, \) and \( T_2 \) represent \( A, B, C, \) and \( D, \) respectively. Except \( \sum L_x \), other variables are known. Then, we have

\[
\sum L_x + H_2 I_h + I_1 T_1 < \Delta D + \ell < \sum L_x + H_1 I_h + I_2 T_2 \quad (2)
\]
or
\[ \Delta D - H_1 I_h - I T_2 - \ell < \sum L_x < \Delta D - H_2 I_h - I T_1 - \ell \] (3)
where \( \ell \) is a small real number representing the error caused by the following reasons. Vehicles cannot always travel along the centerlines of road segments, thus there are always differences between traveled distance and the length of traveled network path. The value of \( \ell \) is generally small and difficult to be precisely anticipated. An alternative is to count \( \ell \) in error ranges of anchor points, which is one reason why we should employ a large OFF when creating buffer circles. The larger the OFF is, the larger the \( H_1 I_h \) and \( I T_2 \) are, the smaller the \( H_2 I_h \) and \( I T_1 \) are, and the broader the range of \( \sum L_x \) is. Therefore, Eq.3 can be changed to the following one:
\[ \Delta D - H_1 I_h - I T_2 < \sum L_x < \Delta D - H_2 I_h - I T_1 \] (4)

Based on Eq.4, an elliptic selection zone can be built. The two distinct foci of the ellipse are respectively \( I_h \) and \( I_t \), and the major axis is \( \Delta D - H_2 I_h - I T_1 \). According to the characteristics of an ellipse, all road segments which may compose network paths meeting Eq.4 must be contained by the ellipse, as shown in Fig.3(a).

The road segments in a selection zone can be further reduced. (1) If a road segment is not connected to any other road segments in the selection zone at either end point (i.e. intersection), and its two end points are not head intersection or tail intersection, respectively, the road segment should be removed. (2) If a road segment is unidirectional, and only one of its end points are connected to other road segments, and the other one is neither head intersection nor tail intersection, it should be removed. For example, if road segment \( R_0 \) in Fig.3(a) is unidirectional, then it is impossibly involved in the unknown network path, and thus \( R_0 \) is removed as shown in Fig.3(b). Furthermore, for the second scenario, the reduction is a propagation process because a road segment being removed might lead to its connected road segments meeting the second scenario and being removed, too. Fig.3(b) is the reduction result of Fig.3(a). With this selection zone and after the reduction, we can search for the unknown network path based on only 8 road segments, instead of the whole network.

### 2.3 Search candidate network paths

We use a brute-force searching algorithm to identify candidate network paths. All network paths from the head intersection to the tail intersection are examined, and their lengths are calculated. A network path is recorded as a candidate network path if its length \( \sum L_x \) meets Eq.4. Otherwise, it is excluded from further analysis. It is required that road segments which compose these network paths be within a selection zone, and this requirement accelerates the computation of this algorithm.

The algorithm requires connectivity information of a road network. A unidirectional road segment is stored as one record including its ID, its start and end intersections’ IDs, and its length, while a bidirectional road segment is simply treated as two unidirectional road segments which have the same centerline and length but different IDs and inversed end points. Furthermore, in order to accelerate the access to connectivity information, a turn table is created to record the connectivity information at intersections\[16\]. Each record in the turn table includes three fields, an intersection’s ID, a ‘from’ road segment’s ID, and a ‘to’ road segment’s ID. Therefore, vehicles can only move from the former (‘from’) road segment through the intersection to the latter (‘to’) road segment. The major part of this algorithm is a recursive procedure ‘FindNetworkPath’ illustrated in Fig.4, where the initial value of \( I_c \) (current intersection) is \( I_h \) (head intersection); \( m \_ path \) recording the current network path...
is empty; ∑L recording the length of m_path is 0; and variables in Eq.4 are also used.

The above algorithm can be used to find all candidate network paths for the trajectory part from Ih to It. A constraint of this algorithm is that a unidirectional road segment can only appear once in a candidate network path; otherwise, the computational time will be unacceptably long. Fig.5 illustrates several scenarios which can or cannot be handled by this algorithm. In scenario A, a vehicle visits one road segment only once, which can be handled. In scenario B, a vehicle visits a bidirectional road segment twice, which can also be handled because this algorithm treats it as two different unidirectional road segments. In scenario C, a vehicle makes a U-turn in the middle of the road segment, which cannot be handled, but it is usually prohibited in urban road network. Scenarios D and E cannot be handled, but they rarely happen during normal and continuous driving. Scenario F can be handled because the three visits of the vehicle on the same road segment happen in different trajectory parts.

### 2.4 Screen candidate network paths

If the number of candidate network paths is more than one, a screening process is employed to determine the most possible one through the following steps.

- **Candidate network paths** are sorted by the ascending differences between their lengths and the mean (∑ΣLx) of the value domain of ∑Lx, i.e.

  \[
  MΣΣx = \frac{(ΔD - H2Ih - I1T1) + (ΔD - H1Ih - I2T2)}{2} \quad (5)
  \]

- If there is no non-anchor point between the current two anchor points, the first one of the sorted candidate network paths is the result. Otherwise, starting from the first one, examine each candidate network path until it is found that the examined network path has sequentially included one of the concerned road segments of every non-anchor point.
For example, in Fig.3(b), suppose that there are two candidate network paths, path \( a \): \([R_1, R_3, R_4, R_8]\) and path \( b \): \([R_1, R_2, R_6, R_7]\). Let \( \sum L_a \) and \( \sum L_b \) denote the lengths of paths \( a \) and \( b \), respectively. \( \sum L_a - M \sum L_x \geq \sum L_b - M \sum L_x \), i.e. path \( b \) has more probability of being the unknown network path to approach the trajectory part between \( I_2 \) and \( I_3 \) than path \( a \). There is one non-anchor point (\( P_2 \)) between \( P_1 \) and \( P_3 \) (as shown in Fig.2) and \( P_2 \) has 3 concerned road segments in the selection zone, i.e. \( R_3, R_4, \) and \( R_5 \). The unknown network path must include at least one of them. Eventually, path \( b \) is excluded, while path \( a \) is selected. The final result is described as follows: between \( P_1 \) and \( P_2 \), the vehicle first travels on \( R_x \) towards \( I_2 \), then travels along path \( a \), and finally enters \( R_y \) from \( I_3 \).

When the unknown network path between a pair of contingent anchor points is determined, \( I_t \) in this pair and the next anchor point respectively become \( I_h \) and \( I_t \) composing the next pair of anchor points. The proposed approach is continually applied until the last anchor point is reached.

3 Experiments

3.1 Data collection and processing

The road network of Tai Po in Hong Kong is used as the underlying network including 1,210 road segments and 1,099 intersections. The road centerline data come from the Lands Department of Hong Kong which creates the data from 1:1,000 paper maps. After preprocessing, we convert the data format to shape files which can be read by ESRI’s software products. Each road segment or intersection is assigned a unique ID. Topological information of the road network is saved in road segments’ records and also in a turn table.

GPS data and odometer data are collected through a vehicle equipped with a GPS receiver and an odometer. Six groups of tracking data along the same network path are respectively collected at different frequencies, including 1Hz, 0.05Hz, 0.02Hz, 0.01Hz, 0.008Hz, and 0.005Hz. Parts of these data are illustrated in Fig.6. The general moving direction is from the left side to the right side. The lower the frequency is, the sparser the tracking points are. When the frequency is lower than 0.02Hz, the tracking points become sporadic like those collected in LBS.

According to Figs.6(a) and 6(b), we can visually identify the actual trajectory and network path traveled by the vehicle. The purpose of this experiment is to calculate the unknown network paths for the other four groups with sporadic tracking points and evaluate the results.

3.2 Implementation of the proposed algorithm

A prototype is developed to implement the proposed algorithm with Java and ESRI’s MapObject. In particular, the map-matching process based on the data at a frequency of 0.008Hz (Fig.6(e)) is described in detail. The proposed approach separately deals with each trajectory part divided by anchor points, and the map-matching processes of different trajectory parts are independent of each other and, therefore, errors will not accumulate or propagate. Thus, we only demonstrate the map-matching process of one trajectory part. As shown in Fig.7(a), there are four tracking points. We respectively name them points \( A, B, C, \) and \( D \) from the left side to the right side, and they are also traveled by the vehicle in the same sequence.

First, a buffer circle is created for each tracking point according to the parameters in Table 1. MER is determined by the GPS data of individual tracking point and Eq.1. \( \delta \) is 4, and PDOP varies from 2.6 to 6.7. OFF is 7, because most road segments in this part of the road network have four lanes with a width no more than 14
meters. The resulting buffer circles are shown in Fig.7(a). It is remarkable that points A and D are anchor points, while points B and C are non-anchor points.

![Fig.7 Buffer circles and selection zone](image)

**Table 1 Parameters of buffer circles**

| Point | δ/m | PDOP/m | MER/m | OFF/m | Radius/m | Concerned road segments’ IDs |
|-------|-----|--------|-------|-------|----------|-------------------------------|
| A     | 4   | 2.6    | 10.4  | 7     | 17.4     | 461                           |
| B     | 4   | 6.2    | 24.8  | 7     | 31.8     | 504, 508, 564                 |
| C     | 4   | 6.7    | 26.8  | 7     | 33.8     | 589, 590, 591, 596            |
| D     | 4.7 | 18.8   | 7     |       | 25.8     | 628                           |

Second, for the trajectory part from anchor points A to D, the head intersection and the tail intersection are determined, and the range of the unknown network path’s length (\(\sum L_x\)) between the two intersections is calculated according to odometer data and Eq.4. These parameters are listed in Table 2. Based on them, a selection zone is created and shown in Fig.7(b), where two black points are I₁h and I₁t, respectively, and road segments within the selection zone are highlighted. A reduction process is applied to the selection zone to further reduce possible road segments (not illustrated in this figure). After reduction, the number of possible road segments changes from 290 to 167.

**Table 2 Parameters of the selection zone**

| ID of I₁h | ID of I₁t | ΔD/m | \(H_1I₁h/m\) | \(H_1I₁t/m\) | \(I₁T₁/m\) | \(I₁T₂/m\) | Range of \(\sum L_x/m\) | \(M\sum L_x/m\) |
|-----------|-----------|------|---------------|---------------|-----------|-----------|-------------------------|-----------------|
| 333       | 520       | 1993 | 216           | 195           | 61        | 109       | [1668, 1737]            | 1702.5          |

Third, based on the selection zone, several candidate network paths have been figured out, two of them whose lengths are closer to \(M\sum L_x\) than others are illustrated in Fig.8. The path in Fig.8(b) is closer to \(M\sum L_x\) than the other one, but it does not include any concerned road segments of non-anchor point B (as shown in Table 1). On the other hand, the path in Fig.8(a) includes road segments 504 and 508, which are two concerned road segments of point B, and also includes the other non-anchor point’s concerned road segments. Therefore, the path in Fig.8(a) is eventually selected, and according to Fig.6(a), this path is just the actual path traveled by the vehicle.

![Fig.8 Two candidate network paths](image)

### 3.3 Results and discussion

The same procedure is applied to other low-frequency tracking data. Fig.9(a) illustrates the error rate of each dataset, which equals to the quotient of dividing falsely-matched trajectory’s length by the whole length. Fig.9(b) illustrates the average traveled
distance between two contingent anchor points. It is remarkable that there is a positive correlation between the two variables. In conclusion, errors are relevant to the following issues:

(1) The more the parallel road segments are, the more the network paths with approximate lengths are, and the higher the error rate is.

(2) The sparser the tracking points are, the higher the error rate is.

(3) The more the anchor points are, the less the error rate is.

The first issue is determined by the geometric features of the underlying road network, the second one the frequency of receiving tracking points, and the third one the positioning technologies. Considering the last two issues in the context of LBS, there is no fixed frequency of receiving tracking points, and thus it is impossible to define a frequency threshold for the proposed algorithm. However, with the growth of LBS, the frequency will increase correspondingly. Furthermore, the advances of positioning technologies will increase the number of anchor points.

We further compare the average computational time for the map matching of a trajectory part between two contingent anchor points with the proposed searching algorithm (section 2.3) under three scenarios in Fig.10(a). The three scenarios are defined in Fig.10, where scenario A represents the proposed approach, and the reduction process refers to removing impossible road segments from selection zones (section 2.2). The number of recursive times in Fig.10(b) refers to the number of executing recursive function ‘FindNetworkPath’. As shown in Fig.10(a), the computational time under scenario A is the shortest among all scenarios, especially when the frequency is lower than 0.008Hz. From Figs.9 and 10, we can conclude that the factors affecting the computational time may include the number of recursive times, the number of possible road segments, and the length of a trajectory part.

4 Conclusion

The paper proposes an economical and fast map-matching algorithm to sporadic tracking points collected in LBS. Most of existing map-matching algorithms overlook these tracking points, though the derived trajectories from them can be used in many traffic studies or applications.
The proposed algorithm derives trajectories from sporadic tracking points through augmenting GPS data with odometer data. The data of traveled distance provided by odometers are compared with the lengths of candidate network paths in order to find the most approximate network path approaching the trajectory of a vehicle. Through analyzing the concerned road segments of tracking points, we can figure out anchor and non-anchor points, which are used to reduce the possible road segments composing candidate network paths and accelerate the searching, computing, and screening processes for these network paths to a great extent. The proposed algorithm is validated by a series of experiments. In particular, the approach of selecting possible road segments with selection zones dramatically saves the computational time.

The performance of the proposed algorithm varies with the geometric features of the underlying road network, the positioning accuracy, and the frequency of receiving tracking points. With the advances of positioning technologies and the growth of LBS, its performance will increase correspondingly.

The proposed algorithm is supposed to be executed in LBS centers or other traffic management centers. The derived trajectories are useful and critical to long-term or offline traffic applications. During the process, however, one should consider to protect the motorists’ privacy because their trajectory information is collected and analyzed. It is easy to remove personal identification information from derived trajectories. Furthermore, LBS companies can sell tracking points or derived trajectories to DOTs, and use the income to partly cover motorists’ LBS fees. This may serve as a potential driver to the growth of LBS.

References

[1] Jagoe A (2003) Mobile location services: the definitive guide[M]. Upper Saddle River, NJ: Prentice Hall
[2] Barron K, Collins J, Derr R, et al. (2004) The future of travel time data-a paradigm shift[OL]. http://depts.washington.edu/abh220/reports/FinalTrvTime2-3-2004.pdf
[3] Maguire D J, Goodchild M F, Rhind D W (1991) Geographical information systems: principles and applications [M]. London: Longman
[4] Scott C (1994) Improving GPS positioning for motor vehicle through map matching[C]. The Seventh International Technical Meeting of the Satellite of Navigation, Salt Lake City, Utah
[5] Kim S, Kim J-H (2001) Adaptive fuzzy-network-based c-measure map-matching algorithm for car navigation system[J]. IEEE Transactions on Industrial Electronics, 48(2): 432-441
[6] Taylor G, Uff J, Al-Hamadani A (2001) GPS positioning using map-matching algorithms, drive restriction information and road network connectivity[C]. GIS Research UK 2001 9th Annual Conference, Wales
[7] White C E, Bernstein D, Kornhauser A L (2000) Some map matching algorithms for personal navigation assistants[J]. Transportation Research Part C, 8: 91-108
[8] Zhao Y L (1997) Vehicle location and navigation systems [M]. Boston, Mass: Artech House
[9] Yu M, Wu C, Li Z (2002) A simplified map-matching algorithm for in-vehicle navigation unit[J]. Geographic Information Sciences, 8(1): 24-30
[10] Li X, Lin H, Zhao Y B (2005) A connectivity-based map matching algorithm[J]. Asian Journal of Geoinformatics, 5(3): 69-76
[11] Blewitt G, Taylor G (2002) Mapping dilution of precision (MDOP) and map-matched GPS[J]. International Journal of Geographical Information Science, 16(1): 55-67
[12] Taylor M A P, Bonsall P W, Young W (2000) Understanding traffic systems: data, analysis, and presentation[M]. Aldershot: Ashgate
[13] Li J, Taylor G, Brunsdon C, et al. (2004) A test-bed simulation for GPS and GIS integrated navigation and positioning research: bus positioning, using GPS observations, odometer readings and map matching[C]. 12th Int. Conf. on Geoinformatics, Sweden
[14] Li J, Taylor G, Kidner D B (2005) Accuracy and reliability of map-matched GPS coordinates: the dependence on terrain model resolution and interpolation algorithm[J]. Computer & Geosciences, 31(2): 241-251
[15] Taylor G, Blewitt G, Steup D, et al. (2001) Road reduction filtering for GPS-GIS navigation[J]. Transactions in GIS, 5(3): 193-207
[16] Miller H J, Shaw S L (2001) Geographic information systems for transportation: principles and applications[M]. New York: Oxford University Press