

Deriving average delay of traffic flow around intersections from vehicle trajectory data

Minyue ZHAO^{1,2}, Xiang LI (✉)²

¹ Shanghai Key Laboratory for Urban Ecology and Sustainability, East China Normal University, Shanghai 200062, China

² Key Laboratory of Geographical Information Science (Ministry of Education), East China Normal University, Shanghai 200062, China

© Higher Education Press and Springer-Verlag Berlin Heidelberg 2012

Abstract Advances of positioning and wireless communication technologies make it possible to collect a large number of trajectory data of moving vehicles in a fast and convenient fashion. The data can be applied to various fields such as traffic study. In this paper, we attempt to derive average delay of traffic flow around intersections and verify the results with changes of time. The intersection zone is delineated first. Positioning points geographically located within this zone are selected, and then outliers are removed. Turn trips are extracted from selected trajectory data. Each trip, physically consisting of time-series positioning points, is identified with entry road segment and turning direction, i.e. target road segment. Turn trips are grouped into different categories according to their time attributes. Then, delay of each trip during a turn is calculated with its recorded speed. Delays of all trips in the same period of time are plotted to observe the change pattern of traffic conditions. Compared to conventional approaches, the proposed method can be applied to those intersections without fixed data collection devices such as loop detectors since a large number of trajectory data can always provide a more complete spatio-temporal picture of a road network. With respect to data availability, taxi trajectory data and an intersection in Shanghai are employed to test the proposed methodology. Results demonstrate its applicability.

Keywords trajectory data, traffic flow information, intersection

1 Introduction

Nowadays, methods of obtaining valid dynamic traffic

information have evolved into a hot issue in traffic related fields. In the past few decades, we have witnessed evolution of traffic data collection methods. In the past, on-site collection methods were widely used, but they were often time-consuming and of low accuracy. With the development of computer technology and video processing technology, loop detectors, closed-circuit televisions, etc., have been adopted. However, problems still exist with these fixed-point detectors, such as excessive investment in hardware, limited coverage of facilities, heavy workload of maintenance and installation of facilities, and low data accuracy. These drawbacks directly lead to a problem that the collected traffic information cannot completely meet the needs of research and management of modern transportation. With the advances of positioning and wireless communicating technologies, trajectory data of vehicles collected with GPS receivers becomes a fundamental part for detecting present traffic status which makes it possible to optimize traffic organization. For example research on estimating traffic parameters based on buses as floating cars and the application of floating car data system in Britain (Ruey et al., 2001; Cowan and Gates, 2002; Tantiyanugulchai and Bertini, 2003). Floating cars generally refer to buses and taxis driving on the main roads in the city, which are equipped with GPS receivers. GPS receivers can continuously record locations of moving vehicles at a given frequency. Typical trajectory data are a series of GPS positioning points in time order containing attributes of position (i.e., latitude and longitude), speed, direction information, etc. With the march of time and increased acquisition frequency, large volume of trajectory data can be obtained, and a complete spatio-temporal picture covering the whole road network can be provided.

With the rapid development of urbanization, traffic congestion as well as traffic accidents, energy waste, and environment pollution become ubiquitous and permanent problems that plague cities. In this paper, we attempt to relieve these problems through deriving traffic information

from vehicle trajectory data composed of large number of discrete positioning points for traffic study. Instead of the whole road network, we put effort on the portion of intersections where vehicles and pedestrians gather, pass through, and change direction. Traffic interference, delay, congestion, and accidents are prone to take place in intersections. Optimal traffic organization around intersections is of vital importance to improve traffic capacity, avoid traffic congestion, and reduce traffic accidents. This paper focuses on the methodology of extracting average delay of traffic flow around intersections in a road network.

In literature, some work has been conducted using GPS data to estimate traffic status, travel route time, and delay of intersections such as HCM (US road capacity manual), Australian method, Canadian method, modified HCM, and TRANSYT8 (Taylor et al., 2000; Cheu and Lee, 2001; Kerner et al., 2005). In addition, Li designed a system that can calculate delay automatically based on image processing technology (Li et al., 2009). On the foundation of existing delay research, Liu has further divided three traffic conditions of under-saturated, critical-saturated, and over-saturated into six traffic situations and then Liu presents a method of calculating intersection delay under signal control (Liu et al., 2007). In general, methods of calculating intersection delay include investigation methods, analytical methods, and simulation methods. Investigation methods require a large number of sampling data but significant deviation may still exist in the calculation results. Analytical methods rely on mathematical models with assumptions which may make it difficult to precisely reflect actual situations. Simulation methods simulate changes of traffic flow through computer software and adjust control parameters to meet requirements of different projects (Chen et al., 2005).

Most of the above methods depend on data collected with fixed detectors, such as ring loop, microwave equipment, infrared equipment, etc., whereas it is sometimes inadequate to estimate traffic status based on data of a particular place alone even if all intersections in a road network were covered by fixed detectors. To overcome these shortcomings, floating data are introduced to the dynamic traffic fields. Floating vehicle technology was introduced by Wardrop and Charles Worth of England Road Research Institute in 1954 (Turner et al., 1998). From the perspective of data collection, the data collected with a floating car should correctly show the real traffic status. A floating car should be the representative vehicle in the traffic flow of a city (Ding, 2004). Xiong proposed two methods, the density-based and the deceleration point-based, to dynamically delineate the traffic flow information around intersections based on floating car data, which is utilized to make analysis of traffic delay at intersections (Xiong, 2009). Zhu designed an intersection delay estimation model based on vehicle trajectory data after considering the characteristics and scopes of applications

of models to meet the complex and inconstant urban transportation environment. However, the estimation range of this model is limited (Zhu and Li, 2011).

Based on existing work, we propose an effective method in this paper, which first figures out trajectory data relevant to given intersections, then derives trips in terms of individual vehicle and direction, and finally estimates average delay of traffic flow around the intersection over time with a proposed algorithm.

The structure of this paper is organized as follows: Section 2 introduces several data structures involved in our approach. Section 3 describes approaches to derive traffic flow information from trajectory data. Section 4 demonstrates the process of calculating average delay of traffic flow around intersections at different periods of time within a day, and analyzes the results. Section 5 concludes the paper and proposes further research direction.

2 Data structure

2.1 Intersection

An intersection data structure is used to record the positions of an intersection and its adjacent road segments. Let I denote an intersection, where $I = \langle P_I, ListP, R, ListW \rangle$. P_I denotes the position of the intersection point, where $P_I = \langle x_I, y_I \rangle$. $ListP$ is a point set with m adjacent intersection points which locate on the connected road segments of P_I , where $ListP = \{P_1, P_2, \dots, P_m\}$, for any element P_i in $ListP$, $P_i = \langle x_i, y_i \rangle$, $i \in [1, m]$. R represents the minimum distance from intersection point P_I to the other adjacent intersection points in $ListP$. $ListW$ is a list of road widths, where $ListW = \{W_1, W_2, \dots, W_m\}$.

2.2 Trajectory

A trajectory data structure records attributes of discrete positioning points. Let GPS denote a positioning point, where $GPS = \langle PointID, TermID, x, y, Time, Speed, Direction \rangle$. $PointID$, $TermID$, x , y , $Time$, $Speed$, and $Direction$ denote the identity of each point record, vehicle identity, x and y coordinate, acquisition time, instantaneous speed, and direction angle of the trajectory points, respectively.

2.3 Trip

A trip data structure contains a set of positioning points in time order belonging to a single vehicle passing through an intersection. Let T denote a trip, where $T = \langle TripID, ListGPS, Road, TimeQuantum, DelayTime \rangle$. $ListGPS$ is a set with p sample positioning points in time order, where $ListGPS = \{GPS_1, GPS_2, \dots, GPS_p\}$. $TripID$, $Road$, $TimeQuantum$, and $DelayTime$ denote trip identity, identity

of the road to which the trip belongs, time identity of the trip, and delay of the trip, respectively. Divide one day into 24 intervals, and incrementally number them, that is to say each hour represents one interval. According to *Time* in *ListGPS*, the interval that *Time* is within represents *TimeQuantum*.

3 Methodology

3.1 Delineate intersection range

To estimate average delay of traffic flow around intersections, we first delineate the estimation range of each intersection. In view of the huge trajectory data which can reduce the efficiency of data processing, we determine trajectory data within the delineated range for a given intersection. As shown in Fig. 1, an intersection buffer is established to fulfill the trajectory data determination.

Two parameters, R and W , are used to define an intersection buffer, where R represents the minimum distance from intersection point P_I to another adjacent intersection point in *ListP*, and W , the maximum of *ListW*, represents the buffer radius of the related road segments. First, a circle with center P_I and radius R is created to build buffer one. Since a large amount of invalid trajectory data, such as taxi trajectory data in a residential area, might still be covered by buffer one, a refining process is needed. Linking P_I and other adjacent intersection points P_1, P_2, P_3, P_4 in *ListP* to get corresponding road axes, we can delineate buffer two containing four rectangles with a bilateral offset W . The overlay of buffer one and two determines the estimation range of the intersection which is used to distinguish related positioning points from other invalid points.

3.2 Trips extraction

Vehicle trajectory data collected with GPS receivers is a discrete representation of vehicle status (position, speed, direction, etc.), whereas continuous changes of vehicle status can be obtained through deriving trips from discrete trajectory data with the following algorithm:

- 1) Define *VehicleTrajectory* and a time threshold S ;
- 2) Sort trajectory data by *TermID* and *Time*;
- 3) i, j, k are integers for iterations, and initially they are 0.

If $i < p$ (p represents the number of positioning points within the intersection buffer, and initially it is 0), $j < q$ (q represents the number of positioning points in *VehicleTrajectory*, and initially it is 0), $k < m$ (m represents the number of positioning points in T , and initially it is 0), repeat the following procedure:

- Add GPS_i into *ListGPS*.
- If $GPS_i. TermID = GPS_{i+1}. TermID$, then add GPS_{i+1} into *ListGPS*.
- $VehicleTrajectory = \langle TermID, ListGPS \rangle$.
- Clear *ListGPS*, add *VehicleTrajectory. ListGPS [j]* into *ListGPS*.
- If $VehicleTrajectory. ListGPS [j + 1]. Time - VehicleTrajectory. ListGPS [j]. Time = S$, then add *VehicleTrajectory. ListGPS [j + 1]* into *ListGPS*.
- Assign *ListGPS* to T .
- Calculate distance d_k from P_I to each positioning point in T , if $d_2 - d_1 < 0$ and $d_m - d_{m-1} > 0$, then T belongs to the intersection.
- $i = i + 1, j = j + 1, k = k + 1$.

In the above algorithm, first, sort positioning points with the same *TermID* in time order to get the trajectory of each vehicle. Define $VehicleTrajectory = \langle TermID, ListGPS \rangle$. Second, each *VehicleTrajectory* may consist of more than one trip within a certain intersection, considering that a

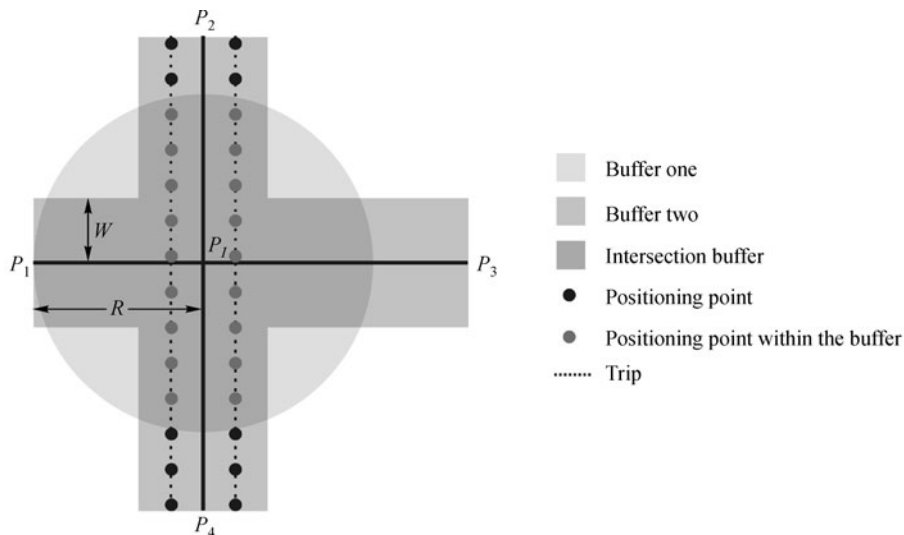


Fig. 1 Intersection buffer

vehicle may pass through the same intersection more than once during the given period of time. Therefore, trips are further extracted from *VehicleTrajectory* in the following way: Since positioning points are often collected at a fixed frequency (e.g., 1 point per 10 s in the sampling data), a time threshold S is defined and if the time interval between two continuous points in the same group is larger than S , then record the two points in two different trips. Finally, for each trip, determine whether it belongs to the intersection P_I or not. The following rule is adopted: If the distance between each positioning point to the intersection P_I decreases and then increases by time, then the trip is counted; otherwise, it is abandoned.

3.3 Calculate average delay

The iteration integer, i , is initially set to 0. The variable a represents acceleration. If $i < m$ (m represents the number of positioning points in T), repeat the following procedure:

- $Speed_i = T.ListGPS[i]$. $Speed$, $i \in [1, m-1]$.
- if $Speed_i > V$ and $Speed_{i+1} > V$, then $t_i = 0$.
- else if $Speed_i < V$ and $Speed_{i+1} < V$, then $t_i = TimeInterval$.
- else if $Speed_i > V$ and $Speed_{i+1} < V$, then $t_i = TimeInterval * (Speed_{i+1} - V) / (Speed_{i+1} - Speed_i)$. (Since $S_{i+1} = S_i + a * TimeInterval$; $S_{i+1} = V + a * t_i$, t_i can be sorted out.)
- else if $Speed_i < V$ and $Speed_{i+1} > V$, then $t_i = TimeInterval * (V - Speed_i) / (Speed_{i+1} - Speed_i)$. (Since $S_{i+1} = S_i + a * TimeInterval$; $V = S_i + a * t_i$, t_i can be sorted out.)
- $T.DelayTime = Sum(t_i)$.
- $i = i + 1$.

First, calculate the delay of each trip. Employ a speed threshold V to determine whether a vehicle is moving or stopped, and assume that there are p positioning points in a trip. For any positioning point i , $Speed_i = T.ListGPS[i]$. $Speed$, $i \in [1, p-1]$. $TimeInterval$ represents the data collection frequency. There are four cases:

Case 1: if $Speed_i > V$, and $Speed_{i+1} > V$, then $t_i = 0$;

Case 2: if $Speed_i < V$, and $Speed_{i+1} < V$, then $t_i = TimeInterval$;

Case 3: if $Speed_i > V$, and $Speed_{i+1} < V$, then $t_i = TimeInterval * (Speed_{i+1} - V) / (Speed_{i+1} - Speed_i)$;

Case 4: if $Speed_i < V$, and $Speed_{i+1} > V$, then $t_i = TimeInterval * (V - Speed_i) / (Speed_{i+1} - Speed_i)$;

The following example demonstrates the above process. Fig. 2 is sample data and Table 1 shows the result:

Assume speed threshold $V = 10$, $TimeInterval = 10$, and we can respectively calculate t_1-t_{12} .

Then, according to the given period of time, determine the *TimeQuantum* that the given period of time belongs to, and the trip with the same *TimeQuantum* counts, and thus we can calculate the number of trips n and the sum of delay t of all the trips in this period of time. We then finally

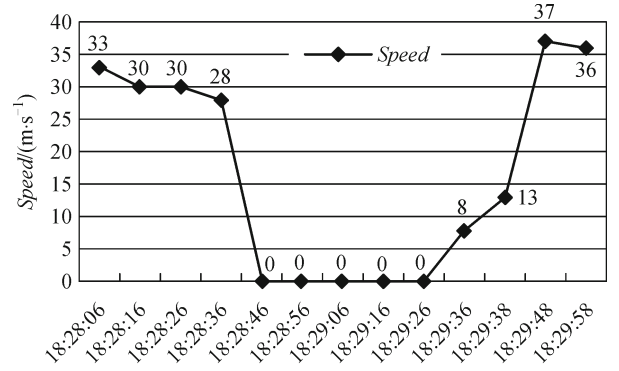


Fig. 2 Line chart of speed versus time

Table 1 Delay time of each situation

i	$Speed_i / (m \cdot s^{-1})$	$Speed_{i+1} / (m \cdot s^{-1})$	Case	t_i / s
1	33	30	1	0
2	30	30	1	0
3	30	28	1	0
4	28	0	3	3.57
5	0	0	2	10
6	0	0	2	10
7	0	0	2	10
8	0	0	2	10
9	0	8	2	10
10	8	13	4	4
11	13	37	1	0
12	37	36	1	0

Note: $T.DelayTime = Sum(t_i) = 57.57$

calculate average delay of traffic flow around intersections $t' = t / n$.

3.4 Map matching

The objective of map matching is to assign each trip to corresponding road segments in order to estimate aggregated average delay based on individual turns around an intersection. The following approach is applied:

For any intersection $I = \langle P_I, ListP, R, ListW \rangle$, i is an integer for iterations, and initially it is 0, if $i < p$ (p represents the number of adjacent positioning points in $ListP$), repeat the following procedure:

- List the linear equation of P_I and P_i in $ListP[i]$.
- Seek the two intersection points of linear equation $(x - P_I, x_I) / (P_i, x_i - P_I, x_I) = (y - P_I, y_I) / (P_i, y_i - P_I, y_I)$ and circle equation $(x - P_I, x_I)^2 + (y - P_I, y_I)^2 = R^2$.
- Calculate distances from positioning point to intersection points, and retain the intersection point with the closer distance.

Calculate distances from positioning point to p intersection points retained. The positioning point belongs to the road that the intersection point with the shortest distance belongs to.

4 Case study

4.1 Study area

In 2008, there were up to 48641 taxis in Shanghai (Shanghai Municipal Statistical Bureau, 2008), and about 27000 of them are equipped with GPS receivers (Shanghai City Comprehensive Transportation Planning Institute, 2008). In our experiments, trajectory data collected by 1950 taxis on October 21, 2007 is employed. There are 15439163 positioning points in total and sampling collection frequency is 0.1 Hz.

The intersection of Deping road and Yushan road in Pudong district, Shanghai, as shown in Fig. 3, is selected as an example to test the potential of our approach. Deping road extends in a north-west-south-east direction, and Yushan road in north-east-south-west direction. This intersection is a crossroad, associated with four road segments. We derive average delay of traffic flow around the intersection in different periods of time, and then compare the results.



Fig. 3 Indicated map of intersection

4.2 Data processing

Raw trajectory data includes 15439163 positioning points. Intersection coordinates $P_I = (21363026.1406, 3459213.75281)$, adjacent intersections coordinate set $ListP = \{(21362848.1, 3459042.497), (21362902.84, 3459411.893), (21363240.15, 3459337.628), (21363173.05, 3459013.533)\}$. We selected 3888 positioning point with the intersection buffer defined by P_I and $ListP$. We extracted 345 corresponding trips passing

through the intersection. Partial results follow:

```
TripID = 4;
ListGPS
= {10, 10079, 21362931, 3459118, 18:45:47, 37, 44
  11, 10079, 21362999, 3459186, 18:45:57, 30, 45
  12, 10079, 21363062, 3459238, 18:46:07, 30, 54
  13, 10079, 21363147, 3459284, 18:46:17, 51, 60};
```

Use trip data as input parameter to derive delay of each trip in different time within the day which consists of 15 periods of time from 7 am to 10 pm with an interval of one hour. Divide trips into 8 groups according to road segments of this intersection and calculate the average delay of each group, i.e. the average delay of the traffic flow around the intersection in this period of time.

4.3 Results and analysis

Figure 4 illustrates the changes of average delay on different road segments and directions at different periods of time.

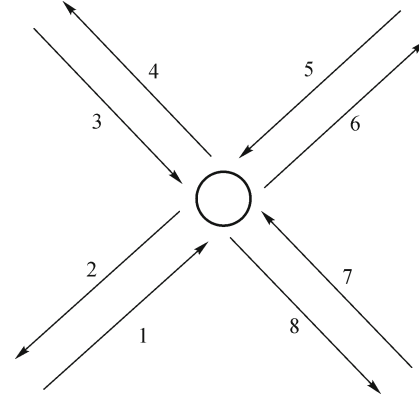


Fig. 4 Directional road segments

We can conclude that average delay of different roads in the order from the Fig. 5 reaches peak in 16:00–17:00, 21:00–22:00, 20:00–21:00 and 7:00–8:00. In addition, traffic on roads of different directions is relatively heavy between 7:00 and 9:00. The reason is as follows: Northwest of Deping the road is close to the Deping road subway station resulting in congestion between 7:00–8:00. Southeast of Deping road is close to Xiangshan elementary school and Xiangshan middle school, leading to congestion between 7:00–8:00. Therefore, average delay estimated by the proposed method can primarily reflect the geographical and traffic characteristics of intersections.

5 Conclusions

This paper proposes a method to efficiently process trajectory data and derive average delay of traffic flow around intersections at different periods of time. We prove

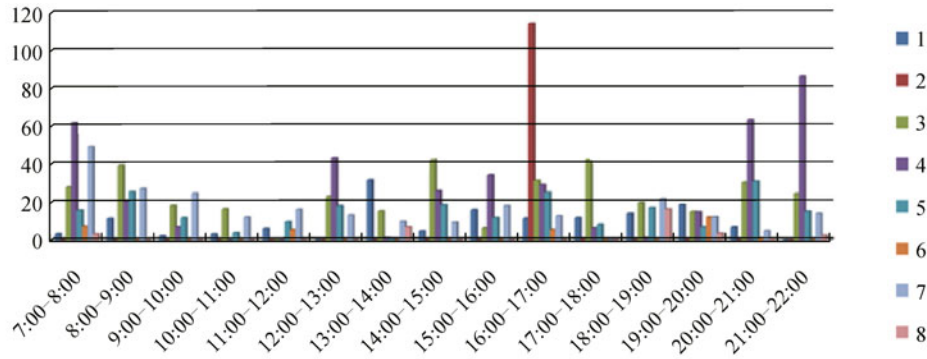


Fig. 5 Average delays on different road segments and directions at different periods of time

a possible way to derive useful traffic information from real-time dynamic data. In future work, we may take the influence of traffic signals into account; also we may consider applying this approach to other intersections in a road network to obtain an integrated thematic map on turn delay around intersections with spatial and temporal characteristics. Further applications can be explored, such as estimating vehicle emissions around intersections.

Acknowledgements This research was sponsored by Open Grant from Shanghai Key Laboratory for Urban Ecology and Sustainability (SHUES), the National Natural Science Foundation of China (Grant No. 41271441), Director Grant of Key Laboratory of Geographical Information Science (No. KLGIS2011C01), and Shanghai Natural Science Foundation (No. 11ZR1410100).

References

- Chen S, Guo J, Wang X, Mao B (2005). Comparison of calculation method of intersection delay. *Beijing Jiaotong University Journal*, 29: 3 (in Chinese)
- Chen R L, Lee D H (2001). An arterial speed estimation model fusing data from stationary and mobile Sensors. In: *Proceedings of IEEE Conference on ITS*. Aug 25–29, Oakland, USA, 573–578
- Cowan K, Gates G (2002). Floating vehicle data system—realization of a commercial system. In: *11th International Conference on Road Transport Information and Control*. UK: London, 187–189
- Ding K (2004). Link travel time estimation based on integration GPS-GIS. In: *World Engineers' Convention*, Nov 2, Shanghai, China
- Kerner B, Demir C, Herrtwich R (2005). Traffic state detection with floating car data in road net works. In: *Proceedings of IEEE Conference on ITS*. Sept 16, Tucson, USA, 700–705
- Li Z, Li N, Liu F (2009). An effective calculating method of signalized intersection delay. In: *2009 International Conference on Digital Image Processing, ICDIP*. Mar 07–09, Bangkok, Thailand, 211–215
- Liu G, Zhai R, Pei Y (2007). A calculating method of intersection delay under signal control. In: *Proceedings of Intelligent Transportation Systems Conference*. Sept. 30–Oct. 3, Seattle, USA, 648–653
- Ruey L C, Der-Horng L, Chi X (2001). An arterial speed estimation model fusing data from stationary and mobile sensors. In: *Proceedings of 2001 IEEE Intelligent Transportation Systems*. Aug 25–29, Oakland, USA, 573–578
- Shanghai City Comprehensive Transportation Planning Institute (2008). *Optimization of traffic monitoring based on information technology in major cities* (in Chinese)
- Shanghai Municipal Statistical Bureau (2008). *Shanghai Statistical Yearbook 2008*. Beijing: China Statistics Press (in Chinese)
- Tantiyanugulchai S, Bertini R L (2003). Arterial performance measurement using transit buses as probe vehicles. In: *Proceedings of IEEE International Conference on Intelligent Transportation Systems*. Oct 12–15, Shanghai, China, 102–107
- Taylor MAP, Woolley J E, Zito R (2000). Integration of the global positioning system and geographical information systems for traffic congestion studies. *Transportation Research*, 8(1–6): 257–285
- Turner S, Eisele W L, Benz R J, Holdener D J (1998). *Travel Time Data Collection Handbook*. Office of Highway Information Management Federal Highway Administration U.S. Department Of Transportation
- Xiong Y (2009). Methods on dynamic delineation of intersections based on floating car data. In: *Proceedings of the 2009 12th International IEEE Conference on Intelligent Transportation Systems*. Oct 4–7, St. Louis, USA, 6
- Zhu W, Li N (2011). Estimation model of trip time and intersection delay time based on GPS data. *Shenyang Architecture Journal (Science Edition)*, 27:1 (in Chinese)