Spatiotemporal Segmentation of Metro Trips Using Smart Card Data

Fan Zhang, Juanjuan Zhao, Chen Tian, Chengzhong Xu, *Senior Member, IEEE*, Xue Liu, *Member, IEEE*, and Lei Rao

***Abstract*—Contactless smart card systems have gained universal prevalence in modern metros. In addition to its original goal of ticketing, the large amount of transaction data collected by the smart card system can be utilized for many operational and management purposes. This paper investigates an important problem: how to extract spatiotemporal segmentation information of trips inside a metro system. More specifically, for a given trip, we want to answer several key questions: How long does it take for a passenger to walk from the station gantry to the station platform? How much time does he/she wait for the next train? How long does he/she spend on the train? How long does it take to transfer from one line to another? This segmentation information is important for many application scenarios such as travel time prediction, travel planning, and transportation scheduling. However, in reality, we only assume that only each trip's tap-in and tap-out time can be directly obtained; all other temporal endpoints of segments are unknown. This makes the research very challenging. To the best of our knowledge, we are the first to give a practical solution to this important problem. By analyzing the tap-in/tap-out event pattern, our intuition is to pinpoint some special passengers whose transaction data can be very helpful for segmentation. A novel methodology is proposed to extract spatiotemporal segmentation information: first, for nontransfer trips, by deriving the boarding time between the gantry and the platform, and then, for with-transfer trips, by deriving the transfer time. Evaluation studies are based on large-scale real-system data of the Shenzhen metro system, which is one of the largest metro systems in China and serves millions of passengers daily. Onsite investigations validate that our algorithm is accurate and that the average estimation error is only around 15%.**

***Index Terms*—Intelligent transportation systems, metro systems, smart card, smart city, trip segmentation.**

I. Introduction

NOWADAYS, metro systems [1] have become one of the most preferred public transit services [2]. Compared with other services, metro has the benefits of high efficiency, large volume, and fast speed. Recently, contactless smart card systems have gained universal prevalence in modern metros. Compared with traditional magnetic cards or paper tickets, the smart card is a more secure and convenient method for authentication and fare collection.

The transaction data collected by the smart card system can be utilized for many operational and management purposes. For metro systems, both the tap-in and tap-out records for each trip are usually saved in the database. Each record contains at least the time, the station, and the card's ID of the transaction. With these data, we can measure user travel behaviors and their possible variances for the purpose of customer management [3]–[5]. We can also analyze the access data to improve transit planning or scheduling [6]–[8].

This paper investigates an important emerging problem: Given the smart card tap-in and tap-out data, how do we extract spatiotemporal segmentation information of metro trips? Shown in Fig. 1(a) is an example of a metro user, e.g., Alice's nontransfer trip in Line 1 (the dotted line): After tap-in at the metro gantry, Alice takes L_1 s to walk to the tap-in platform; before the departure of the next available train, Alice waits L_2 s; she travels on the train for L_3 s (including the dwell time at the intermediate stations) and arrives in the tap-out platform; walking from the platform to the tap-out gantry takes L_4 s; she waits at the tap-out gantry for L_5 s, due to passenger congestion, before she finally leaves the system. Fig. 1(b) is Alice's one-transfer trip from Line 1 to Line 2 (the wide line), and there are more segments: L_6 s for transferring to a Line-2 platform; L_7 s have passed before the next Line-2 train departs; Alice also spends L_8 s on the second train. Spatiotemporal segmentation of trips with multiple transfers can be modeled similarly. In this model, L_1, L_4, and L_6 are of particular importance. They are dominated by the building structures of each metro station, hence independent of moving passengers or trains. For convenience of presentation, we use the term *boarding time* to refer to both L_1 and L_4 and the term *transfer time* for L_6.

Manuscript received May 6, 2014; revised December 6, 2014; accepted February 27, 2015. Date of publication March 13, 2015; date of current version March 10, 2016. This work was supported in part by the China National Basic Research Program (973 Program) under Grant 2015CB352400; by the National Natural Science Foundation of China under Grant 61202107, Grant U1401258, and Grant 61202303; by the Natural Science Foundation under Grant CCF-1016966; by the National High Technology Research and Development Program of China (863 Program) under Grant 2014AA01A702; by the Natural Science Foundation of Hubei Province under Grant 2014CFB1007; by the National Key Technology Research and Development Program of China under Grant 2012BAH46F03; and by the Fundamental Research Funds for the Central Universities. The review of this paper was coordinated by Dr. P. Lin. *(Corresponding author: C. Tian.)*

F. Zhang and J. Zhao are with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, and also with Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen, China.

C. Tian is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: alexandretian@gmail.com).

C. Xu is with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, and also with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202 USA.

X. Liu is with the School of Computer Science, McGill University, Montreal, QC H3A 0E9, Canada.

L. Rao is with General Motors Research Laboratories, Warren, MI 48090-9055 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2015.2409815

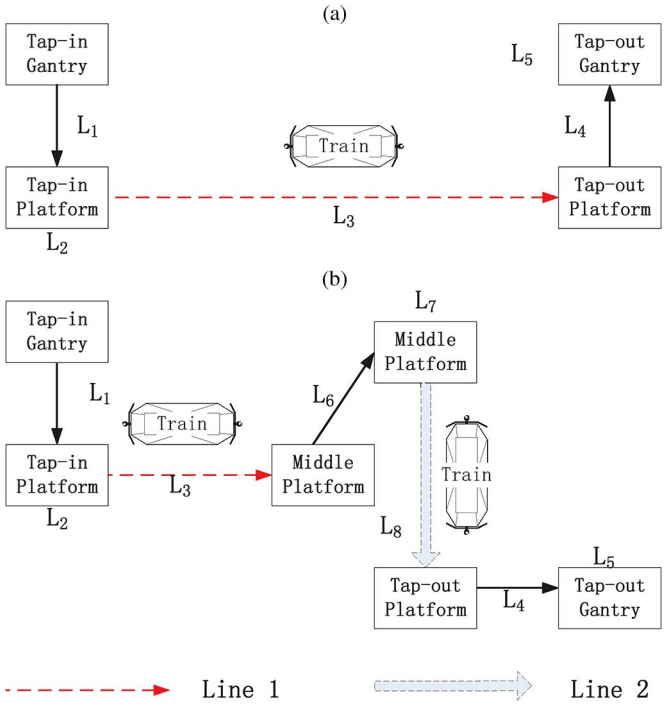

Fig. 1. Spatiotemporal segmentation of Alice's (a) nontransfer trip and (b) one-transfer trip.

More specifically, for a given trip, we want to segment it to several travel segmentations, with both location and time information. This spatiotemporal segmentation information is important in many application scenarios. With successfully deduced L_1, L_4, and L_6 values of every station, we have developed several exciting motivation cases (see the details in Section II).

For most modern metro systems, only each trip's tap-in and tap-out time can be directly obtained, and all other temporal endpoints of the segments are unknown. By subtracting the tap-in time from the tap-out time, the whole trip duration is composed of $L_1 + L_2 + L_3 + L_4 + L_5$ for a nontransfer trip or $L_1 + L_2 + L_3 + L_6 + L_7 + L_8 + L_4 + L_5$ for a one-transfer trip or even more segments for a multitransfer trip. Given only the start and end time, the problem is how to deduce the boundaries between two consecutive segments. To solve this problem, we provide an efficient yet effective spatiotemporal segmentation solution.

The intuition of our solution is as follows: There are some special passengers (termed as Border-Walkers in the paper) in the system, and we utilize their transaction data for segmentation. To the best of our knowledge, we are the first to give a practical solution to this important problem. The contributions of this paper include the following.

- We define the role and illustrate the special functions of Border-Walkers; a set of novel algorithms is proposed to identify Border-Walkers by analyzing the tap-in/tap-out event pattern (see Section IV).
- We further propose a novel methodology to extract spatiotemporal segmentation information, first for nontransfer trips by deriving the boarding time between the gantry and the platform and then for with-transfer trips by deriving the transfer time (see Section V).
- We study our approach using large-scale data collected from the Shenzhen metro system, which is one of the largest metro systems in China; it serves millions of passengers daily. We also present the detailed system design, which realizes our algorithm (see Section VI).
- We perform a large-scale onsite investigation (i.e., we manually take measurements and acquire the trip segmentation information). The measured results validate that our algorithm is accurate and that the average estimation error is only around 15% (see Section VII).

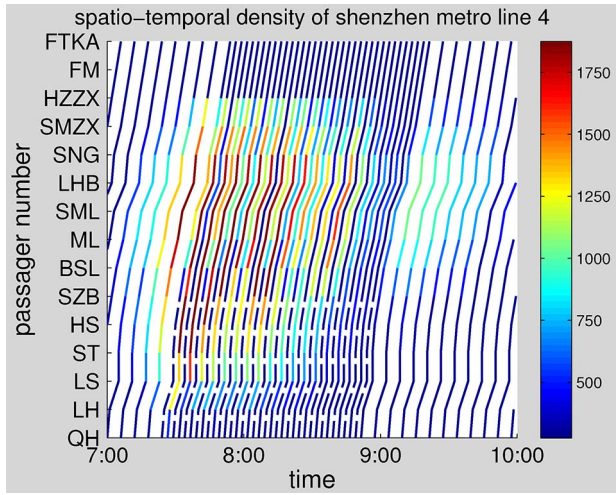

Fig. 2. Spatiotemporal passenger density of Line-4 trains from 7:00 A.M. to 10:00 A.M.

Section III presents the overview of our solution. Sections IV–VII discuss in detail each challenge and solution. Section VIII discusses related work. Section IX concludes this paper.

II. Motivating Applications

Here, we present several exciting use cases, as the motivation example for this paper. They are all our ongoing projects, with the objective of transforming our designed algorithms to real applications that benefit the public transportation.

A. Real-Time Train Density Estimation

Many passengers care more about comfort than travel duration [9], [10]. In a metro system, if the passengers waiting on the platform can be informed of the real-time crowding information of the following (several) trains (e.g., via the in-site LED display), some might change their travel plans by getting on an earlier or later and more comfortable train with fewer people aboard.

Currently, this service is just unavailable. With only the tap-in and tap-out information, the authority has no method yet to figure out the approximated population in each running train. Shown in Fig. 2 is the passenger spatiotemporal density of Line 4 from a typical day: between morning peak hours (i.e., 7:00 A.M. to 10:00 A.M.) and between stations BSC and SNG; the deeper the red color, the more people on a train.

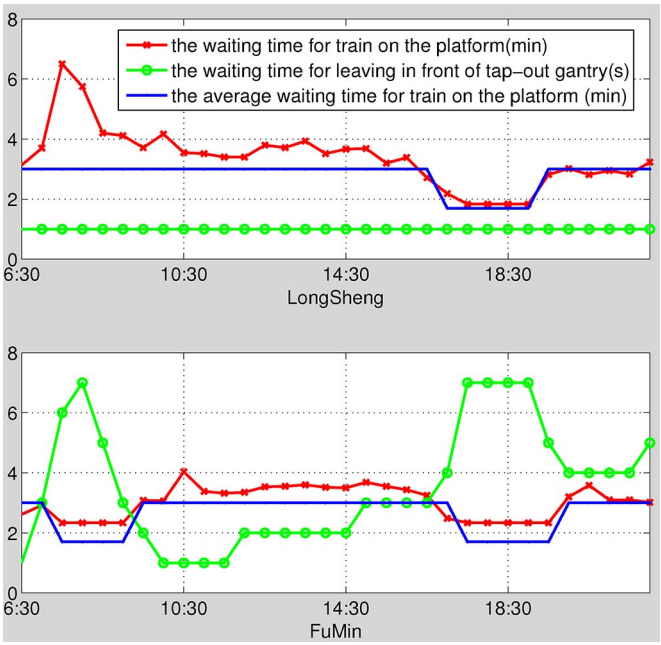

Fig. 3. Waiting time in the tap-in platform and before the tap-out gantry.

The densities are unevenly distributed: We can clearly find that red and blue density interleave among consecutive trains. After investigation, we find that interzone trains are added for the morning peak between the SZB and FTKA stations; they interleave with the normal scheduled trains and are much less populated since they skip the first several stations. However, passengers on a platform are uncertain about the real-time crowdness of the next coming train and the next next coming train. Mostly, people just choose to board on the next train: To them, without information, the next train might be even more crowded; if this is the case, then waiting is just a waste of time; as a result, most people simply take the next available train.

By travel segmentation, we can already accurately estimate the historic spatiotemporal density of every operating train from the records (as shown in Fig. 2). More importantly, combined with passengers' origin–destination (OD pair) prediction [11], [12], we can predict the destination of each passenger when he/she gets into the system; with that information and the historical segmentation results, we then can perform real-time passenger density estimation for each running train. We leave the details of this ongoing project to future works.

B. Personalized Travel Planning

The exact spatiotemporal status of a passenger actually also depends on other factors: the physical condition of a passenger (e.g., disabled or not), peak or low traffic hours, etc. With the help of the algorithms in this paper, we can perform *Personalized Travel Planning*, instead of the general travel planning service currently provided (e.g., Bing Map and Google Map).

Our algorithms can extract information whose value is generally stable: for example, the average L_1 (walk-in), L_4 (walk-out), and L_6 (transfer) values of every station for normal people under nonpeak hours. We can then analyze the variability of every segmentation in the time domain. In Fig. 3, based on our

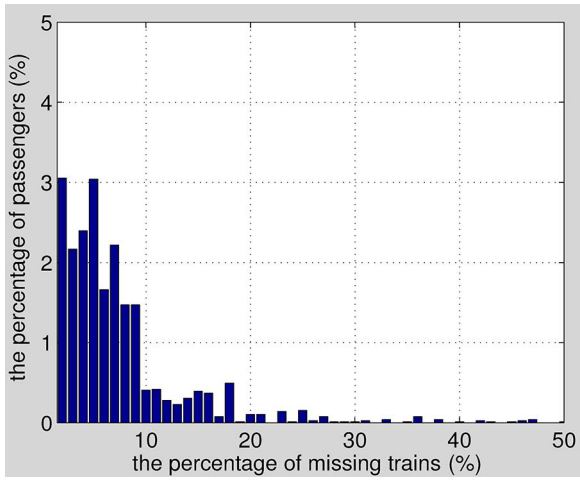

Fig. 4. Individual factors.

algorithms, we demonstrate the extra waiting time in the tap-in platform and before the tap-out gantry for a typical day in two stations. The average expected waiting time (blue line) in the tap-in platform is calculated as half the interval between consecutive trains. Some parts of the average line are lower (usually at peak hours), since extra trains are added; hence, the intervals are reduced.

In LongSheng station, the derived waiting time in the tap-in platform (red line) is always higher than expected, since this station is crowded, and many passengers have to wait for several trains before getting on. Only in the evening peak does the waiting line matches the average line, due to the effects of added trains. As a comparison, the derived waiting time of the FuMin station is almost the same as the expected time: This station has extra trains for both morning and evening peaks. As a comparison, tap-out waiting in the LongSheng station (green line) seldom exists for almost the whole day because passengers getting off at this station are few, whereas for the Fumin station, there is extra waiting time before the tap-out gantry for almost the whole day, particularly for morning and evening peaks.

We also analyzed the difference among individual passengers, by segmenting each individual's travels in nonpeak hours. For each person, we sum up the numbers of trips he/she misses a train, which a passenger at normal speed should get in. Fig. 4 shows that 80% of passengers seldom miss any train. However, nearly 1% of passengers are clearly slower in action compared with the average: some even with a missing probability as high as 50%.

With all these factors extracted, we are developing a project that could provide Personalized Travel Planning based on both hour consideration and historical individual record.

C. Metro Acquaintance

A social phenomenon is called "The Familiar Stranger." For example, Alice and Bob have different OD pairs; they live in different districts and work in different districts. However, they happen to transfer at the same station at approximately the same time every workday; they are familiar with each other, and they have feelings about each other, while none of them dare to make the first contact [13].

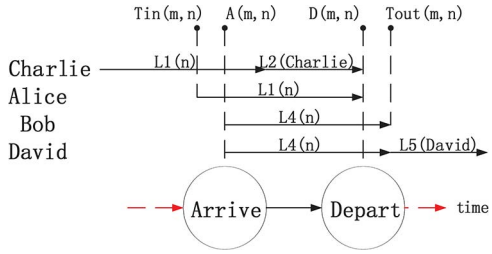

Fig. 5. System model from a train's view.

"Metro Acquaintance" is an ongoing project with the objective of establishing such a special social network. Directly from historical records, by travel segmentation, we calculate the meet probability, when and where, that each pair of passengers have met each other, either in the same platform, or in the same train. By entering their card ID in the web entry, people can establish a metro social network; the connection recommendation of the social system is mostly based on the meet probability deduced by the travel segmentation algorithms in this paper.

III. Overview

A. Modeling

We model the system from a train's point of view. As shown in Fig. 5, let us suppose n is a station of a metro line; we also suppose that a specific train m travels in one direction and stops by n; let us denote the time of its arrival and departure as $A(m,n)$ and $D(m,n)$, respectively. Note here that we abuse the term train m to represent, instead of a physical entity, a logical entity as the combination of three attributes: line, direction, and train sequence number.

We assume that the walking speed is constant for all passengers: In this case, L_1 and L_4 are station dependent, which we can denote as $L_1(n)$ and $L_4(n)$, respectively. As a comparison, L_2 and L_5 are passenger dependent; we denote them as $L_2(passenger)$ and $L_5(passenger)$, respectively.

We illustrate the trajectories of four fictional passengers in Fig. 5 as follows.

- Charlie enters station n, walks $L_1(n)$ s, boards train m, and waits for $L_2(Charlie)$ s before train m departs.
- Alice enters station n later, also walks $L_1(n)$ s but boards train m exactly at the departure time $D(m,n)$.
- Bob alights from train m exactly at time $A(m,n)$ when train m stops at station n, walks $L_4(n)$ s, and goes through the station gantry immediately before all other travelers.
- David alights the train, also walks $L_4(n)$ s but waits for $L_5(David)$ s in the tap-out list before he leaves the station.

B. Intuition and Challenges of Solution

Charlie and David represent normal passengers: They spend extra time in the system waiting, both at the platform, and before the tap-out gantry. As a comparison, Alice does not waste extra time waiting at the platform, i.e., $L_2(Alice) = 0$; Bob do not waste extra time waiting before the tap-out gantry, i.e., $L_5(Bob) = 0$. Alice and Bob are the special passengers we termed Border-Walkers in the paper; their transaction data can be very helpful for segmentation.

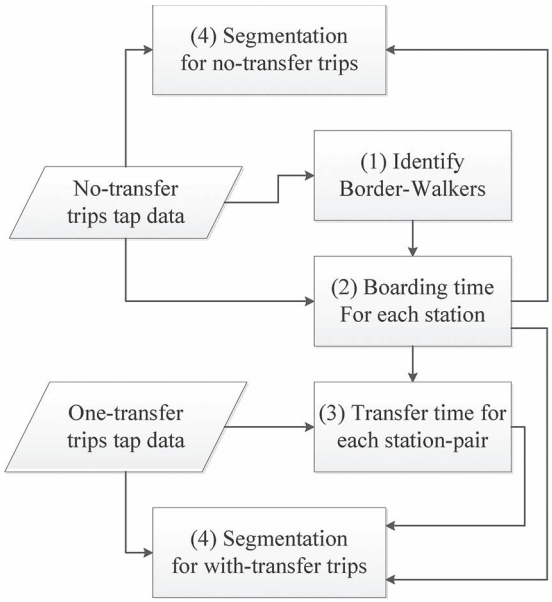

Fig. 6. Solution scheme.

Suppose for each (m,n) pair, there are two transaction records in the smart card data: Alice's tap-in time $T_{in}(m,n)$ and Bob's tap-out time $T_{out}(m,n)$. As shown in Fig. 5, we can derive (1), which relates Border-Walkers data with train timing $A(m,n)$ and $D(m,n)$, for each (m,n) pair. Thus

$$\begin{aligned} &(i)\ T_{out}(m,n) = A(m,n) + L_4(n) \\ &(ii)\ T_{in}(m,n) = D(m,n) - L_1(n). \end{aligned} \tag{1}$$

We already have $T_{in}(m,n)$ and $T_{out}(m,n)$; if $A(m,n)$ and $D(m,n)$ can be fixed, then we can derive $L_1(n)$ and $L_4(n)$ for station n. With all the values of $A(m,n)$, $D(m,n)$, $L_1(n)$, and $L_4(n)$, it seems straightforward, at least for nontransfer trips, to perform spatiotemporal segmentation information.

However, there are challenges to be overcome: first, how to find these Border-Walkers (i.e., Bob and Alice) from all the transaction records. Second, even with their data, the segment calculation cannot directly rely on published train arrival/departure schedule. In many cases, only the schedules of the first/last trains of each line are available to the public. Moreover, there is no guarantee that each train would be punctual at a second level, e.g., an unexpected contact between a train gate and a passenger's body can delay the scheduled departure time for 10 s. In other words, $A(m,n)$ and $D(m,n)$ cannot be directly fixed from public source; instead, they should be derived as well.

The solution architecture is shown in Fig. 6. The deduction first focuses on the nontransfer tap-in/tap-out data: From all the records, we find the Border-Walkers for each (m,n) combination (step 1). With this information, we then derive the boarding time for each station (step 2). Now, we incorporate tap-in/tap-out data with one-transfer: The transfer time for each station pair can be derived (step 3). Eventually, we can segment the trips in our data set (step 4).

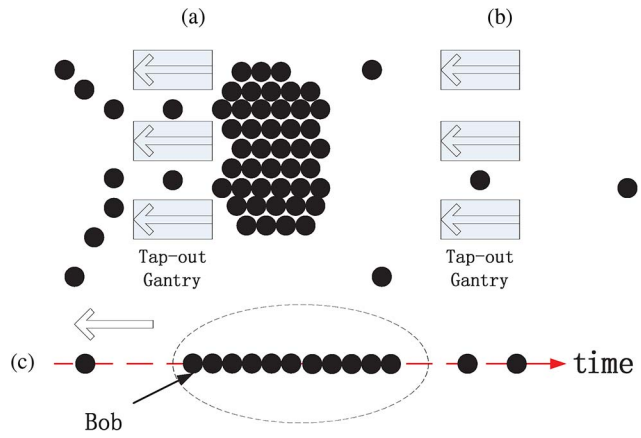

Fig. 7. Tap-out events. (a) Busy time. (b) Idle time. (c) Temporal pattern.

IV. Pinpoint Border-Walkers

A. Finding Bob

Bob can be found by mining the tap-out events. The key insight here is that tap-out events usually form a periodical idle–busy–idle pattern. As shown in Fig. 7(a), when a train m arrives, there can exist many passengers that leave the train together; they walk the same distance to the gantry and tend to congest right in front of it; the resulting tap-out events are relatively very frequent in the time domain. As a comparison, the tap-out events are sparsely distributed between two such busy times, as shown in Fig. 7(b).

Theoretically, it is straightforward to find Bob. All tap-out events in the station are mapped to the time axis, as shown in Fig. 7(c); the classic DBSCAN algorithm is adopted to cluster the points [14]; the dense clusters, which are almost evenly spaced in the time domain (due to the evenly-spaced train schedule), can be identified. Each dense cluster corresponds to the arrival of a train, and we consider in each cluster the passenger with the earliest tap-out time as the special Bob for which we are looking.

B. Finding Alice

Alice can be found after Bob. After identifying Bob, all passengers inside this dense cluster are correlated with a specific train. Suppose there is a train m that travels in one direction and stops by station n. For all the subsequent stations $n+1, n+2, \ldots, N$, tap-out passengers that belong to train m can be identified. Among them, the subset of passengers tapping in at station n are grouped together. We consider that with the latest tap-in record as the special Alice for which we are looking.

C. Handling Exceptions

In practice, there are more challenges. We present in Fig. 8 the tap-out event data of a metropolitan metro system in China. The graph shows a partial set of stations of one line, with four time periods in the same day; the time duration are all 15 min. Fig. 8(a), from 08:15 to 08:30, shows a part of the morning peak; Fig. 8(b), from 16:15 to 16:30, is the normal state of the system; Fig. 8(c), from 18:30 to 18:45, is the evening peak; Fig. 8(d), from 22:15 to 22:30, shows the low hours of a day.

Overall, the periodical idle–busy–idle pattern is quite clear in most cases. There are more implications that can be found. First, the interval between consecutive trains is time dependent: Compared with Fig. 8(b) and (d), Fig. 8(a) and (c) have more clusters in 15 min, which implies shorter train intervals (this is reasonable for morning/evening peaks).

Second, there are some types of exceptions that hinder the clustering step as follows.

- (*Type 1*) There are too many events in some cases. The train arrivals can be extremely frequent in the morning peak, such as at station *XiangMiHu* shown in Fig. 8(a). It is clear that the clustered long line of events contains passengers from several consecutive trains.
- (*Type 2*) There are too few events in some cases. Station *DaXin* demonstrates regular clusters in Fig. 8(a), whereas it is almost impossible to identify any reasonable cluster in Fig. 8(b)–(d).
- (*Type 3*) There are passengers moving in abnormal patterns. A passenger Ethan in haste can run from the gantry to the platform; as a result, he might get on train m, instead of on $m+1$ if he moves at a normal speed.

Let us denote Bob's tap-out time of train m at station n as $T_{out}(m,n)$. A whole day's $T_{out}(m,n)$ values thus form an $M \times N$ matrix (see Fig. 9). *Type 1* and *Type 2* exceptions cause missing entries in the matrix.

The matrix has hidden structures: For example, $T_{out}(m,n) - T_{out}(m,n-1)$ and $T_{out}(m-1,n) - T_{out}(m-1,n-1)$ are correlated since they are both dominated by the travel time from station $n-1$ to n; moreover, $T_{out}(m+1,n) - T_{out}(m,n)$ and $T_{out}(m+1,n+1) - T_{out}(m,n+1)$ are correlated since they are both dominated by the interval between train m and $m+1$. Leveraging the presence of these certain types of structures and redundancy in collected data, compressive sensing [15] can be used to interpolate the matrix to handle *Type 1* and *Type 2* exceptions.

Type 3 exceptions are handled by anomaly detection [16]. Sticking to the Ethan example, we can analyze the relationship between the tap-in time and the belonging train of passengers; Ethan can be identified as an anomaly, instead of being treated as Alice.

V. Derive Boarding and Transfer Time

Using these special passengers' data, we propose a novel methodology to extract spatiotemporal segmentation information, both for nontransfer trips by deriving the boarding time between the gantry and the platform and for with-transfer trips by deriving the transfer time.

A. Segment Nontransfer Trips

The challenge is to derive the boarding time $L_1(n)$ and $L_4(n)$ for each station n. From (1), we could get

$$T_{out}(m,n) - T_{in}(m,n) = L_1(n) + L_4(n) - (D(m,n) - A(m,n)). \quad (2)$$

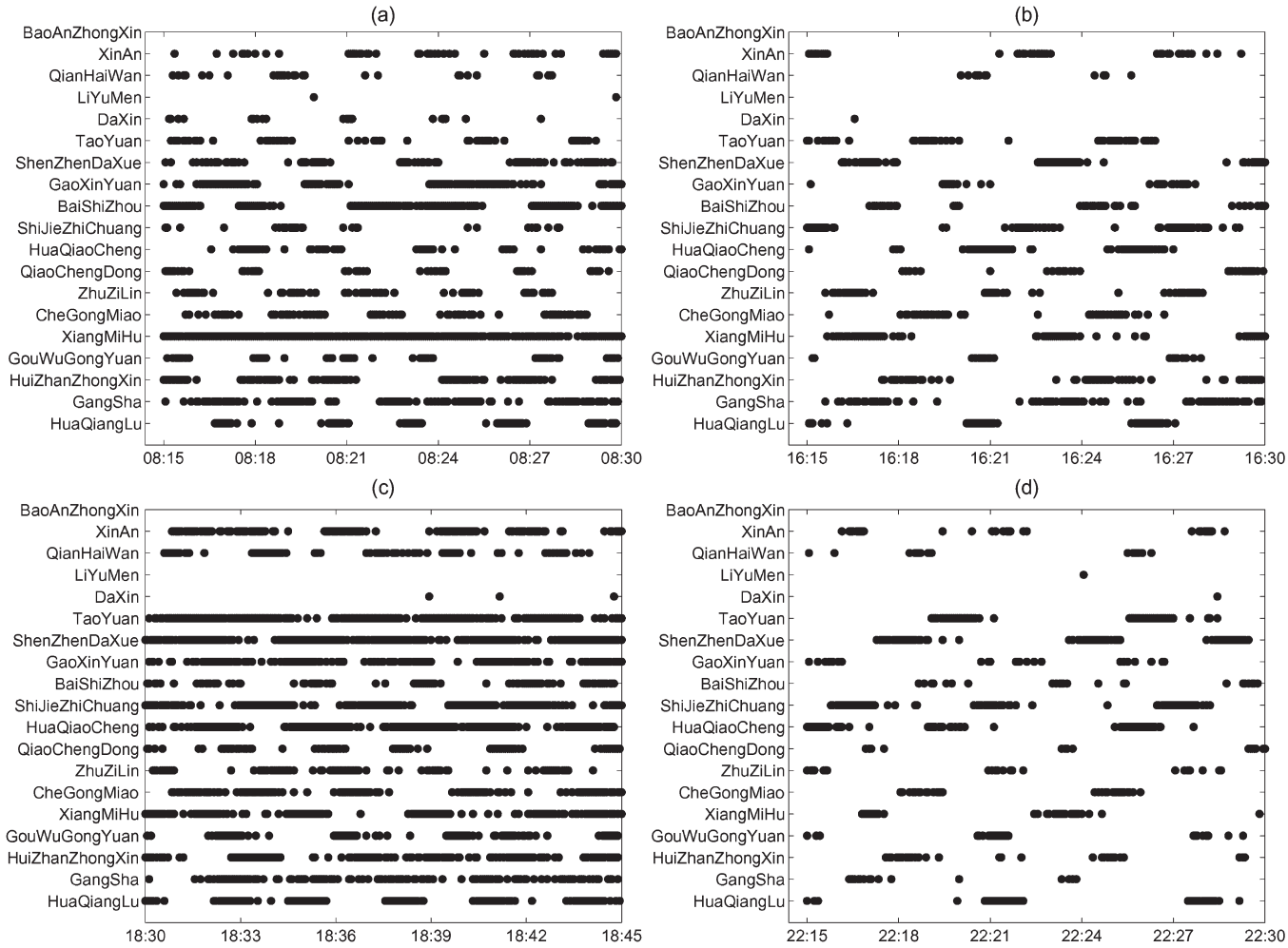

Fig. 8. Tap-out event pattern. (a) Morning peak. (b) Normal. (c) Evening peak. (d) Low hours.

	1	...	n−1	n	n+1	...	N
1		...				...	
⋮	...	...	...	...	...	...	...
m−1	...	...	$T_{out}(m-1, n-1)$	$T_{out}(m-1, n)$	$T_{out}(m-1, n+1)$	...	...
m	...	...	$T_{out}(m, n-1)$	$T_{out}(m, n)$	$T_{out}(m, n+1)$	...	...
m+1	...	...	$T_{out}(m+1, n-1)$	$T_{out}(m+1, n)$	$T_{out}(m+1, n+1)$	...	...
⋮	...	...	...	...	...	...	...
M		...				...	

Fig. 9. Tap-out value matrix.

Two characteristics can be exploited to solve (2). First, for almost all metro lines in one direction, the passengers' route to the platform and the route from the platform in the same station are the same. It is safe to assume that $L_1(n)$ is equal to $L_4(n)$; hence, we approximate both $L_1(n)$ and $L_4(n)$ to be $L_p(n)$.

Second, to reduce operational complexity, the dwell time (i.e., $D(m,n) - A(m,n)$) is generally fixed for each station; for the same reason, the value options are also limited. Compared with $L_p(n)$, dwell time values are relatively smaller. Our interactions with transit agencies reveal that the typical dwell time is around tens of seconds for small stations and is estimated at a larger value for large stations. It is safe to approximate dwell time by a fixed value $L_w(n)$. In our system (see Section VI), we use onsite investigations to estimate $L_w(n)$.

Now, we have

$$L_p(n) = \left(T_{out}(m,n) - T_{in}(m,n) + L_w(n)\right)/2. \quad (3)$$

If $T_{out}(m,n)$, $T_{in}(m,n)$, and $L_w(n)$ can be fixed, then for each line station, the segment length $L_p(n)$ between the gantry and the platform can be derived. The timing for each train can be calculated by

$$\begin{aligned} A(m,n) &= T_{out}(m,n) - L_p(n) \\ D(m,n) &= T_{in}(m,n) + L_p(n). \end{aligned} \quad (4)$$

Now, we have the estimated time values for each $A(m,n)$/$D(m,n)$ pair.

For each nontransfer trip, we can have spatiotemporal segmentation. Assume that there is a trip from station n_1 to n_2; the passenger performs tap-in at $t(n_1)$ and tap-out at $t(n_2)$. We can get his boarding time L_1/L_4 as $L_p(n_1)/L_p(n_2)$. Based on the tap-out event's belonging cluster [see Fig. 7(c)], the passenger's train m can be identified. We can then get how long the passenger waits on the tap-in platform L_2 as $D(m,n_1) - t(n_1) - L_1$. The travel time L_3 in train m is $A(m,n_2) - D(m,n_1)$; the waiting time L_5 can be derived as $t(n_2) - A(m,n_2) - L_4$.

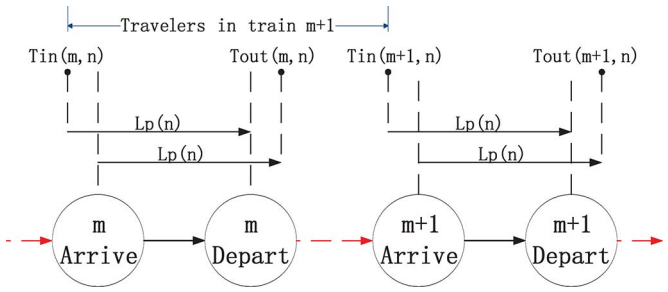

Fig. 10. System model from a station's view.

B. Segment With-Transfer Trips

The challenge is to derive the transfer time L_6 for each possible transfer option. To reduce interference, we only use one-transfer trips for derivation. Let us suppose that the tap-in/transfer/tap-out stations are n_1, n_2, and n_3, respectively, and we suppose that there are two trains, i.e., m_1 on the first line k_1 and m_2 on the second line k_2. A passenger gets on train m_1, then arrives at station n_2 at time $A(m_1, n_2)$, and leaves n_2 at time $D(m_2, n_2)$ on train m_2. Note that in the context of transfer trips, the term *station* is a logical entity as the combination of three attributes: line, direction, and the physical station.

For each transfer trip, we can easily figure out the tap-out train m_2 by the same approach previously mentioned. However, how do we figure out the passenger's tap-in train m_1? Let us first model the system from a station's point of view. Fig. 10 shows time advances from left to right. Let us also suppose that n is a station; a specific train m arrives and departs at $A(m, n)$ and $D(m, n)$, respectively; the next train $m + 1$ arrives and departs at $A(m + 1, n)$ and $D(m + 1, n)$, respectively. In our approach, passengers with tap-in time between $T_{in}(m, n)$ and $T_{in}(m + 1, n)$ are assumed to enter train $m + 1$, i.e., m_1 can be fixed.

The method is to group all passengers transferred from line k_1 to line k_2 at station $n2$: For each such trip, we can get its $A(m_1, n_2)$ and $D(m_2, n_2)$ values. All values of $D(m_2, n_2) - A(m_1, n_2)$ are calculated. It is assumed that the trip with the least $D(m_2, n_2) - A(m_1, n_2)$ value has no waiting time (i.e., $L_7 \approx 0$). We take this as the transfer time $L_6(n_1, n_2)$ for each n_1, n_2 pair. Based on this, we can perform segmentation for each transfer trip, similar to that of the nontransfer trips.

C. Postprocessing

There are many trains running one after another on each line, and a station is visited by 100–200 trains daily. Generally speaking, with each train's transaction data, we can obtain a value of the boarding time $L_p(n)$ for each station. As a result, for each $L_p(n)$, we actually obtain a group of values. The problem is how to converge this set of values to a single $L_p(n)$ result. The same is the situation of transfer time calculation.

For a single $L_p(n)$, most obtained values are close enough. We did find abnormal values, which can be several hours. The classic DBSCAN algorithm is again adopted to cluster the points [14]. After excluding the exceptions, the minimum value in the cluster is chosen as the optimal candidate for boarding time or transfer time.

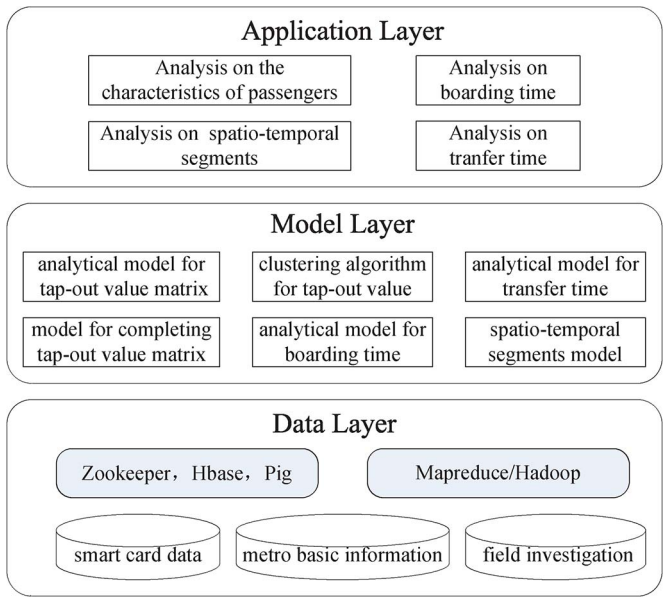

Fig. 11. System architecture.

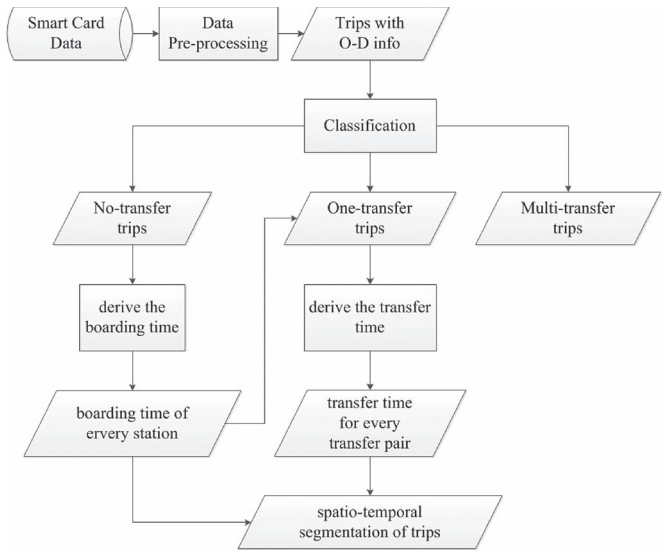

Fig. 12. Processing flowchart.

VI. Prototype and Data Set

A. Processing System

Our spatiotemporal segmentation algorithm processes a large amount of data and, correspondingly, requires intensive sorting and grouping operations. Fig. 11 shows that the implemented system has three layers: the data layer, the model layer, and the application layer.

- The data layer is used mainly for storage purposes and MapReduce [17]/Hadoop [18] job processing. There are three kinds of data: smart card transactions, metro basic information, and intermediate and final results. The storage part uses HDFS [19] and Hbase [20], which is a distributed, scalable, and big data store. The data layer accepts MapReduce jobs from the model layer and effectively accomplishes them by data processing. Some

Fig. 13. Metro graph of Shenzhen.

software tools, such as PIG [21] and HIVE [22], are also used. The results are sent back to the model layer or stored to HDFS.

- The model layer is used to accept the query requests from the application layer; a query is translated to a series of MapReduce jobs. It uses the smart card and other data from the data layer to build the analytical model: clustering model for tap-out events, analytical model for tap-out value matrix, analytical model for boarding time, etc.
- The application layer performs model analysis, such as analysis on the characteristics of passengers, boarding time, spatiotemporal segments, transfer time, etc.

B. Flowchart

The processing flowchart is shown in Fig. 12. The details of data preprocessing and classification steps are given below.

TABLE I
TRANSACTION RECORD FORMAT

Field	Value
CardID	Anonymous unique card id
TrmnlID	For metro, it represents the stop id
TrnsctTime	Transaction date and time
TrnsctType	Transaction type

Data Preprocessing: Every trip contains one tap-in event and one tap-out event; this step joins them together out of the transaction data by matching card ID and time. Moreover, the redundancy should be removed, and the inconsistency should be solved in smart card data.

Classification: This step is to divide the trips from step 1 into three categories: nontransfer trips, one-transfer trips, and multitransfer trips. As previously mentioned, only nontransfer and one-transfer trips are used for boarding/transfer time extraction.

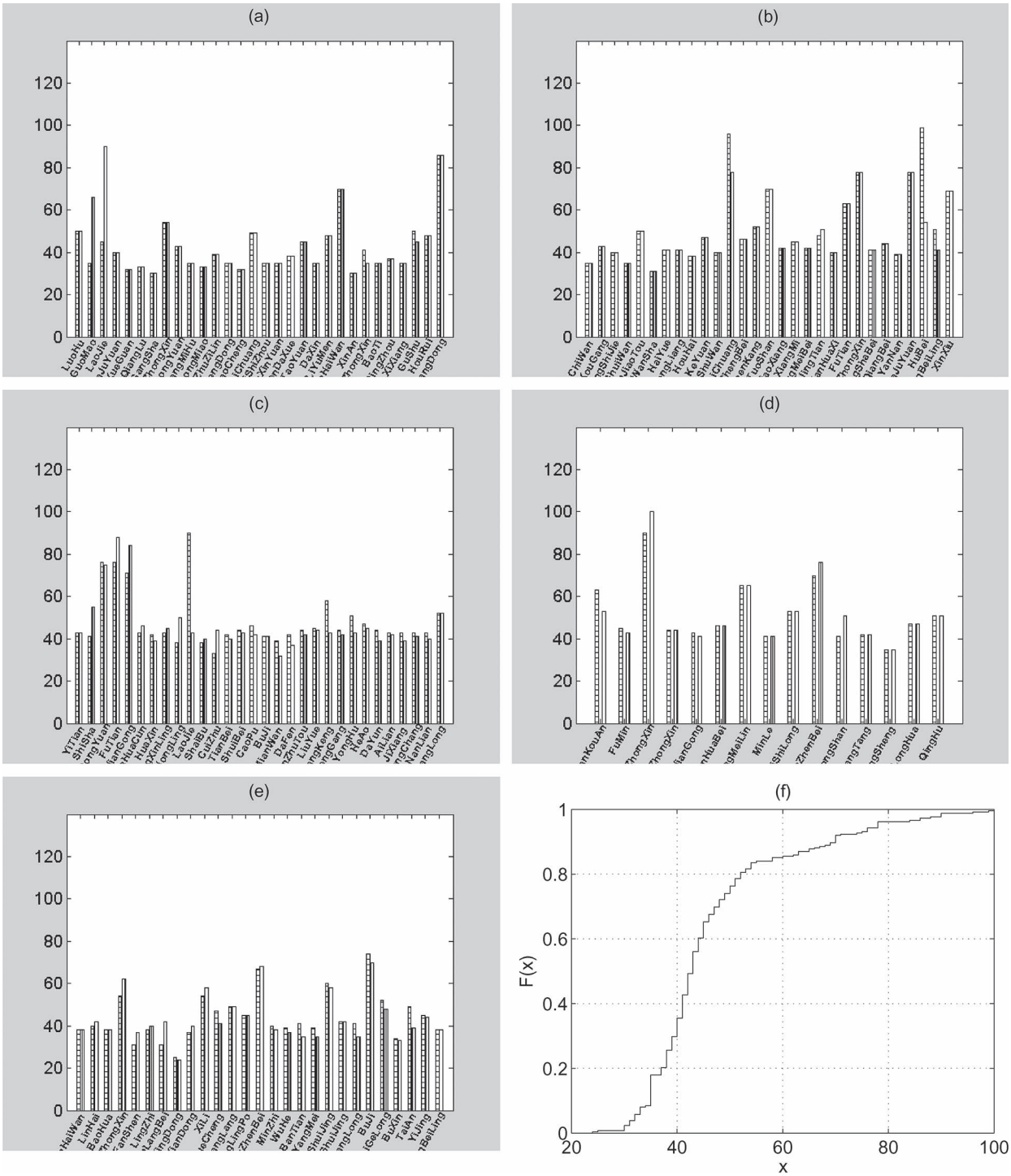

Fig. 14. Boarding time. (a) Line 1—Green. (b) Line 2—Orange. (c) Line 3—Blue. (d) Line 4—Red. (e) Line 5—Purple. (f) Distribution.

C. Data Set

Our data set is the metro transaction data from Shenzhen, China. There have been over ten million public transit smart cards issued, and these smart cards can be used for both the bus and metro systems. Metro alone has, on average, 2.5 million transactions per day, which is estimated to be one third of the total public transit load. Fig. 13 shows the existing five lines by the end of 2013; there are eight more lines coming over the next seven years.

The data set contains two months' metro transaction records from September 1 to October 30, 2013. Each record represents a single card-swipe event, either tap-in or tap-out. The four import fields are listed in Table I. The data are about 500 MB per day or 15 GB per month.

We perform a dwell time onsite investigation for all five lines and get the resulting $L_w(n)$ for our algorithm. The result is that for small stations, the dwell time is around 20 s, whereas for large stations, it is around 35 s.

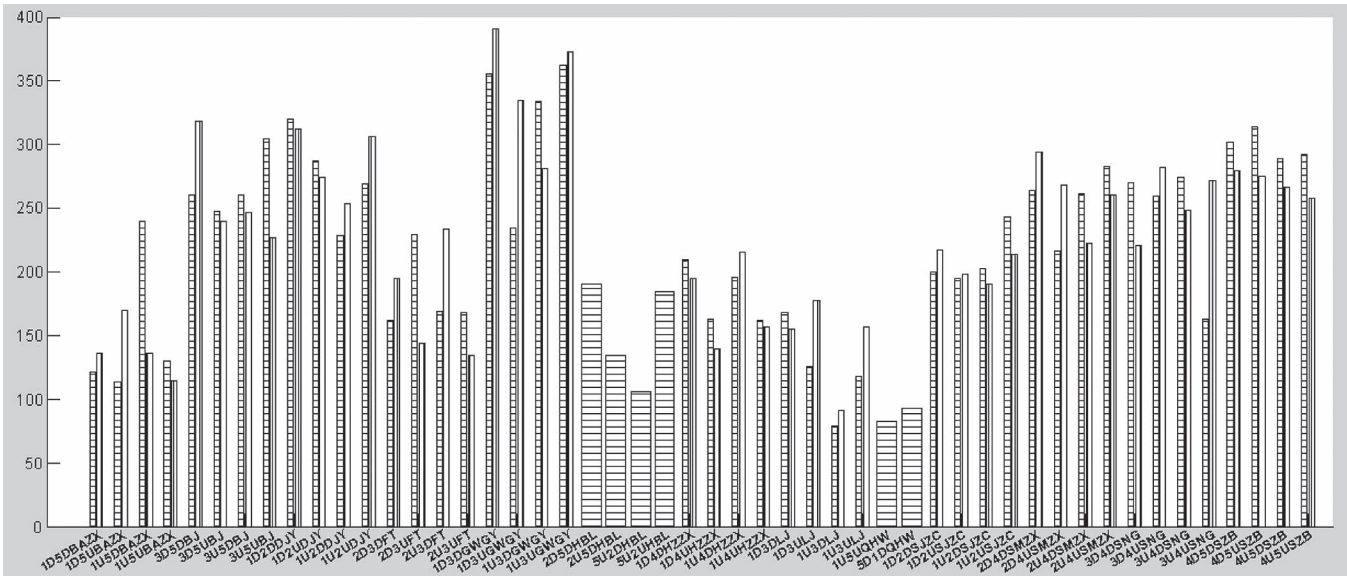

Fig. 15. Transfer time at each station. We have to abbreviate the information on the x-axis for better presentation. For example, **1D5UBAZX** (second item from left to right in the x-axis) denotes the following: from Line **1 D**ownward direction, transfers to Line **5 U**pward direction, in **B**ao**A**n**Z**hong**X**in station.

VII. Empirical Evaluation and Validation Results

Here, we present the empirical results by applying our approach to Shenzhen metro system analysis. First, we analyze the obtained boarding/transfer time. A large-scale onsite investigation is performed, which validates our design. Finally, we segment the trips to show the effectiveness of our approach in analyzing passenger travel patterns.

A. Boarding Time

For each station, the boarding time of two opposite directions is derived separately. The obtained boarding time of five lines is shown in Fig. 14(a)–(e). There are two bars in each station: Each represents one direction.

First, let us observe the boarding time of Line 1. The boarding time pattern of Line 1 is typical, compared with other four lines [see Fig. 14(b)–(e)].

A first observation is that the time of both directions in the same station is mostly comparable. The reason is that most stations have their trains, of the same line, stop at the opposite sides of the same platform. There are also exceptions, i.e., LaoJie and GuoMao. We perform onsite investigation in the two stations. The reason is that their platforms of opposite directions are located in different layers of the same building, due to architectural constraints. The identification of these exceptional cases indirectly verify the accuracy of our approach.

Second, for most stations, the boarding time is less than 60 s. Only very few stations have large boarding time, as that of QianHaiWan and JiChangDong stations. These are large stations with transfer functions: The building design is larger, hence resulting in prolonged walking time.

Shown in Fig. 14(f) is the total cumulative distribution function (cdf). As we can see, for 70% stations, the boarding time is less than 60 s; for only 5%, the boarding time is larger than 100 s.

B. Transfer Time

Fig. 15 shows the transfer time of every possible choice. Again, there are two bars for each n_1, n_2 pair: Each represents one direction. There are several wide bars: In these stations, there is only one-way transfer from one line to another; usually, such a station is the end point of a line.

As we can observe, mostly, the transfer time is less than 90 s. However, the transfer time related to the ShenZhenBei station is extremely high, compared with others. The reason is that the metro station itself is actually fully embedded in the Chinese High Speed Rail station. The huge building makes the transfer between lines a really long journey.

C. Validation

To verify the correctness and reliability of the methodology given in this paper, we have performed a large-scale onsite investigation. Fig. 16 shows the results' comparison between the investigation and our approach: The x-axis is the value of boarding/transfer time; the y-axis is the total cumulative distribution function value. As shown in the figure, our approach can effectively and accurately derive the boarding/transfer time.

The numerical results are also given in Table II. Regarding the boarding time, our average estimation error is between 10% and 20% for each line; on average, the error is only around 15%, which validates that our algorithm is relatively accurate. With regard to the transfer time, the results are even better: The average estimation error is less than 13%.

D. Segmentation

Based on the obtained boarding and transfer time, we can extract the spatiotemporal segments of trips. The trip segmentation results are shown in Fig. 17. For each trip, the obtained $L_1/L_2/L_3/L_4/L_5$ values are transformed into the

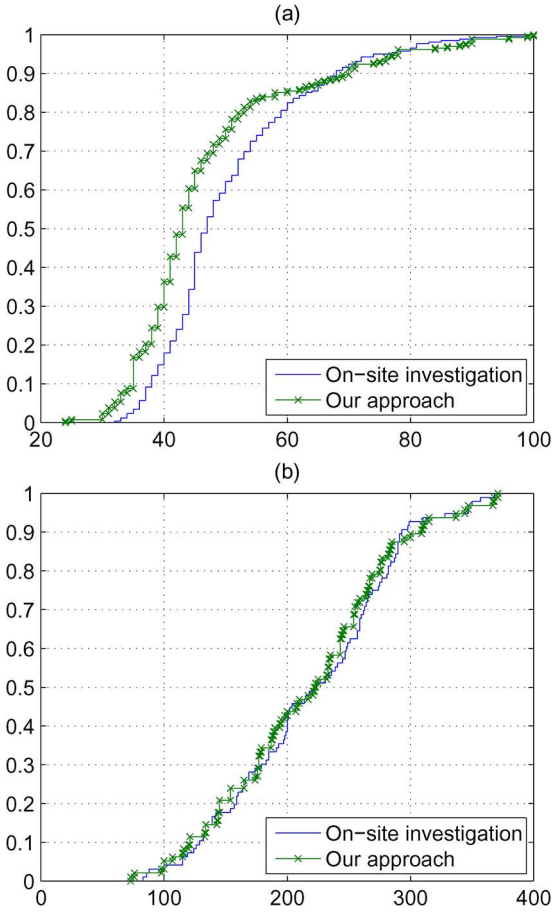

Fig. 16. Onsite validation. (a) Boarding time cdf. (b) Transfer time cdf.

TABLE II
ESTIMATION ERRORS

	Mean Error
Line 1 Boarding	13.95%
Line 2 Boarding	19.97%
Line 3 Boarding	10.1%
Line 4 Boarding	17.4%
Line 5 Boarding	18.5%
All Boarding	15.78%
All Transfer	12.74%

corresponding percentile format. For each line, the percentile values are averaged over all trips in our database.

For nontransfer trips, the onboard time L_3 dominates: It accounts for around 70%–80% of the total trip time. Notice that L_3 includes both the train travel time and the dwell time at intermediate stations. The boarding time is similar for all lines: L_1/L_4 normally takes around 3%–5% of the total trip. The waiting time at the tap-in platform L_2 is different among lines: Less than 12% is spent for Line-1 travelers, whereas for Line 4, this value is nearly 19%. The reason is that Line 1 is the most crowded line, and the interval between trains is smaller. As a comparison, the waiting time at the tap-out platform (i.e., L_5) is negligible: only 1%–2% for every line.

The one-transfer trips' segmentation is shown in Fig. 18. The trips with transfers are typically longer. Boarding time (L_1 and L_4) and the waiting time (L_2, L_7, and L_5) are similar in value

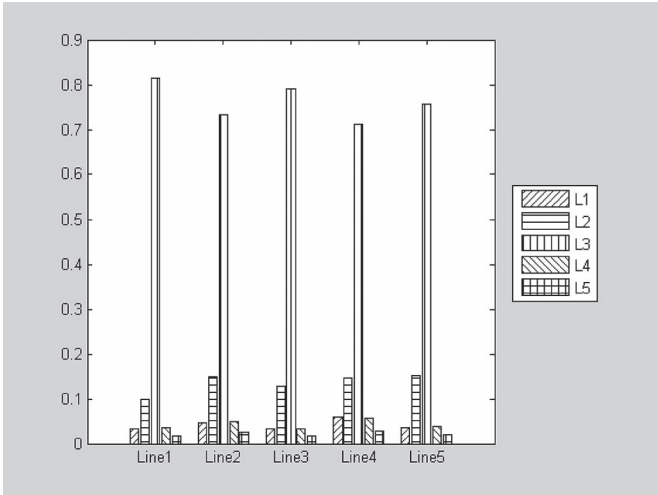

Fig. 17. Spatiotemporal segmentation of no-transfer trips.

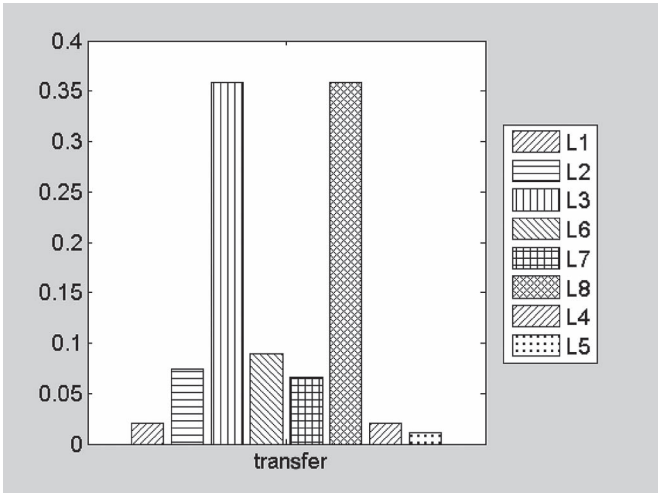

Fig. 18. Spatiotemporal segmentation of one-transfer trips.

compared with nontransfer trips; as a result, their percentiles decrease in the whole trip. The average onboard time $L_3 + L_6$ is over 72%. The average transfer time is 84 s, which is around 4% of the trip.

VIII. RELATED WORK

Most travel behavior analyses focus on the aggregated traffic patterns. The two peak access patterns are common during weekdays: People go to work in the morning and go home in the evening. As a comparison, transactions are more evenly distributed during weekends [3], [4]. There are some existing analyses that also used data from Shenzhen [5], [23], [24]. They focus on the aggregated temporal usage of the whole metro system. In [24], the spatial characteristics of the station usage model are studied. Our previous work [5] studies the aggregated temporal and spatial travel patterns of passengers by mining smart card data. Different from all previous works, this paper focuses on independent trips: We plan to segment each transaction trip into travel segments.

One of our previous works studies the passenger density for public bus [8]. However, unlike metro, the bus boarding and departing time can be easily identified by the card tap events. Sun *et al.* estimate the spatiotemporal density inside a metro system. However, their system model is oversimplified: They assume a uniform boarding time and dwell time for every station. Furthermore, their algorithm requires physical distance among stations as input and does not deal with the transfer time between lines [25]. As a comparison, in our paper, the boarding time is derived for every station separately, which is a more practical work. We require the estimation (instead of explicit value) of the dwell time of each station as input, which is much easier to get via field investigation. We also derive transfer time between lines.

There are other works dedicated to the analysis of transportation systems based on mining large-scale data, such as freight trunks [26], taxis [27], public buses [28], and even electrical vehicles [29].

IX. Conclusion

In this paper, we have investigated an important problem: how to extract spatiotemporal segmentation information of metro trips by only utilizing the tap-in and tap-out information. To the best of our knowledge, we are the first to provide a practical solution to this important problem. We first propose a set of novel algorithms to identify Border-Walkers by analyzing the tap-in/tap-out event pattern; based on that, we further propose a novel methodology to extract spatiotemporal segmentation information: first for nontransfer trips by deriving the boarding time between the gantry and the platform and then for with-transfer trips by deriving the transfer time. We study our approach in the case of Shenzhen metro system and perform a large-scale onsite investigation. The onsite measured results validate that our algorithm is accurate and that the average estimation error is only around 15%.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments.

References

[1] C. Mayet *et al.*, "Comparison of different models and simulation approaches for the energetic study of a subway," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 556–565, Feb. 2014.

[2] "Urban rail transit in China," Wikipedia, the free encyclopedia. [Online]. Available: http://en.wikipedia.org/wiki/Urban_rail_transit_in_China

[3] B. Agard, C. Morency, and M. Trépanier, "Mining public transport user behaviour from smart card data," in *Proc. 12th IFAC Symp. INCOM Control Probl.*, 2006, pp. 17–19.

[4] C. Morency, M. Trépanier, and B. Agard, "Analysing the variability of transit users behaviour with smart card data," in *Proc. IEEE ITSC*, 2006, pp. 44–49.

[5] J. Zhao, C. Tian, F. Zhang, C. Xu, and S. Feng, "Understanding temporal and spatial travel patterns of individual passengers by mining smart card data," in *Proc. IEEE 17th ITSC*, 2014, pp. 2991–2997.

[6] B. Agard, C. Morency, and M. Trépanier, *Mining Smart Card Data From an Urban Transit Network*. Hershey, PA, USA: IGI Global, 2009.

[7] M.-P. Pelletier, M. Trépanier, and C. Morency, "Smart card data use in public transit: A literature review," *Transp. Res. C, Emerging Technol.*, vol. 19, no. 4, pp. 557–568, Aug. 2011.

[8] J. Zhang *et al.*, "Analyzing passenger density for public bus: Inference of crowdedness and evaluation of scheduling choices," in *Proc. IEEE 17th ITSC*, 2014, pp. 2015–2022.

[9] I. Ceapa, C. Smith, and L. Capra, "Avoiding the crowds: Understanding tube station congestion patterns from trip data," in *Proc. ACM SIGKDD Int. Workshop Urban Comput.*, 2012, pp. 134–141.

[10] S. Liu, Y. Liu, L. Ni, M. Li, and J. Fan, "Detecting crowdedness spot in city transportation," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1527–1539, May 2013.

[11] Y. Li and M. J. Cassidy, "A generalized and efficient algorithm for estimating transit route ODs from passenger counts," *Transp. Res. B, Methodol.*, vol. 41, no. 1, pp. 114–125, Jan. 2007.

[12] B. Li, "Markov models for Bayesian analysis about transit route origin–destination matrices," *Transp. Res. B, Methodol.*, vol. 43, no. 3, pp. 301–310, Mar. 2009.

[13] N. Belloni, L. E. Holmquist, and J. Tholander, "See you on the subway: Exploring mobile social software," in *Proc. Extended Abstracts CHI Human Factors Comput. Syst.*, 2009, pp. 4543–4548.

[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, pp. 226–231.

[15] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[16] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surveys*, vol. 41, no. 3, p. 15, Jul. 2009.

[17] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[18] T. White, *Hadoop: The Definitive Guide: The Definitive Guide*. Newton, MA, USA: O'Reilly Media, 2009.

[19] D. Borthakur, *HDFS Architecture Guide*. HADOOP Apache Project. Forrest Hill, MD, USA, 2008. [Online]. Available: http://hadoop.apache.org/common/docs/current/hdfs\ignorespacesdesign.pdf

[20] L. George, *HBase: The Definitive Guide*. Newton, MA, USA: O'Reilly Media, 2011.

[21] A. F. Gates *et al.*, "Building a high-level dataflow system on top of map-reduce: The pig experience," *Proc. VLDB Endowment*, vol. 2, no. 2, pp. 1414–1425, Aug. 2009.

[22] A. Thusoo *et al.*, "HIVE: A warehousing solution over a map-reduce framework," *Proc. VLDB Endowment*, vol. 2, no. 2, pp. 1626–1629, Aug. 2009.

[23] L. Liu, A. Hou, A. Biderman, C. Ratti, and J. Chen, "Understanding individual and collective mobility patterns from smart card records: A case study in Shenzhen," in *Proc. 12th IEEE ITSC*, 2009, pp. 1–6.

[24] Y. Gong *et al.*, "Exploring spatiotemporal characteristics of intra-urban trips using metro smartcard records," in *Proc. 20th Int. Conf. IEEE GEOINFORMATICS*, 2012, pp. 1–7.

[25] L. Sun, D.-H. Lee, A. Erath, and X. Huang, "Using smart card data to extract passenger's spatio-temporal density and train's trajectory of MRT system," in *Proc. ACM SIGKDD Int. Workshop Urban Comput.*, 2012, pp. 142–148.

[26] J. Huang, L. Wang, C. Tian, F. Zhang, and C. Xu, "Mining freight truck's trip patterns from GPS data," in *Proc. 17th IEEE ITSC*, 2014, pp. 1988–1994.

[27] Y. Li, C. Tian, F. Zhang, and C. Xu, "Traffic condition matrix estimation via weighted spatio-temporal compressive sensing for unevenly-distributed and unreliable GPS data," in *Proc. 17th IEEE ITSC*, 2014, pp. 1304–1311.

[28] L. Yin, J. Hu, L. Huang, F. Zhang, and P. Ren, "Detecting illegal pickups of intercity buses from their GPS traces," in *Proc. 17th IEEE ITSC*, 2014, pp. 2162–2167.

[29] Z. Tian *et al.*, "Understanding operational and charging patterns of electric vehicle taxis using GPS records," in *Proc. 17th IEEE ITSC*, 2014, pp. 2472–2479.

Fan Zhang received the Ph.D. degree in communication and information system from Huazhong University of Science and Technology, Wuhan, China, in 2007.

He is an Associate Professor with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. From 2009 to 2011, he was a Postdoctoral Fellow with the University of New Mexico, Albuquerque, NM, USA, and the University of Nebraska—Lincoln, Lincoln, NE, USA. His research topics include big data processing, data privacy and network security, and wireless networks.

Juanjuan Zhao received the M.S. degree from the Department of Computer Science, Wuhan University of Technology, Wuhan, China, in 2009. She is currently working toward the Ph.D. degree with Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China.

From 2009 to 2012, she was a Research Assistant with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. Her research interests include cloud computing, big data processing, streaming-data processing, data fusion techniques, big-data-driven systems, and spatiotemporal data mining.

Chen Tian received the B.S., M.S., and Ph.D. degrees from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2000, 2003, and 2008, respectively.

He is currently an Associate Professor with the School of Electronics Information and Communications, Huazhong University of Science and Technology. From 2012 to 2013, he was a Postdoctoral Researcher with the Department of Computer Science, Yale University, New Haven, CT, USA. His research interests include network function virtualization, data center networks, distributed systems, Internet streaming, and big data processing for smart cities.

Chengzhong Xu (SM'14) received the Ph.D. degree from the University of Hong Kong, Pokfulam, Hong Kong, in 1993.

He is currently a Professor with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA. He also holds an adjunct appointment with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, as the Director of the Institute of Advanced Computing and Data Engineering. He has published more than 200 papers in journals and conferences. His research interest is in parallel and distributed systems and cloud computing.

Dr. Xu was a Best Paper Nominee at the 2013 IEEE High Performance Computer Architecture and the 2013 ACM High Performance Distributed Computing Conferences. He serves on the Editorial Board of a number of journals, including the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the *Journal of Parallel and Distributed Computing*, and *Science China Information Sciences*. He received the Faculty Research Award, the Career Development Chair Award, and the WSU President's Award for Excellence in Teaching. He also received the Outstanding Overseas Scholar Award from the National Science Foundation of China.

Xue Liu (M'06) received the Ph.D. degree in computer science from the University of Illinois at Urbana—Champaign, Champaign, IL, USA.

He is a William Dawson Scholar and an Associate Professor with the School of Computer Science, McGill University, Montreal, QC, Canada. He has also worked as the Samuel R. Thompson Chaired Associate Professor with the University of Nebraska—Lincoln, Lincoln, NE, USA, and HP Labs, Palo Alto, CA, USA. He has published over 150 research papers in major peer-reviewed international journals and conference proceedings in his areas of interest. His research interests include computer and communication networks, real-time and embedded systems, distributed systems, cyberphysical systems, and green computing.

Dr. Liu received the Best Paper Award from the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS in 2008 and the First Place Best Paper Award from the 2011 ACM Conference on Wireless Network Security. His research has been reported by news media, including the *New York Times, Computer World, The Register, Huffington Post, CBC, New Scientist*, and *MIT Technology Review's Blog*. He serves on the Editorial Board of the IEEE TRANSACTIONS OF PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.

Lei Rao received the Ph.D. degree from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2010.

She is a Senior Researcher with General Motors Research Laboratories, Warren, MI, USA. In 2011, she was a Research Associate with the School of Computer Science, McGill University, Montreal, QC, Canada. Her research interests include statistical learning and big data, connected vehicles, green computing, smart grids, and mathematical modeling and optimization. Her papers have appeared in the PROCEEDINGS OF THE IEEE, the IEEE TRANSACTIONS ON SMART GRIDS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the *IEEE International Conference on Computer Communications, the ACM/IEEE International Conference on Cyber-Physical Systems*, and the *IEEE International Conference on Communications*.