

SubTrack: Enabling Real-time Tracking of Subway Riding on Mobile Devices

Guo Liu*, Jian Liu[†], Fangmin Li[‡], Xiaolin Ma*, Yingying Chen[†] and Hongbo Liu[§]

*WuHan University of Technology, Wuhan, P.R.China

[†]Stevens Institute of Technology, Hoboken, NJ, USA

[‡]Changsha University, Changsha, P.R.China

[§]Indiana University-Purdue University Indianapolis, Indianapolis, IN, USA

Email: guoliuwut@gmail.com, jliu28@stevens.edu, lfm@ccsu.edu.cn, maxiaolin0615@whut.edu.cn, yingying.chen@stevens.edu, hl45@iupui.edu

Abstract—Real-time tracking of subway riding will provide great convenience to millions of commuters in metropolitan areas. Traditional approaches using timetables need continuous attentions from the subway riders and are limited to the poor accuracy of estimating the travel time. Recent approaches using mobile devices rely on GSM and WiFi, which are not always available underground. In this work, we present SubTrack, utilizing sensors on mobile devices to provide automatic tracking of subway riding in real time. The real-time automatic tracking covers three major aspects of a passenger: detection of entering a station, tracking the passenger’s position, and estimating the arrival time of subway stops. In particular, SubTrack employs the cell ID to first detect a passenger entering a station and exploits inertial sensors on the passenger’s mobile device to track the train ride. Our algorithm takes the advantages of the unique vibrations in acceleration and typical moving patterns of the train to estimate the train’s velocity and the corresponding position, and further predict the arrival time in real time. Our extensive experiments in two cities in China and USA respectively demonstrate that our system can accurately track the position of subway riders, predict the arrival time and push the arrival notification in a timely manner.

I. INTRODUCTION

Subway riding remains as a major convenient means of public transportation in many years and presents a strong growing trend as the population of urban cities increases. Most of the passengers spend their time on reading, playing games, watching videos, listening to music, or simply dozing off while taking subways [1]. Current subway administration only offers trip maps and voice announcements to indicate the upcoming stops, and such information can be easily ignored by the passengers in a noisy and crowded train environment. As a result, people could easily miss their stops unless they are fully attentive during their subway rides. It will be convenient and helpful for passengers to obtain the stop times automatically and track their subway trips. Based on the fact of the prevalence usage of mobile phones these days, we seek a solution that can enable real-time tracking of subway riding on mobile devices, providing intelligent information of the passenger’s location underground and predicting the arrival time.

To enable accurate arrival time prediction and notification, it is critical to keep track of the velocity and position of the

subway train where the passenger rides currently in real time. Intuitively, the personal mobile devices with built-in GPS could be utilized to perform the tracking task, however, the GPS signals are too weak to provide reliable location results underground. Furthermore, several studies [2, 3] use either GSM signal strengths or barometer readings along the subway line to track subway train riding. However, GSM signals are not always available underground, making it hard to scale. Barometer-based approach is difficult to use when the stations are built at the similar horizontal planes. New approaches explore the possibility of using inertial sensors embedded in mobile devices to detect the train dynamics [4, 5] together with fixed timetables. However, passengers need to manually trigger the stop estimation process on their phones. And these approaches depend on timetables, which cannot reflect the real-time traffic situation and various dwelling time at train stops, resulting in only coarse-grained train stop estimation.

In this paper, we make use of the existing inertial sensors on mobile devices to take one step forward by developing a real-time fully-automatic passenger position tracking and arrival notification system without requiring passenger’s involvement. The basic idea is to track the motion states of the subway train, including both velocity and position, through accumulated accelerometer readings over time, and then predict the arrival time to the next stop based on the current motion state and a subway station map. To facilitate such a system design, several key challenges should be addressed: (1) *departure time detection and station identification*: To automate the initialization of a subway ride tracking, it is critical to pinpoint the departure time and station information as the starting point; (2) *smartphone posture alignment*: We need to understand the smartphone’s posture before proceeding to the inertial sensor data collection, otherwise the motion states will not be accurately derived; (3) *velocity drift calibration*: Velocity drift in inertial systems is inevitable as indicated in previous investigations [6]. An effective calibration method should be developed to mitigate the impact of velocity drift for real-time and accurate tracking.

To trigger the tracking of subway riding automatically without passenger’s involvement, SubTrack needs to detect a

passenger enters a subway station and determine the station information. We find each subway station is associated with one or two cell tower IDs (Cell-ID). These Cell-IDs are stable inside each subway station. This useful phenomenon provides us an unique opportunity to detect whether the passenger has entered the subway station by examining the specific Cell-ID connections. In this work, we develop a passenger station entrance detection mechanism to accurately pinpoint when and which station the passenger enters.

In order to accurately track the motion states of the subway train, including both velocity and position, we need to calibrate the accumulated velocity and distance which have large drift errors due to biased motion sensor readings on mobile phones. Existing studies [6, 7] show that the accumulated drift error in inertial navigation system increases linearly over the time, which suggests that such a linear relationship could be determined if some actual information (e.g., velocity) in the middle of the subway ride is known. Fortunately, the train stops from time to time, which indicates that we can capture the stationary period (i.e., zero velocity) when it stops at each station. Specifically, we use an acceleration energy based approach to detect each subway train stopping period, and exploit it to estimate the velocity drift gradient (i.e., velocity drift during a sensor reading sample period) and calibrate the accumulated velocity/distance in real time. With the calibrated traveling distance, we can then predict the arrival time to the next stop based on a subway station map. By integrating all the above components and findings, SubTrack could perform accurate velocity and position estimation, and predict the arrival time and send the arrival notification in a timely manner.

The following contributions are made in this work:

- We propose SubTrack, which leverages inertial sensors on mobile devices to automatically depict the whole subway riding trip for each passenger in real-time including passenger station entering detection, passenger’s position estimation during the train ride, and arrival time prediction.
- We design a passenger station entrance detection mechanism to accurately determine when the passenger enters a subway station and which station it is.
- We develop a real-time calibration scheme to mitigate velocity drift error introduced by biased motion sensor readings to provide accurate passenger position estimation and further enable stop arrival time prediction.
- We align mobile devices held in arbitrary postures by passengers with the moving direction of the subway train.
- Extensive experiments are conducted in two cities, one in China and the other in USA, to validate the performance of SubTrack. The experimental results demonstrate the feasibility and efficiency of SubTrack.

II. CHALLENGES & SYSTEM OVERVIEW

Smartphone-based localization has been widely studied in many different scenarios [4, 8]. But as far as we know,

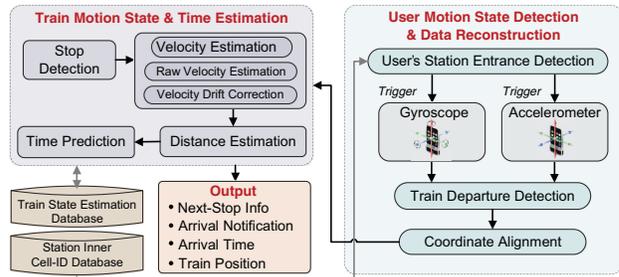


Fig. 1. Overview of SubTrack system flow.

few work has addressed the problem of localization in underground public transportation systems, where GPS signal and wireless infrastructure are not always available. Existing work mainly relies on the built-in inertial sensors to allow smartphones to determine their location substantially [4, 5]. But some strong assumptions of these approaches, such as fixed smartphone postures and reliable sensor readings, prevent them from practical use. Therefore, we propose a generic real-time subway rider tracking system leveraging the build-in inertial sensors of mobile device. In this section, we first point out several key challenges of the system design, and then introduce the proposed system work flow.

A. Challenges

To ensure real-time and accurate subway rider tracking, the following key challenges should be dealt with in the system.

Smartphone Posture Alignment. How the smartphones is carried along with the passenger has substantial impact on the inertial sensor readings. Particularly, if the smartphone is not placed in parallel with the moving direction of the train, it is difficult to correctly derive the motion state of subway train based on the raw sensor readings, not even to mention the smartphone of changing postures. As such, the posture of smartphone is very critical so that an effective posture alignment scheme needs to be developed to convert the raw sensor readings to the coordinate system better serving the subway rider tracking.

Passenger Station Entrance Detection. As the first step to track the subway rider, we need to accurately determine when the passenger enters subway station and which station it is. It is intuitive to apply existing GPS-based and wireless infrastructure-based localization methods to underground transportation system. However, GPS signals can not penetrate through dense earth, while wireless infrastructures are rarely available underneath. So an intelligent entrance detection scheme should be integrated into the SubTrack system to accurately identify the starting point of the trip.

Unreliable Sensor Readings. Due to the inherent flicker noise in the electronics and in other components susceptible to random flickering [6], inertial sensors, including both accelerometer and gyroscope, will produce biased measurements over time, which lead to inaccurate velocity estimation results. It is not realistic to remove such bias inside the sensor themselves. Therefore, an effective data calibration

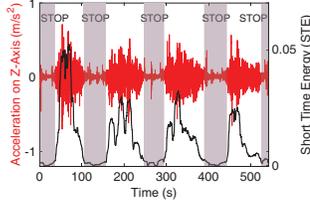


Fig. 2. Illustration of train stopping detection using short time energy (STE).

scheme will be proposed to compensate the bias in the sensor measurements, and thereby improve the accuracy of velocity estimation.

B. System Overview

Given the above challenges, we propose a real time fully-automatic subway riding tracking system, SubTrack. It aims to track the motion states of the subway train, including both velocity and position, through accumulated accelerometer readings over time, and then predict the arrival time to next train stop given the current motion state and subway map. As illustrated in Figure 1, SubTrack includes two major components: *Passenger Motion State Detection & Data Reconstruction* and *Train Motion State & Arrival Time Estimation*. In *Passenger Motion State Detection & Data Reconstruction*, the smartphone of a particular passenger periodically scan nearby cell-tower IDs (Cell-IDs) until it finds one matched record in the *Subway Station Inner Cell-ID Database*, which is built based on historical Cell-ID collections from all subway stations. Once the Cell-IDs match, the departure station of the passenger will be identified accordingly, and then the SubTrack system will collect the inertial sensor readings (i.e., accelerometer and gyroscope) subsequently. Meanwhile, the system will also remind the passenger to manually input his/her destination, which is used for arrival time predication and corresponding notification push.

The raw sensor data is then sent to *Train Departure Detection* module to find out whether the passenger is aboard, then the sensor data before the passenger gets on the train will be removed. Next, SubTrack converts the sensor readings from the smartphone's coordinate to the subway train's coordinate via *Coordinate Alignment* module so that SubTrack can correctly derive the train's motion status.

Inside the core component *Train Motion State & Arrival Time Estimation*, SubTrack first derives the real-time velocity of subway train through accumulating the accelerometer readings over time. Next, we need to identify the stopping periods of the train, which serve as the stationary reference points (i.e., zero velocity) to calibrate the velocity estimation. Integrating both the stationary reference points and specific historical velocity from *Train State Estimation Database*, the velocity drift error will be mitigated based on the linear model between the drift error and time. The *Train State Estimation Database* is built off-line based on the historical train motion state information (i.e., raw and calibrated velocity, traveling distance and reference points), and it could

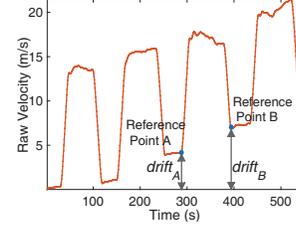


Fig. 3. Accumulated raw velocity of a subway train between 5 stations.

be updated automatically after the passenger taking subway. Given the calibrated velocities, we could keep tracking the train's position accurately. Furthermore, we can also inform the passenger how many train stops remained to his/her final destination, and predict the arrival time to the next train stop based on public geographic information (i.e., subway map).

III. TRAIN MOTION STATE TRACKING AND ARRIVAL TIME PREDICATION

In this section, we focus on detecting whether the train stops, tracking train's position and predicting arrival time using a mobile device (e.g., smartphone). In order to support the above modules, there are many building blocks including *Passenger Station Entrance Detection*, *Train Departure Detection* and *Coordinate Alignment* which are discussed in Section IV and Section V.

A. Train Stopping Detection

Train stopping detection aims to detect when the train has stopped and how long it stays at the train stops (i.e., stopping periods). More importantly, the stopping periods could serve as the reference points (i.e., zero velocity) to calibrate the estimated velocity of the train, which will be discussed in Section III-B.

We observe that the train has minute vibrations along with z -axis on the go, which results in the jitter of z -axis accelerometer readings. We thus can utilize the short time energy (STE) [9] of the acceleration readings on z -axis to detect whether the train has stopped. It is important to note that we use the gravity-aligned acceleration values, which is discussed in Section V, instead of raw accelerometer readings to eliminate the impact of mobile device's arbitrary postures. In addition, the acceleration readings may also be affected by the changing postures (e.g., holding the phone and walking in the train). But it only results in the low frequency sensor readings, i.e., less than 2 Hz [10], whereas the train on the go produces a much higher frequency sensor readings. To eliminate the impact of changing postures, we adopt a Butterworth high-pass filter with cut-off frequency 50 Hz in the velocity estimation module. Figure 2 depicts the z -axis accelerometer readings and corresponding STE on a running train. It is easy to find that the STE of the accelerometer readings on z -axis is much lower when the train has stopped. Therefore, the stopping period of a subway train should be identified with a carefully designed threshold.

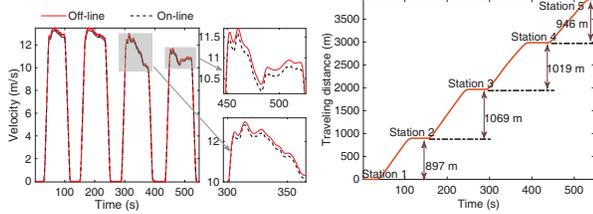


Fig. 4. Estimated velocity of subway train between 5 stations. Fig. 5. Estimated traveling distance of subway train between 5 stations.

B. Velocity Estimation

In order to track the train's position and predict the arrival time to next train stop, we need to have real-time estimation on the train's running speed.

1) *Raw Velocity Estimation*: The basic idea of raw velocity estimation is to accumulate the acceleration readings over time along the moving direction of the train. We assume the train starts moving at the time $\tau = 0$, and the accelerometer readings $acc(\tau)$ are sampled at a constant rate f sample/sec. Then the train's velocity at time $\tau = t$ can be expressed as:

$$vel(t) = vel(0) + \sum_{\tau=0}^t \frac{1}{f} \times acc(\tau), \quad (1)$$

Figure 3 shows an example of velocity estimation for a five-stop trip. The estimated velocity should fall back to zero when the train has stopped. However, due to the biased accelerometer readings, we still observe non-zero velocities at the train stops, called velocity drift error, which are indicated at the "Reference Points" in Figure 3. Further, such drift error follows an increasing trend over time.

2) *Velocity Drift Calibration*: In order to mitigate the velocity drift error, we develop an on-line (i.e., real-time) velocity calibration scheme based on the proposition that the accumulated drift error increases linearly over time, which has been verified by many existing studies [6, 7].

To derive the linear relationship between the drift error and time, we need to identify some reference points with deterministic velocity during the trip, and then fit them to a linear regression model. As shown in Figure 3, given two reference points A and B , which correspond to the stopping period, with zero velocity, the linear velocity drift can be simply calculated based on the velocity difference between two reference points. However, the drift error can not be obtained in real-time, since it has to wait until next reference point to get the second deterministic velocity to calculate the velocity difference.

To overcome the above limitation on velocity drift calibration, we estimate the velocity drift gradient Δvel (i.e., velocity drift during Δt) based on historical raw velocity information between neighboring subway stations, which can be obtained from Equation 1:

$$\Delta vel = \frac{1}{N} \sum_{k=1}^N \frac{drift_B^k - drift_A^k}{L_k}, \quad (2)$$

where $drift_A^k$ and $drift_B^k$ are the velocity drifts of two neighboring reference points (i.e., stopping periods at two adjacent stations), while we have N pairs of such neighboring

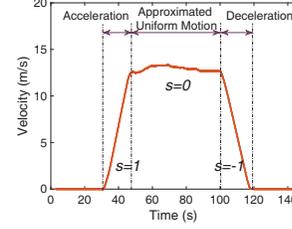


Fig. 6. Typical velocity pattern between two adjacent stations.

reference points from the dataset contributed by passengers' historical data or crowdsourcing. L_k is the number of sensor samples between the k^{th} pair of reference points. Based on the velocity drift gradient Δvel , the calibrated velocity $vel_{cali}(t)$ that is drift error-free could be obtained in real time:

$$vel_{cali}(t) = vel(t) - drift_A - (t - t_A) \times \Delta vel, \quad (3)$$

where $vel(t)$ is the estimated velocity at time t , $drift_A$ is the velocity drift at reference point A (i.e., the previous reference point before t) at time t_A . Figure 4 demonstrates an example of the velocity estimation for five-stop subway ride with both our proposed real-time velocity estimation method (i.e., on-line) and the method in [7] (i.e., off-line). We can find the calibrated velocities from the two methods are very similar to each other, so it validates the effectiveness of our proposed method on eliminating the velocity drift.

C. Traveling Distance Estimation

Given the calibrated velocities, we could derive the current position of the train through the traveling distance estimation and thereby predict the arrival time to next train stop. Specifically, the traveling distance of a subway train can be obtained based on the integration on the calibrated velocity over time. The calculated traveling distance at time t from last train stop can be represented as:

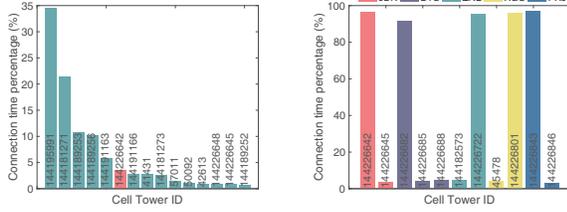
$$dis(t) = dis(0) + \sum_{\tau=0}^t \frac{1}{f} \times vel_{cali}(\tau). \quad (4)$$

Figure 5 gives an example of the estimated traveling distance over 5 train stops. The estimated stop-to-stop distances are 897m, 1069m, 1019m, and 946m, which achieve the average error as low as 53m by comparing with the official subway construction map [11]. Note that there is no accumulated error in our distance estimation since the distance of each stop-to-stop segment is estimated independently.

D. Arrival Time Prediction

We next perform arrival time prediction to the next train stop. Accordingly, the proposed system will keep updating the arrival time to the final destination in real-time, and notify the riders to prepare to get off near arrival.

In our empirical study in the two cities of China and USA, we observe that the train usually experiences three motion phases between two adjacent train stops as shown in Figure 6: *acceleration* ($s = 1$), *approximate uniform motion* ($s = 0$) and *deceleration* ($s = -1$). Moreover, the absolute values of acceleration are almost identical during *acceleration* and *deceleration*, and the motion pattern of the train between two



(a) Outside of a station (b) Inside of 5 stations
Fig. 7. Cell-ID connection time outside and inside of subway stations.

adjacent stations is approximately symmetric with respect to the middle point of two adjacent train stops.

According to our experimental observations, we assume that the absolute accelerometer readings during *acceleration* and *deceleration* are a constant value, acc . Based on the distance to next station that is obtained in Section III-C, we can predict the arrival time to the next train stop at different motion phases of the train as follows:

$$T_{arrival}(t) = \begin{cases} \frac{vel_{uni}}{acc} + \frac{d}{vel_{uni}} - t, & s = 1 \\ \frac{vel_{uni}}{acc} + \frac{d - dis(t) - \frac{vel_{uni}^2}{2acc}}{vel_{cali}(t)}, & s = 0 \\ \frac{vel(t)}{acc}, & s = -1 \end{cases}, \quad (5)$$

where acc is the accelerometer readings which can be directly obtained on the smartphone, and d is the actual distance between the adjacent stations. $vel_{cali}(t)$ and $dis(t)$ are the calibrated velocity and distance, vel_{uni} is the uniform velocity which can be measured when $s = 0$ or obtained from historical velocity when $s = 1$. Note that the motion status of the train (i.e., s) can be obtained by applying a predefined threshold to the short time energy of the train's well-aligned moving acceleration, since the short time energy during the acceleration and deceleration is much larger than that of the *approximated uniform motion*.

The arrival time to next train stop can be predicted in real-time in Equation 5. We can further predict the arrival time to the final destination by integrating historical remaining time information to the rest of the stations. In addition, our algorithm will also intelligently adjust the arrival time predication at every train stop, and prompt notification when the train is approaching to the final destination.

IV. DETECTION OF PASSENGER MOTION STATE BEFORE BOARDING

In this section, we focus on the passenger's motion state detection. Specifically, two main topics are included: 1) *Passenger's station entrance detection*. It aims to determine when the passenger enters the subway station and which station it is; 2) *Train departure detection*. We will detect whether the train departs by differentiating the accelerometer readings caused by the moving train from those corresponding to the walking passenger.

A. Passenger Station Entrance Detection

In the proposed SubTrack system, it is critical to pinpoint the departure time and departure station as the starting point of the trip via subway. Since the GPS signal is too

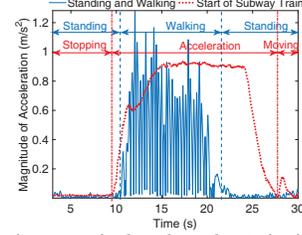
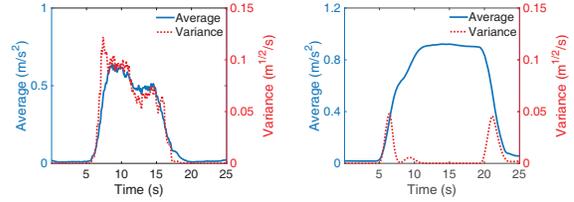


Fig. 8. Acceleration magnitude when the train is accelerating or the passenger is walking in the station before the train starts to move.



(a) Before train accelerates (b) During train accelerates
Fig. 9. Average and variance of acceleration before and during train starts.

weak to provide reliable location information underground, instead we present a Cell-ID based approach to perform the passenger station entrance detection.

Our empirical study observes that the Cell-ID that associates with the smartphone is usually unique and stable inside each specific subway station. It is also much different from the associated Cell-ID around the subway station entrance but outside the station, since the base stations deployed inside the subway station have more reliable wireless links than those outside.

Figure 7(a) presents the Cell-ID scanning results when the passenger walks around but outside a subway station entrance (i.e., *JDK* station), and the Cell-ID in red color is the one when the passenger is inside the subway station as shown in Figure 7(b). We can find that when the passengers are at the station entrance yet not getting in, their smartphones could associate with multiple Cell-IDs, and switch among some of them frequently. Note that it is also difficult for the passenger to connect to the Cell-ID inside the station. Figure 7(b) reports the Cell-IDs that the smartphones could associate with when the passengers stay inside the subway stations. Note that each color corresponds to one particular subway station. It is obvious that the smartphones maintain a long-term connection with one unique Cell-ID (e.g., 98% of the time connected with 144226642) inside each station. We could infer that those Cell-IDs must be specifically deployed inside the stations. Inspired by the above observations, we can determine whether the passenger enters a subway station by periodically examining the duration (i.e., λ seconds) that his/her smartphone has been in association with a unique Cell-ID inside the subway station.

However, it is a different story for some old subway stations (e.g., a few in NYC, USA). The Cell-ID associations around the station entrances are similar as what we observed previously, but there is no cellular signal inside these stations. In other words, the smartphone will lose the associations with

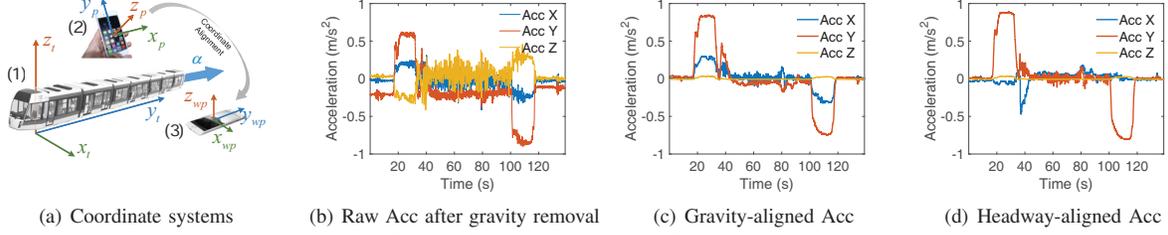


Fig. 10. Align coordinate with the subway train. (a) Different Coordinate systems. (b) Raw acceleration on three axes after gravity removal. (c) Gravity-aligned acceleration of the train. Only z axis of Acc is aligned with the subway train's z axis (gravity direction). (d) Headway-aligned acceleration of the train. All three axes of Acc are aligned with the subway train.

any Cell-ID when the passenger enters the subway station. Given the above observation for old subway stations, we need to determine the subway station the passenger is approaching before the cellular connections are lost.

B. Train Departure Detection

In order to pinpoint the departure time of the train, we first need to determine whether the train that the passengers board is on the go. We notice that the accelerometer readings have different patterns before and after the train starts moving. Therefore, it could help to determine when the train departs by differentiating the accelerometer readings.

Figure 8 shows the acceleration magnitude of the acceleration readings after gravity removal before and during the subway train moves. We can see that the train keeps accelerating with constant acceleration around $0.9m/s^2$ after it departs the station. On the contrary, the acceleration solely caused by human walking varies randomly. The above observation inspires us to detect the train's departure by examining the average and variance of acceleration. Figure 9 depicts the average and variance of accelerometer readings in sliding windows. We can find that when the passenger walks, both the average and variance of acceleration maintain at high level. On the other hand, the average and variance during train accelerating keep stable, but the variance is low. Thus, given appropriate thresholds on the acceleration average and variance, we can detect the departure of the train in real time.

V. COORDINATE ALIGNMENT

Coordinate alignment works after the passenger boards the train. It aims to align the orientation of the mobile device with the moving train, so that the motion state of the train can be understood directly based on the sensor readings. As depicted in Figure 10(a), our goal is to convert the arbitrary pose $\{x_p, y_p, z_p\}$ of mobile device to a well aligned one $\{x_{pw}, y_{pw}, z_{pw}\}$, which is parallel with the pose of the train $\{x_t, y_t, z_t\}$. Note that y axis of the well aligned smartphone is the train's moving direction. The proposed coordinate alignment scheme consists of two main parts: *gravity-alignment* and *headway-alignment*.

1) *Gravity-alignment*: Gravity-alignment aims to convert the pose of mobile device to a *middle status*, only z axis is aligned with gravity direction, between its current pose and well aligned pose. We first sense and align the current

posture of the smartphone by using quaternion [12], which is obtained from the API of CMAttitude [13]. The derived quaternion represents the measurement of the smartphone's pose relative to the *middle status* [13]. Thus the gravity-alignment is completed as follows:

$$[\mathbf{0}, \hat{a}_1] = \hat{q}([\mathbf{0}, \hat{a}]) (\hat{q}^{-1}) = [\mathbf{0}, \hat{a}_{1x}, \hat{a}_{1y}, \hat{a}_{1z}], \quad (6)$$

where $\hat{a} = [a_x, a_y, a_z]$ is the time series of acceleration, and $\hat{q} = [s, x, y, z]$ is the quaternion defined in CMAttitude. Figure 10(c) plots the gravity-aligned acceleration (i.e., \hat{a}_1) while the train passes two adjacent stations, in which the acceleration on z axis is aligned with z_t axis with gravity filtered. We observe that the moving state of the train is converted into \hat{a}_{1x} & \hat{a}_{1y} plane (the same as x_t & y_t plane). The next step of our alignment scheme is to transfer the moving state in \hat{a}_{1x} & \hat{a}_{1y} plane to y axis of the well aligned smartphone to facilitate velocity estimation.

2) *Headway-alignment*: The goal of headway-alignment is to align the gravity-aligned acceleration to the train's coordinate, so that the train's moving state can be obtained from the sensor readings on single axis (y axis). Given a *known acceleration* in the x_t & y_t plane, we can eliminate the ambiguity of acceleration direction estimation in \hat{a}_{1x} & \hat{a}_{1y} plane.

We observe that the acceleration during accelerating and decelerating is always in parallel with the moving direction of the train (i.e., y_t), so it has little impact on x_t and z_t of the train. Thus the headway-alignment is equivalent to maximizing the absolute value of acceleration on y_p by rotating the acceleration in \hat{a}_{1x} & \hat{a}_{1y} plane when the subway train accelerates or decelerates.

Utilizing the variant trigonometric function described in [14], we can derive the deviation angle θ that the acceleration in \hat{a}_{1x} & \hat{a}_{1y} relative to the *known acceleration* that is parallel with y_t . Thus we can calculate a quaternion \hat{q}_1 from the specific angle: $\hat{q}_1 = [\cos(\theta), \sin(\theta) \times [0, 0, 1]]$. Then we can align the acceleration in \hat{a}_{1x} & \hat{a}_{1y} plane with the *known acceleration* as follows:

$$[\mathbf{0}, \hat{a}_2] = \hat{q}_1([\mathbf{0}, \hat{a}_1]) (\hat{q}_1^{-1}), \quad (7)$$

where \hat{a}_2 is the well aligned acceleration. Figure 10(d) presents the well-aligned acceleration while the train passes two adjacent stations. We can observe that the motion states including acceleration, deceleration, etc, of subway train are simply reflected along the y axis, which provides the convenience for velocity estimation of the train.



(a) *City-1*, subway lines 2&4 (b) *City-2*, subway lines A&E
Fig. 11. Subway lines that we conduct our experiments in two cities.

TABLE I
INFORMATION ABOUT THE FOUR SUBWAY LINES.

Line	# 2	# 4	# A	# E
Total length (km)	27.7	33.2	49.7	25
# of stations	21	28	36	32
Avg. station-2-station dis. (km)	1.38	1.22	1.42	0.81
Avg. station-2-station time (min)	2.6	2.48	2.65	1.9

VI. EXPERIMENTAL EVALUATION

In this section, we first introduce the experimental methodology, and then evaluate the overall performance of our proposed SubTrack system from several aspects, including distance estimation, arrival time estimation, passenger station entrance detection and train stopping detection.

A. Experimental Methodology

We prototyped SubTrack on smartphones running iOS 9 (i.e., iPhone 5s, 6 and 6 Plus). The experiments are carried out in two cities, one in China (subway lines No. 2 & No. 4) and the other in USA (subway lines No. A & No. E), with 8 participants over a four-months time period. Figure 11 shows the four subway lines where the experiments are conducted. The detailed information of the subway map, including the length of the line, total number of stations, average distance and average duration between adjacent stations, can be found in Table I [15, 16]. In the experiments, we ask every participant to carry the smartphone installed with SubTrack and take the subway with both random departure and destination stations for the four lines in two cities. There is no special requirement for the participants on their movement and holding styles of the smartphones while taking the train. Before the train stops at a particular station, SubTrack should push a notification on the smartphone about the estimated arrival time to the next train stop, and the participants are asked to mark the train stops manually if the notification misses that train stop. All the system outcomes, including passenger station entrance, arrival time prediction, distance estimation and train stopping detection, will be stored in the memory for statistical analysis while the participants are conducting experiments. In addition, three different thresholds indicating the duration for stable Cell-ID association (i.e., $\lambda = 5, 15, 25s$) are utilized in the passenger station entrance detection.

B. Metrics

We use the following five metrics to evaluate the performance of SubTrack.

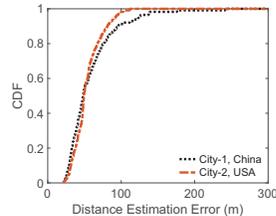
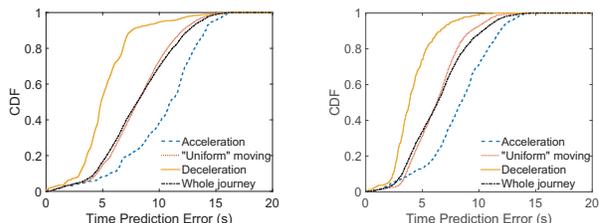


Fig. 12. Performance of traveling distance estimation.



(a) *City-1*, China (b) *City-2*, USA

Fig. 13. CDF of time prediction errors in different motion phases.

Distance Estimation Error. The distance estimation error is defined as the difference of the estimated train traveling distance and the actual distance the train travels.

Time Prediction Error. The time prediction error is defined as the difference between the predicted and the real arrival time when SubTrack pushes the arrival notification.

Precision/Recall of Passenger Station Entrance Detection. Precision is the fraction of the retrieved instances (i.e., the experiments that passengers are detected entering in the station) that are relevant, and recall is the fraction of the relevant instances (i.e., all experiments that passengers indeed enter in the station) that are retrieved.

Stopping Detection Accuracy. The stopping detection accuracy is the fraction of all train stops that are detected.

C. Distance Estimation Accuracy

We first study the distance estimation accuracy of SubTrack and compare the performance with SubwayPS [5] that relies on the train schedule on timetable. The overall performance of SubTrack significantly outperforms SubwayPS. Specifically, as shown in Figure 12, we observe that the mean distance estimation error and 90% estimation error are less than 56m and 100m in both cities. In contrast, according to the timetable of the subway line No.2 in the *City-1* (China) [4], the mean distance estimation error of SubwayPS is as high as 483m (35% of average distance 1380m between two adjacent stations), which is over seven times larger than that for SubTrack. It is also worthwhile to point out that the distance estimation error can be over 100m in few cases for SubTrack. This is because the subway train takes sharp turns at several places, resulting in severe acceleration fluctuation and thereby incorrect accelerometer readings. The above results confirm that SubTrack can achieve highly accurate distance estimation.

D. Arrival Time Estimation Accuracy

We next examine the arrival time estimation accuracy at different motion phases of the train. We first present

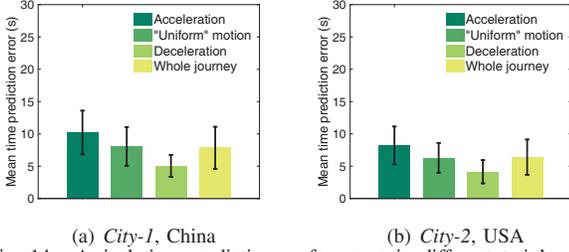


Fig. 14. Arrival time prediction performance in different train's motion phases.

overall arrival time prediction error shown as the black curves in Figures 13(a) and 13(b). We observe that median prediction errors are less than $6.4s$ and $7.8s$, while 90% prediction error are less than $12s$ and $11s$ in *City-1* and *City-2*, respectively. Further, as shown in Figure 14, we find that the mean prediction time errors and standard deviations are still less than $11s$ and $3.5s$ in both two cities. Comparing to the existing work [4] that has the mean error of $45s$, our proposed system has significant improvement on the arrival time prediction.

We also observe that the mean arrival time prediction errors are different at three different train's motion phases (i.e., accelerating, uniform motion, and decelerating). The average and standard deviation of the prediction errors at different motion phases and the overall ride are presented in Figure 14. The difference on mean prediction error is caused by different prediction strategies at the three running phases. In the *acceleration* phase, the prediction accuracy highly depends on the accuracy of the vel_{uni} from the historical data between adjacent stations. The average velocity during uniform moving phase, vel_{uni} , differs from time to time due to random spinning and skipping of the train wheels. Therefore, the arrival time prediction varies in wide range and the mean prediction error is the largest in all three phases. In the *uniform motion* phase, the average prediction error is less than that in the acceleration phase, since the train usually stays at a stable motion state velocity during this phase. It also implies that the arrival time prediction error will be relatively high if the velocity varies severely during this phase. In the *deceleration* phase, the arrival time prediction error stays low for most of the time. This is because the arrival time in this phase is predicated based on the current velocity and deceleration estimation without any historical data, which may introduce some drift errors. These above results demonstrate the effectiveness on the accurate arrival time prediction of SubTrack.

E. Cell-ID based Station Entering Point Detection

Figure 15 depicts the performance of our station entering point detection method with different duration thresholds in two cities. We can find that the detection precision is relatively low (i.e., 0.872) when the threshold is fixed at $\lambda = 5$ seconds, but the recall is very high (i.e., 0.986). This implies that, the inner cell-id of the station will be more likely to be matched with lower threshold when the passenger entering a

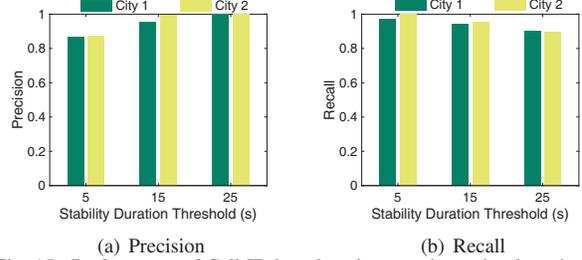


Fig. 15. Performance of Cell-ID based station entering point detection with three different thresholds in two cities.

TABLE II
TRAIN STOPPING DETECTION ACCURACY

Sampling Rate (Hz)	20	40	60	80	100
Detection Accuracy	0.911	0.932	0.955	0.970	0.981

subway station. However, the false triggering will also happen more frequently when the passenger walks by stations. In order to reduce the false triggering, we increase the duration threshold to $\lambda = 25$ seconds. The false triggering could be completely eliminated, however, it also decreases the recall rate. The decreased recall rate indicates that SubTrack can not determine whether the passenger is already inside the station even after the passenger has entered. This is because the connection to cell-tower may be lost for seconds due to the poor signal quality at some spots inside the stations.

According to above analysis, a tradeoff between good precision and good recall is made by setting the threshold at $\lambda = 15$ seconds. Given the updated threshold, most of the false triggering are eliminated while the entering point can also be detected with the average precision at 0.975. In addition, the passenger can also manually start SubTrack in case that the entering point detection fails.

F. Train Stopping Detection

Table II illustrates the train stopping detection accuracy with different sampling rates on motion sensors. We observe that the detection accuracy increases as the sampling rate increases. In particular, the detection accuracy reaches 0.955 when the sampling rate is higher than 60 Hz. Note that if the passengers prefer to save the energy for a longer battery lifespan, the detection accuracy can still maintain high level by reducing the sampling rate a little bit (e.g., 40 Hz). Comparing to the existing work [2, 3, 17], our system provides much better performance on the train stopping detection.

VII. RELATED WORK

In this section we will review the related work, which can be categorized as follows:

Subway Train Stopping Detection. Previous studies [2, 3, 17, 18] propose different ways to perform subway train stopping detection with mobile devices. The authors in [2] take acceleration variance and GSM information along the subway line as features to detect the train stopping. However, GSM signals are not always available underground, making it hard to scale. Moreover, barometers are adopted to detect train stopping in [3, 18]. However, the detection accuracy is very

low, since the pressure differences across different stations at similar horizontal plane are not significant. Furthermore, Nerimi [17] utilizes WiFi fingerprints to achieve acceptable detection accuracy. However, it highly relies on widely pre-deployed WiFi access points, which are not always available at subway stations.

Velocity Estimation Using Smartphones. Existing studies [19, 20] could estimate vehicle velocity by using GPS, GSM signal strength or motion sensors of mobile devices. In particular, velocity estimation of vehicles using GPS module in mobile devices has been developed [19, 20]. In addition, the authors in [21] estimate the speed of vehicles by matching time-series GSM signal strength data to a known signal strength trace from the same road. However, all of these approaches will not work underground. The signals being used are either too weak to provide reliable location information or not always available along with the subway tunnels underground. An alternative way is to use built-in motion sensors in the mobile devices [7]. However, this approach cannot achieve real-time velocity drift calibration with limited reference points in the subway traveling scenario.

Traveling Distance Estimation. A few studies [4, 5] propose to track the train’s position using the timetable of subway. In particular, these approaches track the train’s position according to the number of passed stations and the passed time period from the first station. However, they could only provide coarse-grained distance estimation with the minute level accuracy of the timetable [5]. In addition, the distance estimation accuracy may also greatly affected if the whole ride is not exactly follow the timetable.

Different from existing work, our SubTrack system designs velocity estimation, distance estimation and train stopping detection solely relying on the built-in inertial sensors in a smartphone. Through continuously tracking the motion states of the subway train, our system is able to provide real-time tracking of subway riding and accurate arrival time prediction for the subway riders.

VIII. CONCLUSION

In this paper, we propose SubTrack aiming to achieve real-time automatic tracking of subway riding leveraging the built-in sensors on mobile devices. The proposed system could accurately depict the whole subway riding trip of a passenger including detecting passenger entering a station, tracking the train/passenger’s position and estimating the arrival time of subway stops. In particular, SubTrack employs the unique cell tower identification to first detect when a passenger enters a station and then exploits inertial sensors on the passenger’s mobile device to derive the travel status of the train. A real-time velocity calibration method is developed to mitigate the velocity drift error resulted from the inherent bias in the motion sensors. Our algorithm takes the advantages of the accumulated acceleration readings and the typical moving patterns of the train to estimate the train’s velocity and position, and further predict the arrival time of the train in

real time. Extensive experiments conducted in two cities, one in China and the other in USA, validate the effectiveness and efficiency of the proposed SubTrack system. The results demonstrate that SubTrack can accurately track the position of subway riders, predict the arrival time and push the arrival notification in a timely manner.

IX. ACKNOWLEDGEMENT

This work is supported by the National Science Foundation of China under grant nos. 61502361 and 61373042, and the National Science Foundation under grant nos. CNS1514436 and CNS1409767.

REFERENCES

- [1] “Activities while commuting,” <https://www.ericsson.com/res/docs/2015/consumerlab/ericsson-consumerlab-commuters-expect-more.pdf>.
- [2] K. Yu, H. Zhu, H. Cao, B. Zhang, E. Chen, J. Tian, and J. Rao, “Learning to detect the subway station arrival for mobile users,” in *Intelligent Data Engineering and Automated Learning-IDEAL*. Springer, 2013, pp. 49–57.
- [3] T. Watanabe, D. Kamisaka, S. Muramatsu, and H. Yokoyama, “At which station am i?: Identifying subway stations using only a pressure sensor,” in *ISWC*, 2012.
- [4] A. Thiagarajan, J. Biagioni, T. Gerlich, and Jakob Eriksson, “Cooperative transit tracking using smartphones,” in *SenSys*, 2010.
- [5] T. Stockx, B. Hecht, and J. Schöning, “Subways: Towards smartphone positioning in underground public transportation systems,” in *SIGSPATIAL*, 2014.
- [6] O. J. Woodman, “Pedestrian localisation for indoor environments,” Ph.D. dissertation, University of Cambridge, 2010.
- [7] H. Han, J. Yu, H. Zhu, Y. Chen, J. Yang, Y. Zhu, G. Xue, and M. Li, “Senspeed: Sensing driving conditions to estimate vehicle speed in urban environments,” in *INFOCOM*, 2014.
- [8] P. Zhou, Y. Zheng, and M. Li, “How long to wait?: predicting bus arrival time with mobile phone based participatory sensing,” in *MobiSys*, 2012.
- [9] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals*. Prentice Hall, 1978.
- [10] L. Zhang, P. H. Pathak, M. Wu, Y. Zhao, and P. Mohaptra, “Accelword: Energy efficient hotword detection through accelerometer,” in *MobiSys*, 2015.
- [11] “Information about subway line No. 4 in Wuhan, China.” https://en.wikipedia.org/wiki/Line_4,_Wuhan_Metro.
- [12] C. Doran and A. Lasenby, *Geometric Algebra for Physicists*. Cambridge University Press, 2003.
- [13] “Apple CMAAttitude Class quaternion,” https://developer.apple.com/library/prerelease/ios/documentation/CoreMotion/Reference/CMAAttitude_Class/#//apple_ref/occ/instp/CMAAttitude/quaternion.
- [14] S. Goldt, S. van der Meer, S. Burkett, and M. Welsh, “The linux programmer guide,” *Linux Documentation Project*, 1995.
- [15] “Timetable of subway line in Wuhan, China.” http://ditie.114huoche.com/WuHan/x_221/.
- [16] “Timetable of subway line in Manhattan, USA.” <http://www.mta.info/schedules>.
- [17] I. Lee, G. Yoon, and D. Han, “Nerimi: Wifi-based subway navigation system,” in *Intelligent Radio for Future Personal Terminals (IMWS-IRFPT)*. IEEE, 2011, pp. 1–2.
- [18] S. Hyuga, M. Ito, M. Iwai, and K. Sezaki, “An online localization method for a subway train utilizing the barometer on a smartphone,” in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2016.
- [19] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Toledo, and J. Eriksson, “Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones,” in *SenSys*, 2009.
- [20] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. Bayan, M. Annaram, and Q. Jacobson, “Virtual trip lines for distributed privacy-preserving traffic monitoring,” in *MobiSys*, 2008.
- [21] G. Chandrasekaran, T. Vu, A. Varshavsky, M. Gruteser, R. P. Martin, J. Yang, and Y. Chen, “Vehicular speed estimation using received signal strength from mobile phones,” in *UbiComp*, 2010.