HybridBaro: Mining Driving Routes Using Barometer Sensor of Smartphone

Myounggyu Won, *Member, IEEE*, Ashutosh Mishra, *Student Member, IEEE*, and Sang H. Son, *Fellow, IEEE*

***Abstract*— Recent research showed that human mobility is characterized by reproducible patterns, i.e., humans tend to travel a few known places. Timely identification of these *significant journeys* has prospects for emerging intelligent applications like real-time traffic route recommendation and automated HVAC systems. Existing mobile systems, however, utilize energy-hungry sensors like GPS and gyroscope to detect significant journeys, which make it hard to keep such systems running to continuously monitor driving routes. To address this issue of energy efficiency without compromising the performance, in this paper, a hybrid mobile system based on the barometer sensor of a smartphone is developed. Distinctive elevation signatures of driving routes are captured using the smartphone barometer sensor that is exceptionally energy-efficient and position/orientation-independent. Degraded accuracy due to flat areas with minimal elevation changes is offset by developing an adaptive algorithm that opportunistically obtains GPS locations for a very short period of time when such flat areas are detected in real time. Using over 150 miles of field data, it is demonstrated that the proposed mobile system achieves the mean detection accuracy of 97% with the mean false positive rates of 1.5%.**

***Index Terms*— Driving route detection, mobile computing, driver information systems.**

I. Introduction

RECENT research showed that human mobility is characterized by a high degree of regularity [1], *i.e.,* humans tend to travel a few known places like their home or place of work. Effective identification of these "*significant journeys*" has significant implications on emergence of new smart applications. For example, better driving routes that avoid traffic congestion may be suggested to the user if we know where the user is headed; also, the user's arrival time can be estimated based on the user's significant journey. This estimated arrival time can be sent to the user's smart home HVAC controller to adjust the room temperature to the desired comfort level prior to arrival of the user [2].

Manuscript received June 21, 2017; accepted July 18, 2017. Date of publication August 2, 2017; date of current version September 8, 2017. This work was supported in part by the Global Research Laboratory Program through NRF under Grant 2013K1A1A2A02078326, and in part by the DGIST Research and Development Program (CPS Global Center) through the Ministry of Science, ICT and Future Planning. An earlier version of this paper was presented at the Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems. The associate editor coordinating the review of this paper and approving it for publication was Prof. Subhas C. Mukhopadhyay. *(Corresponding author: Myounggyu Won.)*

M. Won and A. Mishra are with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD 57006 USA (e-mail: myounggyu.won@sdstate.edu; ashutosh.mishra@sdstate.edu).

S. H. Son is with the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology, Daegu 42988, South Korea (e-mail: son@dgist.ac.kr).

Digital Object Identifier 10.1109/JSEN.2017.2734919

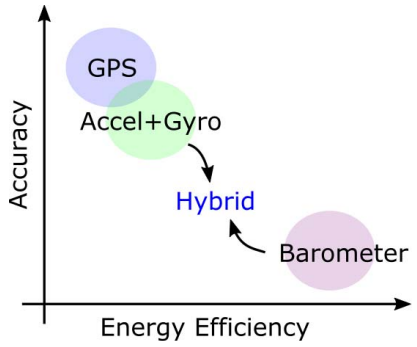

Fig. 1. Motivation for system design to achieve both energy efficiency and high detection accuracy especially for flat areas.

Numerous mobile systems have been developed to detect *significant places* where users spend most of their times during daily life routines [3], [4]. However, little attention was paid to identification of *significant journeys*. Recently mobile systems that are designed to identify significant journeys have been introduced [5]–[7]. Nawaz and Mascolo utilized the accelerometer and gyroscope of a smartphone to detect significant journeys [5]. More specifically, a significant journey was identified based on the distinctive angular speed variations caused by turns and bends of roads. A major issue is the energy efficiency because the gyroscope is known to be one of the most "power-hungry" sensors that is inappropriate to keep it turned on continuously. This energy-efficiency issue cannot be simply addressed by allowing the user to plug in the smartphone in a charging port of a vehicle because it places extra efforts for users to remember and plug in the phone every time he or she makes a journey. To tackle this problem, barometer-sensor-based approaches have been introduced [6], [7]. Extremely energy-efficient barometer sensor of a smartphone is used to identify significant journeys based on the distinctive elevation signatures of routing paths. However, these solutions suffer from limited detection accuracy on flat areas that do not have noticeable elevation changes.

As shown in Figure 1, the GPS-based solutions offer very high accuracy at the cost of degraded energy efficiency, while the barometer sensor-based approaches are very energy efficient at the cost of reduced accuracy. *In this article, we aim to strike the balance between these two lopsided approaches in order to maximize both the energy efficiency and accuracy as illustrated in Figure 1.* To this end, we design, implement, and evaluate a mobile system that utilizes the barometer sensor of a smartphone to detect significant journeys with minimal energy consumption while dynamically adapting to flat areas

by opportunistically performing very brief measurements of GPS locations to increase the overall detection accuracy. The proposed solution is the first hybrid mobile system that balances the energy efficiency and the accuracy in identifying significant journeys.

More specifically, in order to detect significant journeys, air pressure data are continually sampled while driving. Collected air pressure data are compared with previously obtained air pressure data sets that represent user-specified significant journeys. A number of key challenges are addressed to implement the first stand-alone mobile system that effectively identifies significant journeys based on the barometer sensor. First, a journey is represented with air-pressure data sequences of different sizes because of varying travel times, *e.g.,* due to different traffic conditions. An effective pattern matching algorithm is investigated that is computationally light-weighted to run on a mobile device and is capable of effectively performing pattern matching for data sequences with heterogeneous sizes. Additionally, a new mechanism is developed to find the beginning and end of a journey to ensure that the pattern matching algorithm is executed only during a journey to save energy. Another important challenge is to effectively reduce errors contained in measured air pressure data. To cope with this challenge, an algorithm is developed that calibrates the air pressure data of significant journeys. Furthermore, based on the observation that flat areas with no distinctive air pressure changes significantly affect the accuracy of the proposed system, a hybrid algorithm is proposed that adaptively and opportunistically utilizes momentary GPS readings to improve the accuracy at the cost of slightly increased energy consumption. Overall, we make the following contributions:

- The first stand-alone hybrid mobile system for detecting significant journeys based on the embedded barometer sensor of a smartphone is developed.
- An energy-efficient mechanism to determine the beginning and end of a journey is developed.
- An algorithm for calibrating air pressure data for significant journeys is designed to improve the detection accuracy.
- A hybrid approach is presented to cope with flat areas with no significant elevation changes.
- Real-world experiments with 150miles of field data are performed to validate the energy efficiency and accuracy of the proposed system.

This paper is organized as follows. In Section II we review the literature on position tracking algorithms for smartphones and introduces the state-of-the-art on detecting significant journeys. In Section III, we describe the benefits of using barometer sensors and identifies distinctive elevation signatures of significant journeys. Section IV presents an overview and the details of the proposed system. Real-world experiments are performed and the results are reported in Section V. We then conclude in Section VI.

II. Related Work

Vehicle positioning is a compelling research area even to date. Numerous works rely on sensors mounted in the car. Sensor fusion of accelerometer and magnetometer was performed to effectively estimate the vehicle trajectory [8]. To manage the non-line-of-sight conditions for GPS, an ultra wide-band (UWB) data were integrated with raw global navigation satellite systems data [9]. Hsu *et al.* utilized consistency check between clean measurements and multipath measurements to improve the positioning accuracy [10]. Vehicle motion sensors were integrated with differential GPS carrier-phase measurements to achieve high accuracy [11].

While a smartphone can be used as an interface to vehicle's sensors [12], modern smartphones are equipped with advanced sensors that are feasible for accurate estimation of vehicle trajectory. Smartphone-based position tracking systems can be classified largely into two categories: user location tracking and user trajectory tracking. User location tracking systems focus on localizing a phone, often without using the GPS module because it uses too much energy, whereas trajectory tracking systems are designed to identify trajectories of user locations.

Improving the energy efficiency and localization accuracy were primary design objectives of many user location tracking systems. A-loc dynamically adapts the energy efficiency depending on different accuracy requirements for varying user locations [13]. SensLoc uses contextual information about places for localization rather than tracking user's raw coordinates [3]. RAPS adapts the GPS duty cycle based on the location-time history of a user acquired from information sources such as cell-tower-RSS blacklisting, and accelerometers [14]. A number of techniques were designed to improve the energy efficiency including utilization of less power-intensive sensors, and adaptive adjustment of system-wide parameters [15]. Error-tolerant map matching and dead reckoning algorithms were designed to improve the localization accuracy [16]. SmartDC improves the energy efficiency based both on mobility learning/prediction, and adaptive adjustment of the duty cycle [17]. LEAP achieves increased energy efficiency by offloading delay-tolerant GPS signal processing to the cloud [18].

Compared with user location tracking systems, trajectory tracking systems are focused more on mining user's trajectory rather than localizing the current user location. A joint trajectory and position tracking system was proposed [19]. Multiple mechanisms were employed for efficient trajectory detection including trajectory simplification, efficient trajectory update protocols, and sensor management strategies. CAPS detects the user trajectory by comparing the sequence of cell-ID and GPS position pairs for a trajectory with the currently obtained cell-ID sequence [20]. CTrack enables trajectory estimation based on cellular base station (GSM) fingerprints combined with data collected from low-power sensors of a smartphone [21]. However, the radio interface to GSM radios is controlled by proprietary drivers limiting access to complete GSM fingerprints on most devices.

Nawaz and Mascolo introduced the new concept of *significant journeys* reinforcing the significance of user trajectory detection systems [5]. They developed a mobile system that uses only the accelerometer and gyroscope of a smartphone to identify significant journeys. However, the energy-efficiency

issue arises since the gyroscope sensor needs to be turned on continuously. This energy efficiency issue has been addressed by utilizing the extremely energy efficient barometer sensor of a smartphone [6], [7]. The key idea was that a driving path has distinctive elevation signatures that can be captured with only the barometer sensor.

More specifically, in our previous work [7], we developed a stand-alone mobile system based on the embedded barometer sensor of a smartphone that efficiently detects significant journeys. Compared with our previous work, this article makes the following contributions: (1) a more advanced algorithm to detect the beginning and end of a significant journey in various circumstances including convertibles, vehicles with open windows, *etc.*; (2) a hybrid algorithm that effectively copes with flat areas; (3) an algorithm to reduce the noise due to in-vehicle AC; (4) an energy consumption analysis for the proposed system.

III. Preliminaries

In this section, the barometer sensor of a smartphone is characterized compared with other types of sensors, and the distinctive elevation signatures of driving routes are demonstrated to motivate the use of the barometer sensor for identifying significant journeys.

A. Barometer Sensor of Smartphone

Advantages of using a barometer sensor over other types of sensors have been reported in [22]. Similarly, we aim to obtain the following advantages by using a barometer sensor as a core system component of the proposed system:

- **Position independence:** Compared with the state-of-the-art significant journey detection algorithm based on the accelerometer and gyroscope [5], a barometer-sensor-based approach is position independent. More specifically, it is not affected by the position and orientation of a phone, thus not requiring any additional means (*i.e.*, overheads) to offset the dependencies. In addition, a barometer sensor is not affected by frequent minor movements of a phone.
- **Simpler calibration:** The elevation *pattern* of a driving route is used, *i.e.,* relative changes in altitude values are used rather than absolute values, which cancel out the errors between subsequent measurements. In addition, driving routes typically last for a short period of time. Thus the impact of measurement errors due to varying weather conditions is not significant.
- **Low power:** The barometer sensor is extremely energy-efficient due to its low sampling rate and simpler processing requirement. Muralidharan *et al.* demonstrated that the barometer consumes between 3% to 23% less power than the accelerometer [23]. Caroll and Heiser reported the power consumption of different kinds of sensors on a Samsung Galaxy S3 (Figure 2) [24].

B. Distinctive Elevation Signatures of Driving Routes

Two sample routes denoted by Path A and Path B were chosen to demonstrate the distinctive elevation signatures of

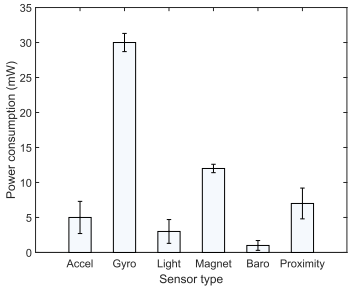

Fig. 2. Power consumption of various sensors on a Samsung Galaxy S3.

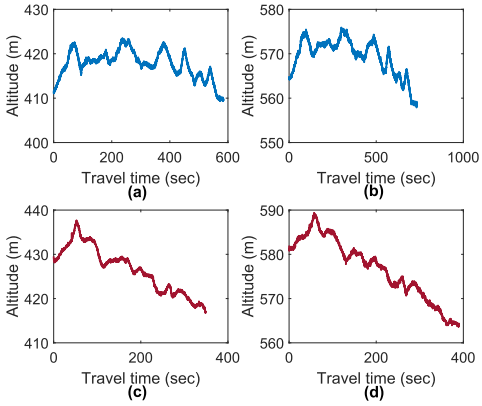

Fig. 3. Comparison of altitude values for Path A and Path B.

driving routes (See the map in Figure 15). Air pressure data of both paths were measured using Samsung Galaxy S6 with the sampling interval of 180ms. Figures 3(a) and 3(b) display the altitude data of Path A obtained in different days. Similarly, Figures 3(c) and 3(d) illustrate the altitude data of Path B. The measurement duration was different due to varying traffic conditions. It was also observed that the absolute altitude values were different for each measurement mainly due to varying weather conditions. An interesting observation was that distinctive elevation patterns were observed for both paths regardless of varying measurement duration and absolute altitude values. These unique elevation patterns of driving routes are the key motivation for developing the barometer-sensor based significant journey detection system.

IV. System Design

In this section, an overview of the proposed system is presented followed by the details of each system component.

A. System Overview

An overview of the proposed system is presented as a state diagram in Figure 4. The system has five main states: Power_ON, Matching, Calibration, Brief GPS, and Noise Removal states. When a phone is turned on, the system goes into the Power_ON state in which it continually checks if a journey has started (Section IV-B). If the system finds that a journey has started, the system state is transitioned to the Matching state. A main task in this state is to collect air pressure data in a window and perform pattern matching, *i.e.,* comparing the collected data with air pressure data sets that represent user-specified significant journeys (Section IV-C). This pattern matching process is continually executed

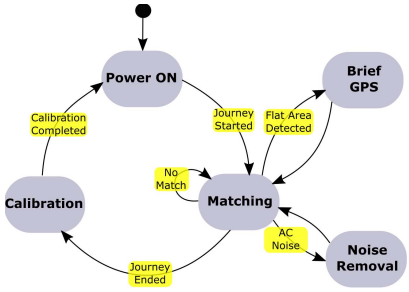

Fig. 4. An overview of the proposed system.

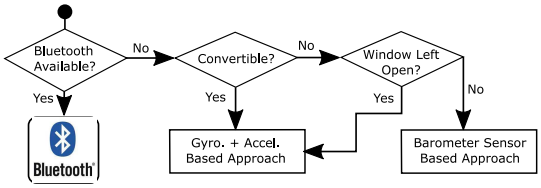

Fig. 5. Flow chart of algorithm to detect beginning and end of journey.

whenever the window is filled with air pressure data, and it is stopped when a journey is ended. In particular, if a flat area with limited variation of air pressure values is detected, the system instantaneously goes into the Brief GPS state to obtain GPS locations for a short period of time (3 seconds in our experimental setting), and it goes back to the Matching state (Section IV-D). In addition, noise in collected data is removed in the Noise Removal state, and goes back to the Matching state (Section IV-E). Finally, if the system finds that a journey has ended, the system state is changed to the Calibration state. In this state, if a significant journey is detected, air pressure data for the corresponding significant journey is calibrated by incorporating newly collected air pressure data (Section IV-F). This calibration process is needed to more accurately represent elevation signatures of significant journeys.

B. How to Determine the Beginning and End of a Journey?

Continually running a pattern matching algorithm to detect a significant journey consumes a lot of energy. Thus, we need to ensure that the pattern matching algorithm is executed only when a journey has started, and it should be stopped when a journey is completed. A naturally arising question is then: *how to detect the beginning and end of a journey in an energy efficient manner without involving user intervention*. In this section, we address this research question.

A key idea to detect the beginning and end of a journey is to leverage the in-vehicle Bluetooth communication system. More specifically, when a vehicle is started, a smartphone is connected to the vehicle's Bluetooth communication system and when the vehicle is turned off, it is disconnected from the system indicating the end of a journey. An advantage of this method is that not only drivers but also passengers are allowed to perform significant journey detection. In addition, to take into account vehicles with no Bluetooth system, we propose the following algorithm. The proposed algorithm is displayed as a flow chart in Figure 5. As shown, if the Bluetooth communication system is not available, users are provided with two options for detecting the beginning and end of a jorney, *i.e.,* the Gyroscope+Accelerometer-based approach

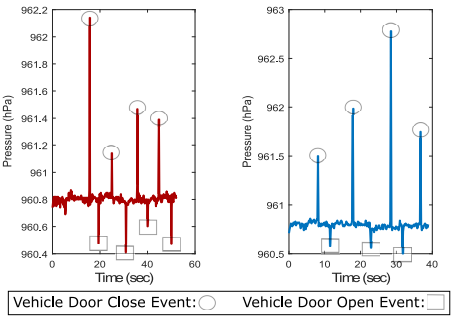

Fig. 6. Air pressure measurements for vehicle door close/open activities: Honda Pilot (left) and Volkswagen Passat (right).

and the Barometer-based mechanism. The former is used for convertible vehicles, and the latter is used for general vehicles. The reason for having two different approaches depending on whether the vehicle is a convertible or not is described in more details. Selection of appropriate approaches is presented to users as a questionnaire when the app is first started.

If the Bluetooth communication system is not available, we need to use different sensors to detect the event of the beginning or end of a journey. In order to enable continuous monitoring of this event, utilizing power-hungry sensors like gyroscopes and GPS modules is not a viable option. Thus, we develop an algorithm that uses only the barometer sensor to detect this event. The key idea is to exploit distinctive air pressure changes in the vehicle cabin that occur immediately after the vehicle door is closed or open. Vehicle cabins are sealed air tight to reduce noise entering from outside the vehicle; thus a pressure pulse wave is created by the vehicle-door close/open event [25]. Since perceptible air pressure increase lowers the perceived quality of a vehicle, air extractors are installed to relieve the air pressure increase [26]. This explains why only instantaneous air pressure spikes are observed.

Air pressure data (hPa) measured with Samsung Galaxy S6 are depicted in Figure 6. The sampling rate for the barometer sensor of the smartphone was 5Hz. The circles in the figure indicate a sharp increase of cabin air pressure when the vehicle-door-close event occurred. Also, as indicated by the squares in the figure, the cabin air pressure significantly decreased when the vehicle-door-open event occurred. Interestingly, despite the presence of the air control system of a vehicle that is designed to minimize the difference between the in-vehicle and outside air pressure, these sharp increases/decreases were reliably captured with a smartphone. Another interesting aspect is that this mechanism can be used not only for drivers but also for passengers regardless of their positions in the vehicle. Consequently, the door-close event is used as an indicator of the beginning of a journey, and the door-open event followed by no subsequent door-close events is counted as the end of a journey. Basically these vehicle door close and open events are used as the on/off switch for our pattern matching algorithm.

Given the results showing that the vehicle door close/open events are closely associated with sharp air pressure changes, a simple threshold-based approach is adopted to detect these events. The threshold is experimentally determined, and the detection accuracy as well as the false positive rates of the

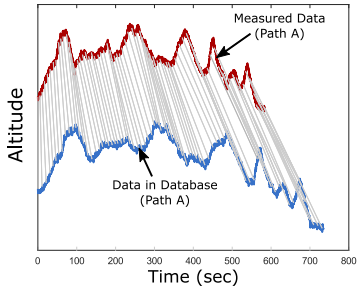

Fig. 7. Illustration of the DTW algorithm.

proposed approach are measured (Section V-B). A challenge is that the accuracy is significantly impacted, or the algorithm would not work at all, when a vehicle window is open. Although in most cases, users keep their vehicle windows closed when they leave their cars, it may not be always the case. In addition, there are convertible vehicles that make the barometer-based approach unusable. To handle these particular scenarios, users are provided with an option that allow them to detect the beginning and end of a journey using different sensors, *i.e.,* a combination of an accelerometer and gyroscope at the cost of increased energy consumption. The details of how to detect the event of entering and exiting a vehicle based on the combination of these two sensors are described in our previous work [27].

C. How to Match Pressure Data Streams With a Significant Journey?

Once the beginning of a journey is detected, the system state is changed to Matching and starts to run the pattern matching algorithm. To manage a large amount of incoming air pressure data in real time, a sliding window is used. More specifically, a sliding window of size WS is defined. When this window is filled with air pressure data, the collected data are compared with data sequences of significant journeys. A challenge in performing pattern matching is that two data sequences to be compared may have different sizes. We address this challenge by applying the Dynamic Time Warping (DTW) algorithm which effectively computes dissimilarity between two data sequences that have different sizes [28], [29]. The algorithm has been very useful in many areas of research that deal with pattern matching for data sequences of varying sizes (*e.g.,* bioinformatics [30], and gesture recognition [31]). Figure 7 illustrates an execution of the algorithm. The algorithm warps two data sequences optimally to perform matching and consequently computes the *DTW Distance* that represents dissimilarity between two data sequences. The DTW distance, denoted by $D(X, Y)$, between two data sequences $X = (x_1, x_2, \cdots, x_n)$ and $Y = (y_1, y_2, \cdots, y_m)$ is calculated as follows.

$$
\begin{aligned}
D(X, Y) &= f(n, m) \\
f(t, i) &= ||x_t - y_i|| + \min \begin{cases} f(t, i-1) \\ f(t-1, i) \\ f(t-1, i-1) \end{cases} \\
f(0, 0) &= 0, f(t, 0) = f(0, i) = \infty \\
&(t = 1, \cdots, n; i = 1, \cdots, m),
\end{aligned}
$$

where $||x_t - y_i||$ is the distance between two points x_t and y_i. Given the DTW distance, the problem of identifying a significant journey is defined as follows that is similar to the *Best-match query* which was first introduced by Sakurai *et al.* [32].

Definition 1: **(Best-match query problem)** Given Sequences X of length n and Y of length m, find the subsequence $X[t_s : t_e]$ whose DTW distance from Y is the smallest among those of all possible subsequences $X[t : j]$, that is, $D(X[t_s : t_e], Y) \leq D(X[t : j], Y)$ for any pair of $t = 1, \cdots, n$ and $j = t, \cdots, n$.

A straightforward method to solve this problem is to calculate the DTW distances for all subsequences, *i.e.*, $X[t_s : t_e]$. However, the time complexity of this approach is too high as $O(n^3 m)$. It is not an appropriate solution to run on smartphones with limited power. To address this challenge, we adopt a solution proposed in [32] in which the combination of "star-padding" and "subsequence time warping matrix" was used to dramatically reduce the space and time complexity. We refer readers to [32] for more details. Consequently, the DTW distances between a collected data sequence and that for each significant journey is computed, and significant journey(s) for which the DTW distance is smaller than a predefined threshold is considered as a match for this measured data sequence. An optimal threshold is experimentally determined in Section V-D.

D. How to Deal With Flat Areas?

Since the proposed system relies on elevation signatures of driving routes, the detection accuracy is significantly impacted by flat areas of routes. In this section, a hybrid approach is proposed to address this accuracy issue. The key idea is simple: when noticeable changes in measured air pressure data are not observed for a certain period of time, the GPS module of a smartphone is turned on momentarily to obtain GPS readings for a short duration of time. After these brief measurements of GPS locations, the GPS module is immediately turned off to avoid excessive energy consumption. These GPS locations are utilized as additional resources to decide a significant journey. Consequently, the hybrid approach improves the detection accuracy at the cost of slightly increased energy consumption due to the short-lived GPS measurements. The energy consumption analysis for various system components is presented in Section V-A.

Recall that users are required to drive along their desired significant routes to build their own air pressure database of significant journeys. More specifically, when users make their first trips along their significant routes, air pressure data are collected and saved in the database for corresponding significant journeys. In addition to these air pressure data, in order to run the hybrid algorithm, the GPS locations of significant journeys are also collected in the database during their initial trips. Consequently, given the air pressure data and the GPS locations in the database defined for each significant journey, momentarily measured GPS readings are used to support the pattern matching algorithm to increase the detection accuracy as illustrated in Figure 8. More specifically,

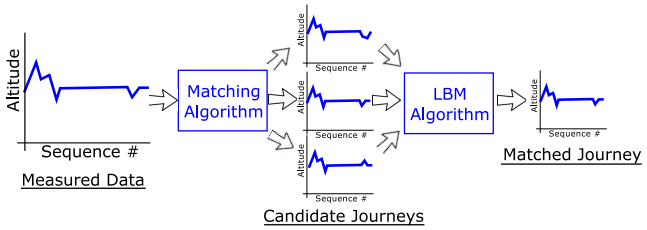

Fig. 8. Illustration of matching algorithm integrated with the LBM algorithm.

the matching algorithm selects candidate significant journeys based on measured air pressure data. It is possible that the matching algorithm outputs a number of candidate significant journeys especially if there are flat portions on the route. This is where the hybrid algorithm (*i.e.,* the Location-based Matching Algorithm) comes in to improve the accuracy of identifying the significant journey by using those brief GPS readings that were measured in flat areas.

The Location-based Matching (LBM) algorithm is described in Algorithm 1. The input for this algorithm is candidate significant journeys generated by the matching algorithm, and the brief GPS readings that were collected along with measured air pressure data (Line 1). The algorithm outputs a matching significant journey (Line 2).

Algorithm 1: Location-based Matching (LBM) Algorithm

1: Input: $L = \{l_0, l_1, \ldots, l_n\}$ // brief GPS readings , $\mathbb{X} = \{X_1, X_2, \ldots, X_m\}$ // candidate significant journeys
2: Output: X_k // matching significant journey
3: **for** $i = 0$ to $i = m$ **do**
4: $\delta_i = dist(L, X_i)$ // $dist(L, X_i)$ is the sum of the minimum distance from each $l_j \in L$ to X_i.
5: **end for**
6: $\text{argmin}_i(\delta_i)$, $0 \leq i \leq m$
7: return X_i

The LBM algorithm itself is simple. For each candidate significant journey X_i, the distance $dist(L, Xi)$ is calculated, where $L = \{l_0, ...l_n\}$ is the set of brief GPS readings (Lines 4-5). The distance $dist(L, X_i)$ is defined as follows:

$$dist(L, X_i) = \sum_{j=1}^{n} d(l_j, X_i).$$

where $d(l_j, X_i)$ is the Euclidean distance between a point l_j and the line segment X_i, more specifically a line segment consisting of a set of GPS locations for the significant journey X_i. The algorithm returns a significant journey X_i that has the smallest $dist(L, X_i)$ (Lines 6–7).

E. Noise From Airconditioner

In-vehicle AC injects air flows into the vehicle cabin, and it may affect the accuracy of air pressure measurement. To investigate the effect of the in-vehicle AC, we varied the AC power levels of a vehicle and drove along a significant route to measure the variations of air pressure. Experimental results show that there is an overall increase in the air pressure as we

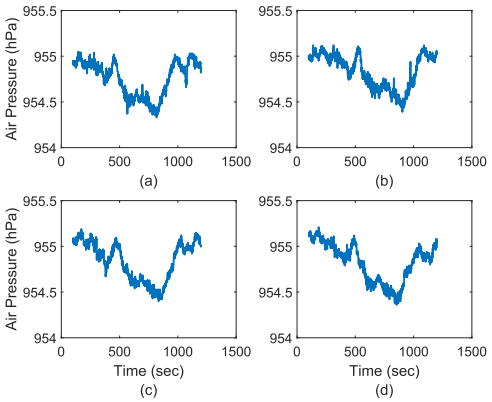

Fig. 9. Air pressure measurements for different AC power levels: (a) 1; (b) 2; (c) 3; and (d) 4.

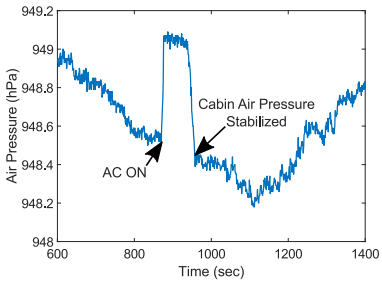

Fig. 10. A peak of air pressure when the AC power level is rapidly increased, or window of a fast running vehicle is open.

increase the AC power level (Figure 9). However, interestingly, it was also observed that regardless of the AC power level, the air pressure "pattern" was preserved.

Another notable observation was that a spike in the air pressure measurements appeared when the AC power level was rapidly increased (*e.g.,* AC power level from 0 to 4) as shown in Figure 10. This kind of a spike was not observed when we slightly increased the AC power level. We determine that such a spike would influence the accuracy of the significant journey detection because those spikes essentially alter the air pressure pattern. We propose an algorithm to remove such spikes. The key idea to deal with these spikes is that when the spike is negligibly small, such spike can be ignored; On the other hand, if a significantly high spike is observed, it is detected and removed from the air pressure data set.

Algorithm 2 describes the spike removal algorithm. Once air pressure data in a window denoted by $X = \{x_1, \ldots, x_n\}$ is collected, the data is quickly checked for any outlier due to the rapid increase of the AC power level as described in the overview of the system (Figure 4). The data is divided into k equal-sized air pressure data sets, *i.e.,* $X^1_{sub} \cup X^2_{sub} \cdots X^k_{sub}$ (Line 3). And then the mean absolute deviation (MAV) is calculated sequentially for each data set. If the calculated MAV for a data set X^i_{sub} is greater than a threshold λ, it indicates that there is an outlier. We then use the first air pressure measurement of X^i_{sub} as the beginning of the outlier data set, denoted by x_{begin} (Lines 6-9). The algorithm continues to calculate MAV for subsequent data sets, and identifies the last data set for which the MAV is greater than the threshold λ.

Algorithm 2: Air Pressure Spike Removal Algorithm

1: Input: $X = \{x_1, x_2, \ldots, x_n\}$ // air pressure data in a window.
2: Output: $Y = \{y_1, y_2, ..., y_m\}$ // air pressure data with no spikes.
3: $X \leftarrow X^1_{sub} \cup X^2_{sub} \cdots X^k_{sub}$
4: $flag \leftarrow FALSE$
5: **for** $i = 0$ to $i = k$ **do**
6: **if** $MAV(X^i_{sub}) > \lambda$ && $flag == TRUE$ **then**
7: $x_{begin} = x_1$ of X^i_{sub}
8: $flag \leftarrow FALSE$
9: **end if**
10: **if** $MAV(X^i_{sub}) > \lambda$ && $flag == FALSE$ **then**
11: $x_{end} = x_1$ of X^i_{sub}
12: **end if**
13: **end for**
14: **if** $x_{begin} \neq 0$ && $x_{end} \neq 0$ **then**
15: $X_{interpolate} \leftarrow$ Interpolate(x_{begin}, x_{end})
16: **end if**
17: $Y \leftarrow X - \{x_{begin} \sim x_{end}\} \cup X_{interpolate}$
18: Return Y

This time, the algorithm sets the first air pressure measurement of this data set as the end of the outlier data set denoted by x_{end} (Lines 10-12). If both x_{begin} and x_{end} have been successfully identified, we perform a simple interpolation to connect the two points since such a spike lasts only for a short duration of time, consequently obtaining a air pressure data set with the outlier eliminated (Lines 14–18).

F. Is Initial Data Set for Significant Journey Accurate?

Users are provided with options on how to build elevation data sets for significant journeys. More specifically, they are allowed to make a trip along their significant routes, or they can use the global digital elevation models (DEMs), *e.g.*, Google Maps Elevation API [33]. A challenge is that these methods do not guarantee for users to obtain the ground truth elevation data. The accuracy of latest DEMs was evaluated, *e.g.,* the Advanced Spaceborne Thermal Emission Reflectometer DEM (ASTER GDEM2), and DEMs based on the Shuttle Radar Topography Mission (SRTM) [34]. The results show that most DEMs have high root mean square height errors between 4∼9m. And obviously, user-obtained elevation data are not always accurate. We attempt to address this challenge based on the results of existing research that demonstrated the relationship between repeated measurements of barometer sensor data and the estimation accuracy of a road grade [35], [36]. More specifically, a mechanism is proposed that continually calibrates the initial air pressure data sets for significant journeys as users make more journeys along their significant routes.

Suppose $X = (x_1, \cdots, x_m)$ be a air pressure data sequence of size m that needs to be compared with a data set of size n for a significant journey denoted by $Y^{\text{FULL}} = (y_1, \cdots, y_n)$. The significant journey calibration algorithm is depicted in Algorithm 3. In this algorithm, the data sequence X that was

Algorithm 3: Significant Journey Calibration Algorithm

1: Input: frame size $\leftarrow \Delta$, $X = (x_1, \cdots, x_m)$, $Y^{\text{FULL}} = (y_1, \cdots, y_n)$
2: Output: $\overline{Y} = \overline{Y}_1 \cup \overline{Y}_2 \cdots \cup \overline{Y}_s$
3: $Y \leftarrow \text{DTW}(X, Y^{\text{FULL}})$
4: $\{\overset{\triangledown}{X}, \overset{\triangledown}{Y}\} \leftarrow \text{BMF}(X, Y, \Delta)$, where
$\overset{\triangledown}{X} = \overset{\triangledown}{X}_1 \cup \overset{\triangledown}{X}_2 \cup \cdots \cup \overset{\triangledown}{X}_s$, $X \subseteq \overset{\triangledown}{X}$
$\overset{\triangledown}{Y} = \overset{\triangledown}{Y}_1 \cup \overset{\triangledown}{Y}_2 \cup \cdots \cup \overset{\triangledown}{Y}_s$, $Y \subseteq \overset{\triangledown}{Y}$
$|\overset{\triangledown}{X}_i| = |\overset{\triangledown}{Y}_i| = \Delta$, $i \in \{1, 2, \cdots, s\}$
5: **for** $i = 0$ to $i = s$ **do**
6: $\overline{Y}_i \leftarrow \text{avg}(\overset{\triangledown}{X}_i, \overset{\triangledown}{Y}_i)$
7: **end for**

identified as a significant journey is integrated into the original data set for the significant journey Y^{FULL}, resulting in a new data set $\overline{Y}$. First, the DTW algorithm is used to find the subsequence Y of Y^{FULL} that is closest to the input sequence X (Line 3) in terms of the DTW distance. This process ensures that only the necessary part of the original data set for the significant journey is calibrated. Once a subsequence Y is found, both X and Y are divided into frames of size Δ, which is done in the BMF function. This function then determines a frame of Y for each frame of X such that the DTW distance is the smallest (Line 4). This way the algorithm prepares s pairs of frames that can be averaged for calibration. Each pair of frames (*i.e.,* $\overset{\triangledown}{X}_i$ and $\overset{\triangledown}{Y}_i$) is averaged to form a calibrated segment $\overline{Y}_i$ for $\overline{Y}$ (Lines 5–6). Consequently, the calibrated segment $\overline{Y}$ is integrated with the original data sequence Y^{FULL}. Since this algorithm is executed after a journey is completed, it does not influence the real-time performance of the proposed system.

V. Experimental Results

We conducted experiments to evaluate the performance of WiTraffic, which was implemented on Samsung Galaxy S6 (Samsung Galaxy Nexus for energy consumption measurements). These experiments were carried out in the city of Brookings, SD, United States, which is located in a fairly flat area according to the elevation data for counties [37]. Air pressure data were collected from more than 150 miles of roads in this city.

We first performed a microbenchmark on the power consumption of WiTraffic to evaluate its energy efficiency. We then selected experimentally system parameters, *i.e.,* the optimal window size and threshold. Based on selected system parameters, we measured the detection accuracy for seven different significant journeys with varying in-vehicle AC power levels. We then analyzed the effect of the proposed calibration scheme on the detection accuracy, followed by the performance analysis of the execution time of the matching algorithm on different smartphones. Experiments were concluded with the performance evaluation of the proposed hybrid algorithm designed for flat areas of the city. More specifically, the performance of our hybrid solution was compared with a non-hybrid approach which is based only on the barometer sensor [7]. The results indicated

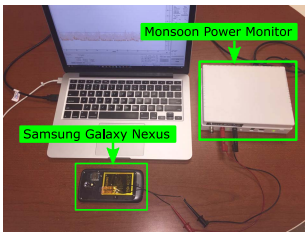

Fig. 11. Experimental setting for power measurement.

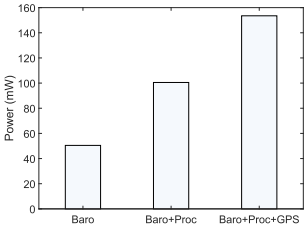

Fig. 12. Power consumption of the system components.

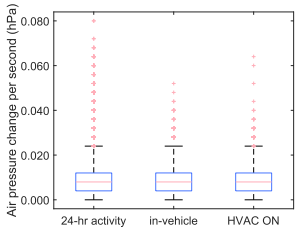

Fig. 13. Air pressure difference between subsequent measurements for Samsung Galaxy S6.

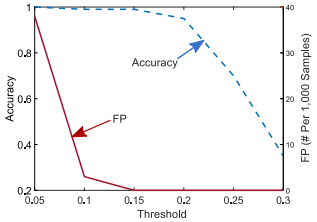

Fig. 14. Accuracy of vehicle door event detection.

that the proposed hybrid approach improved the average detection accuracy (95.6% to 97%) and the false positive rates (2.3% to 1.5%) in our experimental settings.

A. Energy Consumption

We used the Monsoon power monitor [38] to evaluate the energy efficiency of the proposed system. Since the battery of Galaxy S6 was not removable, we used Samsung Galaxy Nexus for this experiment. Figure 11 displays our experimental setting. In this experiment, we used the window size of 130 seconds in accordance with the results presented in Section V-C. To simulate the execution of the hybrid algorithm in the lab environment, the GPS module was periodically and briefly turned on every 1 minute and GPS locations were obtained for 3 seconds. We ran the system for 10 minutes and measured the average power consumption of the proposed system.

The results are depicted in Figure 12. The average power consumption for the barometer sensor was 50mW. During this 10-min experiment, the matching algorithm was executed 4 times, and the average power consumption for that was 50.2mW exhibiting a noticeably small amount of energy consumption, comparable to that for the barometer sensor. Regarding momentary measurements of GPS locations, the average power consumption was 53mW. We observed that these brief GPS readings consumed much lower energy compared with the scenario where the GPS module continuously operates (*i.e.,* about 150mW). Consequently, the total power consumption to run the system extensively for 10 minutes (*i.e.,* "extensive" in that a flat area was detected every minute) was 153.2mW, which is reasonably low power consumption enough to continuously operate the proposed system.

B. Accuracy of Vehicle Door Event Detection

In this section, we evaluate the performance of the proposed algorithm that is designed to detect the beginning and end of a journey. Recall that the algorithm utilizes either of the in-vehicle Bluetooth communication system, accelerometer+gyroscope, or barometer sensor depending on different use cases. In particular, we have reported the accuracy of the accelerometer+gyroscope-based approach for detecting the event of driver entering/existing a vehicle in our previous paper [27]. This section focuses on the performance evaluation of the barometer-sensor-based approach.

The vehicle door/close event is detected if the air pressure difference between subsequent measurements is greater than a predefined threshold. To determine an appropriate threshold, we measured the air pressure difference from a number of general environments *i.e.,* in a stationary/moving vehicle, in a vehicle with AC on, and 24-hour daily activity at home. The results obtained with Samsung Galaxy S6 are depicted in Figure 13. These results coincide with the literature [23] in that short-term air pressure variations (within tens of seconds) are reasonably small. More precisely, air pressure measured every 180ms gave very small variations. It was observed that the barometer sensor of Samsung Galaxy S6 maintained a majority of air pressure differences being smaller than 0.05hPa. Consequently, we tested the event detection accuracy with the threshold greater than 0.05hPa.

We then measured the accuracy of the vehicle door event detection by varying the threshold. Specifically, we sat in a stionary/moving vehicle carrying a smartphone and repeatedly performed door open/close actions 100 times. The event detection accuracy was defined as:

$$\text{Accuracy} = \frac{\text{the number of successful event detection}}{\text{the total number of door close/open actions}}.$$

In addition, the false positive rates (FP) were measured by counting the number of false detection while continuously driving along Path A and Path B. Specifically, FP was defined as the number of false detection per 1000 samples. The results are depicted in Figure 14. A small threshold of 0.05 resulted in very high accuracy at the cost of high false positive rates of about 40 instances per 1,000 samples. Consequently, it was

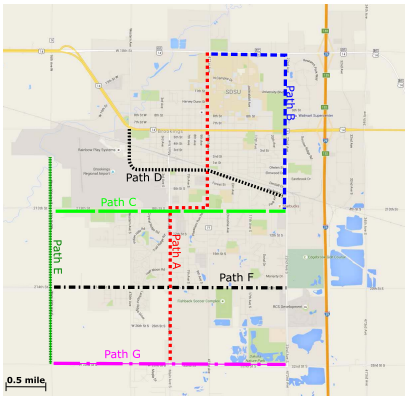

Fig. 15. Selected driving routes for evaluation of the proposed system.

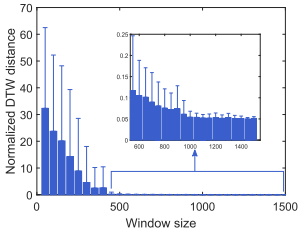

Fig. 16. Effect of the window size on the detection accuracy.

determined that a threshold of 0.15 resulted in high detection accuracy with nearly 0 FP instances in our experimental setting.

C. Determining Window Size

To evaluate the accuracy of significant journey detection, an appropriate window size needs to be determined. In this experiment, we collected four different air pressure data sets for Path A and Path B, respectively (Figure 15). One of the four air pressure data sets for each path was used to represent a significant journey.

To determine an appropriate window size, we used the *normalized DTW distance* (*i.e.*, DTW distance divided by the length of a matching subsequence) by varying the window size. More specifically, we randomly and uniformly extracted 1,000 subsequences from each data set. The normalized DTW distance for each subsequence was calculated by performing pattern matching between the subsequence and its corresponding significant journey. The results are depicted in Figure 16. It was observed that the normalized DTW distance decreased as the window size increased because there were more air pressure samples with then increased window size. The results also indicated that too small window size (*e.g.*, smaller than 500) is not appropriate since it resulted in many false alarms. On the other hand, a window size must not be too large because it increases the delay for significant journey detection. From our experiments, we observed that the window size of 650 to 700 yielded adequate results. A window size of 650 means air pressure measurements for about 130 seconds, *i.e.*, our pattern matching algorithm is first performed 130 seconds after a journey is started.

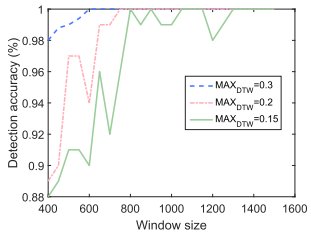

Fig. 17. Effect of threshold on the detection accuracy.

D. Determining Threshold

A significant journey is detected if the calculated normalized DTW distance is smaller than a threshold denoted by $\mathsf{MAX}_{\mathsf{DTW}}$. Also note that multiple candidate significant journeys can be selected by the algorithm. Similar to the experiment that was performed for determining the window size, we randomly and uniformly selected 1,000 subsequences with varying window sizes from each data set for Path A and Path B. These subsequences were compared with their corresponding significant journeys, and the detection accuracy was obtained (See Section V-E for the definition of the detection accuracy). We repeated this experiment by varying the threshold. Figure 17 displays the results. We observed that regardless of the threshold, small window sizes (*e.g.*, smaller than 500) resulted in low detection accuracy. It was also observed that too small thresholds (*e.g.*, < 0.2) resulted in degraded accuracy, and if the threshold is too high, it increases the false alarm rates. Consequently, in this experiment, we noted that a threshold of 0.3 gave high accuracy while maintaining the false alarm rates low.

E. Detection Accuracy

This section evaluates the detection accuracy of the proposed system. For this set of experiments, elevation data for 5 more driving routes were used. These routes are depicted in in Figure 15 labeled as Path C,..., Path G . Elevation data sequences of varying window sizes were randomly extracted from Path A and Path B. The extracted data were matched with significant journeys for all routes to measure the **detection accuracy** and **false positive rates (FP)**. In particular, a false positive was recorded if a significant journey has been detected but it was neither Path A nor Path B. More specifically, the detection accuracy and the false positive rates are defined as follows.

$$\text{Detection Accuracy} = \frac{TS - NF - NN}{TS}.$$

$$\text{False Positive Rates} = \frac{NF}{TS}.$$

Here TS represents the total number of data sequences that were compared with significant journeys; NF indicates the total number of false positives; and NN is the total number of none-detection.

Figures 18 and 19 display the results for the extracted data sequences compared with the significant journeys for Path A and Path B respectively by varying the window size. It was observed that both the detection accuracy and the false

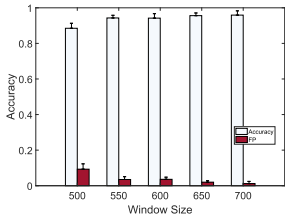

Fig. 18. Detection accuracy and false positive rates for Path A.

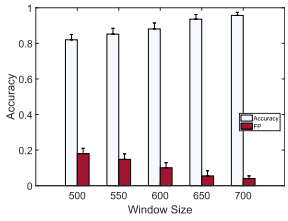

Fig. 19. Detection accuracy and false positive rates for Path B.

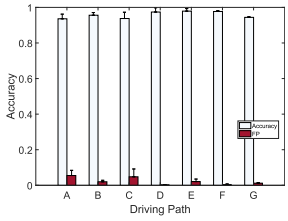

Fig. 20. Detection accuracy and false positive rates for all paths.

positive rates were gradually improved as the window size was increased. The results coincide with what was presented in Section V-C. With the selected window size of 650, the detection accuracy was about 95%, and the false positive rates were around 1%∼5%. The same experiment was performed with the selected window size for other 5 driving routes. Figure 20 depicts the results. It was observed that the average detection accuracy of all paths was 95.7% and the average false positive rates were 2.3%. Overall, these experimental results indicate that the proposed system has high detection accuracy even though the test site (*i.e.,* the city of Brookings) is located in a relatively flat area [37]. The accuracy will be further improved with the proposed hybrid algorithm that is designed to cope with flat areas. Experimental results for the hybrid algorithm are presented in Section V-I.

F. Effect of AC Noise

In this section we evaluate the effect of AC noise on the performance of the proposed algorithm. We selected Path A for this experiment. We collected air pressure data on this path for each AC power level, *i.e.,* 1, 2, 3, and 4. Figure 21 displays the accuracy results. Although the AC was turned on, the detection accuracy was fairly high. The reason is because significant journeys are detected based on air pressure patterns rather than absolute air pressure values. Furthermore, any spikes due to rapid adjustment of AC power are effectively eliminated by the spike removal algorithm. Another observation was that the accuracy slightly decreased with higher AC power levels.

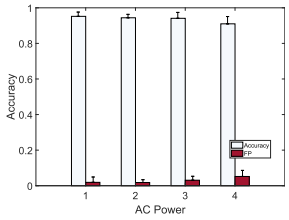

Fig. 21. Effect of noise due to AC on detection accuracy.

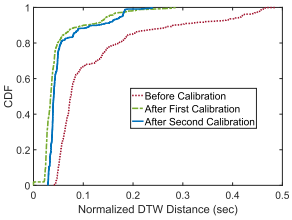

Fig. 22. CDF of DTW distance before and after calibration.

Similarly, the false positive rates were marginally increased with increased AC power levels. Overall, the results indicate acceptable accuracy and false positive rates in the presence of noise due to the in-vehicle AC.

G. Effect of Calibration of Significant Journey

Once a significant journey is detected, the corresponding air pressure data are incorporated into the detected original data set for the significant journey for data calibration purpose. This section evaluates the effect of this data calibration. For this experiment, air pressure data sets for Path A and Path B were used, and the normalized DTW distances for 1,000 randomly and uniformly generated data sequences of size 500 were calculated for both before and after calibration. Figure 22 displays the results. Noticeable improvement in the normalized DTW distance was observed when the first calibration was completed decreasing the normalized DTW distance by up to 53%. This improvement diminished for the second calibration decreasing the normalized DTW distance by up to 15%.

H. Execution Time of Pattern Matching Algorithm

One of the major concerns for the proposed system is the execution time of the DTW matching algorithm. Even though we integrated an algorithm with very fast processing speed [32], the execution time on a smartphone is unknown. We wrote Java code for the matching algorithm that was originally written in the C programming language to run the proposed system on an Android platform. We then measured the execution time for running the matching algorithm on Samsung Galaxy S6 which was equipped with Android OS Lollipop v.5.0, Quad-core 1.5 GHz Cortex-A53 processor, and 3GB of RAM. The algorithm ran 100 times for different window sizes. In particular, based on the accuracy experiments, we used window sizes > 500 to measure the execution time. Figure 23 depicts the results. The average execution time for Galaxy S6 was 0.09, 0.11, and 0.13 seconds for window sizes

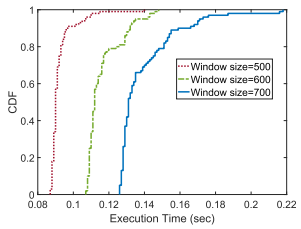

Fig. 23. CDF of execution time on Samsung Galaxy S6.

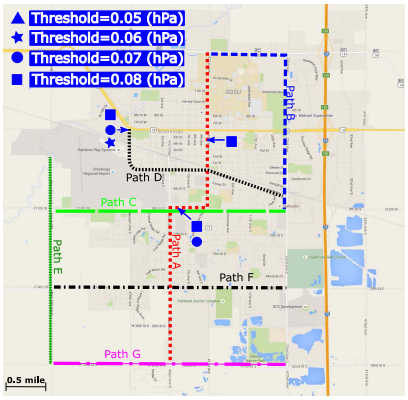

Fig. 24. Locations where brief GPS readings were made.

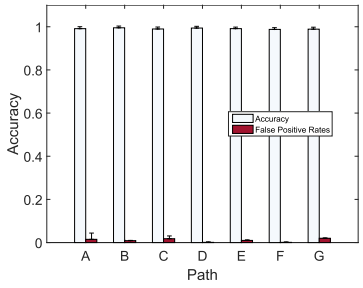

Fig. 25. Detection accuracy after applying the hybrid algorithm.

of 500, 600, and 700, respectively. More specifically, when we increased the window size by 100, we observed about 20% increase in the execution time.

I. Improving Accuracy on Flat Areas

The performance of the hybrid algorithm for flat areas is evaluated. Recall that the GPS locations are collected when air pressure data do not change noticeably. Based on the window size determined in Section V-C, the standard deviation for air pressure data in a window is measured when a window is filled with air pressure data. The calculated standard deviation was compared with varying thresholds, and it was determined to turn on the GPS module. Figure 24 shows the points where the GPS module was briefly turned on with varying thresholds. These flat areas are suspected to have affected the accuracy results presented in Section V-E. Figure 25 depicts the accuracy results after applying the hybrid algorithm. The results show that in comparison with the non-hybrid solution the mean detection accuracy increased from 95.7% to 97%, and the false positive rates decreased from 2.3% to 1.5%.

VI. Conclusion

We have presented the design, implementation, and evaluation of a barometer sensor-based mobile system that detects significant journeys. The proposed system is integrated with a hybrid algorithm that adaptively utilizes GPS measurements to handle flat areas in order to increase the detection accuracy in flat areas. The proposed system has potential to be integrated with other intelligent transportation applications. For example, the significant journeys of vehicles can be shared with other vehicles via vehicle-to-vehicle (V2V) communication, allowing them to derive optimal real-time driving routes computed based on those significant journeys. It is also possible that the proposed system can be integrated with V2V-based traffic control systems [39], [40] to reduce traffic jams.

References

[1] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[2] J. Lu *et al.*, "The smart thermostat: Using occupancy sensors to save energy in homes," in *Proc. 8th ACM Conf. Embedded Netw. Sens. Syst.*, 2010, pp. 211–224.

[3] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava, "SensLoc: Sensing everyday places and paths using less energy," in *Proc. 8th ACM Conf. Embedded Netw. Sensor Syst.*, 2010, pp. 43–56.

[4] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 481–490.

[5] S. Nawaz and C. Mascolo, "Mining users' significant driving routes with low-power sensors," in *Proc. 12th ACM Conf. Embedded Netw. Sens. Syst.*, 2014, pp. 236–250.

[6] B.-J. Ho, P. Martin, P. Swaminathan, and M. Srivastava, "From pressure to path: Barometer-based vehicle tracking," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy-Efficient Built Environ.*, 2015, pp. 65–74.

[7] M. Won, S. Zhang, A. Chekuri, and S. H. Son, "Enabling energy-efficient driving route detection using built-in smartphone barometer sensor," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2378–2385.

[8] R. Hostettler, W. Birk, and M. L. Nordenvaad, "Joint vehicle trajectory and model parameter estimation using road side sensors," *IEEE Sensors J.*, vol. 15, no. 9, pp. 5075–5086, Sep. 2015.

[9] H. M. Georges, Z. Xiao, and D. Wang, "Hybrid cooperative vehicle positioning using distributed randomized sigma point belief propagation on non-Gaussian noise distribution," *IEEE Sensors J.*, vol. 16, no. 21, pp. 7803–7813, Nov. 2016.

[10] L.-T. Hsu, H. Tokura, N. Kubo, Y. Gu, and S. Kamijo, "Multiple faulty GNSS measurement exclusion based on consistency check in urban canyons," *IEEE Sensors J.*, vol. 17, no. 6, pp. 1909–1917, Mar. 2017.

[11] T. B. Karamat, M. M. Atia, and A. Noureldin, "Performance analysis of code-phase-based relative GPS positioning and its integration with land vehicle's motion sensors," *IEEE Sensors J.*, vol. 14, no. 9, pp. 3084–3100, Sep. 2014.

[12] M. Reininger, S. Miller, Y. Zhuang, and J. Cappos, "A first look at vehicle data collection via smartphone sensors," in *Proc. IEEE Sens. Appl. Symp. (SAS)*, Apr. 2015, pp. 1–6.

[13] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, 2010, pp. 285–298.

[14] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based positioning for smartphones," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, 2010, pp. 299–314.

[15] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, 2010, pp. 315–330.

[16] X. Zhu, Q. Li, and G. Chen, "APT: Accurate outdoor pedestrian tracking with smartphones," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2508–2516.

[17] Y. Chon, E. Talipov, H. Shin, and H. Cha, "SmartDC: Mobility prediction-based adaptive duty cycling for everyday location monitoring," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 512–525, Mar. 2014.

[18] H. S. Ramos, T. Zhang, J. Liu, N. B. Priyantha, and A. Kansal, "LEAP: A low energy assisted GPS for trajectory-based services," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 335–344.

[19] S. Bhattacharya, H. Blunck, M. B. Kjærgaard, and P. Nurmi, "Robust and energy-efficient trajectory tracking for mobile devices," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 430–443, Feb. 2014.

[20] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-ID sequence matching," in *Proc. 9th Int. Conf. Mobile Syst., Appl., Services*, 2011, pp. 293–306.

[21] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proc. 8th USENIX Symp. Netw. Syst. Design Implement.*, 2011, pp. 267–280.

[22] K. Sankaran, M. Zhu, X. F. Guo, A. L. Ananda, M. C. Chan, and L.-S. Peh, "Using mobile phone barometer for low-power transportation context detection," in *Proc. 12th ACM Conf. Embedded Netw. Sens. Syst.*, 2014, pp. 191–205.

[23] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal, "Barometric phone sensors: More hype than hope!" in *Proc. 15th Workshop Mobile Comput. Syst. Appl.*, 2014, p. 12.

[24] A. Carroll and G. Heiser, "The systems hacker's guide to the galaxy energy usage in a modern smartphone," in *Proc. 4th Asia–Pacific Workshop Syst.*, 2013, p. 5.

[25] D. A. Wade, P. Elia, J. A. Marleau, P. Repp, and R. E. Newton, "Vehicle door closure cabin pressure relief system," U.S. Patent 9 656 534, May 23, 2017.

[26] K. L. McCarthy and S. Gumate, "Air extractors," U.S. Patent 2010 0 216 384, Aug. 26, 2010.

[27] H. Park, D.-H. Ahn, M. Won, S. H. Son, and T. Park, "Poster: Are you driving?: Non-intrusive driver detection using built-in smartphone sensors," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 397–400.

[28] H. Sakoe and S. Chiba, "A dynamic programming approach to continuous speech recognition," in *Proc. 7th Int. Congr. Acoust.*, vol. 3. 1971, pp. 65–69.

[29] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.

[30] F. Hermans and E. Tsiporkova, "Merging microarray cell synchronization experiments through curve alignment," *Bioinformatics*, vol. 23, no. 2, pp. e64–e70, 2007.

[31] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive Mobile Comput.*, vol. 5, no. 6, pp. 657–675, 2009.

[32] Y. Sakurai, C. Faloutsos, and M. Yamamuro, "Stream monitoring under the time warping distance," in *Proc. IEEE 23rd Int. Conf. Data Eng. (ICDE)*, Apr. 2007, pp. 1046–1055.

[33] *Google Maps Elevation API*. Accessed on May 11, 2015. [Online]. Available: https://developers.google.com/maps/

[34] M. Rexer and C. Hirt, "Comparison of free high resolution digital elevation data sets (ASTER GDEM2, SRTM v2.1/v4.1) and validation against accurate heights from the Australian national gravity database," *Austral. J. Earth Sci.*, vol. 61, no. 2, pp. 213–226, 2014.

[35] B. Y. Boroujeni and H. C. Frey, "Road grade quantification based on global positioning system data obtained from real-world vehicle fuel use and emissions measurements," *Atmos. Environ.*, vol. 85, pp. 179–186, Mar. 2014.

[36] B. Y. Boroujeni, H. C. Frey, and G. S. Sandhu, "Road grade measurement using in-vehicle, stand-alone gps with barometric altimeter," *J. Transp. Eng.*, vol. 139, no. 6, pp. 605–611, 2013.

[37] *Topographic Prominence*. Accessed on Nov. 20, 2015. [Online]. Available: http://www.cohp.org/prominence/

[38] *Monsoon Power Monitor*. Accessed on Nov. 20, 2015. [Online]. Available: https://www.msoon.com/LabEquipment/PowerMonitor/

[39] M. Won, T. Park, and S. H. Son, "FuzzyJam: Reducing traffic jams using a fusion of fuzzy logic and vehicular networks," in *Proc. IEEE 17th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1869–1875.

[40] M. Won, T. Park, and S. H. Son, "Lane-level traffic jam control using vehicle-to-vehicle communications," in *Proc. IEEE 17th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 2068–2074.

Myounggyu Won (M'13) received the Ph.D. degree in computer science from Texas A&M University at College Station in 2013. He was a Post-Doctoral Researcher at the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology, South Korea. He is currently an Assistant Professor with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD, USA. His research interests include wireless mobile systems, vehicular ad hoc networks, intelligent transportation systems, and wireless sensor networks. He received the Graduate Research Excellence Award from the Department of Computer Science and Engineering, Texas A&M University, in 2012.

Ashutosh Mishra (S'17) received the B.S. degree in electrical, electronics and communications engineering from Rajiv Gandhi Prodyogiki Vishwavidyalaya. He is currently pursuing the M.S. degree in computer science with South Dakota State University. His research interests are in mobile computing and wireless sensor networks.

Sang H. Son (M'85–SM'98–F'13) received the B.S. degree in electronics engineering from Seoul National University, the M.S. degree from KAIST, and the Ph.D. degree in computer science from the University of Maryland at College Park, College Park. He is currently the Dean of the Graduate School and Director of the CPS Global Center, Daegu Gyeongbuk Institute of Science and Technology. He was a Professor at the Computer Science Department, University of Virginia, and WCU Chair Professor at Sogang University. He was a Visiting Professor with KAIST, the City University of Hong Kong, the Ecole Centrale de Lille, France, Linkoping University, Sweden, and the University of Skövde, Sweden.

His research interests include cyber physical systems, real-time and embedded systems, database and data services, and wireless sensor networks. He has authored or co-authored over 300 papers and edited/authored four books in these areas. He is an IEEE Fellow and a member of both the Korean Academy of Science and Technology and the National Academy of Engineering of Korea. He has served as the Chair of the IEEE Technical Committee on Real-Time Systems during 2007–2008. He is a Founding Member of the ACM/IEEE CPS Week, and serving as a member of the Steering Committee for RTCSA, Cyber Physical Systems Week, and SEUS. He is serving as an Associate Editor of the *ACM Transactions on Cyber Physical Systems*, and has served on the Editorial Board of the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and *Real-Time Systems Journal*. He received the Outstanding Contribution Award form Cyber Physical Systems Week in 2012. His research has been funded by the Korean Government, the National Research Foundation, the U.S. National Science Foundation, DARPA, the Office of Naval Research, the Department of Energy, the National Security Agency, and IBM.