Robust and Accurate Smartphone-Based Step Counting for Indoor Localization

Fuqiang Gu, *Student Member, IEEE*, Kourosh Khoshelham, Jianga Shang, *Member, IEEE*, Fangwen Yu, and Zhuo Wei

***Abstract*— Robust and accurate step counting is important for indoor localization algorithms that rely on smartphone inertial sensors. Existing solutions for step counting do not consider users' false walking state (e.g., when a user in still state uses her phone for texting, playing games, and watching movies), which results in the over-counting problem. In this paper, we propose a robust and accurate step counting algorithm to solve the overcounting problem caused by false walking. Experimental results show that the proposed algorithm outperforms the commonly-used peak detection-based method and can improve the step counting accuracy by 6.56% for normal walking case, 9.54% for free walking case, and by 58.92% for false walking case, which is a significant improvement in the accuracy and robustness. We also compare the proposed method with popular commercial step counting applications including S Health, i-Health, and Pedometer++, which shows that our method can achieve better accuracy.**

***Index Terms*— Indoor localization, dead reckoning, user motion state recognition, step counting, step detection, smartphones.**

I. Introduction

INDOOR localization has attracted much attention from both academia and industry because of its widespread applications such as shopping guide, location-based social networks, and smart car parking [1]–[3]. Pedestrian Dead Reckoning (known as PDR) is one of the most popular indoor localization methods since it can provide real-time locations of a user or object given an initial location, and does not rely on any infrastructure. The PDR method uses the readings of inertial sensors, which have been built in most modern smartphones, to compute the heading and step length of the user in order to estimate the location in real time.

One of the big challenges of the PDR method is accurate and robust estimation of the number of steps that a user has taken. When a user walks normally along a path, it is not difficult to obtain a relatively accurate estimation of steps. However, if a user conducts a period of false walking (which means that a user in still state uses her phone for texting, watching movies, playing games, etc.) apart from normal walking, it is challenging to obtain a robust and accurate estimation of steps. The accuracy of step counting has a direct influence on the accuracy of indoor localization methods relying on inertial sensors and many other phone applications such as health and fitness related applications.

Although a lot of research work has been done in the improvement of step counting algorithms, existing step counting solutions tend to assume that the phone is put in a fixed position or the user does not change her motion states (e.g., changing from walking to still, and then to walking), which is not always the case in real world. The researchers in [4] have considered the influence of different phone carrying ways on the performance of step counting algorithms, but they did not consider the false walking problem (which means that the counter reports a step event while the user does not walk in reality). The lack of consideration for false walking would undoubtedly lead to the over-counting problem.

To address this problem, we propose a robust and accurate step counting algorithm, which is based on the peak detection method. However, conventional peak detection-based methods count steps by simply counting peaks detected, which only use the temporal constraint to remove false peaks that do not accurately reflect users' steps. They cannot work well when dealing with false walking case. In this study, we first analyze the step features, namely periodicity, similarity, and continuity. Then, by putting constraints on these features, we can remove noisy data caused by false walking, thereby improving the accuracy of step counting.

II. Related Work

Step counting can be done by using camera [5], accelerometer [6]–[8], commercial pedometers [9], etc. Here we focus on the step counting methods using the built-in smartphone accelerometer, which can be mainly categorized as: threshold setting [10], [11], peak detection [12], [13], correlation analysis [4], [14] or spectral analysis.

The threshold-based methods work by judging whether the accelerometer readings satisfy a certain threshold condition, which further include simple threshold methods and zero velocity update (ZUPT) methods. The former detects a step

Manuscript received January 25, 2017; accepted March 13, 2017. Date of publication March 22, 2017; date of current version May 5, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 41271440 and Grant 61402199, and in part by the China Scholarship Council-University of Melbourne Research Scholarship under Grant CSC201408420117. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ferran Reverter.

F. Gu and K. Khoshelham are with the Department of Infrastructure Engineering, University of Melbourne, Parkville VIC 3010, Australia (e-mail: fuqiangg@student.unimelb.edu.au; k.khoshelham@unimelb.edu.au).

J. Shang and F. Yu are with the Faculty of Information Engineering, China University of Geosciences and the National Engineering Research Center for Geographic Information System, Wuhan 430074, China (e-mail: jgshang@cug.edu.cn; yufangwen@cug.edu.cn).

Z. Wei is with the Huawei's Shield Laboratory, International Pte Ltd., Singapore 117674 (e-mail: wei.zhuo@huawei.com).

Digital Object Identifier 10.1109/JSEN.2017.2685999

by comparing the amplitude of acceleration, its low-filtered part or other measures inferred from acceleration with the preset thresholds [15], while the latter does this via making use of the fact that each foot is regularly static for a short portion of time during normal walking motion, which allows to remove the accumulated error from the accelerometer output [16]. However, it is difficult to designate a unified threshold value that works for different smartphone's poses and different users.

Instead of using a threshold-based condition for counting steps, the peak detection-based methods compute the number of steps by counting the peaks of accelerometer readings [17]–[20]. These methods usually first extract local peaks in the amplitude of acceleration, and then apply a temporal constraint on the detected peaks to reduce overcounting. After this operation, each remaining peak is counted as one step. However, the results of peak detection-based methods may not be accurate during a transition of the step modes or smartphone's poses.

The correlation analysis method counts steps by calculating and comparing the correlation coefficients between two neighboring windows of accelerometer data [4], [6], [14]. In these methods, the acceleration is transformed from time domain to frequency domain by using discrete Fourier transform, dynamic time warping, or auto-correlation. However, these methods are computationally expensive and hence are not applicable to smartphone-based applications.

III. The Proposed Method

The proposed method is based on the peak detection method since it has a good performance [6]. We improve the conventional peak detection-based method by applying three constraints (namely, periodicity, similarity, and continuity) for removing the effect of false walking, thereby improving the accuracy of step counting.

A. Peak Detection

The smartphone accelerometer readings present a repetitive and periodical pattern when the user is walking, as depicted in Fig. 1. By utilizing the repetitiveness and periodicity of the user's walking, we can compute how many steps users traveled and further infer the distance they moved.

To reduce the effect of smartphone's orientation, we only use the amplitude of the acceleration:

$$acc_t = \sqrt{{acc_{x_t}}^2 + {acc_{y_t}}^2 + {acc_{z_t}}^2} \tag{1}$$

where acc_{x_t}, acc_{y_t} and acc_{z_t} are the acceleration at time t along x-axis, y-axis, and z-axis, respectively. Based on the amplitude of the acceleration, we can extract the local maxima by checking whether the peak detection condition is met as follows:

$$\begin{aligned} peak_t = \ & (acc_t | acc_t >= (acc_{t-K} : acc_{t-1}) \\ & \&\& \ acc_t >= (acc_{t+1} : acc_{t+K})) \end{aligned} \tag{2}$$

where K is a threshold that is used to control that the detected peaks reflect the user's walking periodicity. Its value is determined by the both sampling rate of the accelerometer and the user's walking speed. Different values of K can have an influence on the detection of peaks that reflect the user's steps, as shown in Fig. 2, therefore selecting an appropriate value is important. A small value may introduce some false peaks while a large one may ignore some correct peaks. In this study, the sampling rate of the smartphone accelerometer is 50 Hz, and the time period for each step ranges from 0.5 seconds to 0.7 seconds.

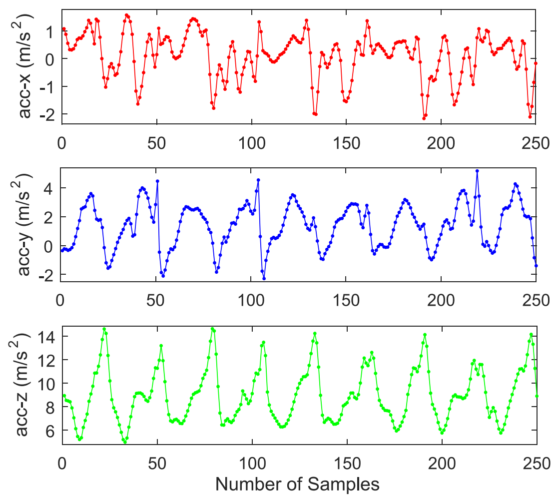

Fig. 1. Accelerometer readings during a user's walking. (The user walks with the phone in the hand without swinging.)

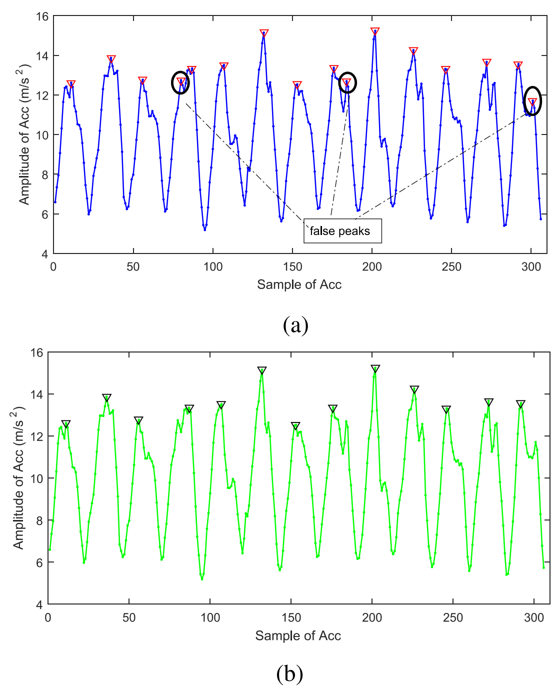

Fig. 2. The influence of the value of K on the peak detection. (a) K=5. (b) K=10.

B. Step Feature Analysis

Although the peak detection-based method has a good performance for step counting when the user walks normally, it does not consider the case of false walking. In this study, we impose three constraints on the walking characteristics in order to reduce the influence of false walking.

- **Periodicity**. The periodicity is defined as the time difference between two neighboring peaks of accelerometer readings, namely

$$T_i = t_{peak_{i+1}} - t_{peak_i} \tag{3}$$

where T_i is the periodicity, $t_{peak_{i+1}}$ and t_{peak_i} are the time when peak $i+1$ appears and the time when peak i arises, respectively. Experimental test shows that the periodicity for the same motion state (e.g., walking, running, or going upstairs or downstairs) is relatively fixed since users walk or run at a relatively constant speed. However, the periodicity is varying when the user remains still while using the phone arbitrarily, e.g., texting, playing phone games. By limiting the range of walking periodicity to a certain interval, we are able to overcome the over-counting problem to some extent.

- **Similarity**. It is observed that the acceleration peaks for two steps are quite similar when users walk or run naturally. This similarity can be measured by the peak distance between two windows of acceleration, which is expressed as

$$sim_i = -\|peak_{acc}(i+2) - peak_{acc}(i)\| \tag{4}$$

where $peak_{acc}(i+2)$ and $peak_{acc}(i)$ are two neighboring acceleration peaks for the left step or right step. The reason why we compare sensor data from the same foot is that the readings for the left foot are slightly different from those for the right foot. The similarity value is inversely proportional to the distance in the value of two acceleration peaks. The closer the distance, the higher the similarity. To avoid that the similarity in still state becomes very high, we modify equation (4) into

$$sim_i = \begin{cases} -\infty, & \text{if } m_i \text{ or } m_{i+2} \text{ is still state} \\ -\|peak_{acc}(i+2) - peak_{acc}(i)\|, & \\ & \text{otherwise} \end{cases} \tag{5}$$

where m_i and m_{i+2} are the corresponding motion states for $peak_{acc}(i)$ and $peak_{acc}(i+2)$, respectively. This is to guarantee that the similarity constraint works for the false walking case and is not affected by the still state.

- **Continuity**. User motion states are generally smooth and continuous, which means that users do not intermittently change their motion states and each motion state lasts for a certain period of time [21]. However, when a user remains in still state and uses her phone for texting, calling, watching movies, playing games, etc., the motion states reflected by sensor data may not be continuous. This characteristic can be used to help eliminate some types of false walking. We use a boolean variable C_i to measure the continuity of user motion states by judging whether the number of the neighboring $N+1$ windows of acceleration readings with the variance surpassing a threshold δ_{var} is greater than the

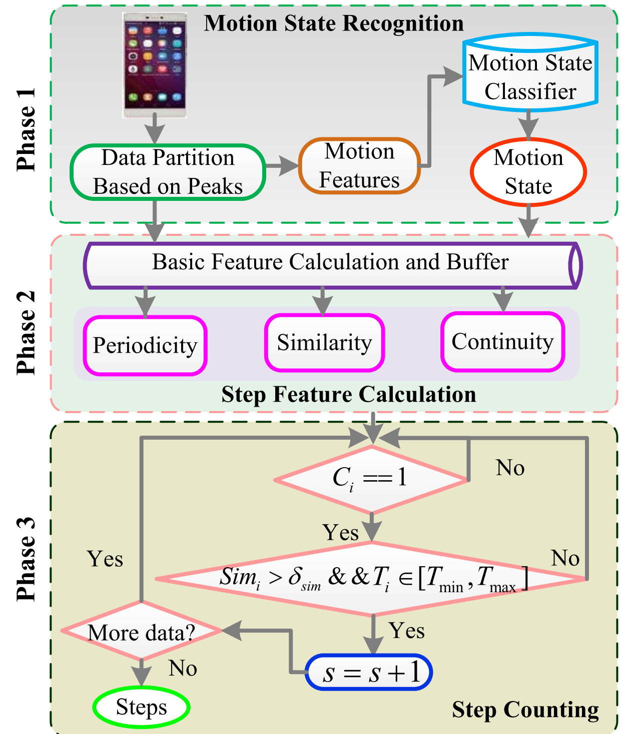

Fig. 3. Flowchart for step counting. Notice that s is the number of steps counted, δ_{sim} is the similarity threshold, $[T_{min}, T_{max}]$ is the periodicity interval of normal walking. Other symbols are the same as described in the section above.

number threshold M, namely

$$C_i = \begin{cases} 1, & \text{if } sum(var(acc(i-N+1:i+1)) > \delta_{var}) >= M \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where N, M, and δ_{var} can be empirically determined. It should be noted that if the condition $var(acc(i)) > \delta_{var}$ is true, then it means that the current state may be normal walking or false walking. In this sense, we can eliminate the influence of still states.

C. Step Counting

The proposed step counting algorithm takes as input accelerometer readings and outputs the number of steps. Firstly, we partition sensor data into different windows by finding the acceleration peaks. Then, the following three phases are conducted to obtain the number of steps that a user has taken, as shown in Fig. 3.

- **Phase 1:** Recognizing user motion states. It is observed that user motion states have an influence on the sensor readings, which will further affect the accuracy of step counting algorithm. For example, the built-in smartphone accelerometer presents a larger variation in the readings when the user is running than when she is walking. We use the decision tree classifier [22] to recognize user motion states (e.g., walking, still) since it is efficient, easy-to-implement, and can achieve a relatively high accuracy. The recognition of more complex user motion states can be found in our prior work [21].

Note that the classifier may not be able to recognize false walking since it also witnesses a variation in the accelerometer readings. Nevertheless, we can eliminate the effect of false walking by imposing constraints on the periodicity, similarity, and continuity. The outputs of this step are user motion state and windowed accelerometer readings with the corresponding timestamps.

- **Phase 2:** Computing step features (namely, periodicity, similarity, and continuity). To do this, some basic features (e.g., Peaks of Acceleration) need to be calculated first, by which we can further compute the three step features. The motion states computed in the first phase are used to calculate the continuity feature according to equation (6). Similarly, we can compute the periodicity and similarity features by using equations (3) and (5), respectively. It should be noted that we use a buffer to store a certain number of last windows of data since the computation of these three features takes as input a sequence of accelerometer readings and motion states, rather than only one single window of data. Specifically, two neighboring windows of accelerometer readings with timestamps are needed to compute the periodicity and similarity features, and $N + 1$ motion states are required to calculate the continuity feature.
- **Phase 3:** Estimating the number of steps by counting acceleration peaks under the three step feature constraints. For each peak detected in the accelerometer data, three constraints should be satisfied for the peak to be counted as a step. The first constraint is the continuity constraint which stipulates that some peaks in false walking state can be removed. This is because when the user stays in the false walking state, the change pattern of the acceleration is not continuous and repetitive. The second constraint is the similarity constraint that helps in removing some peaks in false walking state by checking these peaks with the similarity value surpassing a threshold. The periodicity constraint is used to constrain the periodicity range for walking. If a periodicity falls outside of this range, the corresponding peak will be removed. It should be noted that using a single constraint cannot eliminate all the effect from false walking state, only combining three constraints together can significantly improve the accuracy of the step counting algorithm.

IV. Experiments and Results

A. Experimental Setup

The proposed step counting algorithm was evaluated by a series of experiments conducted in the surroundings of the Infrastructure Engineering building located at the Parkville campus of the University of Melbourne.

In total, two groups of experiments were conducted in arbitrary trajectories, which means that there was no requirement for the user to walk in straight lines. The first group of experiments is to evaluate the performance of the proposed method and to show its superiority over the conventional peak detection-based method. In the first group of experiments, eight volunteers were recruited and asked to walk 300 steps with a smartphone in different motion states. During the experiments, the participants were required to count their steps until they have walked 300 steps, which was regarded as the ground truth for estimating the accuracy of the step counting algorithm. A Samsung Galaxy S III smartphone was used to record the accelerometer readings and corresponding timestamps collected by these eight participants in three cases: (a) normal: the participants walked 300 steps with the phone in a fixed phone pose (e.g., in the trouser's pocket); (b) free: the participants walked 300 steps with the phone in arbitrary phone poses; (c) false: the participants walked 300 steps according to the free walking rules while in some segments they were asked to stop and sway the mobile phone thereby create a false walking state. Each group of data may include some noisy data introduced at the beginning and end of conducting an experiment.

TABLE I
PARAMETER SETTING

Symbol	Value	Meaning
K	15	peak detection threshold
T_{min}	0.3	minimum periodicity threshold
T_{max}	1	maximum periodicity threshold
sim_i	-5	similarity threshold
M	2	window size of continuity
N	4	number threshold of continuity
σ_{var}	0.7	variance threshold for motion recognition

The second group of experiments is to compare the proposed method with three popular commercial step counting applications: S-Health on Sony C6603 [23], i-Health [24] on iPhone 6 plus and Pedometer++ [25] on iPhone 5s. The data for the proposed method were collected by the Samsung Galaxy S III smartphone. The tester took four phones in the hand and walked 300 steps in three cases: (a) normal walking, (b) free walking and (c) false walking.

B. Parameter Setting

The parameters for the proposed method and the conventional peak detection-based method were set as shown in Table I, where all the values are in the standard unit. These values of parameters were set by experimental analysis, which are applicable for different users and walking modes. Since the threshold values of three constraints are important for achieving a more robust and accurate step counting algorithm, here we explain their determination process in detail. These values are important for achieving accurate step counting. For example, a too small lower threshold of periodicity may exclude the data of fast walking and a too large upper threshold may include data of false walking. Therefore, we first compute the periodicity range of the users taking a step based on the data including normal walking and free walking, by which we can obtain proper values for the minimum periodicity threshold and for the maximum periodicity threshold. Then, we calculate the movement similarities for all the neighbouring left or right steps using the data for normal walking and free walking, and set the similarity threshold to a value that will not lead to under-counting for normal walking. The absolute value of similarity threshold should not be too small as it may

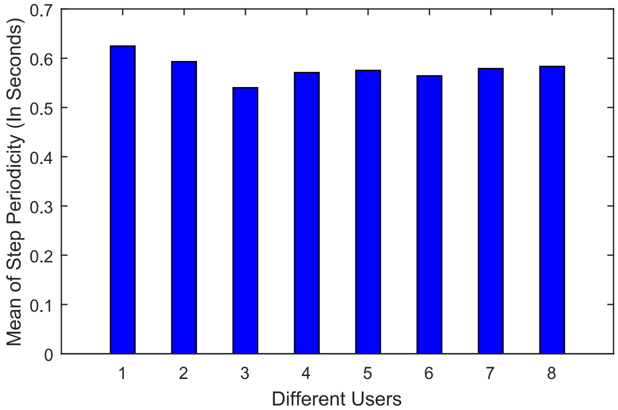

Fig. 4. Step periodicity of different participants. (The users walk naturally with the phone in the pocket.)

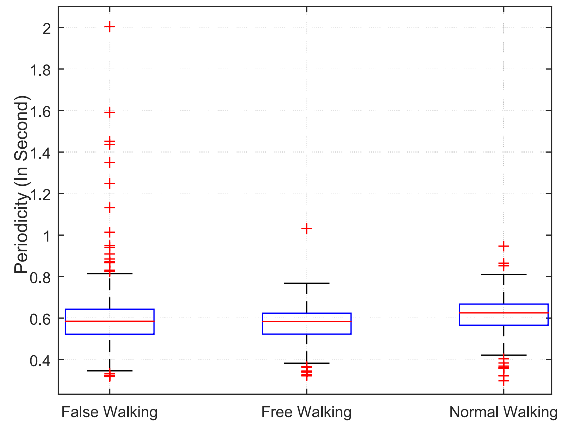

Fig. 5. Periodicity comparison among three cases.

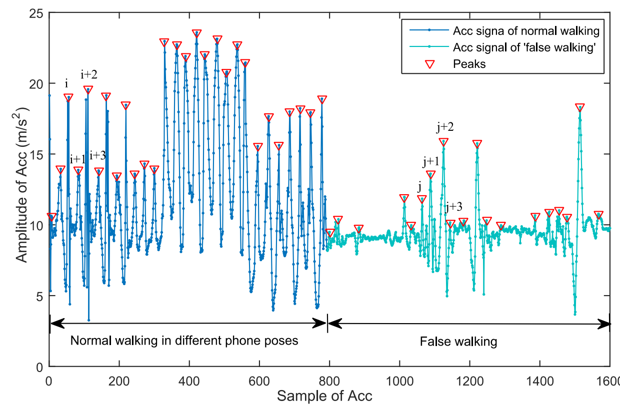

Fig. 6. Signal comparison between normal walking and false walking.

exclude steps taken in normal walking state, while it may not function well if it is too large. Finally, false walking data are used to set the continuity threshold to a value that will not affect the accuracy of normal walking and free walking cases, but is able to reduce the mis-counting for false walking.

C. Periodicity Analysis

To analyze the periodicity in different walking states, we compare the time period of walking one step for each participant. The mean walking periodicity for each participant is shown in Fig. 4, from which we can see that the walking periodicity for different users is quite similar to one another, ranging from 0.5 seconds to 0.65 seconds for each step (one left step or right step). That is to say, we can set thresholds for walking periodicity to rule out some false walking steps that lead to the over-counting problem.

To clearly illustrate how the periodicity thresholds work, we use box plots to compare the statistics of periodicity values in each walking mode, as described in Fig. 5. The data for Fig. 5 were collected by one of the eight volunteers in three walking modes. From Fig. 5, we can see that there are much more outliers of periodicity in the false walking case than those in free walking and normal walking cases. This means that by setting an appropriate threshold interval, we can remove the data with the periodicity falling outside of the interval, which is caused by false walking, thereby improving the accuracy of the step counting algorithm.

D. Similarity Analysis

After using the periodicity constraint to limit the periodicity range of steps, we can eliminate some abnormal steps with periodicity falling outside of the periodicity interval of normal walking. However, we cannot remove those false steps whose periodicity falls into the interval of normal walking only by utilizing the periodicity constraint. Thus, we also utilize the similarity constraint (as described in section III) to further assist addressing the over-counting problem caused by false walking. We compare the similarity of steps among normal walking with that among false walking.

Fig. 6 shows the acceleration signal difference between normal walking and false walking, from which we can see that the acceleration signal for normal walking presents a periodical and repetitive pattern no matter which pose the phone is put in, but there is no such a pattern for false walking state. Specifically, the similarities between two neighboring left steps ($acc(i : i+1)$ and $acc\ (i+2 : i+3)$) or right steps ($acc(i+1 : i+2)$ and $acc(i+3 : i+4)$) in normal walking state are -0.56 and -0.04, respectively. In contrast, the similarities for those in false walking state are -3.65 (for $acc(j : j+1)$ and $acc\ (j+2 : j+3)$) and -4.06 (for $acc(j+1 : j+2)$ and $acc\ (j+3 : j+4)$), respectively, which are significantly different from those in normal walking.

This similarity distribution for different cases (false walking, free walking, and normal walking) is shown in Fig. 7. It is obvious that the box for false walking is wider than those for free walking and normal walking, and there are more similarity outliers in the false walking case. This is because when a user in still state uses her smartphone, the phone's movement pattern is random, so the similarity between two windows of accelerometer readings is lower. By setting a proper similarity threshold (e.g., -5), we can remove steps with abnormal similarity and further enhance the accuracy of step counting.

E. Continuity Analysis

The continuity constraint is used to judge whether a user's motion is continuous or not so that we can determine whether

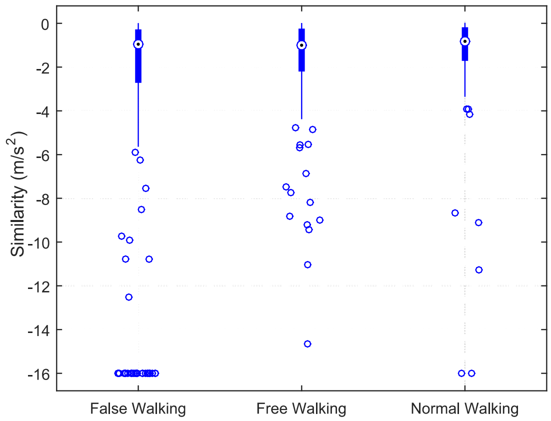

Fig. 7. Similarity comparison among three cases.

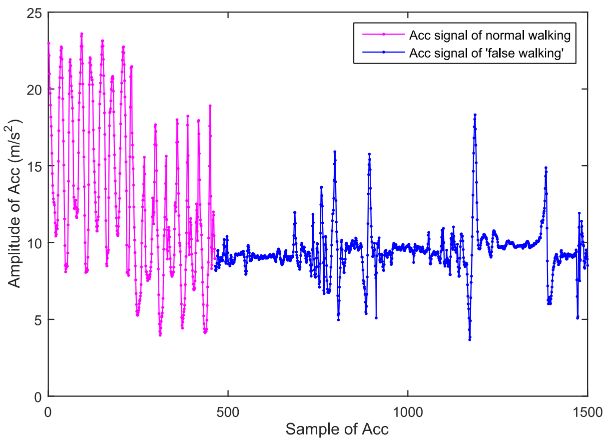

Fig. 8. Signal comparison between normal walking and false walking.

the user is in normal walking or false walking. According to the observation that users do not intermittently switch their motion states, we regard those motions with frequent significant changes in the accelerometer readings as false walking and then remove them to avoid over-counting. Fig. 8 shows the acceleration of normal walking and false walking, from which it is easy to see that the motion states of false walking are not continuous and do not present a periodical and repetitive pattern.

Fig. 9 shows the continuity in three walking modes, from which we can see that the motion states for free walking and normal walking are continuous except a few outliers at the beginning or end of data collection. By contrast, in the false walking case, the motion states are often not continuous during the period of false walking. This information will help analyze periodicity and similarity and further assist in improving the accuracy and robustness of the step counting algorithm.

F. Step Counting Performance

In this section, we present the performance of the proposed step counting algorithm and compare it with the commonly-used peak detection-based method. The reason why we just compare our method with the peak detection-based method is that the peak detection-based method is a typical conventional step counting method, which has a better performance compared with normalized autocorrelation, short term

Fig. 9. Continuity comparison among three cases.

Fourier transform, continuous wavelet transform [6]. Since we focus on solving the over-counting problem caused by false walking in this research, it is reasonable only to compare the performance of the proposed method with one of popular methods. We use the relative error to evaluate the performance, which is defined as

$$e = \frac{|N_e - N_r|}{N_r} * 100\% \tag{7}$$

where N_e is the estimated number of steps, and N_r is the ground truth.

The results are shown in Fig. 10. Each sub-figure shows the corresponding result for case (a), (b), and (c) of the first experiment group, respectively. Generally, for both methods, the errors for normal walking are smaller than free walking, and the largest errors appear in the false walking case. This is because changing the phone poses during false walking introduces some noise, which affects the accuracy of step counting algorithms. Fig. 10 also demonstrates that the error for different users is different since they have varying walking characteristics and different phone use habits. It is interesting to note that in Fig. 10 (c) the error of the peak detection-based method for user 5 reaches up to around 200%, probably because that the user introduced too much noise while conducting the experiment. This huge error was dramatically reduced to 18% by using the proposed method.

It is evident in Fig. 10 that the proposed method outperforms the commonly-used peak detection-based method in all the three cases. The average errors for each group of experiments are given in Table II, from which we can see that compared with the commonly-used peak detection-based method, the proposed method can improve the step counting accuracy by 6.56% for normal walking, 9.54% for free walking, and by 58.92% for data including false walking. This is a significant improvement in the accuracy and robustness.

Fig. 11 shows the comparison between our method and the standard peak-detection-based method. It can be seen that when the user walks normally (can be in different walking speeds), the two methods present similar results. However, when the user introduces noise of false walking (remaining still while using the phone for texting, playing a game,

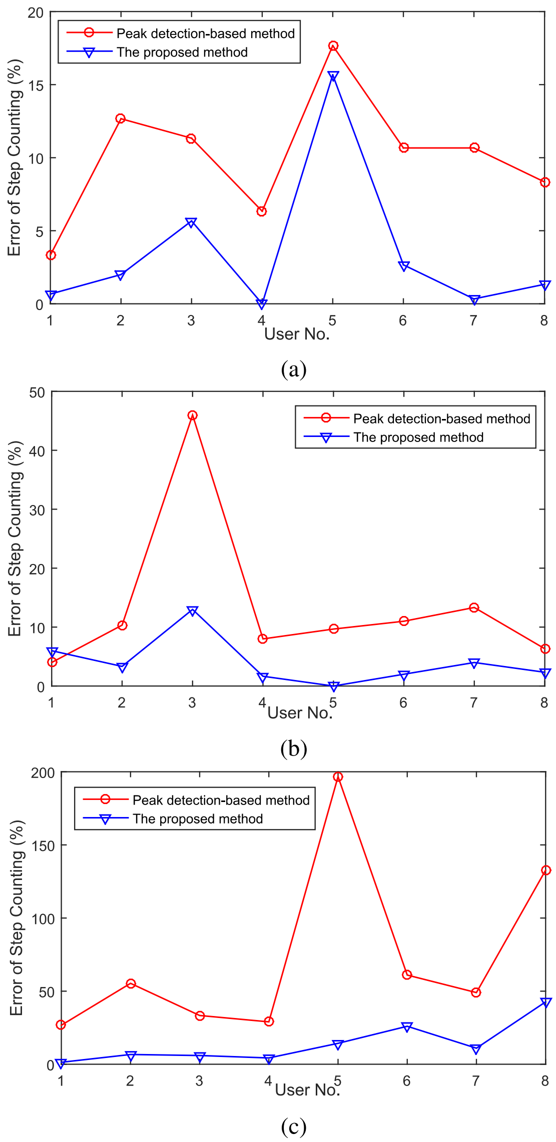

Fig. 10. Performance comparison between the peak detection-based method and the proposed method (a) Normal walking. (b) Free walking. (c) False walking.

TABLE II

AVERAGE ERROR

Experiment Group	Peak Detection -based Method	The Proposed Method	Improvement
Normal walking	10.13%	3.54%	6.58%
Free walking	13.58%	4.04%	9.54%
False walking	72.96%	14.04%	58.92%

and so on), the standard peak detection-based continues wrongly to count steps, which can be avoided by our method.

G. Comparison With State-of-the-Art Applications

To compare the proposed method with popular state-of-the-art step counting applications, another group of experiments were conducted. The commercial applications we compare with include S Health, Pedometer++, and i-Health. The former is popular on Android platform while the latter two are popular on iOS platform.

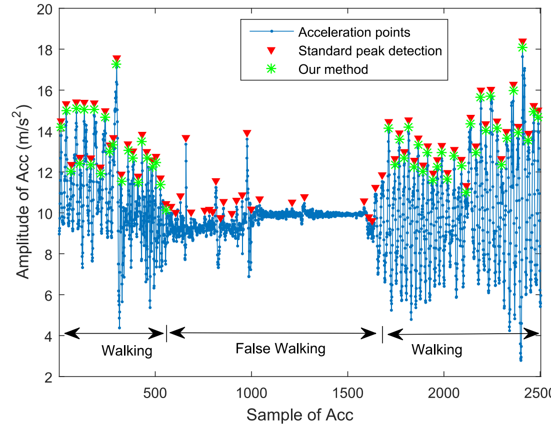

Fig. 11. Comparison between our method and the standard peak detection-based method.

TABLE III

ACCURACY COMPARISON WITH COMMERCIAL APPLICATIONS

Algorithm	Error		
	Normal Walking	Free Walking	False Walking
S Health	2%	1.66%	13.67%
Pedometer++	0.67%	1%	15.33%
i-Health	0.33%	4.33%	18.67%
Our method	0.33%	1%	11.67%

The results are shown in Table III, from which it is clear that all step counting methods tend to perform best when the user walks normally and perform worst when dealing with false walking case. Overall, our method outperforms the three commercial step counting applications especially when dealing with the false walking problem.

V. CONCLUSION AND DISCUSSION

This paper presents an accurate and robust step counting algorithm which is based on peak detection and analysis of step features, including periodicity, similarity, and continuity. The proposed method focuses on solving the over-counting problem caused by false walking, which is ignored in existing applications. The performance of the proposed method is evaluated by a series of experiments. The experimental results show that the proposed method outperforms the commonly-used peak detection-based method. We also compare the proposed method with some state-of-the-art step counting applications, and demonstrate that our methods perform better especially when dealing with false walking case.

A limitation of the proposed method is when a user in still state swings her phone on purpose as she does in normal walking. This problem can be solved by combining measurements from other infrastructures such as WiFi access points [26]–[28]. This is because the WiFi signal strength does not change significantly over time when the user stays at the same location. In the future, we will try to solve this problem by considering different sensors and signal modules in the smartphones.

References

[1] H. Huang and G. Gartner, "A survey of mobile indoor navigation systems," in *Cartography Central Eastern Europe*. Berlin, Germany: Springer, 2009, pp. 305–319.

[2] J. Shang, X. Hu, F. Gu, D. Wang, and S. Yu, "Improvement schemes for indoor mobile location estimation: A survey," *Math. Problems Eng.*, vol. 2015, Mar. 2015, Art. no. 397298.

[3] L. Pei, M. Zhang, D. Zou, R. Chen, and Y. Chen, "A survey of crowd sensing opportunistic signals for indoor localization," *Mobile Inf. Syst.*, vol. 2016, Apr. 2016. Art. no. 4041291.

[4] M. S. Pan and H. W. Lin, "A step counting algorithm for smartphone users: Design and implementation," *IEEE Sensors J.*, vol. 15, no. 4, pp. 2296–2305, Apr. 2015.

[5] K. Ozcan and S. Velipasalar, "Robust and reliable step counting by mobile phone cameras," in *Proc. 9th Int. Conf. Distrib. Smart Cameras*, Sep. 2015, pp. 164–169.

[6] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, Sep. 2013, pp. 225–234.

[7] H. H. Lee, S. Choi, and M. J. Lee, "Step detection robust against the dynamics of smartphones," *Sensors*, vol. 15, no. 10, pp. 27230–27250, Sep. 2015.

[8] J. Shang, F. Gu, X. Hu, and A. Kealy, "APFiLoc: An infrastructure-free indoor localization method fusing smartphone inertial sensors, landmarks and map information," *Sensors*, vol. 15, no. 10, pp. 27251–27272, Oct. 2015.

[9] E. L. Melanson *et al.*, "Commercially available pedometers: Considerations for accurate step counting," *Preventive Med.*, vol. 39, no. 2, pp. 361–368, Aug. 2004.

[10] J. S. Sheu, G. S. Huang, W. C. Jheng, and C. H. Hsiao, "Design and implementation of a three-dimensional pedometer accumulating walking or jogging motions," in *Proc. Int. Symp. Comput., Consum. Control (IS3C)*, Jun. 2014, pp. 828–831.

[11] C. C. Lo, C. P. Chiu, Y. C. Tseng, S. A. Chang, and L. C. Kuo, "A walking velocity update technique for pedestrian dead-reckoning applications," in *Proc. IEEE 22nd Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2011, pp. 1249–1253.

[12] J. Chon and H. Cha, "LifeMap: A smartphone-based context provider for location-based services," *IEEE Pervas. Comput.*, vol. 10, no. 2, pp. 58–67, Jun. 2011.

[13] H. J. Jang, J. W. Kim, and D. H. Hwang, "Robust step detection method for pedestrian navigation systems," *IET Electron. Lett.*, vol. 43, no. 14, pp. 749–751, Jul. 2007.

[14] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Aug. 2012, pp. 293–304.

[15] J. A. B. Link, P. Smith, N. Viol, and K. Wehrle, "Footpath: Accurate map-based indoor navigation using smartphones," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2011, pp. 1–8.

[16] L. Ojeda and J. Borenstein, "Personal dead-reckoning system for GPS-denied environments," in *Proc. IEEE Int. Workshop Safety, Secur. Rescue Robot.*, Sep. 2007, p. 1–6.

[17] H. Zhang, W. Yuan, Q. Shen, T. Li, and H. Chang, "A handheld inertial pedestrian navigation system with accurate step modes and device poses recognition," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1421–1429, Mar. 2015.

[18] M. Susi, V. Renaudin, and G. Lachapelle, "Motion mode recognition and step detection algorithms for mobile phone users," *Sensors*, vol. 13, no. 2, pp. 1539–1562, Jan. 2013.

[19] Y. Chon, E. Talipov, and H. Cha, "Autonomous management of everyday places for a personalized location provider," *IEEE Trans. Syst., Man, Cybern. Part C Appl. Rev.*, vol. 42, no. 4, pp. 518–531, Jul. 2012.

[20] D. Loh, S. Zihajehzadeh, R. Hoskinson, H. Abdollahi, and E. J. Park, "Pedestrian dead reckoning with smartglasses and smartwatch," *IEEE Sensors J.*, vol. 16, no. 22, pp. 8132–8141, Nov. 2016.

[21] F. Gu, A. Kealy, K. Khoshelham, and J. Shang, "User-independent motion state recognition using smartphone sensors," *Sensors*, vol. 15, no. 12, pp. 30636–30652, Dec. 2015.

[22] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, "Decision trees for mining data streams based on the Gaussian approximation," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 108–119, Jan. 2014.

[23] Samsung Electronics Co., Ltd. *S Health*, accessed on Sep. 20, 2016. [Online]. Available : http://shealth.samsung.com

[24] Apple Inc. *Health*, accessed on Sep. 20, 2010. [Online]. Available: www.apple.com/ios/health

[25] Cross Forward Consulting LLC. *Pedometer++*, accessed on Sep. 20, 2016. [online]. Available: pedometerplusplus.com

[26] A. Cramariuc, H. Huttunen, and E. S. Lohan, "Clustering benefits in mobile-centric WiFi positioning in multi-floor buildings," in *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, Jun. 2016, pp. 1–6.

[27] X. Hu, J. Shang, F. Gu, and Q. Han, "Improving Wi-Fi indoor positioning via ap sets similarity and semi-supervised affinity propagation clustering," *Int. J. Distrib. Sens. Netw.*, vol. 2015, Jan. 2015, Art. no. 109642.

[28] G. Caso, L. De Nardis, and M. G. Di Benedetto, "Frequentist inference for WiFi fingerprinting 3D indoor positioning," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 809–814.

Fuqiang Gu (S'16) received the M.Sc. degree in computer science from the China University of Geosciences, Wuhan, in 2013. He is currently pursuing the Ph.D. degree with the Department of Infrastructure Engineering, University of Melbourne. His research interests include indoor localization and navigation, location-based services, activity recognition, and machine learning.

Kourosh Khoshelham received the Ph.D. degree in geomatics from the Hong Kong Polytechnic University. He is a Lecturer with the Department of Infrastructure Engineering, University of Melbourne. He is the Co-Chair of ISPRS WG IV/5 on indoor/outdoor seamless modeling, LBS, and mobility. His research interests include mobile mapping, indoor mapping, building information modeling, and information extraction from images and point clouds.

Jianga Shang (M'12) received the Ph.D. degree in computer system architecture from the Huazhong University of Science and Technology. He is an Associate Professor with the Faculty of Information Engineering, China University of Geosciences, Wuhan. He is a Senior Researcher with the National Engineering Research Center for Geographic Information System. He is a committee member of ISPRS WG IV/5 on indoor/outdoor seamless modeling, LBS, and mobility. His research interests include indoor location-awareness, mobile and context-aware computing, location-based social networks, and geospatial information systems. He is a member of the ACM, China Computer Federation, and GNSS & LBS Association of China.

Fangwen Yu received the M.Sc. degree in computer science from the China University of Geosciences in 2015, where he is currently pursuing the Ph.D. degree in surveying and mapping. His research interests include indoor localization and cognitive navigation, indoor spatial model, and machine learning. He is a student member of ACM and the China Computer Federation.

Zhuo Wei received the B.A. degree from Jilin University, China, and the M.S. and Ph.D. degrees from the Huazhong University of Science and Technology, China. He is currently a Research Scientist with the Huawei's Shield Lab, International Pte Ltd., Singapore.

His interests include image processing and video processing. He was a recipient of the Best Paper Award of CMS 2012.