



## Article

# Smartphone-Based Unconstrained Step Detection Fusing a Variable Sliding Window and an Adaptive Threshold

Ying Xu <sup>1,\*</sup>, Guofeng Li <sup>1</sup>, Zeyu Li <sup>1</sup> , Hao Yu <sup>1</sup>, Jianhui Cui <sup>1</sup> , Jin Wang <sup>1</sup> and Yu Chen <sup>2</sup>

<sup>1</sup> College of Geodesy and Geomatics, Shandong University of Science and Technology, Qingdao 299590, China; guofengli@sdust.edu.cn (G.L.); lizeyu@sdust.edu.cn (Z.L.); yuhao@sdust.edu.cn (H.Y.); jhcui@sdust.edu.cn (J.C.); wangjin@sdust.edu.cn (J.W.)

<sup>2</sup> Shandong Provincial Institute of Land Surveying and Mapping, Jinan 250013, China; chenylt@shandong.cn

\* Correspondence: yingxu@sdust.edu.cn

**Abstract:** Step detection for smartphones plays an important role in the pedestrian dead reckoning (PDR) for indoor positioning. Aiming at the problem of low step detection accuracy of smartphones in complex unconstrained states in PDR, smartphone-based unconstrained step detection method fusing a variable sliding window and an adaptive threshold is proposed. In this method, the dynamic updating algorithm of a peak threshold is developed, and the minimum peak value filtered after a sliding window filter is used as the adaptive peak threshold, which solves the problem that the peak threshold of different motion states is difficult to update adaptively. Then, a variable sliding window collaborative time threshold method is proposed, which solves the problem that the adjacent windows cannot be contacted, and the initial peak and the end peak are difficult to accurately identify. To evaluate the performance of the proposed unconstrained step detection algorithm, 50 experiments in constrained and unconstrained states are conducted by 25 volunteers holding 21 different types of smartphones. Experimental results show: The average step counting accuracy of the proposed unconstrained step detection algorithm is over 98%. Compared with the open source program Stepcount, the average step counting accuracy of the proposed algorithm is improved by 10.0%. The smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold has a strong ability to adapt to complex unconstrained states, and the average step counting accuracy rate is only 0.6% lower than that of constrained states. This algorithm has a wide audience and is friendly for different genders and smartphones with different prices.

**Keywords:** step detection; indoor positioning; unconstrained state; peak detectors; adaptive threshold; variable sliding window



**Citation:** Xu, Y.; Li, G.; Li, Z.; Yu, H.; Cui, J.; Wang, J.; Chen, Y. Smartphone-Based Unconstrained Step Detection Fusing a Variable Sliding Window and an Adaptive Threshold. *Remote Sens.* **2022**, *14*, 2926. <https://doi.org/10.3390/rs14122926>

Academic Editor: Liang Chen

Received: 27 April 2022

Accepted: 17 June 2022

Published: 19 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The Global Navigation Satellite System (GNSS) is dominant in providing outdoor positioning service due to its coverage and high accuracy. However, people spend about 80% of their time indoors every day according to incomplete statistics. During the epidemic, medical staff or volunteers needed to grasp the dynamic position of personnel in isolated hotels and isolated wards in real time. When an indoor fire occurs, rescuers need to know the exact location of trapped people in time. In the construction of a smart city [1–3], and the tracking of pandemics [4], indoor positioning is the basic technology. In short, indoor positioning has broad application prospects. However, GNSS cannot provide services indoor. Researchers have proposed to use pedestrian dead reckoning (PDR) or PDR and wireless sensor fusion to achieve indoor positioning [5]. Smartphone-based step detection is necessary for PDR to determine pedestrian trajectory information [6,7]. In addition, smartphones bring many conveniences to people's lives with their rich functions and applications. Among these functions, step detection plays a role in health care for obese patients, has become a physical therapy to control chronic low back pain [8], monitor the

fall of the elderly [9], and can also be applied to daily fitness training [10]. MI band, Huawei band and other commercial products record users' daily steps, and then give health tips. The commonly used step detection algorithms are zero velocity update (ZUPT) [11,12], autocorrelation analysis [13], peak detection [14], etc.

ZUPT refers to the lower limb being in a static state at a certain time during the walking process, and the walking speed is zero at this time, or the output values of acceleration sensor and angular velocity sensor will be approximately zero when the foot makes contact with the ground. The ZUPT method generally requires the sensor to be fixed in a specific position of the lower limbs, such as calves, feet, etc. [11,12]. Obviously, the smartphone does not have the basic conditions for this method.

When pedestrians walk continuously, there is a high correlation between the front and back gait cycles. The auto correlation analysis detects the number of steps by judging the correlation coefficient of the two cycles. Pan [13] uses auto correlation analysis to calculate steps. The experimental results show that the average step counting accuracy can reach 97.8% when the pedestrians dynamically switch the position carried by the smartphone, but this experiment failed to consider the important factor of changing the movement state of pedestrians at any time on the step-counting accuracy. Additionally, the calculation of correlation coefficient is large, which affects the timeliness of the algorithm.

Peak detection also detected the number of steps according to the periodicity of the pedestrian's continuous walking. Unlike the auto correlation analysis method, the peak detection takes the number of peaks (valleys) generated by the acceleration sensor or gyroscope sensor as steps. However, the pseudo-peak restricts the accuracy of the detection steps of the peak detection method. At present, the pseudo-peak is mainly eliminated by setting the threshold. Xu et al. [7] used a fixed threshold method to remove pseudo-peak, which has high accuracy when pedestrian motion and smartphone carrying mode are constrained. Cho et al. [14] used the sign-of-slope method and average threshold method to realize peak detection; Zhang et al. [15] used the mean value of the acceleration amplitude of the previous window to dynamically update the acceleration threshold; Wang [16] adaptively selects the threshold of acceleration according to the average value of the difference between peaks and valleys in unit time. Ryu et al. [17] proposed an adaptive threshold method, which uses the average value of the first five consecutive peaks as the adaptive threshold. However, these algorithms are easy to cause misjudgment in multiple motion modes. Dirican et al. [18] proposed a threshold-based unconstrained step counting algorithm. Unlike other ways of thinking about setting update thresholds based on acceleration data, this method sets the real and imaginary parts of the data transformed by fast Fourier transform (FFT) to different thresholds and achieves the update of thresholds by averaging the current and previous thresholds. This method achieves an adaptive update of acceleration thresholds and can adapt to a variety of unconstrained states, but the accuracy of the pedometer for running states is only 41.7%.

In addition to the pseudo-peak affecting the accuracy of step counting, selecting the appropriate sliding window is also helpful to indirectly improve the accuracy of step counting. Currently, the sliding window size is mainly based on the time required for a single step, so that the pedestrian can continuously output the step counting results without delay for each step; however, this requires that the size of the sliding window should essentially match the time used for a single step. Ning et al. [19] sets the size of three sliding windows according to the three states of motion of going up and down, walking and running, but the size of these windows are still fixed, so it is difficult to adjust to different users. Kang et al. [20] proposes a method of changing the sliding window, which mainly determines the size of the sliding window adaptively according to the walking frequency, but the accuracy of the walking frequency will affect the determination of the sliding window. It can be seen that it is difficult to ascertain the size of the sliding window when realizing continuous single-step counting. As long as the influence of sliding window size on real-time performance is within an acceptable range, looking for a sliding window which is not based on single-step detection, and solving the problem that a large sliding

window separates the connection between neighboring windows, is also an ideal solution to the problem.

In summary, in the motion environment and motion state, the random transformation of smartphone carrying mode and the complex unconstrained state with interference factors, the step detection algorithm based on peak detection has the problem that the adaptive threshold is difficult to accurately update dynamically and the sliding window is difficult to accurately determine, which restricts the accuracy of peak detection. In view of this, smartphone-based unconstrained step detection fusing a variable sliding windows and an adaptive threshold is proposed in this paper. The algorithm uses the minimum peak filtered by the sliding window as the adaptive threshold to solve the problem that the peak threshold is difficult to update adaptively when the pedestrian state changes. The algorithm is a step detection algorithm of a variable sliding window. It is not based on the commonly used single-step detection of a sliding window, but realizes the variable sliding window on the basis of a fixed sliding window of 1 s, which ensures the close connection between the windows. At the same time, the cooperative time threshold solves the problem that the initial peak and the final peak make it difficult to distinguish the authenticity in the fixed sliding window. Using this algorithm, the accuracy of step counting can be guaranteed under complex unconstrained conditions. There are three aspects of contributions for the smartphone-based unconstrained step detection method proposed in this paper. First, it allows users to carry smartphones at multi points for indoor positioning. Second, users can freely switch the way of carrying and the state of motion for indoor positioning. Third, smartphones with different valences have high step detection accuracy.

The Section 1 introduces the research background and existing problems of the step detection algorithm. The preprocessing process and motion state recognition process of the step detection algorithm are described in the Section 2. Smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold is proposed in the Section 3. The Section 4 evaluates the step counting performance of smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold in constrained and unconstrained states through 50 groups of experiments. The Section 5 summarizes the work of this paper.

## 2. Step Detection Preprocessing and Motion State Recognition

As the sensors in the smartphone can detect the periodic changes of pedestrians, smartphones can detect the step numbers. Both acceleration sensors and gyroscope sensors can detect the periodic change when walking. However, the sensitivity of the gyroscope sensor is depressed, and the acceleration sensor is mostly used for step detection [21].

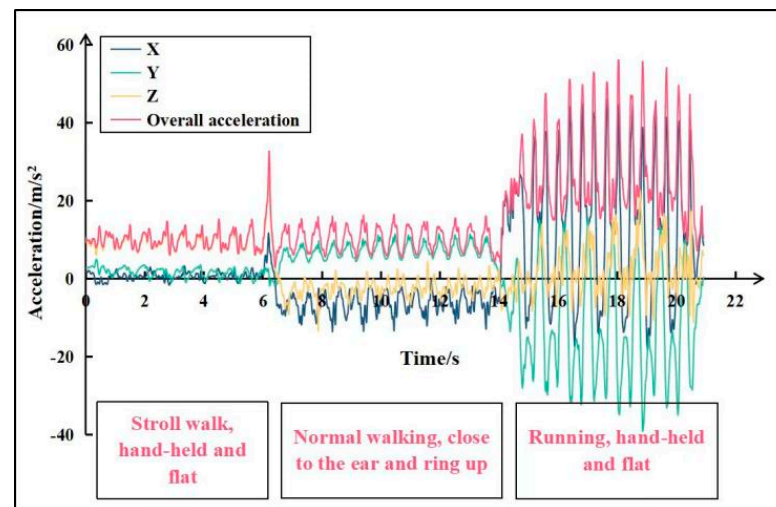
With each step forward, the pedestrian will produce a vertical motion and forward motion. The vertical axis of the three-axis accelerometer will produce an approximate sinusoidal wave, and the number of peaks (valleys) detected can be used as the number of steps of the pedestrian. However, the location of the smartphone carried by pedestrians is changeable, and it is difficult to accurately identify which single axis is in the vertical state, but the influence of sensor attitude can be reduced by calculating the overall acceleration. Formula (1) is the formula for calculating the overall acceleration.

$$a_c(t) = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (1)$$

In the formula,  $a_x$ ,  $a_y$ ,  $a_z$  represent the accelerometer output values of the X-axis, Y-axis and Z-axis at t time, and  $a_c(t)$  represents the overall acceleration.

The signal characteristics of triaxial acceleration and overall acceleration were compared by experiment. Figure 1 shows the triaxial acceleration signal and the overall acceleration signal when walking at will. In the experiment, there are three states: walking and the smartphone is flat, normal walking and putting the smartphone next to the ear, running and swing hand. It can be seen from Figure 1 that the most sensitive axis has undergone three transformations, namely the Z-axis, Y-axis and X-axis, whereas the overall

acceleration signal shows significant periodical changes. Therefore, the overall acceleration is adopted in the step detection algorithm in this paper.

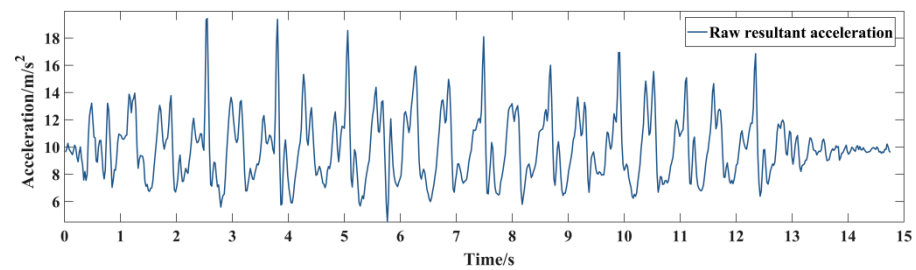


**Figure 1.** Triaxial acceleration and overall acceleration when walking at will.

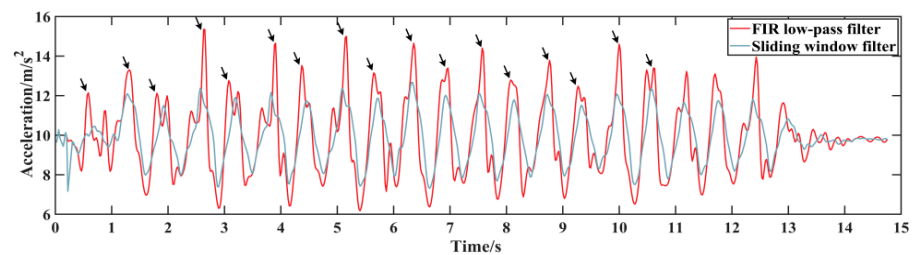
The accuracy of the acceleration sensor of the smartphone is low, which leads to too many burr points of the original overall acceleration signal. It needs to be filtered to reduce the interference of more burrs before step detection. In addition, in the process of movement, pedestrians will be accompanied by multiple motion states. Different motion states often have different peak threshold. Therefore, it is necessary to identify motion state before step detection. The following focuses on the methods of step detection data preprocessing and motion state recognition.

### 2.1. Data Preprocessing of Step Detection

In order to remove the white Gaussian noise, Guo et al. [22] uses the weighted moving average method and the Kalman filter to preprocess the original resultant acceleration data to remove the influence of the Gaussian white noise, and then uses the Butterworth filter to refine the denoising. But the excessive and complicated filtering methods increased the data processing time. Zhang et al. [15] adopted the sliding window filter method to weaken the multi-peak phenomenon, which is a more common and better smoothing method for filter data, but the method loses the characteristics values of the data. Alabadleh et al. [23] used the Kalman filter and high-pass filter to smooth data, which removed gravity and outliers, but the algorithm had some complexity. Liu et al. [24] adopted a low-pass filter to eliminate signal noise, and the low-pass filter can keep the characteristics of the data very well. In this paper, a Finite Impulse Response (FIR) low-pass filter based on a Hamming window [25] is used to preprocess the ensemble acceleration signal, where the order of the filter is 10 and the length of the Hamming window is 11. As the actual output frequency of some smartphones does not match the sampling frequency, the sampling frequency used in the paper is the actual output frequency and the pass-band frequency is 5 Hz. Figure 2 shows the original overall acceleration signal, and Figure 3 compares a sliding window filter (the window size is 15 samples) and a FIR low-pass filter. The FIR low-pass filter retains the large and small peaks of the original overall acceleration signal (shown by the black arrows in Figure 3), reflecting the different characteristics of the left and right footsteps of the human body when walking.



**Figure 2.** Raw overall acceleration signal.



**Figure 3.** FIR low-pass filter and sliding window filter.

## 2.2. Motion State Recognition

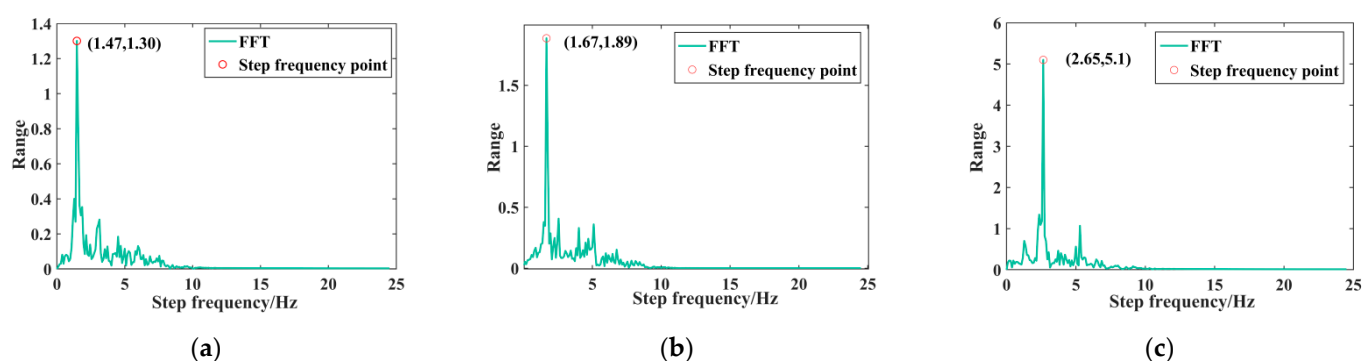
Pedestrians lift one foot off the ground, move to a new position and after that, put it back on the ground, which is known as a single step [17]. Figure 4 shows the decomposition diagram of pedestrian single-step action. Since the step detection algorithm in this paper is mainly applied to PDR, stroll walking, normal walking and running in the previous progress are considered, which are general division methods.



**Figure 4.** Single step action decomposition.

When pedestrians are in different states of motion, the time and the peak acceleration of a single step are different. If a single fixed time threshold and a peak threshold are used to realize step counting, it is difficult to ensure the accuracy of steps. Identifying different states of motion and setting or updating different thresholds according to different states of motion can effectively improve the step counting accuracy. Zhang [26] uses the finite state machine method to distinguish whether the pedestrian is at rest or in motion, but does not make a further division of the motion state. In the study of Chen et al. [27], based on the inherent correlation between the state of motion and the maximum acceleration, the maximum acceleration threshold is set to identify the state of motion of pedestrians. With this method, it is easy to misjudge the state of motion under more complex unconstrained conditions. For example, the acceleration caused by the arm swing of the hand-held smartphone during stroll walking is similar to that of normal walking and when the smartphone is flat. Generally speaking, the step frequency of stroll walking, normal walking and running increases in turn, so the state of motion can be identified based on the step frequency. Using FFT, the time domain information can be converted into frequency domain information, and with the exception of the first DC point, the point with the largest amplitude is taken as the step frequency.

In this paper, through 50 experimental tests of 25 people, it is found that the walking frequency of continuous walking should be less than 1.6 Hz, the walking frequency of continuous normal walking should be less than 2 Hz, and the walking frequency of continuous running is 2–3.5 Hz. Limited by navigation factors and non-competitive state, the frequency above 3.5 Hz is mostly caused by interference factors such as body shaking, typing, video brushing, etc. Therefore, the frequency above 3.5 Hz is regarded as an interference state in this paper. When the state of motion is stable, the frequency calculated by FFT will be similar to the real step frequency. To accurately identify the step frequency, FFT requires at least 256 samples. In this paper, when the number of samples is less than 256, FFT judgment is not enabled. When the number of samples is more than 256, the window size of FFT is set to 256 samples, and the sliding time is 1 s (the actual samples in 1 s). Because the FFT used in this paper requires 256 samples (about 5 s), if there is multiple switching of state of motion in the window, it will reduce the accuracy of FFT to judge the step frequency. In order to prevent large step counting errors caused by misjudgment, the frequency threshold should be as small as possible. Theoretically, the time threshold should correspond to the step frequency, but it is found that in the walking state (stroll walking, normal walking), the time required for a single step is similar to that of running. So in order to avoid misjudging some true peaks because of the time threshold, in this paper, the time thresholds of all exercise states are set to smaller values. According to many experimental tests and references, the spectrum, the step frequency threshold, the time threshold and the peak threshold of walking, normal walking and running are shown in Figure 5 and Table 1.



**Figure 5.** Spectrum of pedestrian stroll walking, normal walking and running: (a) Stroll walking (b) Normal walking (c) Running.

**Table 1.** Empirical values of step frequency threshold, time threshold and peak threshold.

Motion State	Step Frequency Threshold (Hz)	Time Threshold (ms)	Peak Threshold ( $m/s^2$ )
stroll walking	[0,1.6]	>333	>10.6
normal walking	[1.6,1.8]	>333	>11.0
running	[1.8,3.5]	>286	>11.6
interference factors	>3.5	—	—

### 3. Smartphone-Based Unconstrained Step Detection Fusing Variable Sliding Window and Adaptive Threshold

In order to address the problem of the low accuracy of step detection algorithms in a complex unconstrained state, an unconstrained step detection algorithm for smartphones is proposed in this paper. In this algorithm, the minimum peak filtered by the sliding window is used as the adaptive threshold. The algorithm is a step detection algorithm of a variable sliding window, which ensures the close connection between the windows. At the same time, the cooperative time threshold solves the problem that the initial peak and the final peak are difficult to distinguish the authenticity in the fixed sliding window. The flow chart of the algorithm is illustrated in Figure 6, and the steps of the algorithm are as following:

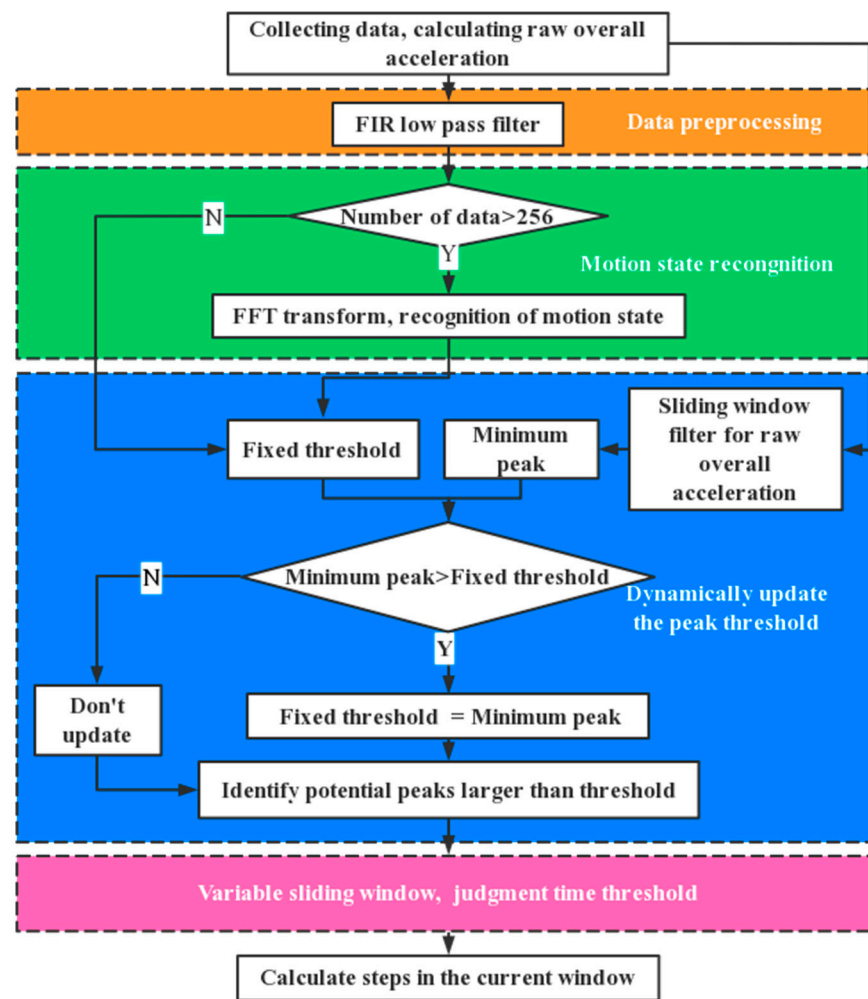


Figure 6. Flow chart of the unconstrained step detection algorithm.

The first step is data preprocessing. The FIR low-pass filter is used to denoise the original overall acceleration signal.

The second step is to identify the motion state and gain the fixed threshold for the third step. The motion state is identified by FFT. The initial peak threshold and time threshold are matched according to different motion states.

The third step is to dynamically update the peak threshold and eliminate the pseudo-peak. It is found that the waveform filtered by a sliding window is smaller than that filtered by a FIR low-pass filter. Based on these characteristics, the minimum peak value detected by sliding window filtering in the window is used in this paper as an adaptive threshold, and the adaptive threshold is used to replace the fixed threshold for step detection based on the waveform after the FIR low-pass filter. This peak threshold dynamic updating algorithm is proposed in detail in Section 3.1.

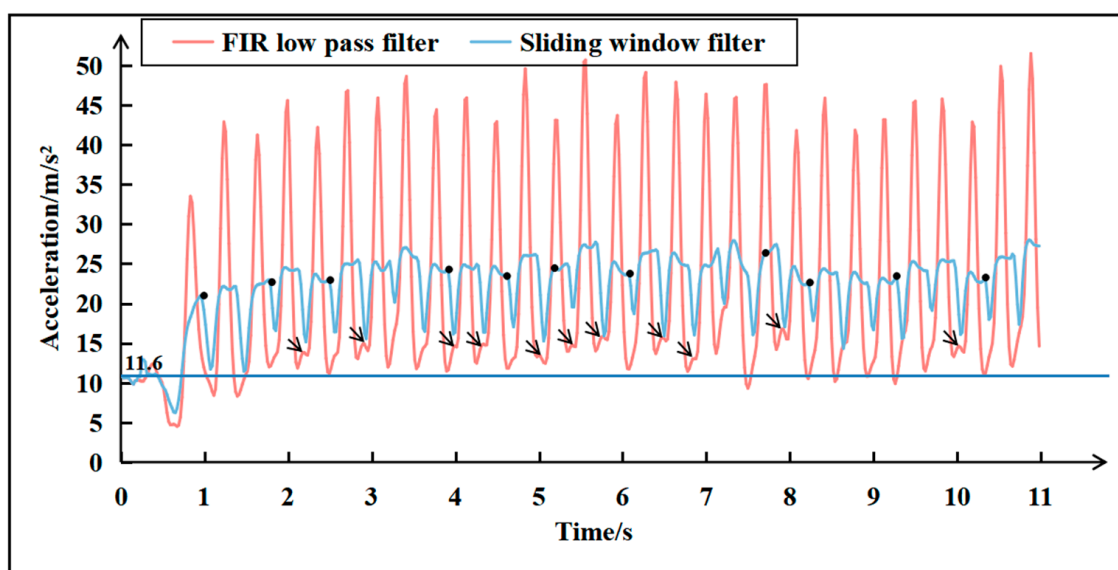
The fourth step is the variable sliding window cooperative time threshold to eliminate the pseudo-peak. Since the fixed sliding window cuts off the connection between the adjacent windows, the discrimination of the initial peak and the final peak of each window is easily missed. To solve this problem, the variable sliding window cooperative time threshold pseudo-peak elimination method is proposed to eliminate the pseudo-peak, and the structure and process of this method will be introduced in detail in Section 3.2.

The fifth step is to calculate the steps for the current window.

### 3.1. Dynamically Update the Peak Threshold

When affected by height, weight, health status, walking habits and other factors, it is difficult to adapt to different users and different unconstrained states only by a fixed threshold. It is found that the waveform filtered by a sliding window is smaller than that filtered by the FIR low-pass filter. To dynamically update the peak threshold, this paper takes the minimum peak detected after a sliding window filter as the adaptive threshold based on this feature. If the adaptive threshold is greater than the empirical threshold, the empirical threshold is used as the adaptive threshold.

Figure 7 is a group of experiments running on flat ground. In the experiment, adaptive peak threshold and fixed peak threshold ( $11.6 \text{ m/s}^2$ ) are used to eliminate pseudo-peak and count steps, respectively. The experimenter ran 29 steps, and the step number results of adaptive peak threshold and fixed peak threshold were 29 steps and 40 steps, respectively.



**Figure 7.** Pseudo-peak elimination by adaptive threshold in running state.

In Figure 7, the arrow marks the pseudo-peak, and the blue horizontal line is the fixed threshold line. The black dots are adaptive thresholds obtained by a sliding window filter. It can be seen that the fixed threshold is difficult to eliminate pseudo-peaks marked by arrows, whereas adaptive thresholds can eliminate them.

### 3.2. Pseudo-Peak Elimination Method with a Variable Sliding Window Cooperative Time Threshold

Because the size of the sliding window determined in this paper is larger than that of the single-step sliding window, there may be several true and pseudo-peaks in the window. The positions of true and pseudo-peaks can be divided into three categories: the initial peak in front of the window, the adjacent peak in the window and the peak at the end of the window.

As shown in Figure 8, the initial peak of the second window and the peak at the end of the ninth window are pseudo-peaks in fact. The fixed sliding window separates the connection between the adjacent windows. If the fixed sliding window of 1 s is adopted, the two pseudo-peaks pointed by the arrow will be misjudged as true peaks. In order to solve this problem, this paper proposes a pseudo-peak elimination method with a variable sliding window cooperative time threshold, which ensures the connection between the adjacent windows, and can judge the starting peak and the end peak. Figure 9 is a flowchart of this method.



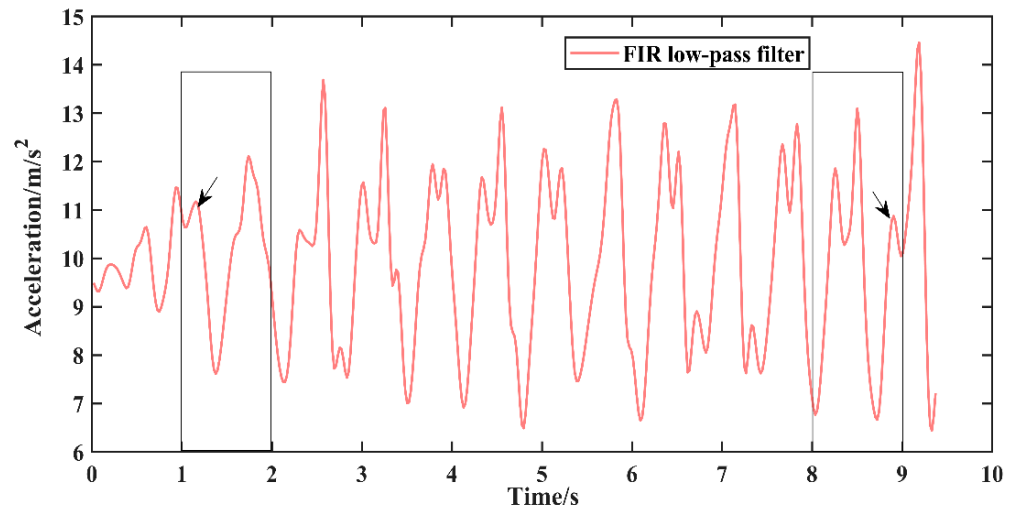


Figure 8. Pseudo-peaks in front of the window and at the end of the window.

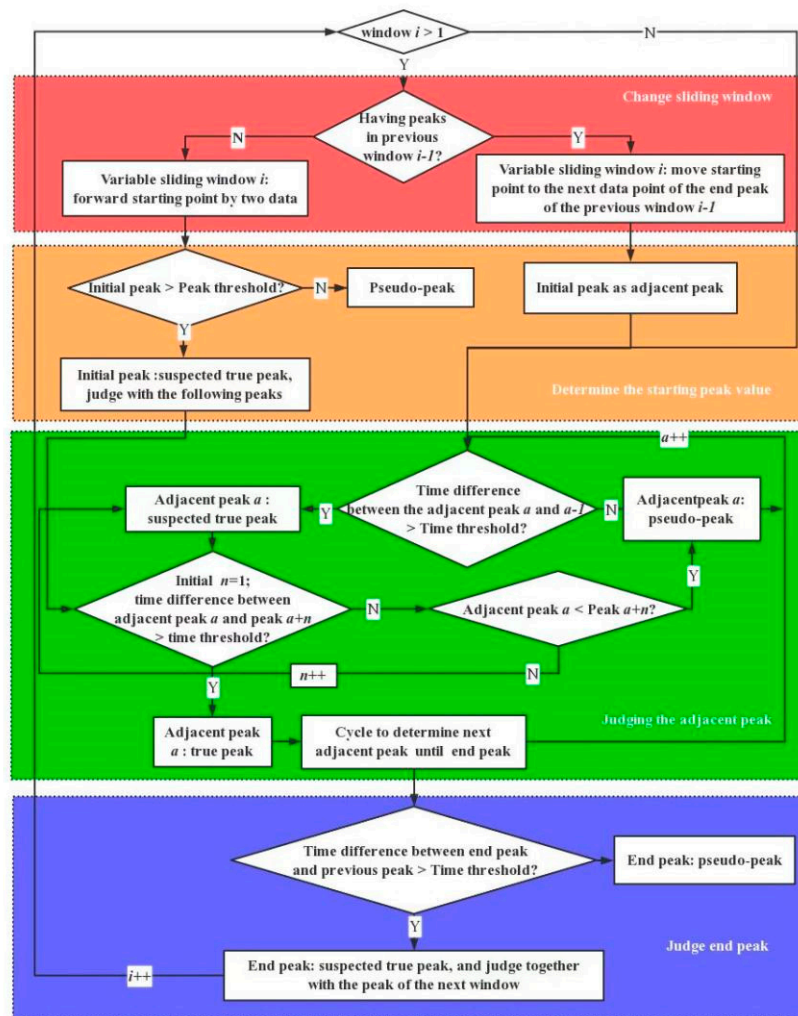


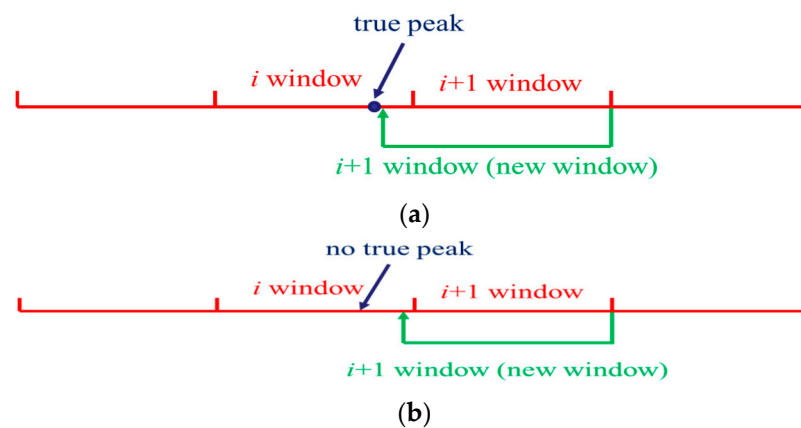
Figure 9. Flow chart of pseudo-peak elimination method with variable sliding window cooperative time threshold.

According to the flow chart, the specific implementation steps of this method are as follows:

Step 1: judging adjacent peaks. If there are two or more peaks in the  $i$  ( $i \geq 1$ ) window, there are neighboring peaks in the window. The time threshold is used to judge the adjacent peaks. If the time difference between peak  $a$  and peak  $a-1$  is less than the time threshold, then peak  $a$  is a pseudo-peak. Otherwise, peak  $a$  is suspected to be a true peak, which needs to be judged cyclically with the subsequent peak  $a+n$  ( $n = 1, 2, 3 \dots$ ). If the time difference between peak  $a$  and peak  $a+n$  is less than the time threshold, then peak value is compared. If peak  $a$  is greater than peak  $a+n$ , peak  $a$  is a suspected true peak, and the program executes  $n++$  to continue the loop judgment. If peak  $a$  is less than peak  $a+n$ , peak  $a$  is a pseudo-peak. If the time difference between peak  $a$  and peak  $a+n$  is greater than the time threshold, peak  $a$  is determined to be the true peak. After achieving the judgment of peak  $a$ , the program ends this cycle and moves to judge the next peak  $a++$ .

Step 2: judging the end peak. If the time difference between the end peak and the previous peak is greater than the time threshold, the end peak is suspected to be the true peak, which needs to be judged together with the starting peak of the next window. Otherwise, the end peak is a pseudo-peak.

Step 3: variable sliding window. If the  $i$  ( $i \geq 1$ ) window contains a peak, the starting point of the  $i+1$  window is set to the next data point of the end peak of the  $i$  window (Figure 10a). If the  $i$  window does not contain a true peak, the starting point of the  $i+1$  window is moved forward two data points (Figure 10b).



**Figure 10.** Schematic diagram of the variable sliding window: (a) The  $i$  ( $i \geq 1$ ) window contains a peak (b) The  $i$  window does not contain a true peak. Note: the green arrows indicate the starting point of new windows.

Step 4: judging the initial peak. If the previous window does not contain a peak, and the initial peak is greater than the peak threshold, the initial peak is suspected to be a true peak, which needs to be judged with the next peak. The judgment method is the same as that of peak  $a$  and the peak  $a+n$  in step 1. If the initial peak is less than the peak threshold, the initial peak is a pseudo-peak. If the previous window contains a peak, the initial peak is equivalent to the adjacent peak, which can be judged by the method in step 1.

Step 5: circularly executing step 1, step 2, step 3 and step 4 until the window ends.

#### 4. Experiment and Analysis

In order to evaluate the accuracy of smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold, 25 volunteers (9 females and 16 males) were recruited. The volunteers' height ranged from 155 cm to 185 cm and the weight ranged from 41 kg to 93 kg. The experimental smartphones were all volunteers' own phones, including 21 models of Huawei, Apple, Samsung, Honor, Xiaomi, Oneplus, Oppo and Vivo. Figure 11 shows 25 smartphone brands and models of the volunteers in the experiment.



Figure 11. Brands and models of 25 smartphones.

#### 4.1. Experimental Setup

The experiment was carried out on the campus of Shandong University of Science and Technology. The four kinds of states of volunteers' motion state, motion environment, smartphone carrying mode and interference factors are shown in Table 2. In total, 25 volunteers were divided into 5 groups to carry out the step detection experiments in a constrained state (smartphones keep flat) and an unconstrained state. The experimental process in a constrained state is shown in Table 3, and the experimental process in an unconstrained state is shown in Table 4. Then, 5 volunteers in each group collected data according to the experimental process of Tables 3 and 4, and accurately counted the actual walking steps. Figure 12 is the schematic diagram of data collected by volunteers in a constrained state, and Figure 13 is the schematic diagram of an unconstrained state. The data collected by the volunteers, as well as attribute information such as height, weight and smartphone models have been uploaded to GitHub (<https://github.com/jackleenotjackma/StepCountingData.git>) (accessed on 25 December 2021).

Table 2. Experimental motion environment, motion state, smartphone carrying mode and interference factor.

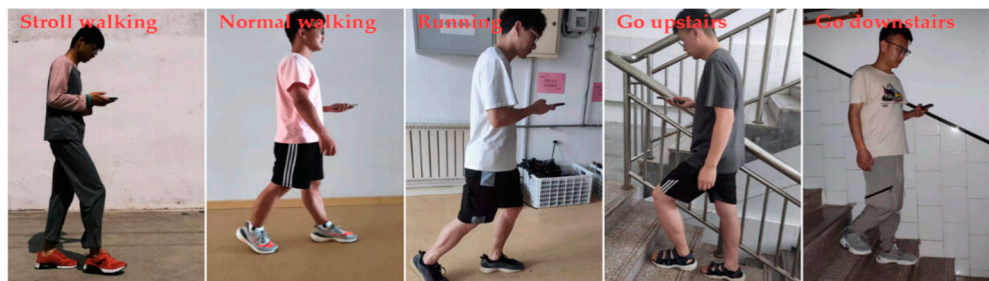
Test Scenario	Motion State	Smartphone Carrying Mode	Interference Factor
Flat ground(A1)	Standing(B1)	Trouser pocket(C1)	Typing(D1)
Go upstairs(A2)	Stroll walking(B2)	Hold and swing hand(C2)	Brush video(D2)
Go downstairs(A3)	Normal walking(B3)	Hand-held and flat(C3)	—
—	Running(B4)	Close to the ear and ringer up(C4)	—

**Table 3.** Experiment process of 5 constrained state groups.

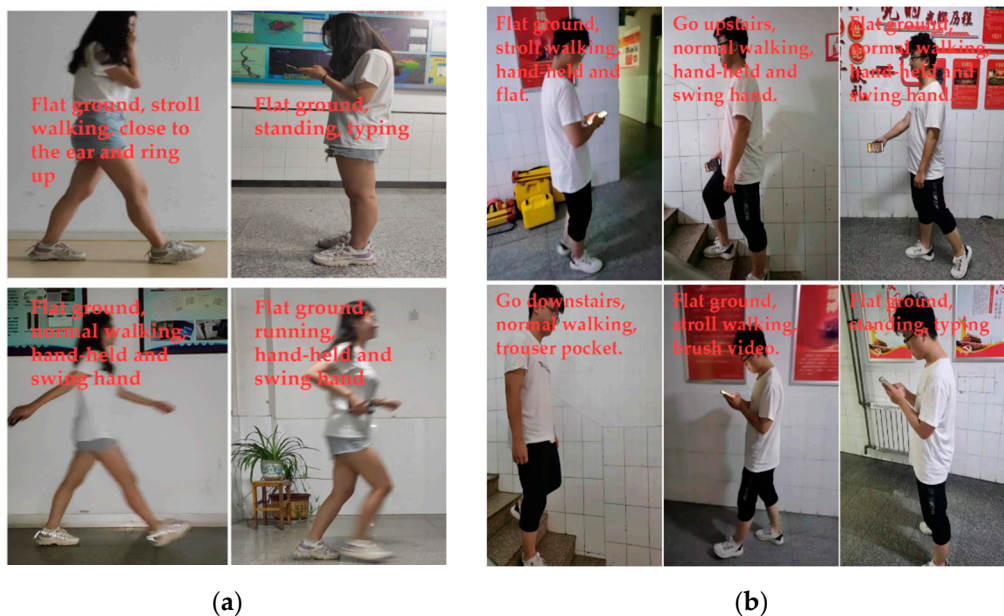
Experimental Group	Experimental Process
Constrained: first group	A1,B2,C3
Constrained: second group	A1,B3,C3
Constrained: third group	A1,B4,C3
Constrained: fourth group	A2,B3,C3
Constrained: fifth group	A3,B3,C3

**Table 4.** Five groups of unconstrained state experiment process.

Experimental Group	Experimental Process
Unconstrained: first group	A1,B1,D2–A1,B2,C2–A1,B3,C3–A1,B4,C1
Unconstrained: second group	A1,B2,C4–A1,B1,D1–A1,B3,C2–A1,B4,C2
Unconstrained: third group	A1,B4,C2–A1,B3,C2–A1,B3,C3–A1,B2,D2–A1,B1,D1
Unconstrained: fourth group	A1,B2,C3–A2,B3,C2–A1,B3,C2–A3,B3,C1–A1,B2,D2–A1,B1,D1
Unconstrained: fifth group	A1,B2,C3–A2,B3,C1–A1,B3,C2–A3,B3,C4–A1,B2,D2–A1,B1,D1



**Figure 12.** Constrained experiment process.



**Figure 13.** Unconstrained experiment process: (a) Schematic diagram of the second group of experimental process (b) Schematic diagram of the fourth group of experimental process.

4.2. Experimental Results and Analysis

A total of 50 sets of data were obtained in the experiment, and the steps were between 72 steps and 343 steps. Unconstrained step count detection algorithm for smartphones is compared with the open source program Stepcount [28], which is well-known on GitHub.

Stepcount is a peak detection algorithm based on an adaptive threshold. The actual step number was taken as the reference value.

The step counting results of 25 volunteers in a constrained state and an unconstrained state are shown in Table 5. Figures 14 and 15 show the step counting accuracy of 25 volunteers in a constrained state and an unconstrained state, respectively. It can be seen from Table 5 and Figures 14 and 15 that under constraint conditions, the lowest step counting accuracy of the proposed smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold is 96.3%, whereas the lowest correct rate of step counting by the Stepcount is only 45.4%. In the unconstrained state, the lowest correct rate of the proposed smartphone-based unconstrained step detection fusing variable sliding window and adaptive threshold is 95.3%, whereas it is only 57.5% for Stepcount. Regardless of the constrained state or the unconstrained state, the step counting accuracy of the proposed smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold is stable, whereas the step counting accuracy of the Stepcount fluctuates greatly. In the constraint condition, the average step counting accuracy of the unconstrained step detection algorithm and the Stepcount are about 99.0% and 90.1%, respectively. In the unconstrained state, the average step accuracy of the unconstrained step detection algorithm and Stepcount are about 98.4% and 87.4%, respectively. In the constrained state and unconstrained state, the step accuracy of the proposed smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold is about 8.9% and 11% higher than that of the Stepcount.

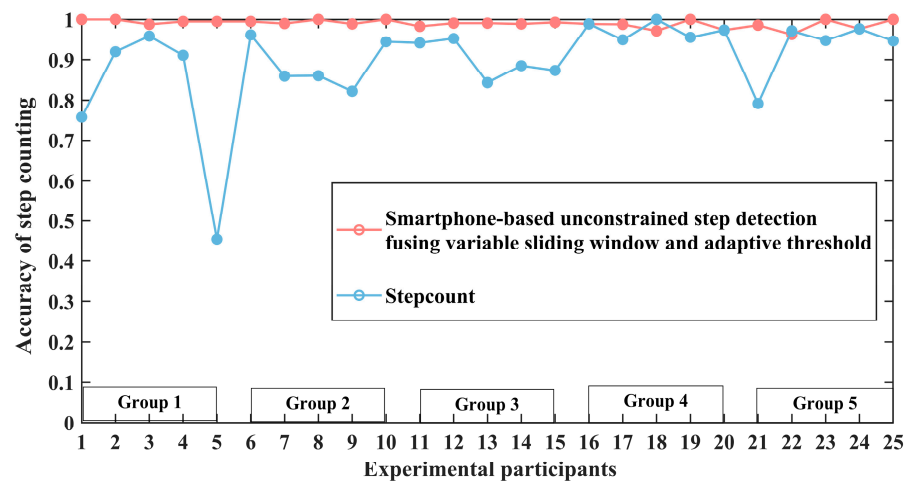


Figure 14. Step counting accuracy in constrained state.

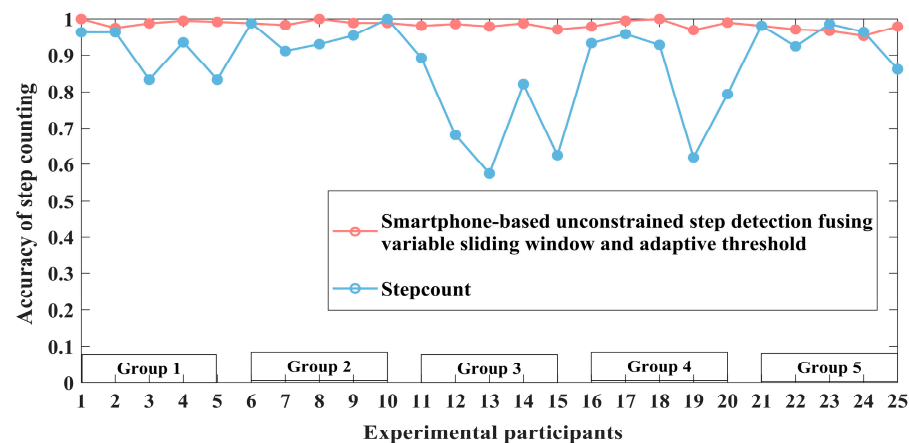


Figure 15. Step counting accuracy in unconstrained state.

**Table 5.** Constrained state and unconstrained state step counting result.

Experimental Group	Actual Steps	Smartphone-Based Unconstrained Step Detection Fusing a Variable Sliding Window and an Adaptive Threshold		Actual Steps	Smartphone-Based Unconstrained Step Detection Fusing a Variable Sliding Window and an Adaptive Threshold		
		Stepcount	Stepcount		Stepcount	Stepcount	
		Correct Rate (%)	Correct Rate (%)		Correct Rate (%)	Correct Rate (%)	
		Constrained State				Unconstrained State	
Group 1	186	75.8	100.0	241	96.3	100.0	
	150	92.0	100.0	240	96.3	97.5	
	170	95.9	98.8	240	83.3	98.8	
	191	91.1	99.5	184	93.5	99.5	
	196	45.4	99.5	240	83.3	99.2	
Group 2	210	96.2	99.5	160	98.8	98.8	
	199	85.9	99.0	180	91.1	98.3	
	150	86.0	100.0	171	93.0	100.0	
	180	82.2	98.9	195	95.4	98.9	
	235	94.5	100.0	182	100.0	98.9	
Group 3	240	94.2	98.3	260	89.2	98.1	
	343	95.3	99.1	210	68.1	98.6	
	230	84.3	99.1	240	57.5	97.9	
	183	88.5	98.9	240	82.1	98.8	
	284	87.3	99.3	240	62.5	97.1	
Group 4	94	98.9	98.9	194	93.3	97.9	
	80	95.0	98.8	212	95.8	99.5	
	107	100.0	97.2	235	92.8	100.0	
	114	95.6	100.0	260	61.9	96.9	
	114	97.4	97.4	209	79.4	99.0	
Group 5	72	79.2	98.6	161	98.1	98.1	
	108	97.2	96.3	170	92.4	97.1	
	96	94.8	100.0	214	98.6	96.7	
	132	97.7	97.7	192	96.4	95.3	
	131	94.7	100.0	237	86.1	97.9	

In order to explore the adaptability of the proposed smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold, the performance of this algorithm is analyzed from the perspective of smartphone price and gender of volunteers.

The smartphones of the 25 volunteers were divided into three types according to prices: mid-low-level (1200–2599 RMB), mid-level (2600–3599 RMB) and high-level (over 3600 RMB). Figure 16 shows the step counting accuracy of the smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold with different prices in the unconstrained state. It can be seen from the figure that the smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold has the best step counting accuracy for high-level smartphones, and there is no obvious difference between mid-level smartphones and mid-low-level smartphones. In the unconstrained state, the average step counting accuracy of the smartphone-based unconstrained step detection fusing variable sliding window and adaptive threshold is 99.1%, 98.0% and 98.0% for high-level, mid-level and mid-low-level smartphones respectively.

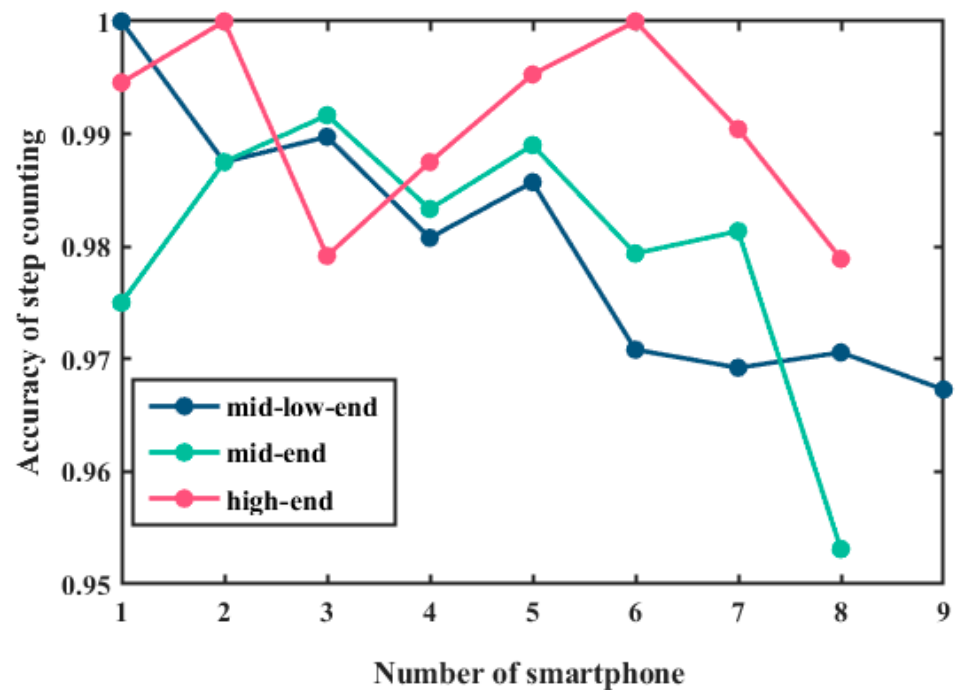


Figure 16. Step counting accuracy rate of smartphones with different prices in unconstrained state.

From a gender perspective, in the unconstrained state, the average step counting accuracy rate of the smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold is 98.2% for male and 98.7% for female, respectively. The difference between them is only 0.5%. It can be seen that the smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold has good adaptability to different genders.

## 5. Conclusions

Aiming at the problem of low step detection accuracy of PDR in an unconstrained state, this paper proposes a step detection algorithm for smartphones. In this algorithm, the pseudo-peaks are eliminated by preprocessing acceleration using a FIR low-pass filter, FFT recognition gait, dynamic updating peak threshold and a variable sliding window cooperative time threshold; and, finally, the step counting is realized.

The FIR low-pass filter is used to denoise the overall acceleration signal. FFT is used to identify stroll walking, normal walking, running and interference state. The minimum peak value after a sliding window filter is used to dynamically update the peak value threshold, which solves for the problem that the fixed peak value threshold has low adaptability in an unconstrained state. A method of a variable sliding window cooperative time threshold is proposed, which ensures the connection between adjacent windows and makes up for the problem that the fixed window cannot judge the initial peak and the end peak. In order to evaluate the step counting performance of the proposed algorithm, 50 experiments in constrained and unconstrained states are conducted by 25 volunteers holding 21 different types of smartphones. The experimental results show that, the average step counting accuracy of the proposed algorithm is 98.4% in an unconstrained state, which is 10.0% higher than that of the result from the open source program Stepcount. This proposed algorithm has a strong ability to adapt to complex unconstrained states and it is friendly for different genders and mobile phones with different prices. In the future, we plan to carry out indoor positioning research with this step detection method and use it in the construction of a smart city and the tracking of pandemics.

**Author Contributions:** Conceptualization, Y.X., G.L. and Z.L.; methodology, Y.X. and G.L.; software, G.L. and Z.L.; validation, Y.X., Z.L., H.Y. and J.C.; formal analysis, G.L., H.Y. and J.C.; investigation, G.L., H.Y., J.C. and J.W.; data curation, G.L.; writing—original draft preparation, G.L.; writing—review and editing, Y.X., H.Y., J.C., J.W. and Y.C.; supervision, Y.X. and Z.L.; project administration, Y.X.; funding acquisition, Y.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Natural Science Foundation of China (42174035), Talent introduction plan for Youth Innovation Team in universities of Shandong Province (innovation team of satellite position and navigation) and Shandong University of Science and Technology school-level scientific research team (2019TDJH103).

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Sobnath, D.; Rehman, I.U.; Nasralla, M.M. Smart cities to improve mobility and quality of life of the visually impaired. In *Technological Trends in Improved Mobility of the Visually Impaired*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 3–28.
2. Nasralla, M.M. Sustainable virtual reality patient rehabilitation systems with IoT sensors using virtual smart cities. *Sustainability* **2021**, *13*, 4716. [[CrossRef](#)]
3. Hazarika, A.; Poddar, S.; Nasralla, M.M.; Rahaman, H. Area and energy efficient shift and accumulator unit for object detection in IoT applications. *Alex. Eng. J.* **2022**, *61*, 795–809. [[CrossRef](#)]
4. Kumar, S.A.; Nasralla, M.M.; García-Magariño, I.; Kumar, H. A machine-learning scraping tool for data fusion in the analysis of sentiments about pandemics for supporting business decisions with human-centric AI explanations. *PeerJ Comput. Sci.* **2021**, *7*, e713. [[CrossRef](#)] [[PubMed](#)]
5. Kuang, J.; Niu, X.; Chen, X. Robust Pedestrian Dead Reckoning Based on MEMS-IMU for Smartphones. *Sensors* **2018**, *18*, 1391. [[CrossRef](#)] [[PubMed](#)]
6. Huang, Y.; Bi, L.; Fu, W.; Yang, R.; Bi, X. A Precise Step Counting Algorithm Based on Acceleration Correlation Analysis. In Proceedings of the 2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, China, 25–27 October 2019; pp. 299–302. [[CrossRef](#)]
7. Xu, L.; Xiong, Z.; Liu, J.; Wang, Z.; Ding, Y. A Novel Pedestrian Dead Reckoning Algorithm for Multi-Mode Recognition Based on Smartphones. *Remote Sens.* **2019**, *11*, 294. [[CrossRef](#)]
8. Lang, A.E.; Hendrick, P.A.; Clay, L.; Mondal, P.; Trask, C.M.; Bath, B.; Penz, E.D.; Stewart, S.A.; Baxter, G.D.; Hurley, D.A.; et al. A randomized controlled trial investigating effects of an individualized pedometer driven walking program on chronic low back pain. *BMC Musculoskel. Dis.* **2021**, *22*, 206. [[CrossRef](#)] [[PubMed](#)]
9. Kang, H.; Lee, B. Comparison of Gait variables and Relative Risk of Falls According to Walking Speed During Flat and Obstacles Walking of Fallers and Non-Fallers in Korean Elderly Women. *Exerc. Sci.* **2022**, *31*, 80–87. [[CrossRef](#)]
10. Du, C. Step-Counting Function of Adolescent Physical Training APP Based on Artificial Intelligence. *Math. Probl. Eng.* **2021**, *2021*, 5582598. [[CrossRef](#)]
11. Xie, D.; Jiang, J.; Wu, J.; Yan, P.; Tang, Y.; Zhang, C.; Liu, J. A Robust GNSS/PDR Integration Scheme with GRU-Based Zero-Velocity Detection for Mass-Pedestrians. *Remote Sens.* **2022**, *14*, 300. [[CrossRef](#)]
12. Wang, Z.; Zhao, H.; Qiu, S.; Gao, Q. Stance-Phase Detection for ZUPT-Aided Foot-Mounted Pedestrian Navigation System. *IEEE/ASME Trans. Mechatron.* **2015**, *20*, 3170–3181. [[CrossRef](#)]
13. Pan, M.; Lin, H. A Step Counting Algorithm for Smartphone Users: Design and Implementation. *IEEE Sens. J.* **2015**, *15*, 2296–2305. [[CrossRef](#)]
14. Cho, Y.; Cho, H.; Kyung, C.M. Design and Implementation of Practical Step Detection Algorithm for Wrist-Worn Devices. *IEEE Sens. J.* **2016**, *16*, 7720–7730. [[CrossRef](#)]
15. Zhang, H.; Duan, Q.; Pan, D.; Bei, H. Integrated iBeacon/PDR Indoor Positioning System Using Extended Kalman Filter. In *Advances in Materials, Machinery, Electrical Engineering (AMMEE 2017)*; Atlantis Press: Amsterdam, The Netherlands, 2017; pp. 9–16. [[CrossRef](#)]
16. Wang, P. Research on Indoor Positioning Technology Based on WiFi and Inertial Sensor. Master's Thesis, Lanzhou Jiaotong University, Lanzhou, China, 2019. [[CrossRef](#)]
17. Ryu, U.; Ahn, K.; Kim, E.; Kim, M.; Kim, B.; Woo, S.; Chang, Y. Adaptive Step Detection Algorithm for Wireless Smart Step Counter. In Proceedings of the 2013 International Conference on Information Science and Applications (ICISA), Pattaya, Thailand, 24–26 June 2013; pp. 1–4. [[CrossRef](#)]
18. Dirican, A.C.; Aksoy, S. Step counting using smartphone accelerometer and fast Fourier transform. *Sigma J. Eng. Nat. Sci.* **2017**, *8*, 175–182.
19. Ning, Y.; Liang, J.; Wang, J. Algorithm of adaptive threshold gait detection in multiple motion states. *J. Chin. Inert. Technol.* **2020**, *28*, 172–178. [[CrossRef](#)]



20. Kang, X.; Huang, B.; Yang, R.; Qi, G. Accurately Counting Steps of the Pedestrian with Varying Walking Speeds. In Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI), Guangzhou, China, 8–12 October 2018; pp. 679–686.
21. Bassett, D.R.; Toth, L.P.; Lamunion, S.R.; Crouter, S.E. Step Counting: A Review of Measurement Considerations and Health-Related Applications. *Sports Med.* **2017**, *47*, 1303–1315. [[CrossRef](#)] [[PubMed](#)]
22. Guo, Y.; Liu, Q.; Ji, X.; Li, G.; Wang, S. Pedestrian Gait Analysis Based on Mobile Phone Accelerometer. *J. Chin. Inert. Technol.* **2017**, *25*, 708–712. [[CrossRef](#)]
23. Alabadleh, A. Step detection algorithm for accurate distance estimation using dynamic step length. In Proceedings of the IEEE International Conference on Mobile Data Management, Daejeon, Korea, 29 May–1 June 2017; pp. 324–327. [[CrossRef](#)]
24. Liu, Y.; Chen, Y.; Shi, L.; Tian, Z.; Zhou, M.; Li, L. Accelerometer Based Joint Step Detection and Adaptive Step Length Estimation Algorithm Using Handheld Devices. *J. Commun.* **2015**, *10*, 520–525. [[CrossRef](#)]
25. Hannah, A.; Agordzo, G.K. A Design of a low-pass FIR filter using Hamming Window Functions in Matlab. *Comput. Eng. Intell. Syst.* **2020**, *11*, 24–30. [[CrossRef](#)]
26. Zhang, H. Research on Bluetooth and Dead Reckoning Based Indoor Positioning Algorithm. Master's Thesis, Chongqing University, Chongqing, China, 2017.
27. Chen, G.; Li, F.; Zhang, Y. Pedometer method based on adaptive peak detection algorithm. *J. Chin. Inert. Technol.* **2016**, *23*, 315–321. [[CrossRef](#)]
28. STEPCOUNT. Available online: <https://github.com/finnfu/stepcount> (accessed on 25 December 2021).