Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.ijamce.com

The Sensor Data Analysis and Pedometer algorithm Design

Aiqiang Yang, Jiaxin Wang, Jianlei Kong*, Tingli Su

Computer and Information Engineering School, Beijing Technology and Business University, Beijing, China

ARTICLE INFO

Article history:
Received 20 August 2018
Accepted 28 October 2018
Available online 25 December 2018

Keywords:
Smart phones
IMU
Gyroscope
Pedometer algorithm
Matlab

ABSTRACT

The development of smart phones continues to drive people's requirements for the diversification of mobile phone functions. As a new tool for people to pay attention to their daily exercise, pedometer has become an important focus of researchers in the functional design of smart phones. This paper mainly acquire and analyze the data of sensors IMU in mobile phones under motion and static state, and use the acquired information to design a new step counting algorithm to make the step counting function more efficient and accurate. This algorithm is more suitable for uniform motion.

1. Introduction

After more than 20 years of development, especially in the past 10 years, with the continuous enhancement of application processor performance and the rapid development of mobile communication technology, mobile phones are no longer a simple communication tool, but people's entertainment and life. In a fast access terminal in the application, the smartphone has profoundly changed the user's lifestyle and become an indispensable part of the user's life[1].

In order to continuously improve the experience of human-computer interaction and meet the needs of users for various life and entertainment, sensors as a basic device for information sensing have been widely applied to smart phones, enabling automatic brightness adjustment, automatic screen change, intelligent anti-missing, and motion.

Sensor makes the functions of smart phones more and more rich, and the human-computer interaction experience is getting better and better. As an important part of smart phones, sensor modules can become a research hotspot in recent years, because they can realize the above basic sensor functions, and on the other hand, because the data collected by each sensor can be used reasonably. Step monitoring, fall detection, behavior recognition and other motion monitoring functions. In particular, the mobile phone step counting function, as one of the application examples of the sensor module, can conveniently and quickly monitor the daily movement of the user[2].

At present, the classical algorithm is based on the mobile phone acceleration sensor peak wave trough detection step counting algorithm, analyzing the walking motion law of the human body, and giving the step counting result of the mobile phone in different postures[3]. Some analysts designed a motion measurement amplitude-based step-by-step algorithm based on zero-speed detection by analyzing the pedestrian motion model and walking posture, which realized the conversion of the carrier coordinate system and the pedestrian geographic coordinate system, eliminating the pseudo zero-speed phenomenon [4].

The structure of this paper is as follows. Section 2 introduces the working principle and application of the sensor related to the step-by-step in the mobile phone. The third section introduces the method of collecting mobile phone sensor data, and analyzes the data of the sensors collected in different states. The fourth section designs different counting algorithms for the characteristics of mobile phone sensors. The fifth section tests and compares the designed algorithms, and gives the experimental results and results analysis, and objectively evaluates the algorithm. The sixth section is conclusion.

2. Cell phone sensor

2.1 IMU

IMU (Inertial Measurement Unit) is a device that measures the angular velocity and acceleration of an object. It usually refers to a combination of three accelerometers and three gyroscopes. The acceleration sensor and the gyroscope are mounted on mutually perpendicular measuring axes, the acceleration sensor detects an acceleration signal of the object in the independent coordinate

* Corresponding author.
E-mail addresses: kongjianlei@btbu.edu.cn (J. Kong)

system of the carrier coordinate system, and the gyroscope detects the angular velocity signal of the carrier relative to the navigation coordinate system, and measures the object in the three-dimensional space. The angular velocity and acceleration are used to determine the pose of an object[5]. IMU has been applied to many fields such as drone navigation systems, motion state pattern recognition[6], etc.

2.1.1 Accelerometer

An acceleration sensor is a device that measures acceleration force by sensing acceleration and converting it into an electrical signal. It is generated based on the acceleration of the motion of the object and the change in capacitance[7]. According to the principle, it can be divided into a linear acceleration sensor and an angular acceleration sensor. The linear acceleration sensor works by using the principle of inertia. According to the formula below,

$$A = \frac{F}{M}$$

where A is the acceleration, F is the inertia force and M is the mass. It is only necessary to measure the inertial force F to obtain the magnitude of the acceleration A. The angular acceleration sensor works by the piezoelectric effect. The external force causes the crystal to deform, and the voltage is generated at the same time of deformation. Therefore, it is only necessary to calculate the relationship between the acceleration and the voltage, and the acceleration can be converted into a voltage form output.

2.1.2 Gyro

The gyroscope is also called an angular velocity sensor, and the rotational angular velocity of the measuring device when it is deflected or tilted. Only the acceleration sensor can detect the linear motion in the axial direction on the mobile phone, and the rotation motion cannot be measured. In addition, the action of the gyroscope can accurately determine the actual motion of the user[8]. The working principle is shown in Fig.1.

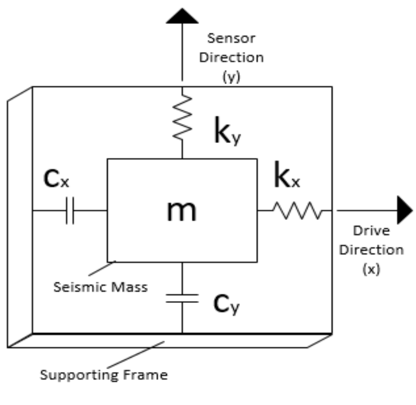


Fig. 1. Measurement Process of Gyro

Actually, the most basic working principle is Hooke's law.

$$F = kx$$

where F is the force, k is a constant factor characteristic of the spring: its stiffness, and x is small compared to the total possible deformation of the spring.

The x direction is the driving direction such that m has a velocity along the x axis, which is known. When there is an angular velocity perpendicular to the xy plane, a Coriolis force is generated in the y direction according to the Coriolis force, and the distance is changed, resulting in a change in capacitance, and the angular velocity of the rotation axis perpendicular to the xy plane can be calculated[9].

3. Data collection and analysis

3.1 Data collection

Install the MATLAB Mobile™ application on an Android device and use the MATLAB Connector to allow a connection between the MATLAB session and MATLAB Mobile™ on an Android device, both of which must be connected by password, DNS or IP on the same network. Then MATLAB can get the data of the sensor in the Android device, and use the MATLAB drawing to visually represent the sensor data.

```
>> connector on
DNS name: bogon
IP address: 10.100.122.223
Use this link to test the MATLAB Connector:
http://bogon:31415/

If the test is successful, but MATLAB Mobile cannot connect,
your computer might have multiple IP addresses.  To determine
the correct one, see Determining the DNS Name or IP Address of a Computer.
```

Fig. 2. Connecting to Matlab Mobile

In the process of collecting data，firstly, using MATLAB Connector to allow connection between the desktop MATLAB session and MATLAB Mobile on the Android device. The device must be able to connect to the desktop, either by being on the same network, using a VPN, or through a similar configuration. Then execute the connector command. The desktop MATLAB shows the details of connecting information, which are shown in Fig.2.

The computer's DNS name and IP address are displayed on the MATLAB Command Window, along with a test link. Click on the test link to ensure that the desktop is ready to connect. The last digits after the colon on the test link correspond to the port that has been opened. After that, the MATLAB Mobile can be connected to the desktop.

```
                   Connected: 1
                     Logging: 0
            InitialTimestamp: ''

   AccelerationSensorEnabled: 0
AngularVelocitySensorEnabled: 0
       MagneticSensorEnabled: 0
    OrientationSensorEnabled: 0
       PositionSensorEnabled: 0
```

```
                   Connected: 1
                     Logging: 0
            InitialTimestamp: ''

   AccelerationSensorEnabled: 1
AngularVelocitySensorEnabled: 1
       MagneticSensorEnabled: 0
    OrientationSensorEnabled: 1
       PositionSensorEnabled: 0
```

Fig. 3. The Connecting State of Sensors

If it is the first time to launch MATLAB Mobile, select Connect to Your Computer. If you previously connected to the cloud or a different desktop, go to the Settings screen and select Add a Computer from the Connect to Your Computers section. On the Add a Computer screen, enter the DNS name or IP address that was displayed in the previous step. Then, enter the Connector password that you previously specified, as well as the port that was opened. Finally, press the Connect button. MATLAB Mobile should now be connected to the desktop MATLAB session. Then use the mobile dev command to create an object that represents your mobile device. The desktop MATLAB shows the details of connecting state of sensors, which are shown in Fig.3.

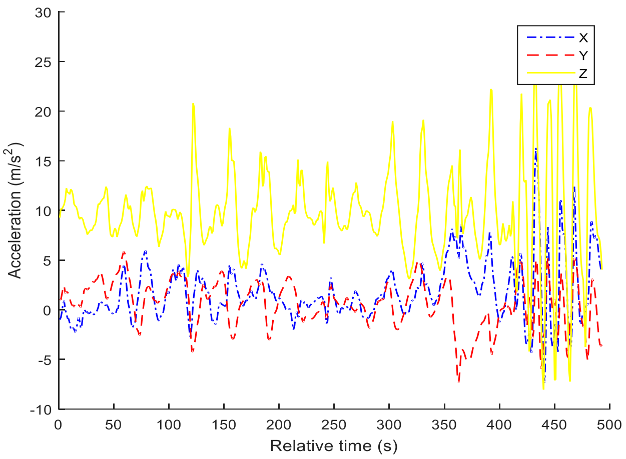


Fig.4 . The measured acceleration data when the sensor is moving

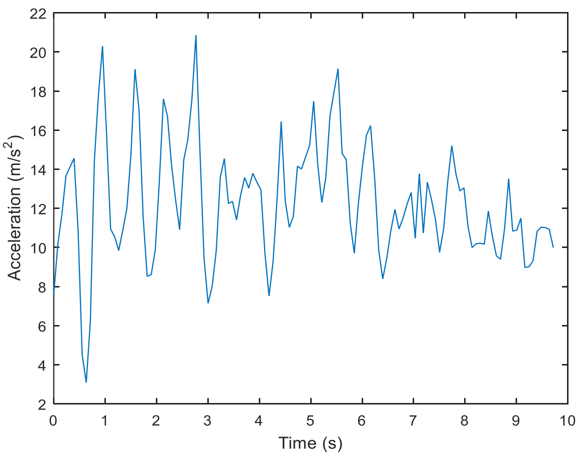


Fig.5. Acceleration scalar value

The displayed output should show Connected: 1, indicating that the mobile dev object has successfully established a connection to the app. Then enable the acceleration sensor on the device. After enabling the sensor, the Sensors screen of MATLAB Mobile will show the current data measured by the sensor. The Logging property allows you to begin sending sensor data to mobile dev. The device is now transmitting sensor data. A pause is included to allow for some measurements to be made before moving to the next steps, but is not required for logging to occur. During logging, the device is held or kept in a pocket while walking around. This generates changes in acceleration across all three axes, regardless of device orientation. The Logging property is used to again to have the device stop sending sensor data to mobile dev. At this point, the data recorded by the mobile phone sensor for this period of time is all transferred to the desktop.

Using accellog to retrieve the XYZ acceleration data and timestamps transmitted from the device to mobiledev. As shown in Fig.4, the logged acceleration data for all three axes can be plotted together.

Then, processing the collected raw acceleration data. To convert the XYZ acceleration vectors at each point in time into scalar values, the magnitude is calculated using the equation below.

$$mag = \sqrt{x^2 + y^2 + z^2}$$

where mag represents magnitude and x, y, z are the three acceleration vectors respectively. This allows large changes in overall acceleration, such as steps taken while walking, to be detected regardless of device orientation. The magnitude is plotted to visualize the general changes in acceleration, which is shown in Fig.5

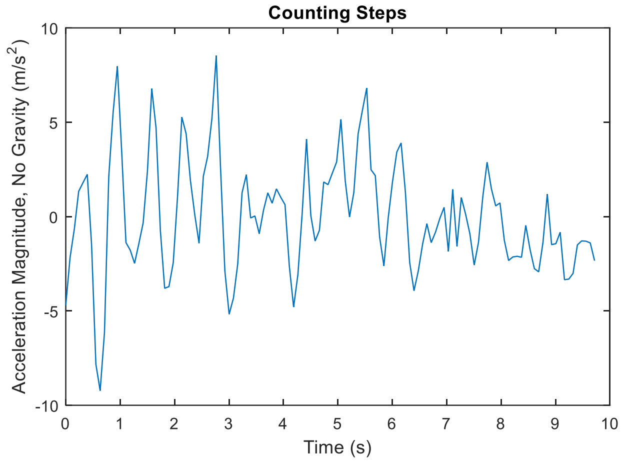


Fig.6. Gravity-free acceleration scalar value

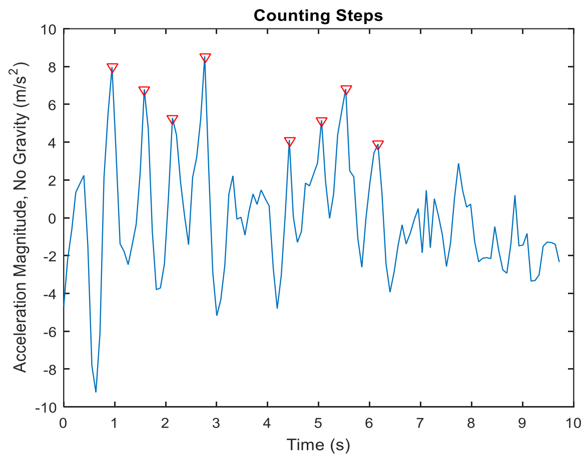


Fig.7. Counting Steps

The plot shows that the acceleration magnitude is not zero-mean. Subtracting the mean from the data will remove any constant effects, such as gravity, using the equation below, and the result isshown in Fig.6.

$$magNoG_i = mag_i - \frac{1}{N}\sum_{k=1}^{N} mag_k$$

where $magNoG_i$ is the ith value of magnitude without gravity, mag_i is the ith value of magnitude calculated before and N is the total number of magnitude values we collected.

The plotted data is now centered about zero, and clearly shows peaks in acceleration magnitude. Each peak corresponds to a step being taken while walking. Findpeaks is a function from Signal Processing Toolbox that is used to find the local maxima of the acceleration magnitude data. Only peaks with a minimum height above one standard deviation are treated as a step. This threshold should be tuned experimentally to match a person's level of movement while walking, hardness of floor surfaces, etc. The number of steps taken is simply the number of peaks found. The peak locations can be visualized with the acceleration magnitude data, shown in Fig.7.

3.2 Data analysis

First analyze the data of the acceleration sensor. The acceleration data of the three axes of the acceleration sensor when the mobile phone is stationary is shown in Fig.4. Since the mobile phone is stationary and flat, there is no force in the X and Y directions, so the acceleration in the X -axis and Y -axis directions is approximately 0; the Z-axis direction has gravity, so there is a gravitational acceleration of approximately 9.8 m/s².

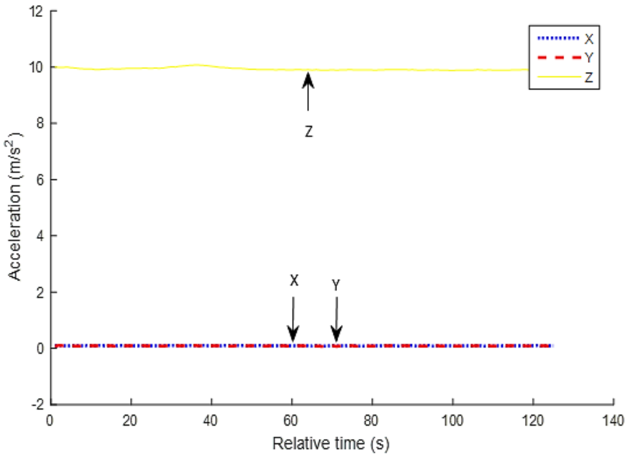


Fig. 8. The measured acceleration data when the sensor is static

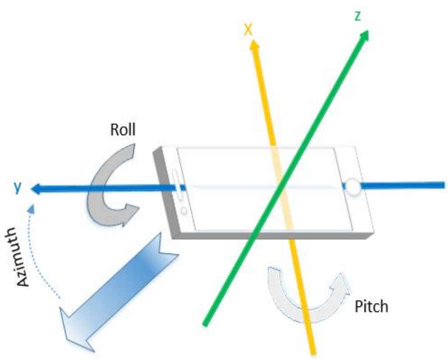


Fig. 9. Mobile phone direction sensor Tri-axis schematic diagram

When the mobile phone is in motion, that is, when the person carrying the mobile phone is walking or running, the acceleration data of the three axes of the acceleration sensor are changed. As shown in Fig.8, the initial acceleration values of the three axes are the respective states in the static state. The acceleration value, walking stride, frequency, leg height, etc. all affect the change of these values, that is, the force acting in these three directions during the walking process is shown in the form of acceleration in the figure, due to the acceleration It changes with changes, so it is different from the data at rest.

The angular acceleration sensor can return the angular acceleration data of the XYZ three axes in units of radians/second. The direction sensor also provides three data, namely azimuth, pitch and roll, azimuth is the azimuth, returning the angle between the magnetic north pole and the y axis when horizontal, pitch refers to the angle between the x-axis and the horizontal plane, and roll refers to the Y -axis and the angle between the water levels. The schematic diagram of the three axes in the mobile phone is shown in Fig.9.

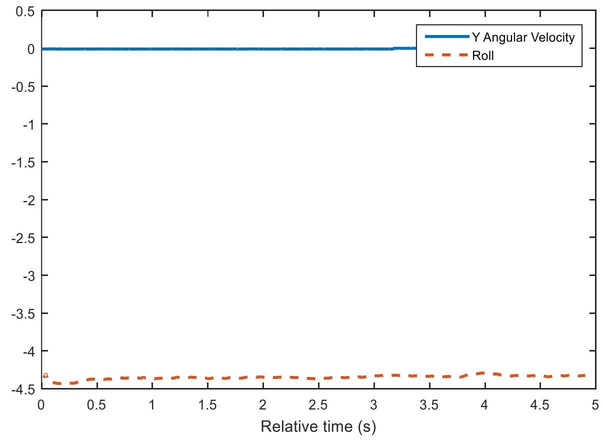


Fig.10. Angular velocity and direction sensor data at rest

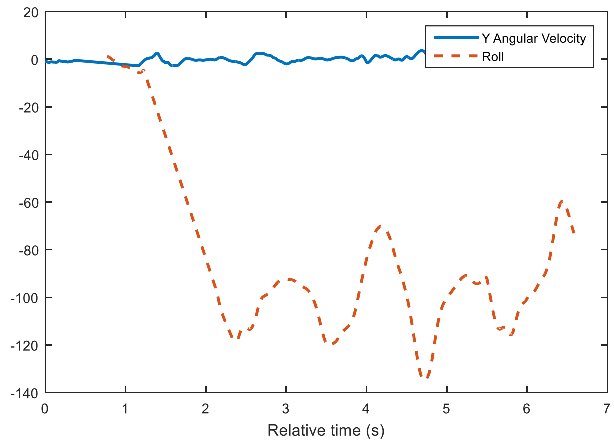


Fig.11. Angular velocity and direction sensor data under motion

Then analyze the data of the angular velocity sensor and the direction sensor when the mobile phone is at rest. As shown in Fig.10, this is the data of the two sensors when the mobile phone is stationary. Here, only the acceleration value of the angular acceleration sensor in the Y -axis direction and the roll value of the direction sensor are displayed.

Since the mobile phone is stationary and flat, no rotational motion occurs, and no angular acceleration occurs in the Y -axis direction, so the angular acceleration value in the Y -axis direction is always maintained at zero. At the same time since the plane of the mobile phone is not absolutely horizontal, roll is the rotation angle of the Y -axis line. Actually, the angle between the X -axis and the horizontal plane is negative when the right axis of the X -axis is higher than the left axis. Therefore, the roll value of the direction sensor at this time is negative, but the value is small.

When the mobile phone is in motion, the data of the angular velocity sensor and the direction sensor will change. As shown in Fig.11, the initial values of the two data are consistent with the values in the static flat state.

When simulating the walking process, the side handshake makes the front of the mobile phone face to the left. It is obviously that the right axis of the X-axis of the mobile phone is higher than the left axis, so the roll value of the direction sensor is a negative value, and correspondingly according to the change of the degree of flipping of the mobile phone. A change has occurred. Since the simulation stops after 15 seconds, the phone is placed flat on the horizontal plane, and the values of both are reset to zero.

4. Pedometer algorithm design

Some scholars combine the peak-to-peak detection algorithm and the traversing intermediate threshold algorithm. The user holds the MEMS device to obtain the accelerometer data, and then the algorithm estimates the number of steps and the trajectory of the walking. The algorithm will judge the possible effective counting points from the acceleration sampling data. Determine whether these points are valid step counts, and then get the number of walking steps. Finally, the walking trajectory of the pedestrian is calculated by combining the step size, the number of steps and the heading information[10]. Others use the zero-crossing detection step-by-step algorithm of the gyroscope, but the step-by-step algorithm will gradually decrease the step accuracy with the increase of the step frequency, and the algorithm has high requirements for the placement position of the mobile phone[11]. Some people analyzed the pedestrian motion model and walking posture, and designed an acceleration measurement amplitude step counting algorithm based on zero-speed detection, which realized the conversion of the carrier coordinate system and the pedestrian geographic coordinate system, eliminating the pseudo zero speed phenomenon[4].

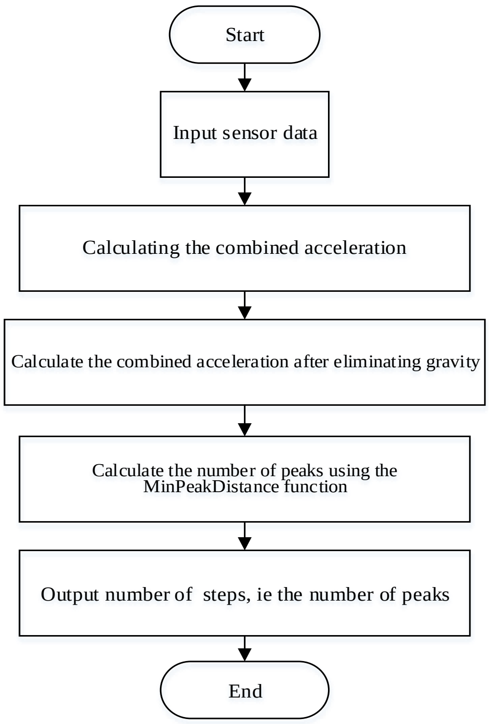


Fig.12. The Flowchart of Algorithm 1

At present, the commonly used step counting algorithm is to convert the vector of the acceleration sensor XYZ three axes into a scalar value, then calculate the step width, remove the gravity acceleration value, find the local maximum value of the acceleration data, and find the local maximum value, that is, the peak value. The quantity is directly equivalent to the number of steps taken. But simply calculating the peak value of the acceleration curve may cause invalid values to be counted, resulting in inaccurate step counting. This method has errors because it does not filter out interference waves, such as subtle vibrations. The algorithm designed in this paper considers the general frequency of human motion and selects multiple peaks in the fixed interval to obtain effective step measurement.

We designed a kernel algorithm and then did some optimization. The process of the two algorithms will be described in detail below.

Firstly, Matlab was used to obtain the three-axis accelerometer data of the smartphone and calculate the resultant acceleration.

Secondly, considering the effect of gravity on the Z-axis acceleration, we calculated the resultant acceleration after removing the gravitational acceleration.

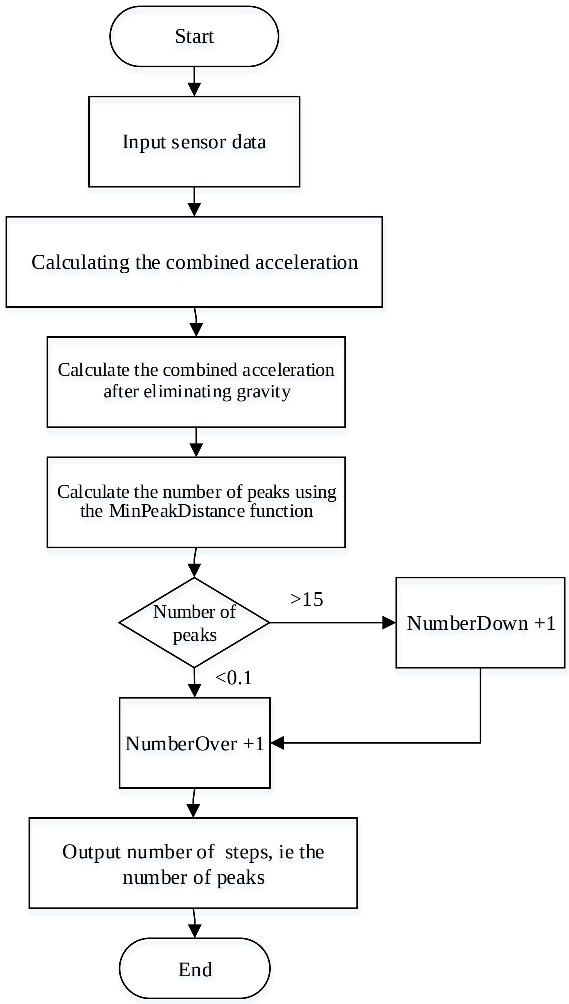


Fig.13. The Flowchart of Algorithm 2

Thirdly, based on the time difference between two steps when a person walks normally, detected the peak value of the resultant acceleration. Finally, the number of peaks detected was the number of steps. The algorithm flow chart is shown in Fig.12.

However, a single consideration of the time difference between the two steps of walking causes a change in the state of motion of the human body, such as a running state, resulting in a large error in the result. On this basis, we have optimized the algorithm, the detailed process is given below.

Firstly, we used Matlab to obtain the three-axis accelerometer data of the smartphone and calculated the resultant acceleration. Secondly, considering the effect of gravity on the z-axis acceleration, we calculated the resultant acceleration after removing the gravitational acceleration again. Thirdly, based on the time difference between two steps when a person walks normally, the peak value of the resultant acceleration is detected, and the mean value of the peak is calculated. Fourthly, considering that the acceleration will not be higher than 0.1 at rest, the running state acceleration will generally exceed 15, and the peak number of the two cases will be calculated separately. Fifthly, when the mean value was less than 10, the motion state was considered to be walking, so the peak number was not calculated in a doubling manner. Instead, the final peak number was calculated in a doubling manner. Finally, we obtained the number of steps. The flow chart of this algorithm is given in Fig.13.

This algorithm considers the algorithm to calculate the number of acceleration peaks in different states.

5. Experimental results

The smartphone used to collect data in this article is OPPO R15, model PACM00, and system model is Android 8.1.0. The software used is the MATLAB Mobile app, and the computer version of MATLAB R2015a.

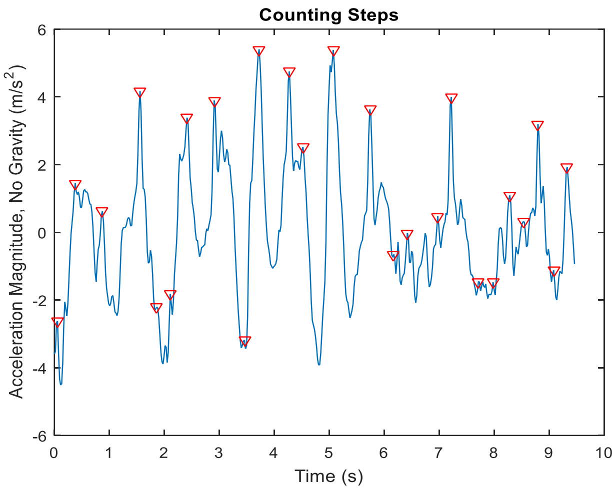


Fig.14. The Peaks of Walking

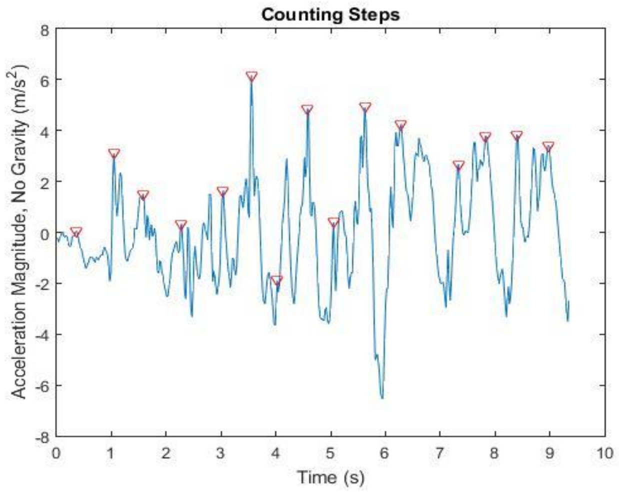


Fig.15. The Firs Original Peaks of Walking with Algorithm 1

Table 1 The Results of Walking with Algorithm 1

Walking			
Number	Real	*Result*	*Error Rate*
1	18	15	16.67%
2	17	14	17.65%
3	19	8	57.89%
4	19	16	15.79%
5	15	14	6.67%
6	16	16	0.00%
7	18	17	5.56%
8	20	13	35.00%
9	14	10	28.57%
10	19	14	26.32%

Table 2 The Results of Running with Algorithm 1

Running			
Number	Real	*Result*	*Error Rate*
1	28	15	46.43%
2	29	12	58.62%
3	27	14	48.15%
4	26	9	65.38%
5	36	14	61.11%
6	34	14	58.82%
7	31	15	51.61%
8	28	14	50.00%
9	20	13	35.00%
10	26	14	46.15%

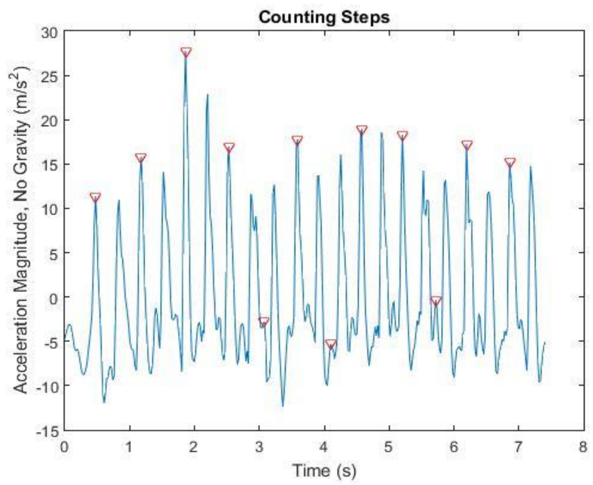


Fig.16. The First Original Peaks of Running with Algorithm 1

Since the normal walking frequency of the human body is 0.5 Hz - 5 Hz an d the single step period is 0.2 s - 2 s, the time difference between the starting points of the two consecutive steps is between 0.2 s and 2 s [12]. The sampling period of the mobile phone sensor used in this paper was 0.019s, and the time of the 12th sampling point was 0.2090s, so it was determined that 12 was the interval of selecting the peak. The sampling time was 10s, and the action was walking. The result is shown in Fig.14. The manual step was 13 steps, and the algorithm result was 25 steps. It can be seen that some errors were also selected as peaks, which causes a large error. By observing the acceleration curve, the interval was changed to 24. Table 1 shows the results of ten tests and Fig.15. gives the first testing result. It is clear that the results were close to the number of manual steps.

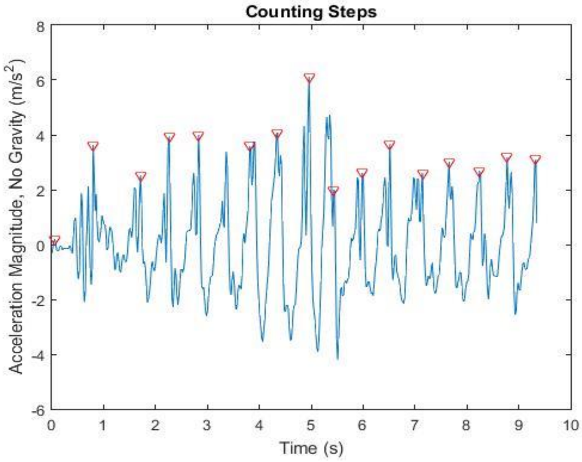


Fig.17. The First Original Peaks of Walking with Algorithm 2

At this point, the motion was changed to run, and we test again. The results are shown in Table 1 and Table 2 and Fig.16. Through the comparison of the results, it was found that the error increases again, which was because some of the effective values were regularly removed due to the large interval. Therefore, based on the division of the action, the number of peaks with the acceleration value more than 15 was recorded, and the case where the acceleration was close to zero was not considered, and the number of peaks less than 0.2 is recorded, and we finally made the rules below to calculate the number of peaks:

(1) When the mean value of the selected peak is greater than 10, that is, when running or brisk walking, the following equation is used to calculate the number of steps:

$$\text{numSteps} = \text{numel(pks)} + \text{NumberOver} * 2 - \text{NumberDown}$$

(2) When the mean value of the selected peak is less than 10, that is,when walking slowly or normally, the following equation is used to calculate the number of steps:

$$\text{numSteps} = \text{numel(pks)} + \text{NumberOver} - \text{NumberDown}$$

where numSteps is the original peak number, NumberOver is the number of peaks whose acceleration exceeds 15, and NumberDown is the number of peaks whose absolute value of acceleration is less than 0.1. This algorithm was used to test the mixing of running and walking. The results have been given in Table 3 and Table 4, and the the first group of testing result are shown in Fig.17 and Fig.18. Table 5 gives a comparison of the results of the different methods combined with the two algorithms. It can be seen that the results of Algorithm 2 are optimal.

Table 3 The Results of Walking with Algorithm 2

Walking			
Number	*Real*	*Result*	*Error Rate*
1	16	15	6.25%
2	19	12	36.84%
3	18	14	22.22%
4	20	6	70.00%
5	11	12	9.09%
6	18	16	11.11%
7	16	10	37.50%
8	20	13	35.00%
9	18	12	33.33%
10	14	13	7.14%

Table 4 The Results of Running and Walking with Algorithm 2

Running and Walking			
Number	*Real*	*Result*	*Error Rate*
1	25	22	12.00%
2	20	14	30.00%
3	21	15	28.57%
4	18	16	11.11%
5	29	11	62.07%
6	26	28	7.69%
7	29	26	10.34%
8	24	25	4.17%
9	23	11	52.17%
10	24	12	50.00%

Table 5 The Comparison of Average

c	*Average Error*
Walking + Algorithm 1	21.01%
Running + Algorithm 1	52.13%
Walking + Algorithm 2	26.85%
Running + Walking + Algorithm 2	26.81%

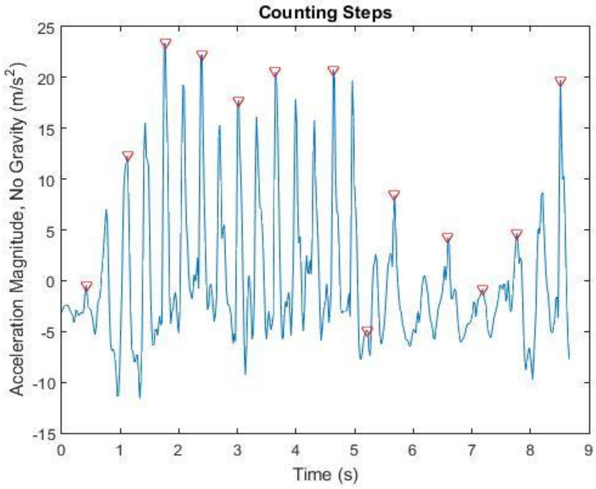


Fig.18. The Original Peaks of Running and Walking with Algorithm 2

6. Conclusions

Pedometers are receiving more and more attention in daily life, and smart phones are now an indispensable device, and the pedometer function included in them has been further developed. In this paper, the mobile phone sensor is briefly introduced, and the pedometer algorithm is designed by using the data of the accelerometer. The test simulation has achieved satisfactory results and the experiments proved that the algorithms are available for uniform motions, such as walking slowly and jogging. If there is an opportunity in the future, algorithm optimization can be performed in terms of the classification of motion types and the fusion of sensor data such as GPS.

References

[1] Xiaorong Chai, Xuemei Lei. 2016. Research on Counting Algorithm Based on Accelerometer and Gyroscope .J. West Leather, 273-277.

[2] Zhijie Duan.2017. Design and Implementation of Mobile Pedometer Based on Android Sensor Module. D. Chongqing University of Posts and Telecommunications.

[3] Haibo Ling, Jing Yang, Xiancun Zhou. 2017. Research on Peak and Trough Step Counting Algorithm Based on Mobile Accelerometer. J. Journal of Sichuan University of Science & Engineering (Natural Science Edition), 21-25.

[4] Guoliang Chen, Zhou Yang. 2017. Step Counting Algorithm Based on Zero Velocity Update. J. Geomatics and Information Science of Wuhan University, 726-730.

[5] Jiandong Wang, Yunhui Liu, Baoquan Song et.al. Design for a PNS and Data Preprocessing for the IMU. J. Electrical measurement & Instrumentation, 19-22.

[6] X.-b. Jin, Xiang N., and Su T., Online Motion Pattern Recognition of Finger Gesture by Inertial Sensor. International Journal of Applied Mathematics in Control Engineering, 2018. 1(1): p. 39-46.

[7] Lei Li. 2012. Research and Application of MEMS Inertial Sensor in Intelligent Remote Control System. D. Ocean University of China.

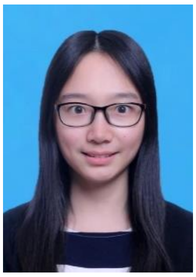
[8] Wei Fei, Zhengshi Liu, Chuanrong Zheng. Basic Principles and Applications of Several Angular Acceleration Sensors. J. Machine Tool & Hydraulics, 149-150.

[9] Xinhua Wu, Hongwei Li.2008. Talking about the Influence and Application of Coriolis Force. J. Journal of Hebei North University (Natural Science Edition), 36-38.

[10] Xizhong Lou, Li Qi, Jun Fang.2017. Study on the pedometer algorithm of MEMS sensor. J. Journal of China Jiliang University, 81-86.

[11] Jayalath S,Abhayasinghe N. 2013. A gyroscopic data based pedometer algorithm. C.Proceedings of the 8th International Conference on Computer Science&Education(ICCSE 2013),Sri Lanka,April 26-28, 551-555.

[12] Bin Huang, Xinhui Wu. 2011. Algorithm Research of Pedometer Based on MATLAB. J. System Simulation Technology, 152-155.



Aiqiang Yang was born in 1996. She is a master degree candidate in School of Computer and Information Engineering in Beijing Technology and Business University. Her major is the control theory and control engineering. She received her bachelor degrees from Beijing Technology and Business University. Her research interests include deep learning and computer vision.

Jiaxin Wang was born in 1995. She is a master degree candidate in School of Computer and Information Engineering in Beijing Technology and Business University. Her major is the detection technology and automation. She received her bachelor degrees from Beijing Technology and Business University. Her research interests include NLP and deep learning.

Jianlei Kong received his B.E. degree in Automation Engineering and the Master degree in Forest Engineering from Beijing Forestry University in 2011 and 2013. In 2016, he received Ph.D. degree of Forest Engineering again from Beijing Forestry University. Since then, he has been a lecturer in College of Computer and Information Engineering, Beijing Technology and Business University. His research lies in the intelligent systems that recognize scenes and obstacles for autonomous robots based on multisensor fusion, supervoxel segmentation and big data tendency analysis.

Tingli Su received her B.E. degree in Mechatronic Engineering and the Ph.D. degree in Control Science and Engineering from Beijing Institute of Technology. Since then, she has been a lecturer in College of Computer and Information Engineering, Beijing Technology and Business University. Her research interests lies in the multisensor fusion, sensor data processing, and big-data based state estimation.