MCEN90032 Project 1: Pedometer using Built-in Accelerometer in Smart Phone

Quang Trung Le (987445)

1 Introduction

1.1 Problem

Pedometer is a device which automatically detects human walking steps. It plays an important role in monitoring the human's daily routine and behavior recognition. There are many methods to design a step counter (SC) and using accelerometer is one of the most popular technique [1]. Because the modern smart phones are equipped with many types of sensors including accelerometer, it can be utilised for SC. However, step counting with smartphone is still challenging [1].

1.2 Aim

The objective of this project is to build a simple pedometer using the data of an accelerometer and count the number of walking step in real-time. For simplification, in this project, the step counter is coded as an application on smart phone (Android or iOS), uses the signal from the build-in accelerometer and shows the steps as people walk.

Although, due to complex behaviours of human (walking,running) and various phone positions, the real pedometer has to deal with different patterns and challenges. In the scope of this project, the phone is put inside a pant's pocket and it will track only the walking activity.

2 Literature Review

Many papers have been discovered and published to solve this problem. They can be divided into three main categories: time domain approaches and frequency domain approaches.

The time domain techniques include thresholding, peak detection, zero-crossing counting and autocorrelation. **Thresholding** is one of the simplest techniques, proposed in [2]. Although thresholding effectively senses at the foot, it is not suitable for various activities because it is difficult to select one optimal threshold for all cases [3]. Meanwhile, **peak detection** and **zero-crossing counting** limitation is dealing with occasional disturbance, which causes many interference peaks [4]. Another technique is **Autocorrelation**, which gives good performance, but resource consuming [4].

Compare to time domain methods, frequency domain approaches such as Fast Fourier Transform (FFT) and continuous/discrete wavelet transforms (CWT/DWT) is robustly more accurate. However, the computational cost is expensive [4].

3 Methodology

3.1 Analysis

As discussed in [5], free movement of smart phone can influence acceleration signal and cause wrong detection in **Thresholding** . However, in the scope of SC with the phone attached inside pant's pocket, this could be a feasible solution. Besides, it is declared that **Peak detection** is one of the best step counting algorithm [4]. Therefore, this project will try to combine two techniques.

The task in this project will follow some steps, which was introduced in [1] and [6]:

1. **Data acquisitions**: The acceleration data is updated in real-time on smart phone.
2. **Data processing**: The raw signal is filtered to obtain the important linear data.
3. **Step detection**: A range of conditions, including Threshold, Time window, Peak detection and first peak detection is checked to obtain the real peaks. The number of steps is then determined to be equals to the number of peaks.

3.2 Algorithm

3.2.1 Data Acquisition

Firstly, because the phone is randomly put into the pockets, it is impossible to track the walking motion by the acceleration in a particular axis of the phone frame. Therefore, the magnitude of the acceleration $magAcc$ is employed to eliminate orientation-dependence. It is calculated using Equation (1), where $xAcc$, $yAcc$, $zAcc$ are the acceleration in x, y and z axis of the phone frame respectively. This parameter will be applied for the later steps.

$$M_{acc} = \sqrt{xAcc^2 + yAcc^2 + zAcc^2} \qquad (1)$$

Figure 1 shows a sample acceleration data of walking activity, where conducted 5 steps within 4 seconds at normal pace. Some features can be extracted from the figure:

- All the steps have a similar pattern.
- All the step have a highest peak followed by a lowest trough.
- The highest peaks occur when the foot hits the floor.
- In each step, there are a few maxima and a few minima.

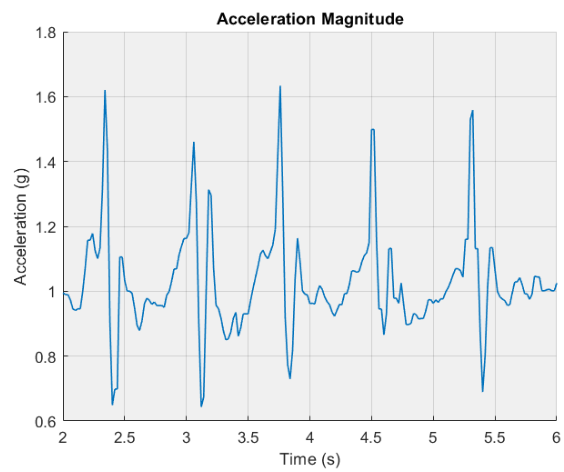


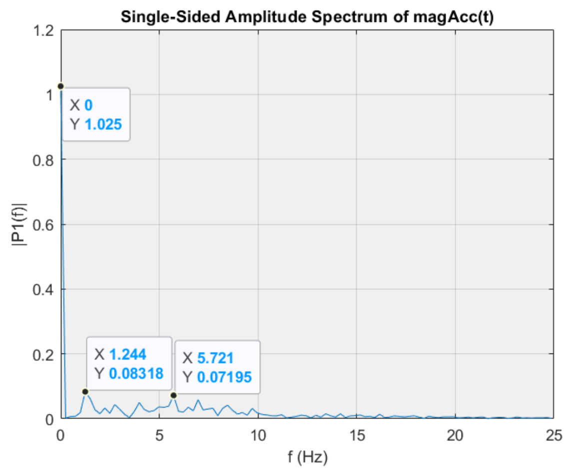
Figure 1: Sample acceleration pattern during walking

The analysis in walking pattern in [7] consolidated the above observations. Besides, the nature of walking stages discovered in [6] also shows that the maximum acceleration in each step is when the foot hit the ground.

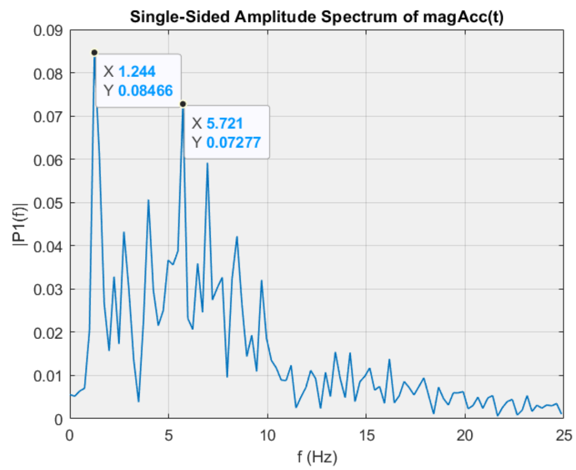
3.2.2 Data Processing

In general, while people walk, the acceleration is the combination of: constant gravity acceleration ($0Hz$) and user-generated linear acceleration ($> 0Hz$). The linear acceleration in x, y and z axis of the phone frame constantly change due to human motion. It can be obtained by eliminating the gravity acceleration from the raw data.

According to [6, 8], the human walking step frequency is in range of 0.5-3 steps/second, which is equivalent to 0.5-3Hz. Ideally, a bandpass filter (0.5Hz - 3Hz) would filter out the linear acceleration. However, in stead of filtering each acceleration component ($xAcc$, $yAcc$, $zAcc$), the filter is applied for the acceleration magnitude M_{acc}.



(a) Frequency analysis on raw data



(b) Frequency analysis after gravity filter

Figure 2: Frequency analysis on sample data

The sample data must be analysed before applying the filter. From Figure 2a, it is obvious that the $0Hz$ is significant in the data. Then, it is essential to filter out the gravity acceleration to obtain the linear acceleration. The remaining acceleration is then analysed, shown in Figure 2b. As can be seen from this figure, apart from the step walking, there is also a considerable range of the vibrations and a filter at cut-off frequency 3Hz would solve. After the band-pass filter is applied, the remaining acceleration magnitude is shown in Figure 3.

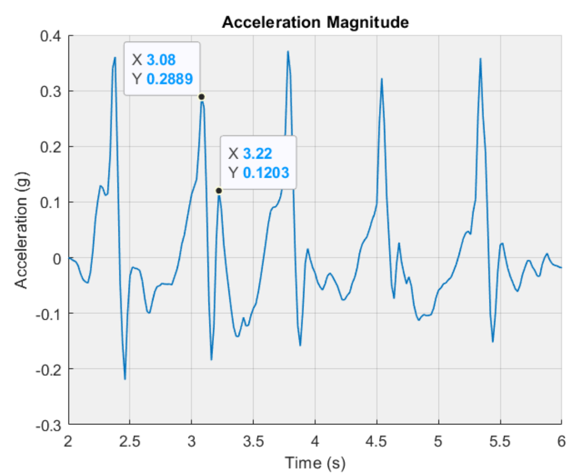


Figure 3: Remaining acceleration magnitude after filtering

3.2.3 Step Detection

This is the key stage to give the accurate step counting. The step is defined as happening when the acceleration meets the "real peak" [6]. The "real peak" in this case is limited by some constraints:

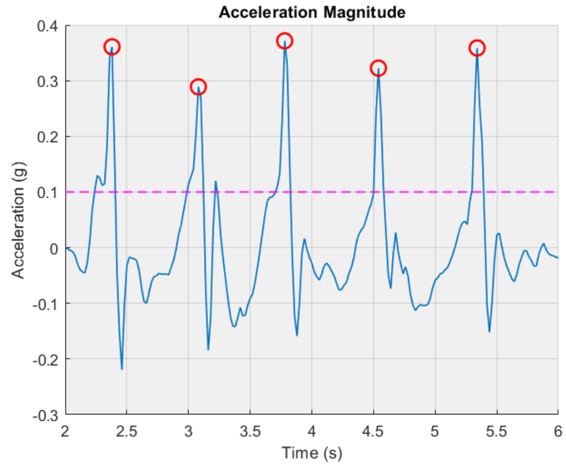


Figure 4: Peak Detection Result for sample data

1. **Threshold** (T_p): A real peak must be larger than a specific threshold. However, this threshold varies for different walking modes. Therefore, numerous experiments in all modes is essential to determine this parameter. For example, in the Normal walking mode in figure 3, all the peaks' magnitude is greater than 0.25. Then threshold could be selected $T_p = 0.25$. However, this parameter must be tested with numerous testcases, especially with slow walking, where the acceleration magnitude is significantly lower than normal and fast walking [1]. In this project, $T_p = 0.1$.
2. **Peak detection**: After reaching the peak, the acceleration decreases. It can be translated into mathematical language as: $M_{acc}[i] < M_{acc}[i-1] \Rightarrow M_{acc}[i-1]$ is a peak candidate. However, the wrong peak can be detected. For example, in figure 3, if $T_p = 0.1$, the first step could be early detected at the lower peak. Therefore, the above condition is not sufficient for peak detection.
3. **Time window** (T_w): The false detection can be overcome by Time window constraint. Since step frequency is $0.5 - 3Hz$, each step duration is 0.33 - 2 seconds. Any potential peak outside of this range will be ignored. This constraint will discard invalid vibrations. In figure 3, because the time between the previous peak (peak 2^{nd} and the false peak is $0.14s$, the false peak will not be counted.
4. **Lone-peak detection**: An occasional movement (not walking) can create a suddenly large acceleration, which should not be counted as a peak. Therefore, if there is no preceding and following sets of steps (within 2 seconds), it will not be counted.

A sample result of step detection is shown in Figure 4.

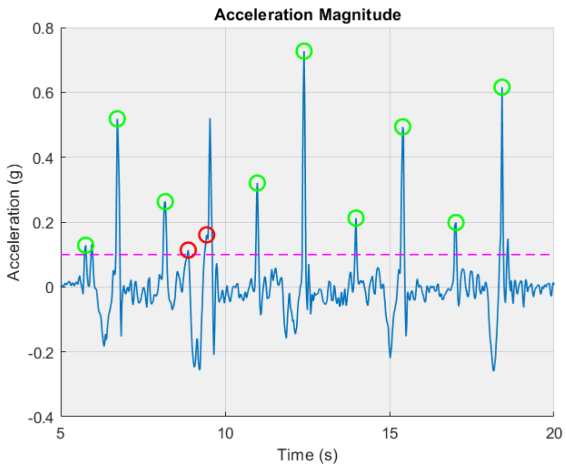
4 Experimental Result and Discussion

To evaluate the efficiency of the algorithm, the experiment was conducted for many times, with different walking modes (slow walking,normal walking, fast walking). The experimental results are shown in Table 1.

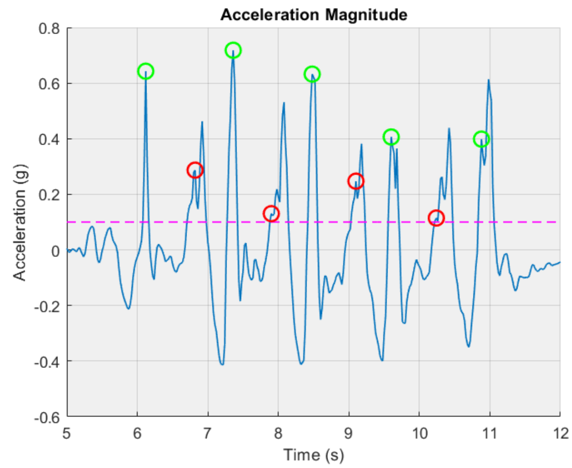
As can be seen in Table 1, the accuracy is significant high in normal walking mode ($> 95\%$). This parameter considerably decreases in slow walking and fast walking, however, it is quite high

Mode/Attempts	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	Average Accuracy
Slow	9/9	10/9	10/9	11/10	12/10	89.6%
Normal	9/9	9/9	10/9	9/9	9/9	97.8%
Fast	11/9	10/9	9/9	10/10	10/9	91.1%

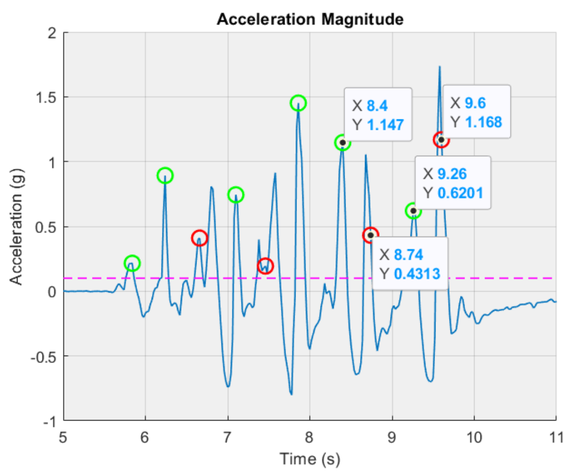
Table 1: Experimental Result on three walking modes



(a) Slow walking



(b) Normal walking



(c) Fast walking

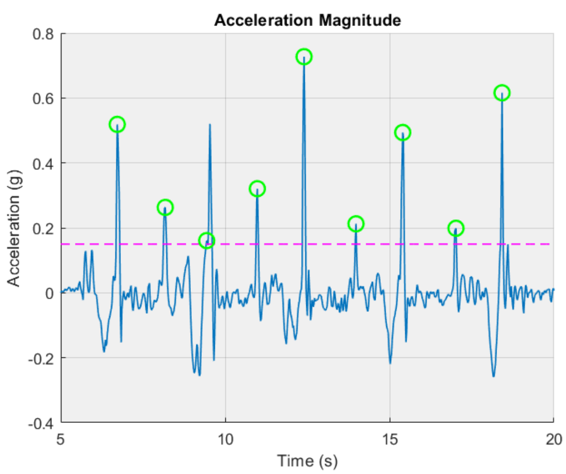
Figure 5: Experimental step detection in different walking modes

(roughly 90%). Analysing the experimental data, the accuracy does not well reflect the the "real peak" detection. Some examples are shown in Figure 6.

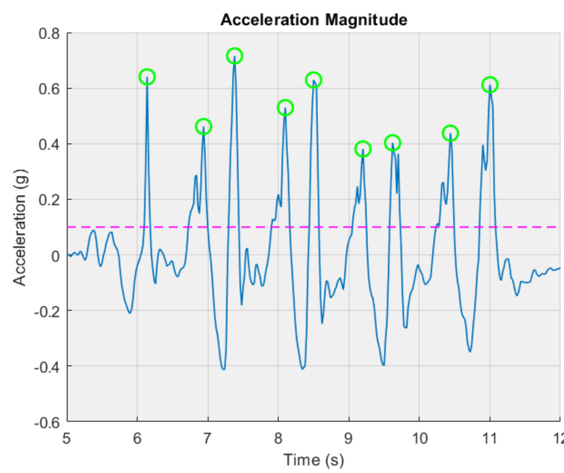
However, a range of issues can be seen in these figures:

- Firstly, the mis-detection can be seen in step 4 in slow walking (figure 5a). Although, the 4^{th} peak is not a real step, the vibration causes the big acceleration magnitude, leading to a step counting. There are many solutions such as: dynamic threshold [6], higher fixed threshold, higher order filter. However, if the threshold increases, using dynamic threshold or fixed threshold, it will discard the low peaks. For example, the real 1^{st} peak in slow walking was discard when the Threshold increases to 0.15, as shown in Figure 6a.

- Secondly, in slow walking, although step 5^{th} is a real step, but the peak is early detected, compared to the real peak. Although this issue does not lead to wrong step counting, it will be a serious problem if the walking pattern is not stable and the vibration is bigger. The false peak detection is also significant in normal walking (peak 2, 4, 6,8) and fast walking (step 3, 5) (as can be seen in figure 5b, 5c). This issue can be solved by "Adjacent peaks selection" [9]. In particular, if there exists more than a peak within a small window (< 0.33 second), the real peak is the peak with the larger magnitude. The new peak detection result is shown in figure 6b, with peak detection accuracy reaches 100%.
- Thirdly, in fast walking, some peak is lately detected (peak 8^{th} and 10^{th} figure 5c). The reason is that, the time window between the pairs of real peaks ($7^{th} - 8^{th}$, $9^{th} - 10^{th}$) is greater than the "Time Window Threshold" ($0.33s$). This threshold was determined based on the maximum walking speed 3 steps/second [8]. Therefore, reducing "Time Window Threshold" will solve the problem. However, it is not consistent with the past papers and the assumptions.



(a) Slow walking with Threshold increase



(b) Normal walking with "Adjacent peaks selection"

Figure 6: Step detection with issues solving

5 Conclusion

Overall, the algorithm conducted in this project returns a fairly well result in step counting (accuracy $> 90\%$). However, it is considerably lower compared to the past researches and there exists many issues. The key reason is the inefficient selection on the threshold for many cases. This limitation has been discussed in many papers [1, 4, 8]. The next step to improve the accuracy is using dynamic threshold instead of fixed threshold as proposed in this project.

References

[1] W. W. Myo, W. Wettayaprasit, and P. Aiyarak, "A more reliable step counter using built-in accelerometer in smartphone," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 2, pp. 775–782, 2018.

[2] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *Journal of Global Positioning Systems*, vol. 3, no. 1-2, pp. 273–279, 2004.

[3] D. Gafurov and E. Snekkenes, "Towards understanding the uniqueness of gait biometric," in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, 2008, pp. 1–8.

[4] X. Kang, B. Huang, and G. Qi, "A novel walking detection and step counting algorithm using unconstrained smartphones," *Sensors*, vol. 18, no. 1, p. 297, 2018.

[5] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013, pp. 225–234.

[6] N. Zhao, "Full-featured pedometer design realized with 3-axis digital accelerometer," *Analog Dialogue*, vol. 44, no. 06, pp. 1–5, 2010.

[7] V. Nagpal, *Android Sensor Programming By Example*. Packt Publishing Ltd, 2016.

[8] V. T. Pham, D. A. Nguyen, N. D. Dang, H. H. Pham, V. A. Tran, K. Sandrasegaran, D.-T. Tran *et al.*, "Highly accurate step counting at various walking states using low-cost inertial measurement unit support indoor positioning system," *Sensors*, vol. 18, no. 10, p. 3186, 2018.

[9] B. Wang, X. Liu, B. Yu, R. Jia, and X. Gan, "Pedestrian dead reckoning based on motion mode recognition using a smartphone," *Sensors*, vol. 18, no. 6, p. 1811, 2018.