Pedometer Placed in the Pocket

5130309753 Junwei Lu

June 26, 2016

Contents

1 Background

This work is an assistant part of the indoor localization project. Some previous work has been done in the scenario of holding the smartphone at hand horizontally when performing the step counting algorithm. In that scenario, users are using smartphone to find the target site via our indoor localization application. A step counted will contribute to some distance of movement in the localization algorithm. However, in this work, we only focus on the step counting algorithm.

In the scenario of the previous work, the gesture of the smartphone is relatively stable. As the smartphone was held at hand horizontally, only the acceleration on the z-axis will change significantly. And in this work, I'm going to study the step counting algorithm when the sensors are placed in the trouser pocket. The scenario is different from the previous work. The users might not always hold the smartphone, however some sensors are put in their trouser pockets. In this way, the pedometer that we design can assist the indoor localization algorithm to keep the trajectory of the user all the time without requiring the users to hold the smartphone at hand.

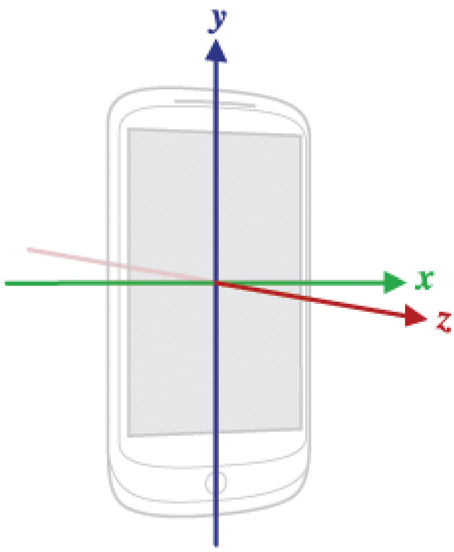

Figure 1: The coordinate system of a smartphone.

2 Basic Approach

2.1 The choice of sensor data

The first work is to choose which sensor data to use. This involves with the shape of the sensor and how the sensor is placed in the pocket. And here, I'll use the onboard sensors of the smartphone as an example and when the shape of sensor is changed, the analysis should be different.

As the smartphone might slide in the pocket, the acceleration data might not be stable. Observing the waveform obtained from the accelerometer when walking in a constant speed along a straight line, we can find that there might be multiple peaks in a step and the magnitude of each steps might be significantly different.

However, as the rotation of the leg is the most significant motion when a person walk, we can utilize the gyroscope data. Considering the gesture of the smartphone, the angular velocity on x-axis might be the most optimal data to utilize.

2.2 Data preprocessing

The waveform of the raw data of the angular velocity on x-axis obtained from gyroscope when a person walk in a constant speed along a straight line has some jitter which would disturb the peak detection. Therefore, some preprocess of the data should be performed to make the waveform smooth.

The first approach is low pass filter. The formula of the low pass filter that we use is shown below where $\alpha \in (0, 1)$.

$$y_n = x_n \times \alpha + y_{n-1} \times (1 - \alpha)$$

And then we are calculating the moving average of the data. A sliding window is maintained which contains the latest k data where k is the window size. The the output value is the average of the data in the window. The formula is showned below.

$$z_n = \frac{x_n + x_{n-1} + \ldots + x_{n-k+1}}{k}$$

The parameters α and k are adjusted via experiments. And the result of preprocessed data is shown in figure 2. As we can see, the preprocessed waveform is quite smooth and can be utilized to perform the peak detection which will be introduced later.

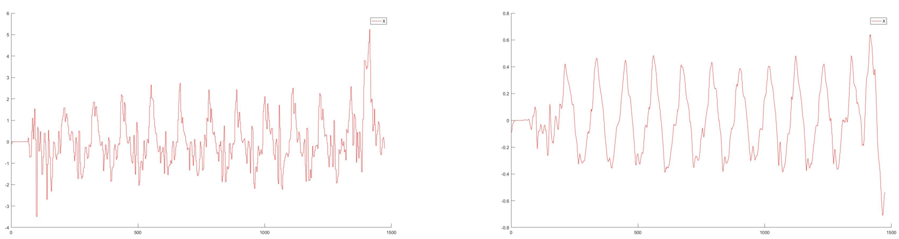

Figure 2: The comparison between the raw data and the preprocessed data

2.3 Peak detection

The step counting algorithm is basically based on peak detection. There are two thresholds that we utilize to detect a peak. The first one is the magnitude threshold. Only the local maximum point above the magnitude threshold could be regarded as a peak. The second one is the temporal threshold. When we detect a local maximum point above the magnitude threshold, we check the time interval between it and the last detected peak. If the time interval is wider than the temporal threshold, the point is regarded as a peak.

In the same way we can also detect the valleys of the waveform and a peak-vally pair represents a step.

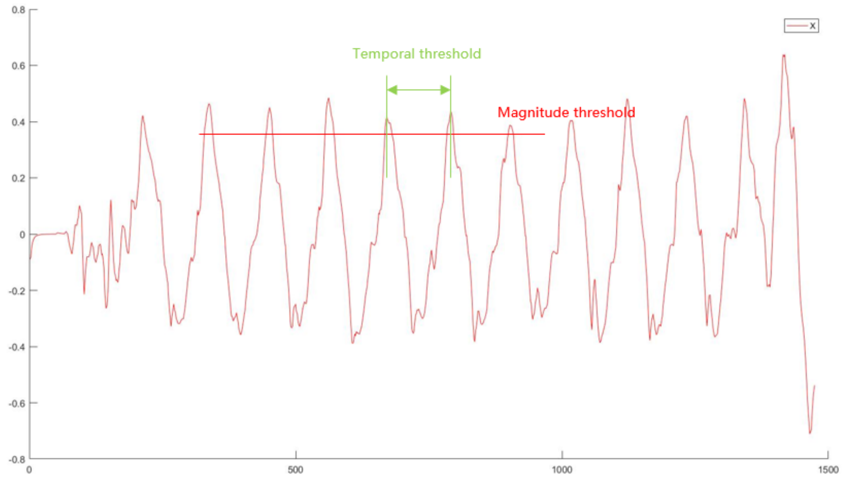

Figure 3: Magnitude and temporal thresholds of a peak detection.

2.4 The effect of the basic approach

I implemented the algorithm as an Android application and perform the test on HTC E8. On average, in each 50 steps, there's no more than 1 step miscounted.

3 Adaptive Threshold Adjustment

As we can see in the previous part, the basic approach performs well when a person walk in a constant speed along a straight line. However, different people might have different magnitude of rotation of the legs. And in different walking mode (walking or running), the magnitude of rotation of the legs might also be different. Therefore, there should be an approach to adjust the magnitude threshold and temporal threshold adaptively.

The basic idea to adaptively adjust the magnitude threshold is to maintain two statistics values μ_a and σ_a. μ_a is set to be the average value of the latest detected peak and valley. And σ_a is set to be the magnitude standard deviation for the recent k samples. And now, the magnitude threshold of peak could be set to be $\mu_a + \frac{\sigma_a}{\alpha}$. The magnitude of the valley could be set to be $\mu_a - \frac{\sigma_a}{\alpha}$.

As for the temporal threshold, the idea is similar. We could maintian two statistics values μ_p and σ_p for peak. μ_p and σ_p are set to be the average and standard variance of the time intervals of the latest k detected peaks. And the temporal threshold of the peak could be set to be $\mu_p - \frac{\sigma_p}{\beta}$. Similarly, we can set the temporal threshold of the valley to be $\mu_v - \frac{\sigma_v}{\beta}$.

Figure 4 shows an ideal case of how the algorithm will adjust the thresholds adaptively and select the peak candidates and valley candidates.

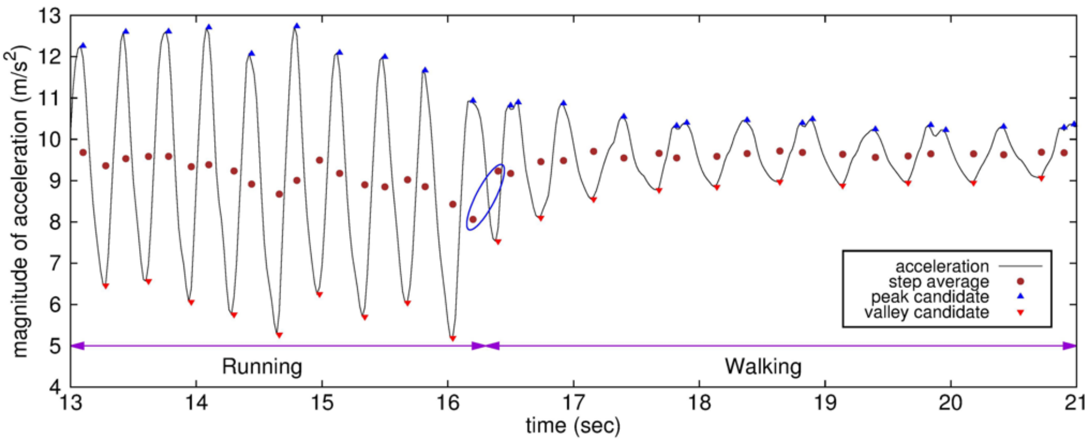

Figure 4: Adaptive threshold adjustment when the walking mode is changed.

References

[1] Step Detection Robust against the Dynamics of Smartphones. Hwan-hee Lee, Suji Choi and Myeong-jin Lee