Estimate Step Count Report

GitHub Link: https://github.com/EshitaKhanna/StepTracker
Video Link: https://drive.google.com/file/d/17Gd66WbGvEO3h9D5EssM_j5o0oleWzQN/view?usp=share_link

Introduction

Step Tracker mobile application estimates the step count for a person, assuming that the phone is held in hand or kept in the pocket, by collecting accelerometer data from the smartphone.

Overview

- **Sensor Used:** Walking involves two stages - The Stance Phase and the Swing Phase [Ref]. The acceleration changes when a person takes a step. When the heel strikes, the body experiences acceleration, and when the foot lifts off the ground a deacceleration is experienced. This change in the magnitude of the acceleration is measured by the accelerometer. Hence, we use Accelerometer data to count the number of steps.

- **Filtering Data:** Accelerometer determines the acceleration by measuring the forces applied to it. However, the force of gravity is always acting which influences the readings of the accelerometer. Therefore to measure real acceleration, the contribution of the force of gravity must be removed from the data. This is done by applying a highpass or lowpass filter to the data [Ref].
  The natural effective frequency of walking range is between 0 -2 Hz (usually) [Ref]. Hence, those frequencies below or above this range are filtered out by applying a bandpass filter on the data.

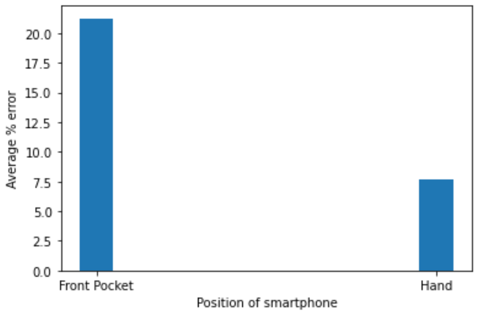
- **Step Counting using Peak Detection Algorithm:** After filtering the data, Peak Detection Algorithm is applied to it. Whenever a step is taken by the user, there is an acceleration produced which is represented by the peaks in the waveform. While walking, the difference in the magnitude of acceleration results in a series of peaks which are detected by the Peak Detection Algorithm and used to estimate the step count for the user.

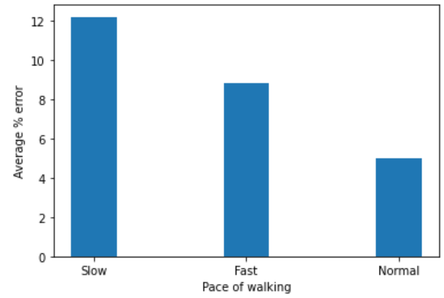
Algorithm

1. The accelerometer data is used to measure the change in the magnitude of the acceleration. This data can be visualized as time-series waveforms, where the peaks of the waveform indicate sudden changes in acceleration, which can be interpreted as steps taken by the person.
2. To measure the actual acceleration effect of gravity needs to be removed. Thus, to remove the contribution of the force of gravity, the mean of acceleration's magnitude is subtracted.
3. Accelerometer measures the acceleration in three directions (X, Y, and Z axes). The data varies according to the orientation of the smartphone. Since we require the total acceleration of the user irrespective of the orientation of the smartphone, we take the magnitude of the acceleration.
4. For walking, the effective frequency is usually less than 2Hz. Hence, bandpass filtering is applied to the time-series waveform to get filtered data that ranges from 0 - 2Hz [Ref].
5. The data is then normalized [Ref].
6. After removing the noises, Peak Detection Algorithm is applied.
   Peaks with a minimum height above one standard deviation are treated as a step. This threshold value is different for every person depending upon the hardness of the ground or the level of movement while walking. The minimum height of the peaks and minimum distance between the peaks is calculated empirically, and Peak Detection Algorithm is then applied. [Ref]
   We take the minimum distance between two windows as 0.25 seconds (4 samples given 16 Hz sampling rate), as the 2 Hz is the upper bound of walking data frequency, and ¼ of the seconds can not have more than one walking phase (stance or swing). Hence, any peaks less than 0.25 seconds apart are considered false peaks.

Offline Data Collection

The data for walking is collected using the Sensor Logger application [Ref]. The data is collected at two locations - in the Hand and in the Front Pocket of the user, and for three different walking speeds i.e. slow, normal, and fast walking speed. The app samples data at 60Hz.
Link to the data: https://github.com/EshitaKhanna/StepTracker/tree/master/data





Link of Python code: https://github.com/EshitaKhanna/StepTracker/blob/master/Step_Tracker.ipynb

Evaluation

The user analysis results, presented in the following tables, compare the accuracy of phone positioning during data collection. When the phone is held in the hand, the results are more accurate, with an average error of 7.66%. However, when the phone is placed in the front pocket of the user, false peaks are added to the waveform, resulting in an average error of 21.25%.

User ID	Gender	Phone location	Ground truth (# steps)	Offline Python Code Steps	% Error
1	Female	Front Pocket	20	23	15
		Hand	20	20	0
2	Male	Front Pocket	10	13	30
		Hand	10	9	10
3	Female	Front Pocket	16	19	18.75
		Hand	15	13	13

In the following table, the accuracy of data collection is compared for different walking speeds. The results indicate that walking at a normal or faster pace gives more accurate results compared to walking at a slower pace. Walking too slowly results in an average error of 12.22%.

User ID	Gender	Speed (Phone in hand)	Ground truth (# steps)	Offline Python Code Steps	% Error
1	Female	Slow	20	23	15
		Fast	20	20	0
		Normal	20	19	5
2	Male	Slow	12	13	8.33
		Fast	15	14	6.66
		Normal	10	9	10
3	Female	Slow	15	13	13.33
		Fast	10	8	20
		Normal	10	10	0

Android App Implementation

App code: https://github.com/EshitaKhanna/StepTracker/tree/master/app/src/main

The Step Tracker application has been implemented in Java.

Limitations

The application is accurate to a certain extent but it has certain limitations. Placing the phone in the pocket or rotating the phone may add false peaks in the waveform which results in extra steps. Walking too slowly also shows false results.

Future Work

The Step Tracker app can be further extended to count the steps more accurately and to overcome the existing limitations. As the walking data is very periodic, finding periodicity between consecutive windows can solve these problems.