IMU-Based Walking Workouts Recognition

Fanuel Wahjudi
EECS International Graduate Program
National Chiao Tung University
Hsinchu City, Taiwan
fanuelwahjudi.edu@gmail.com

Fuchun Joseph Lin
Department of Computer Science
College of Computer Science
National Chiao Tung University
Hsinchu City, Taiwan
fjlin@nctu.edu.tw

***Abstract*— To better accurately estimate the calories burnt during popular walking workouts, it is essential to detect the environment under which these workouts are conducted. To our best knowledge, no gait analysis studies have been done so far for such detection. This research addresses this problem by recognizing walking workouts under different environments based on the foot-mounted inertial sensor. Our objective is to recognize ten different workout activities including walking and brisk-walking under flat surface, ascending/descending staircase and upward/downward slope with no stairs. Our algorithm first identifies the extended foot-flat phase, then uses it as a boundary to extract key important features. Decision Tree, Random Forest and K-Nearest Neighbor machine learning algorithms are evaluated to decide which one works the best along with our algorithm.**

Keywords—Gait analysis; activity recognition; environment detection; walking workouts; machine learning algorithms

I. Introduction

Recently, with the emergence of the Internet of Things (IoT), the development of wearable devices to enable innovative IoT applications is growing rapidly. These wearable devices are often integrated with Inertial Measurement Unit (IMU) including accelerometer and gyroscope. They have been utilized for many applications in a large variety of forms such as watch, glasses, bracelet, ring and even shoes

In this research, we focus on the wearable application of shoe-mounted IMU. Our objective is to recognize whether a walking workout activity is walking or brisk-walking and under what environment this activity is being performed based on the data collected from IMU. We intend to recognize ten different types of walking workouts including walking and brisk walking under five different environments: (1) flat surface, (2) upward in stairs, (3) downward in stairs, (4) upward in a slope (5) downward in a slope.

All these are popular and natural activities in maintaining a person's health and each of these requires a different level of effort which leads to a different result of calories burnt. One of the primary aspects on maintaining health is to keep the calories intake equal to the calories burnt, achievable by regular exercises. By more precisely recognizing each type of walking workout activities, we would be able to more accurately determine the calories burnt by these most popular daily exercises.

To accomplish the recognition, we focus on developing algorithms for (1) identifying the "extended foot-flat phase" because it can be used as a boundary for feature extraction, and (2) extracting useful features that can be used to detect a particular type of walking workout activity. The machine learning models applied in our research include Decision Tree, Random Forest and K-Nearest Neighbor. These models are evaluated to determine which one works the best for our research objective.

The rest of this paper is organized as follows: Section II surveys related work. Section III explains the gait phase. Section IV describes our proposed recognition algorithms. Section V describes the system design and implementation. Section VI shows our experimental results and evaluation. Finally, in Section VII we provide our conclusion and possible future work.

II. Related Work

Signal segmentation and feature extraction are two important techniques in achieving our research objectives. A survey of both techniques is presented below.

A. Signal Segmentation

The fixed window approach has been used by researchers for many years. However, some studies show that this approach has the shortcoming when dealing with an action with mutable and various frame lengths such as gait signal, due to various walking speed of different people. To solve this issue, it is normal to evaluate the longest possible length of an expected gait cycle [3]. Sang et al. [2] used a window length of 2 and 4 second as the expected length for classification while Kwapisz et al. [10] chose 10 second in order to provide sufficient time to capture several repetitions of motion.

The dynamic window approach has been proposed to tackle the issues of the fixed length approach where different window lengths based on the detection of gait cycle is used. Muaaz and Rene [9] estimated the length of a gait cycle by detecting the minimum Euclidean distance. Ghobadi and Esfahani [5] applied an equation to determine segmentation based on the resting states and the velocity magnitude. Chen et al. [7] extracted the peaks in X-axis accelerometer to find the foot flat period, then defined the distance between peaks as one segment.

978-1-5386-4980-0/19/$31.00 ©2019 IEEE

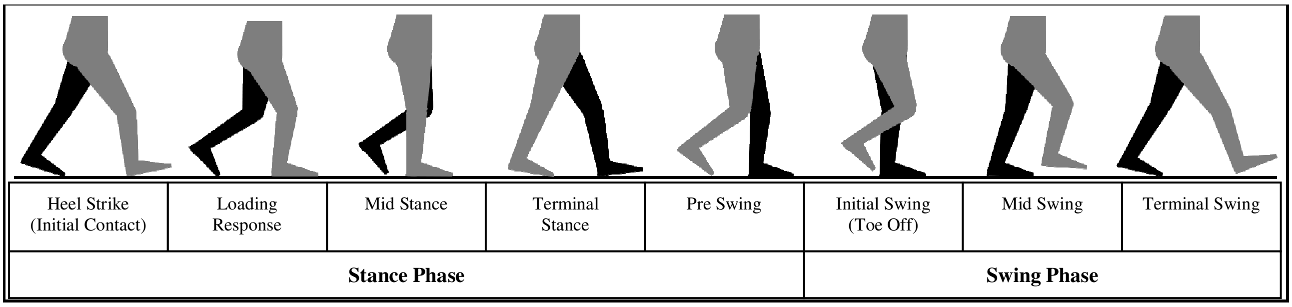

Figure 1. Gait Cycle of Non-Stair Walking

B. Feature Extraction

Feature extraction can be done either in the time domain or in the frequency domain. Feature extraction in time domain is generally based on statistical parameters such as cycle length, zero-crossing, local extrema, mean and standard deviation [3][4][7][10]. Sang et al. [2] in their paper proposed fractal dimension as the feature and evaluated its effectiveness.

Feature extraction in frequency domain requires the use of Fourier transform operations in order to convert a signal from time domain to frequency domain [6]. Among Fourier transform operations, Discrete Wavelet Transform (DWT) and Fast Fourier Transform (FFT) methods are the ones that are most commonly used [3][7].

C. Recognition Algorithm

Ngo et al. [8] relied on the likelihood of heel strike and proposed to use interclass relationship in the feature vector for recognizing five similar gait action classes (namely walking on flat,up/down stairs, and up/down a slope).

On the other hand, we rely on the disengage of Extended Foot Flat (EFF) phase and non-EFF phase to generate specific features for doing similar recognition. In experiments, the proposed algorithms were able to positively recognize ten different walking workouts (namely walking and brisk-walking on flat, up/down stairs and up/down a slope).

III. Gait Phase

In general, when conducting walking and/or brisk-walking, we are doing a repetitive gait cycle pattern. To comprehend this cycle pattern, the gait phases are defined to depict an entire cycle. The gait cycle is a lapse between two consecutive repetitive stages of walking which can be divided into two phases: (1) Stance Phase that occurs when the foot is in contact with the ground surface, and (2) Swing Phase that occurs when the foot swings in the air [1][17][18][19]. Different environments may lead to different sub-phases for each phase. There are different motion patterns in the stance phase and the swing phase for walking upward in stairs, walking downward in stairs, and walking in non-stair cases.

Walking in flat, ascending slope and descending slope are all categorized into non-stair walking. These activities in general have similar movement patterns, as depicted in Figure 1. The stance phase includes five sub-phases: initial contact (heel strike), loading response (foot flat), mid stance, terminal stance, and pre-swing (toe-off) [1][17][19] while the swing is divided into three sub-phases: initial swing, mid swing, and terminal swing [1][17][19].

For both walking ascending and descending a stair, the stance and the swing phases can be divided into three sub-phases and two sub-phases, respectively. However, those sub-phases for ascending and descending are different. The subphases of stance phase for ascending include weight acceptance, pull-up, and forward continuance while the ones for descending include weight acceptance, forward continuance, and controlled lowering [13]. On the other hand, the subphases of swing phase for ascending include foot clearance and foot placement while the ones for descending include leg pull through and foot placement subphases [13].

IV. Recognition Method

Below we introduce the extended Foot Flat (EFF) phase and describe its role on the recognition. Then an algorithm for the EFF phase detection is explained. Finally, we examine gait characteristic of each activity.

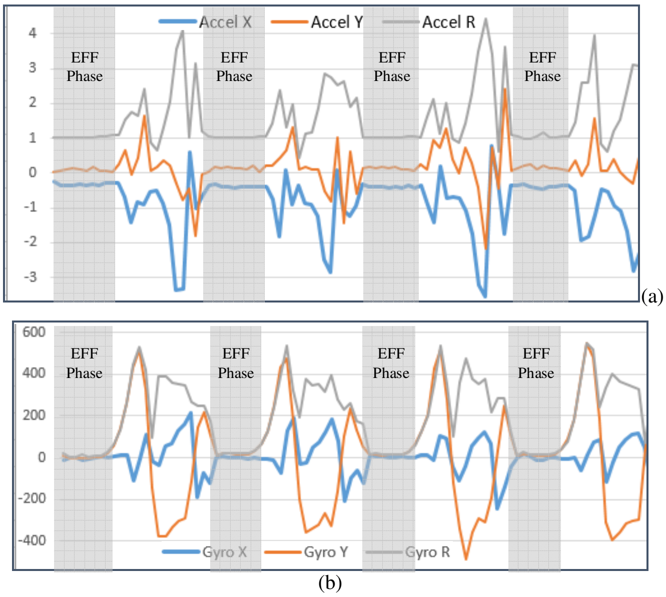

Figure 2. X,Y-axis, and Resultant Signals of (a) Accelerometer and (b) Gyroscope

A. *Extended Foot Flat (EFF) Phase*

Extended Foot Flat (EFF) Phase is a part of Stance Phase when the foot is in full contact with the ground surface. In the non-stair walking such as flat walking, this phase includes both Loading Response (Foot Flat) and Mid Stance subphases (See Figure 1). During the EFF phase, the foot is in stationary form for a small period of time. This phase can thus be identified clearly from both accelerometer and gyroscope signals. As depicted in , accelerometer produces signals with low dispersion and stable values while gyroscope signals will reach zero in the EFF phase due to its stationary nature. The capability of accelerometer in sensing the orientation of a stationary object can be further used to explicate the degree of ground slope such as flat, ascending slope or descending slope.

However, the EFF phase doesn't provide any clue to distinguish walking workouts of ascending and descending the stair. Fortunately, these activities are distinguishable at their swing phase. The swing phase of non-stair walking begins with lifting the foot then swinging it forward at the same level. On the other hand, the swing movement in ascending stair and descending stair will move the foot upward and downward, respectively. This swing phase is outside the EEF phase and thus will be called Non-EFF Phase hereafter.

Distinguishing the EFF Phase and Non-EFF phase holds the key for the recognition of various walking workouts in different environments. By designing an algorithm to detect the Extended Foot Flat phase, we can set the boundaries for feature extraction. Based on carefully selected features, algorithms can be developed to accurately recognize various walking workouts in different environments.

B. *EFF Detection Algorithm*

Our proposed algorithm consists of four major steps: (1) obtaining the resultant value of gyroscope, (2) applying normalization, (3) finding the burst numbers of successive zeros, and (4) examining the distribution of x-axis values from accelerometer to determine whether the periods identified in the previous step can be categorized as an EFF. Details of the detection procedure are discussed below:

1. Acquire the resultant value of gyroscope using the following equation:

$$Gr^2 = Gx^2 + Gy^2 + Gz^2$$

2. Normalize the obtained resultant value. The following formula is applied to restrict the value range of resultant signals between zero (0) and an arbitrarily chosen point a:

$$norm = round\left(a.\frac{x - x_{\min}}{x_{\max} - x_{\min}} \right)$$

3. Find the burst numbers of successive zeroes. The normalization generates zero values during the EFF phase. These burst numbers of successive zeroes are marked as the potential candidates of the EFF phase. Figure 3 shows an example of marked successive zeros (depicted as shadows) in normalized resultant values of gyroscope.

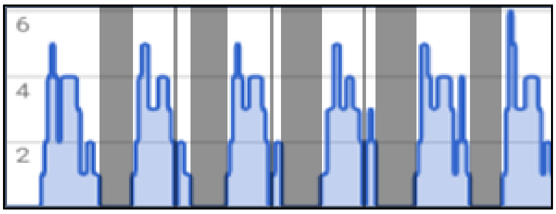

Figure 3. Marked successive zeros

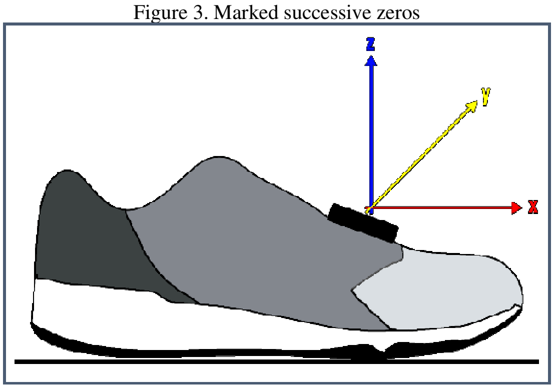

Figure 4. IMU placement and axis orientation on the shoe

4. Examine the distribution of x-axis values from accelerometer. As it is also possible to produce successive zero values during the non-EFF phase, we need to measure the stability of x-axis values of the accelerometer and define the threshold during the period of discovered successive zeros to determine whether the period appertains to the EFF phase. The parity of stable periods in both accelerometer and gyroscope signals during the EFF phase can be utilized as the portent of the EFF phase.

C. *Gait Characteristic*

To provide proper features for recognizing whether an activity is walking or brisk-walking, and identify under what environment this activity is being performed, below we discuss the key distinctions of these activities by examining the pattern divergence of their gait signals.

1) *Walking vs Brisk-walking*

Between brisk walking and normal walking, the movement speed is one of the most striking differences. The speed difference can be shown on both accelerometer and gyroscope signals. The faster the walking pace, the higher the absolute value of their signal produced in general. Brisk walking thus has a higher absolute values in both sensors. Calculation of mean value in both sensors can be used as a typical feature to distinguish between walking and brisk walking.

2) *Non-stair walking vs stair walking*

Stair walking such as ascending stair and descending stair would have different gait cycles than non-stair walking. The key to distinguish ascending stair, descending stair and non-stair walking lies in the swing phase that is contained in the non-EFF phase. Both accelerometer and gyroscope signals show pattern discrepancy on these activities in the non-EFF phase. Figure 5 compares the gyroscope signals of these activities.

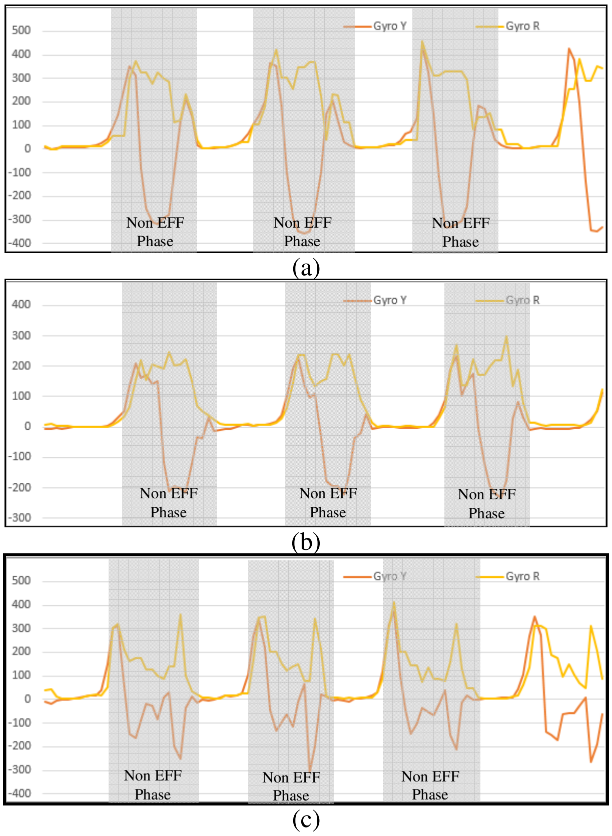

Figure 5. Gyroscope Signals of Walking in (a) Non-Stair, (b) Ascending Stair and (c) Descending Stair

The key difference between these activities is their signal fluctuation, particularly along x-axis, z-axis of accelerometer and y-axis of gyroscope. Consequently, statistical calculations such as standard deviation and variance along those signals in the non-EFF phase can play a key role as they can measure how much signal fluctuates.

3) Flat walking vs slope walking

Flat walking and slope walking (both ascending slope and descending slope) are categorized under non-stair walking. They share the same gait cycle that can be proven from similar movement patterns in both accelerometer and gyroscope signals. It is thus difficult to distinguish these activities based on both accelerometer and gyroscope gait signals.

However, an accelerometer sensor is able to perceive the orientation of a stationary object. In the EFF phase, the foot is flat with the ground surface for a moment. Hence, an accelerometer is able to project the slope of the surface. The accelerometer data on z-axis and x-axis in the EFF phase can be used to tell the track environment where walking workout is conducted, whether in flat, ascending slope or descending slope. Mean value of accelerometer signals in x-axis during the EFF phase is the key.

V. IMPLEMENTATION

Below we discuss the implementation details including system design, sensor configuration, data preprocessing and test design.

Figure 6. Sensor placement on top of the shoes

A. System Design

Our system consists of three devices: an IMU sensor as the wearable device, an Android smartphone, and a desktop PC. An explanation of these three devices is presented below.

1) Wearable Device

In this study, a wearable device is required to be worn by the user on the toe cap of his/her shoe. We decide to use MetaMotionR Board developed by Mbientlab[15], for its low cost and strong development support. It collects and streams raw sensor data during walking workouts and sends those data to a smartphone over Bluetooth. Mbientlab provides open source Software Development Kit (SDK) to enable the utilization of the board [20].

2) Android Smartphone

Due to the need to stream raw sensor data, a smartphone is utilized as a bridge between the wearable device and a desktop. The smartphone uses Bluetooth Low Energy (BLE) to get streaming data from the wearable device. In our experiment, we use a Samsung Galaxy S5 with Android 6.0.1 Operating System (OS).

3) PC Desktop

After the raw sensor data has been acquired and stored in the smartphone, those collected data then are transferred to the desktop for further analysis. The analysis includes data preprocessing and data classification.

B. Sensor Configuration

Sensor placement is very crucial here, different placement leads to orientation inconsistency between the training and test stages. To solve this problem, we eventually decide to ask every subject to wear the same pair of shoes. Moreover, the sensor device is also attached at the same place on the top of laces near the toe vamp of the shoe as shown in Figure 6.

C. Data Preprocessing

Due to its simplicity and effectiveness, we applied overlapping fixed window segmentation, also known as sliding window segmentation, for data preprocessing. In this segmentation, a window represents data sampling at certain timestamp. We decided to use 40 windows in a segment, as it would provide sufficient time to capture the longest possible length of an expected full gait cycle.

Our method extracted seven statistical features in time domain including (1) Mean, (2) Variance, (3) Standard Deviation, (4) Root Mean Square, (5) Zero Crossing Rate, (6) Skewness and (7) Kurtosis values.

Through the data preprocessing, we acquire four signals each from accelerometer and gyroscope including the signals from x, y, and z axes and the resultant value calculated from x, y, z. Thus eight data signals are obtained. However, based on the configured position of sensor (see Figure 4), the angular major movement during walking workouts occurs on y-axis of gyroscope and it would affect the orientation of x-axis and z-axis of accelerometer. Therefore, these signals are the keys for the recognition of the ten workouts under our study. Moreover, resultant signal of both sensors are also worthwhile since their values are obtained from all three axes.

After passing through the EFF phase detection and data segmentation, we are able to classify each of segment windows into three categories: General windows, EFF windows and non-EFF windows.

Table 1. Selected Features

42 Selected Features			
1.	Mean of Accelerometer X in General Windows	22.	Mean of Accelerometer R in non-EFF Windows
2.	Mean of Accelerometer Z in General Windows	23.	Variance of Accelerometer R in non-EFF Windows
3.	Mean of Accelerometer R in General Windows	24.	SD of Accelerometer R in non-EFF Windows
4.	Mean of Gyroscope Y in General Windows	25.	ZCR of Accelerometer R in non-EFF Windows
5.	Mean of Gyroscope R in General Windows	26.	RMS of Accelerometer R in non-EFF Windows
6.	Mean of Accelerometer X in EFF Windows	27.	Skewness of Accelerometer R in non-EFF Windows
7.	Mean of Accelerometer Z in EFF Windows	28.	Kurtosis of Accelerometer R in non-EFF Windows
8.	Mean of Accelerometer X in non-EFF Windows	29.	Mean of Gyroscope Y in non-EFF Windows
9.	Variance of Accelerometer X in non-EFF Windows	30.	Variance of Gyroscope Y in non-EFF Windows
10.	SD of Accelerometer X in non-EFF Windows	31.	SD of Gyroscope Y in non-EFF Windows
11.	ZCR of Accelerometer X in non-EFF Windows	32.	ZCR of Gyroscope Y in non-EFF Windows
12.	RMS of Accelerometer X in non-EFF Windows	33.	RMS of Gyroscope Y in non-EFF Windows
13.	Skewness of Accelerometer X in non-EFF Windows	34.	Skewness of Gyroscope Y in non-EFF Windows
14.	Kurtosis of Accelerometer X in non-EFF Windows	35.	Kurtosis of Gyroscope Y in non-EFF Windows
15.	Mean of Accelerometer Z in non-EFF Windows	36.	Mean of Gyroscope R in non-EFF Windows
16.	Variance of Accelerometer Z in non-EFF Windows	37.	Variance of Gyroscope R in non-EFF Windows
17.	SD of Accelerometer Z in non-EFF Windows	38.	SD of Gyroscope R in non-EFF Windows
18.	ZCR of Accelerometer Z in non-EFF Windows	39.	ZCR of Gyroscope R in non-EFF Windows
19.	RMS of Accelerometer Z in non-EFF Windows	40.	RMS of Gyroscope R in non-EFF Windows
20.	Skewness of Accelerometer Z in non-EFF Windows	41.	Skewness of Gyroscope R in non-EFF Windows
21.	Kurtosis of Accelerometer Z in non-EFF Windows	42.	Kurtosis of Gyroscope R in non-EFF Windows

The defined seven features can be extracted from the signals data in these three different windows to get a lot of combination features. However, not all of these are relevant features. Based on the key distinction explained in Section IV.C, only 42 features would be selected as listed in Table 1.

Feature Numbers 1 to 5 are the key features to distinguish between walking and brisk-walking. Features Numbers 6 and 7 are the key features to distinguish between ascending slope, descending slope and flat environments. And the rest are the key features to distinguish between ascending stair, descending stair and non-stair environments.

D. *Dataset and Test Design*

In our experiment, we ask a total of seven participants to perform ten different workouts in flat surface, stair and slope with no stair. Before performing these walking workouts, each participant is required to wear the same shoes with an attached IMU on the top of the right shoe. For each workout activity, participants are asked to perform the activity in their normal natural pace.

One of participants is asked to perform twice for each workout. For this participant, one set of data collected is used for training while the other set is used for testing. On the other hand, the rest of participants are asked to perform just once for each workout and the data thus collected are used for training. In total we generated 21987 lines for training dataset and 3018 lines for testing dataset.

Our experimental walking workouts are conducted in flat surface, stair and slope with no stair. These three places can be extended into five different environments for our testing.

VI. Result & Evaluation

We apply and compare three different machine learning algorithms including Decision Tree, Random Forest and K-Nearest Neighbor. Three models are built using the selected 42 features in Table 1 and the accuracies of these three machine models are shown in Table 2.

Decision Tree, Random Forest, and K-Nearest Neighbor (KNN) models perform at an average accuracy of 89.16%, 97.11%, and 63.38%, respectively. Generally speaking,

Table 2. Result of Accuracy Testing

Model	Environment	Activity	
		Walking	*Brisk-walking*
Decision Tree	Ascending Slope	66.25%	94.08%
	Ascending Stair	95.45%	62.90%
	Descending Slope	99.50%	90.80%
	Descending Stair	99.09%	83.33%
	Flat	94.99%	85.52%
Random Forest	Ascending Slope	91%	90.14%
	Ascending Stair	100%	100%
	Descending Slope	95.79%	100%
	Descending Stair	100%	100%
	Flat	98.54%	100%
K-NN	Ascending Slope	48.5%	39.90%
	Ascending Stair	97.72%	96.77%
	Descending Slope	41.58%	40.80%
	Descending Stair	95.45%	88.09%
	Flat	80.18%	56.20%

Random Forest provides the highest accuracy. Compared to Decision Tree, it builds an ensemble of Decision Trees and combines them to increases the overall result. It thus overcomes several problems with decision trees, including overfitting and variance between the training and testing data. Hence it normally provides better results than Decision Tree.

On the other hand, compared with KNN, Random Forest can identify what variables are important in the classification while KNN cannot. KNN is based on feature similarity that applies distance metric techniques such as Euclidean distance to find the most similar features. Since we only provide very few features to distinguish non-stairs walking workout, it thus leads to low accuracy on these activities. In the gait analysis area, KNN is more suited for user recognition.

VII. Conclusions & Future Work

We collected the accelerometer and gyroscope gait data using an IMU sensor attached on the shoe and performed the analysis to recognize ten different walking workouts.

We introduced the EFF phase and propose an algorithm to detect it. Then, we utilize both EFF and non-EFF phases in order to recognize these ten workouts.

We discover that the features from the EFF phase play the key role to distinguish workouts in flat, ascending slope, or descending slope. On the other hand, the non-EFF phase features help to distinguish workouts in non-stair, ascending stairs, and descending stairs.

We employ three different machine learning algorithms, Decision Tree, Random Forest and K-Nearest Neighbor for data analysis. Our evaluation concludes that (1) Among three ML algorithms, Random Forest gives the highest accuracy at an average of 97.11%; on the other hand, K-Nearest Neighbor gives the lowest accuracy, thus not recommended for future use, and (2) Among ten walking workouts, recognizing walking workouts in non-stair cases is most difficult since these workouts have similar gait cycle patterns. Hence in average, their recognition accuracies are also the lowest.

Potential future work may include:

- Explore more features and employ more machine learning algorithms to increase the accuracy of activity recognition.
- Add more workout activities such as running into consideration.
- Consider finer granularity of environments such as breaking down the level of a slope based on its degree of the steepness.
- Utilize sensors beside accelerometer and gyroscope to improve the accuracy of the recognition

When dealing with an activity with higher velocity such as running, the EFF detection algorithm will need to be extended to detect much shorter gait cycle.

Moreover, it is possible to use the dynamic window segmentation approach based on the EFF detection to improve the efficiency of our algorithm.

References

[1] W. Tao, T. Liu, R. Zheng, and H. Feng, "Gait analysis using wearable sensors" in Sensors 2012, 12, pp. 2255–2283, February 2012.

[2] V. N. T. Sang, T. Q. D. Khoa, S. Yano, and T. Kondo, "Ascending and Descending Walking Recognition using Smartphone" in MOBIHEALTH'15 Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare, pp. 120-123, London, Great Britain, October 2015.

[3] S. Sprager, and M. B. Juric, "Inertial Sensor-Based Gait Recognition: A Review." In Sensors 2015,15, pp. 22089-22127, September 2015.

[4] N. Ravi, N. Dandekar, P. Mysore, and M.L. Littman, "Activity Recognition from Accelerometer Data" in the 17th conference on Innovative applications of artificial intelligence, Volume 3, pp. 1541-1546, Pittsburgh, Pennsylvania, July 2005

[5] M. Ghobadi and E. T. Esfahani, "Foot-Mounted Inertial Measurement Unit for Activity Classification" in 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 6294-6297, Chicago, 2014.

[6] M. O. Derawi, "Accelerometer-Based Gait Analysis, A survey." in Nor Informasjonssikkerhetskonferanse NISK, 2010.

[7] M. Chen, J. Yan and Y. Xu, "Gait Pattern Classification with Integrated Shoes" in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 833-839, St. Louis, 2009.

[8] T. T. Ngo et al., "Similar gait action recognition using an inertial sensor" 2014 Elseveir Pattern Recognition, Volume 48, pp. 1289-1301, 2015.

[9] M. Muaaz, and R. Mayrhofer, "Accelerometer based Gait Recognition using Adapted Gaussian Mixture Models." in The 14th International Conference on Advances in Mobile Computing & Multimedia (MoMM2016), pp. 288-291, Singapore, Singapore, November 2016

[10] J. R. Kwapisz, G. M. Weiss, and S. A. Moore "Activity Recognition Using Cell Phone Accelerometers", SensorKDD '10, Washington, DC, USA, July 2010.

[11] M. Derawi, and P. Bours, "Gait and activity recognition using commercial phones" 2013 Elseiver Computers & Security, Volume 39, pp. 137-144, 2013.

[12] M. Boutaayamou et al., "Segmentation of gait cycles using foot-mounted 3D accelerometers" in 2015 International Conference on 3D Imaging (IC3D), pp. 1-7, Liege, 2015.

[13] S. J. Abbas and Z. M. Abdulhassan, "Kinematic Analysis of Human Climbing up and Down Stairs at Different Inclinations, 2013

[14] Centers for Disease Control and Prevention, "Measuring Physical Activity Intensity" [Online]. Available: https://www.cdc.gov/physicalactivity/basics/measuring/index.html [Accessed May 2018].

[15] Mbientlab, "MetaMotionR Board" [Online]. Available: https://mbientlab.com/product/metamotionr/ [Accessed May 2018].

[16] J. Taborri, E. Palermo, S. Rossi, and P. Cappa, "Gait Partitioning Methods: A systematic Review" in Sensor 2016, 16, January 2016

[17] T. L. Switaj and F. G. O'Connor, "Chapter 43 - Gait Analysis, In The Sports Medicine Resource Manual", edited by P. H. Seidenberg and A. I. Beutler, W.B. Saunders, Philadelphia, 2008, pp. 536-542.

[18] L. Chaitow and J. DeLany, "Application of Neuromuscular Techniques : Chapter 3 – Gait analysis", Volume 2, 2nd Edition, Churchill Livingstone, July 2011, pp. 61-84

[19] Physiopedia, "Gait" [Online]. Available: https://www.physio-pedia.com/Gait [Accessed June 2018].

[20] Mbientlab, "Tutorials: 6. SDKs" [Online]. Available: https://mbientlab.com/tutorials/SDKs.html [Accessed June 2018].