



Summer Internship Project Report

Assistive Device to map and interpret the local scene for imparting Spatial Awareness in Visually Impaired People

2.11.2015

Submitted by:
Prashant Sinha
B.Tech. (IT and Mathematical Innovations)
72174

Mentor:
Ir. Abhijeet Parmar
Senior Design Fellow
Design Innovation Centre,
University of Delhi

Human Physical Activity Recognition from Acceleration Data under Naturalistic Conditions

Disclaimer

This document describes work undertaken as part of a program of study at the Cluster Innovation Centre of the University of Delhi. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the centre.

CONTENTS

Abstract	5
Acknowledgements	6
1 Introduction	7
1.1 Statement of the Problem	7
1.2 Proposed Solution	7
1.3 Proposed Work	8
2 Motion Tracking using Inertial Measurement Unit	9
2.1 Background	9
2.2 Motion Tracking Device	9
2.2.1 Sensor data fusion	9
2.2.2 Data logging and Visualisation	10
3 Realtime Human Activity Recognition Task	11
3.1 Background	11
3.2 Previous Work	11
3.3 Undertaken Challenge	11
3.3.1 Hardware Complexity Constraint	11
3.3.2 Attitude Independent Recognition (Axis Constraint)	11
3.3.3 Hardware Friendly Classification (Time Complexity Constraint)	12
3.4 Method	12
3.4.1 Acquisition of Activity Dataset	12
3.4.2 Activity Classes	12
3.4.3 Windowing and Overlapping	14
3.4.4 Feature Engineering	14
3.4.5 Axial Features	14
3.4.6 Axial Feature Pooling and Feature Vector	17
3.4.7 Classification using Supervised Learning	17
3.4.8 Realtime Activity Prediction	17
3.5 Feature Classification	17
3.6 Results and Discussions	18
3.6.1 Results	18
3.6.2 Discussions	19
4 Conclusion	20
5 Future Work	21
References	22
Appendix	23

LIST OF FIGURES

Figure 1	Arduino Motion Tracker Shield prototype.	9
Figure 2	IMU Sensor Axes Configuration	10
Figure 3	Annotation and Visualisation dashboard.	10
Figure 4	Methodology Overview	13
Figure 5	Example instance frames for various activities	14
Figure 6	Sample key points selected from raw signal.	16
Figure 7	Comparative classification accuracy for varying window sizes.	18
Figure 8	Confusion Matrix for window length $l = 100$.	18
Figure 9	Parallel coordinates plot of feature vectors.	20
Figure 10	RadViz visualisation of feature vectors.	20

LIST OF ABBREVIATIONS

DFT	Discrete Fourier Transform. 14
DTC	Decision Tree Classifier, uses a decision tree as a predictive model. 18
HAR	Human Activity Recognition. 11 , 17
IMU	Inertial Measurement Unit. 9
RBF	Radial basis function, a real-valued function used as a kernel for multi-class SVM . 18
RFC	Random Forest Classifier, uses a ensemble learning method for classification. 18
SVM	Support Vector Machine, constructs hyperplanes to separate labeled vectors. 4 , 11 , 18

ABSTRACT

Providing accurate and opportune information on people's activities and behaviors is one of the most important tasks in pervasive computing. Innumerable applications can be visualized, for instance, in medical, security, entertainment, and tactical scenarios. In this study, we explore the detection of human activities (walk, run, jog, climb stairs up, climb stairs down, stand still, biking) from data acquired using single tri-axial accelerometer. We address the task of detecting activities irrespective of the user specific sensor position, orientation and body attachment. We compare the classification accuracies in case of orientation uncorrected data in different positions and orientations across multiple individuals. Also, we show that a single device and accelerometer data is sufficient to detect the activities. Additionally we also show significant activity detection accuracies when the user holds the device in the hand or carries it in the pockets, as in a real life scenario. Random Forest classifiers showed the best performance recognizing everyday activities with an overall accuracy rate of 96%. Finally, we present some open problems and ideas that, due to their high relevance, should be addressed in future research.

KEYWORDS

Human Activity Recognition, Feature Extraction, Windowing, Pervasive Computing, Random Forest

ACKNOWLEDGEMENTS

Thank You.

My sincerest gratitude for your support, patience, guidance:

- Ir. Abhijeet Parmar, Academic Supervisor, DIC, for his continuous support and guidance of my Summer Internship Project, for his scientific intuition which made him a constant source of ideas and inspirations.
- Prof. Shobha Bagai, Program Coordinator (B.Tech. ITMI), CIC, for her advice, supervision and immense knowledge, for her critical comments and suggestions for improving my work.
- Prof. B. Biswal, Coordinator, DIC, for providing excellent research environment and infrastructure to carry forward my research.
- All CIC faculty, staff and students who transferred their immense knowledge and skill for exceptionally enriching my intellectual maturity.
- Prof. Alex "Sandy" Pentland, MIT Media Lab, for his willingness to critically review my research method, result and answering my questions about human activity behaviours.
- Prof. Samir K. Brahmachari, Founder Director, CSIR-Institute of Genomics and Integrative Biology, Chief Mentor, CSIR-OSDD Unit, for his qualified remarks which helped me in refining my model and results.

For your time and cooperation:

- Pragya Jaiswal, and Parul Sethi for their precious time for assisting in data collection task.
- Devesh Khandelwal for his insights in programming methods.

To my friends, old, new and future throughout the globe.

Above all, for your love and support:

My Family.

Sincerely,

(Prashant Sinha)

1 INTRODUCTION

1.1 Statement of the Problem

Visually impaired people confront a number of challenges in their daily life - from figuring out if they are at the correct bus stop to reading the label on their soft drink bottle. Navigation, for instance, is very difficult for a person and prevents the user from moving independently or walk safely.

A number of assistive technologies are available for users now a days which help a visually impaired person to perform their day to day activities, such as:

1. Screen Reader: Allows the user to use a computer and mobile phone. It is an example of software based assistance that reads the content of the screen to the user and allows them to use the device as a regular person. It is not perfect, for, it requires the developers of the apps to incorporate the assistive guidelines during the development process.
2. GPS and Navigation: Allows the users to pinpoint their location on map, and also get step by step directions to a specified locations. This helps users to navigate and prevents them from losing their way.
3. Refreshable Braille Display: It consists of electronically actuated dots in a matrix of 4x2 dots per character. The dots raise and depress to form each of the character, therefore making use of braille script to present the characters to the user.
4. Tactile paving on sidewalks, metro stations and shopping complexes; elevators with braille embossed buttons; and announcements in trains also aid into facilitating the visually impaired people.

However, it is still very difficult to navigate and identify objects in new or strange locations, or locations that are not designed keeping accessibility in mind. They'll also require constant assistance from other people, which is, in turn, subject to knowledge and behaviour of other people and has its own social, and mental consequences.

The navigation technologies that are available today can indeed give navigation instructions from a location to another, however, the instructions require personal judgement of the user. These include avoiding hitting various obstacles, staying on side walk, and so on. For a blind person, following these instructions is not less than a challenge either.

Hence, it is required that proper steps be taken to make a visually impaired person partially or fully self reliable and hence support them to live independently.

1.2 Proposed Solution

Through the global penetration of Smart Phones, access to Navigation Services that are assisted though GPS are available very easily and are cost affective. However, the instructions provided by the Navigation Services require personal judgement of the user to follow. This includes avoiding obstacles that are encountered on the way, staying on the sidewalk and in

correct direction and so on. This is not an easy task for a visually impaired person. The services also fail to work in an indoor environment.

Hence, our solution proposes that increasing the local spatial awareness is a key solution to this problem, and would help the visually impaired people gain more self-reliance.

1.3 *Proposed Work*

1. Building a wearable assistive device
2. Motion context awareness using motion trackers
3. Scene description and object detection using 3D point cloud
4. Haptic and auditory interface for interaction

2 MOTION TRACKING USING INERTIAL MEASUREMENT UNIT

2.1 Background

Motion tracking refers to the process of recording the movement of objects. Various passive and active methods may be utilized for achieving this purpose, such as video surveillance, low range RADAR or motion sensors. An Inertial Measurement Sensor is an array of electronic sensors that are used to measure the external forces acting on an object. An IMU may contain an Accelerometer, Gyroscope, Magnetometer, GPS, Pressure Sensor etc.

1. An accelerometer measures the acceleration in a axis.
2. A gyroscope measures the rate of rotation in a particular axis.
3. A magnetometer (also known as a digital compass) measures the magnitude of magnetic field in a particular axis.

Through a combination of these three devices, in the three orthogonal axes, it is possible to accurately reproduce the motion of an object the **IMU** is pivoted to.

2.2 Motion Tracking Device

To track and record the motion data, we built a shield for an Arduino Uno (Figure 1). The motion tracking shield uses a tri-axial **IMU** (Figure 2) having an Accelerometer, a Gyroscope and a Magnetometer for motion tracking. The shield also has a 16 GB onboard flash storage to record the IMU data samples. The schematic for the shield is available in Appendix.

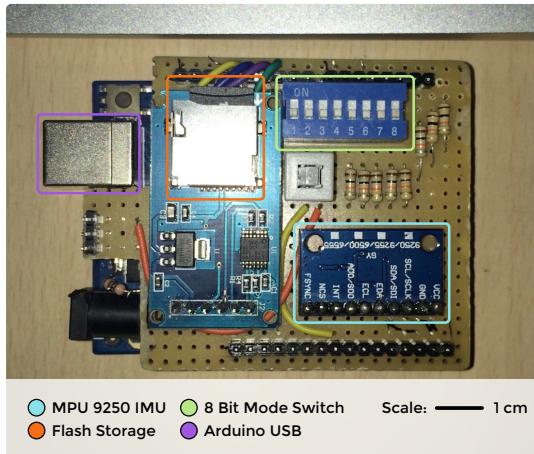


Figure 1: Arduino Motion Tracker Shield prototype.

2.2.1 Sensor data fusion

Data from the three axes of the accelerometer, gyroscope and magnetometer are fused to get the attitude and heading information of the device. The information are given by quaternion angles as Yaw, Pitch and Roll. A Quaternion filter was implemented following the method proposed by [1].

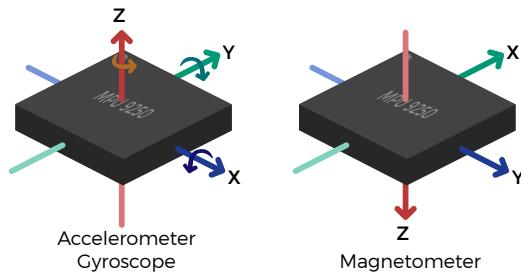


Figure 2: IMU Sensor Axes Configuration

2.2.2 Data logging and Visualisation

We used InfluxDB (Figure 3) for efficient recording and retrieval of time series sensor data. Multiple interface for data logging was provided, which includes UDP channel over WiFi, USB Serial, and an on-board MicroSD card.



Figure 3: Annotation and Visualisation dashboard.

3 REALTIME HUMAN ACTIVITY RECOGNITION TASK

3.1 Background

Providing and predicting information about user's intent, behaviour, and activities is one of the most important task in pervasive computing. Human Activity Recognition (**HAR**) is a problem of inferring the state of the user by exploiting the data obtained from sensors that are already present in many mobile devices. The information provided from **HAR** system can be utilised to provide contextual awareness which can further be utilised in innumerable applications. Context aware mobile apps may optimize their user experience based on the activity that the user is presently performing. It can also be used in various patient and baby monitoring systems to provide accurate and realtime activity data which may prove useful to the caretakers in case of emergencies [2]. Activity information may also be used to remove false positives from pedometer data in fitness tracking applications for more accurate gait analysis.

3.2 Previous Work

Various studies have shown efficiency of machine learning methods for the solution of **HAR** task. [2] proposes an energy efficient method of activity classification using fixed point arithmetic scheme for training a **SVM**. The classification accuracy of their scheme is 89% for 6 activity classes. [3] is the closest to the present study. They take a 6 seconds window to compute the features. The method involves further windowing to calculate the features such as mean trends and windowed mean difference. Their result had an accuracy of 97-98% using kNN and Naive Bayes classifier.

[4] investigate the relation between window length and the classification accuracy, while [5] report success using signal processing and filters on the combined gyroscope and accelerometer data.

[6] use PCA and Cosine transform for the **HAR** task. They report 97% accuracy between 4 activity classes. [7] also use gyroscope and accelerometer data in their study while claiming the decrease in classification accuracy due to the bias arising by shifting the position of the device.

3.3 Undertaken Challenge

3.3.1 Hardware Complexity Constraint

It is required that the task is solved solely through the data from a single triaxial accelerometer. This reduces the hardware complexity.

3.3.2 Attitude Independent Recognition (Axis Constraint)

It is required that the task be solved without any prior knowledge of the location and orientation of the motion tracker. This follows from the fact that the motion tracker could be placed at any spot on the user, for example their pockets, their arm or in their hand.

3.3.3 Hardware Friendly Classification (Time Complexity Constraint)

It is required that the features that are generated from the sensor sample sequence must be computationally low cost. This is required for energy efficient implementation of the technique in various portable devices such as a fitness tracker or a mobile phone.

3.4 Method

The task of human activity recognition (HAR) begins with data collection. However, the raw data in itself is not representative enough to be used for carrying out the task of classification. Thus, we need to prepare the data by cleaning it and carrying out various data preprocessing activities. The preprocessed data still doesn't capture the temporal nature of the task. Hence data segmentation is performed to enhance the discriminative power of the data. Feature extraction, then, converts the signals into the most relevant and powerful features which are unique for the activity at hand (Figure 4). Finally classification algorithms can be trained on the refined data, and a reliable prediction model can be created

3.4.1 Acquisition of Activity Dataset

Motion data was logged through motion tracking device and an iPhone app developed for the purpose at the sampling rate of 50 Hz while multiple human subjects performed various activities as mentioned in section 3.4.2. The location of the devices and the duration of activities were neither controlled nor taken into consideration. The subjects were also allowed to move or use the devices during the activity. These events were also not controlled and did not have any exclusive label. This was done to ensure that the dataset reflected the real-world usage pattern. The starting and ending timestamp of the activities were used to label the logged data.

We also obtained the public dataset [5] and [8] to further increase the volume of our data. Only the accelerometer data was utilised from the obtained dataset.

3.4.2 Activity Classes

We define following activity classes to set the scope of the HAR task.

1. Stationary
 - 1.1. Sitting
 - 1.2. Standing
2. Active
 - 2.1. Walking
 - 2.1.1. Typical
 - 2.1.2. Stair
 - 2.1.2.1. Walking Upstairs
 - 2.1.2.2. Walking Downstairs

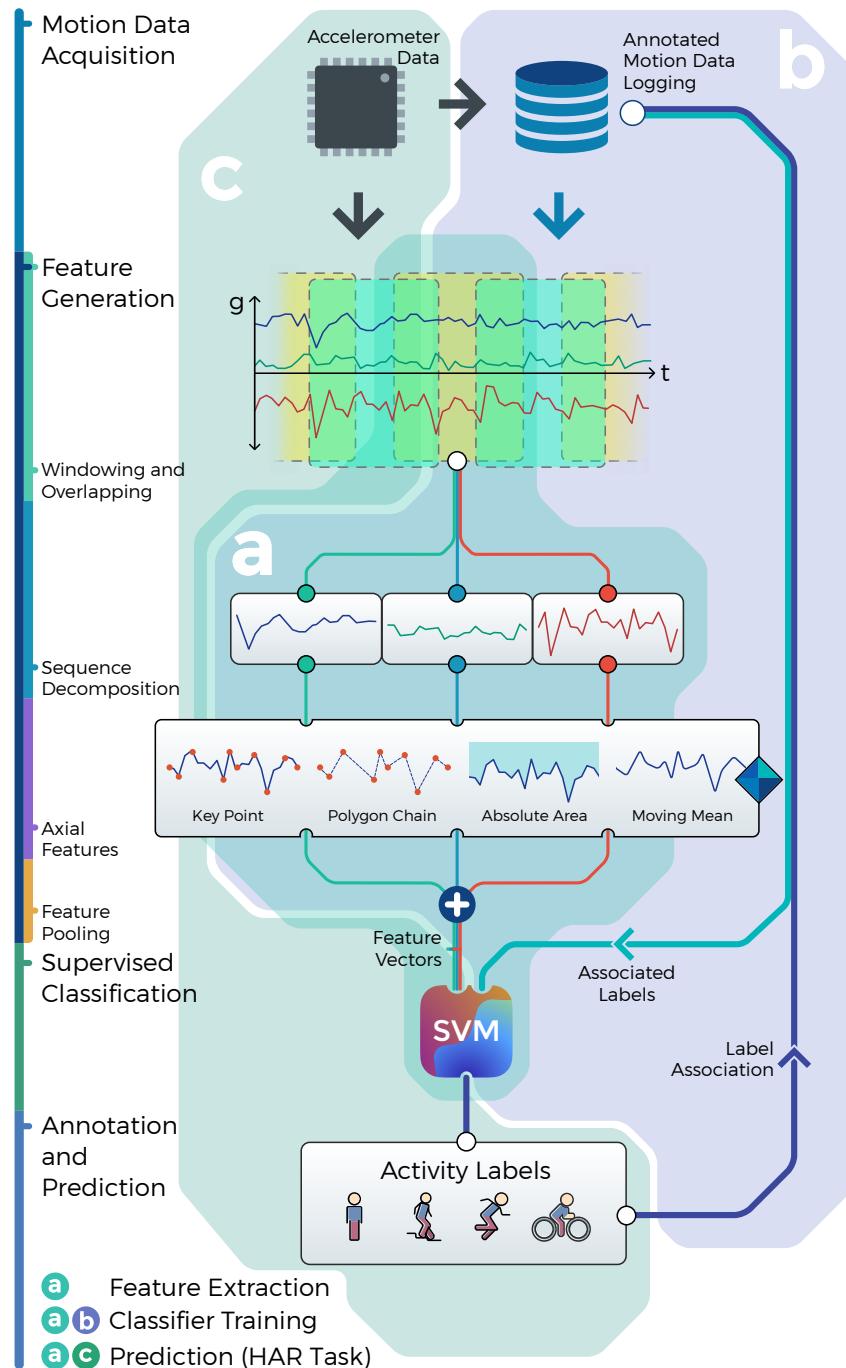


Figure 4: Methodology Overview

2.2. Running

2.2.1. Typical

2.2.2. Jogging

2.3. Biking

3.4.3 Windowing and Overlapping

Due to the nature of task, it is often very difficult to represent a particular activity solely through one sample point. Hence, it is recommended to utilise a sequence of signal samples to represent the activity instances [4].

In the obtained discrete-time sample sequence, we defined a window of length having l samples for all the three axes as activity instance. The windows were also overlapped to reduce any segmentation bias. The value of l , and its effect on classification accuracy is discussed in section 3.6.

3.4.4 Feature Engineering

To reduce the computational complexity and enhance the recognition characteristics, feature generation is required. Overall 5 features were calculated for each axis. To conform to the axis constraints as defined in section 3.3.2, the features from the three axes were combined to get a final feature vector.

To address the time complexity constraint as defined in section 3.3.3, all the features are calculated in time domain, since the frequency domain conversion using discrete Fourier transform (DFT) is a computationally intensive process.

From the figure 5 it can be observed that various activities are inherently periodic in nature. The features are calculated for all the three axes of the activity instances obtained after windowing and overlapping the data.

3.4.5 Axial Features

Mathematically, the activity instance can be represented by a subsequence of discrete acceleration sample sequence.

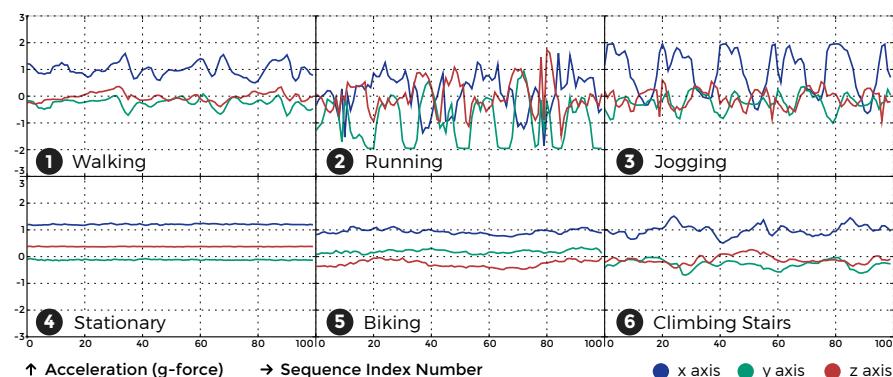


Figure 5: Example instance frames for various activities

Let S_{raw} be a sequence of ordered discrete-time acceleration sample sequence as obtained from the motion tracker. Here,

$$S_{raw} = (\langle a_t^x, a_t^y, a_t^z \rangle) \quad (1)$$

where a_t^x , a_t^y and a_t^z represent the accelerometer reading at instance t , in x , y and z axis respectively.

We define the activity instance S_{act} as the subsequence of S_{raw} having length l . From S_{act} , we further define decomposed axial discrete-time sample sequence by equation 2, and indexed sequence by 3.

$$S_{act}^k = (a_i^k) \quad (2)$$

$$S_{rmp}^k = (\langle i, a_i^k \rangle) \quad (3)$$

where $k \in \langle x, y, z \rangle$ and i is the sequence index.

The axial feature vector is defined in the equation 4.

$$F^k = \langle a_{auc}^k, a_{tss}^k, a_{kp}^k, a_{binnedkp}^k, a_{sm}^k \rangle \quad (4)$$

The feature vector components are discussed below.

3.4.5.1 Absolute area under linearly interpolated curve

Let $f_{[i,j]}^l$ denote the two-point form of a straight line segment joining arbitrary elements in S_{rmp}^k . Here,

$$f_{[i,j]}^l = \left(\frac{a_j^k - a_i^k}{j - i} (x - i) + a_i^k \right) \quad x \in [i, j] \quad (5)$$

We may obtain the linearly interpolated curve function f_{int} as a piecewise function of multiple line segments [9], defined by $f_{[i,j]}^l$. Hence, if l denotes the number of elements in sequence S_{rmp}^k , we get equation 6.

$$f_{int}^k = (f_{[i,i+1]}^l)_{i=0}^{l-1} \quad (6)$$

We take the absolute area under f_{int}^k as a feature, which is given by equation 7

$$a_{auc}^k = \int_0^l |f_{int}^k| dx \quad (7)$$

3.4.5.2 Total sum of squares

Let \bar{a}^k denote the mean of S_{act}^k . Then, we obtain the total sum of square for series S_{act}^k as:

$$a_{tss}^k = \sum_{n=0}^l (a_n^k - \bar{a}^k)^2 \quad \forall a_n^k \in S_{act}^k \quad (8)$$

The quantity obtained from equation 8 is used as a feature.

3.4.5.3 Key point

We define the key points, S_{kp}^k , a subsequence of S_{rmp}^k , where the signal sequence attains relative maximum or minimum value (extrema values).

Due to noisy nature of the signal, the relative extrema are estimated by taking n neighbours of the element currently in consideration. The quantity n is defined as the 'order' of extrema function. This is done to ensure that small fluctuations are not accounted as key points.

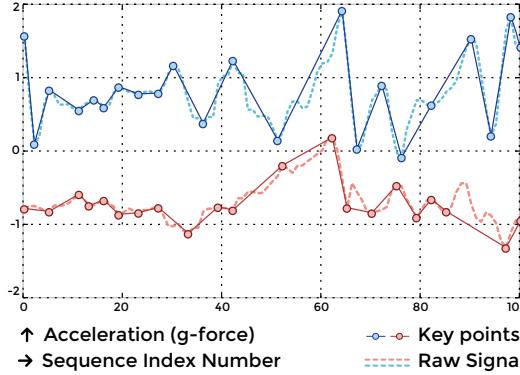


Figure 6: Sample key points selected from raw signal.

3.4.5.4 Key point gradient

Let

$$G_{kp}^k = \left(\frac{a_j^k - a_i^k}{j - i} \right)_{i,j} \quad (9)$$

be a sequence defined from all consecutive elements i and j in key point sequence S_{kp}^k . Here, the elements of sequence G_{kp}^k denote the gradient of a straight line joining the key points.

We take the quantity from equation 10 as a feature.

$$a_{kp}^k = \text{Var}[G_{kp}^k] \quad (10)$$

3.4.5.5 Binned key point gradient

We define gradient binning as the process of grouping straight lines on the basis of their gradient. The grouping ranges are obtained by dividing the I and IV quadrants (within the domain of arctan) of cartesian plane in, say, h equal segments. The binning ranges are defined by the following sequence:

$$G_{bin} = (\langle i, \left[\tan\left(\frac{\pi}{h}i\right), \tan\left(\frac{\pi}{h}(i+1)\right) \right] \rangle)_{i=\lceil \frac{h}{2} \rceil}^{\lfloor \frac{h}{2} \rfloor} \quad i \in \mathbb{Z} \quad (11)$$

The binning function f_{bin} follows from G_{bin} where, the gradients are given index number i , if they fall in corresponding range.

Using f_{bin} , we define equation 12, where, a_i are the elements in G_{kp}^k .

$$G_{binnedkp}^k = (f_{bin}(a_i)) \quad (12)$$

We take the following quantity as a feature:

$$a_{binnedkp}^k = \text{Var}[G_{binnedkp}^k] \quad (13)$$

3.4.5.6 Variance of smoothed series

We use a simple rolling average on S_{act}^k to smooth out the short-term fluctuations in the sequence. We form a new sequence S_{smooth}^k having length $l - n$, where l is the window length and n is the order of rolling average, from S_{act}^k . For the elements e in S_{act}^k , we have

$$S_{smooth}^k = \left(\frac{e_m + e_{m-1} + \dots + e_{m-(n-1)}}{n} \right)_{m=0}^{l-n} \quad (14)$$

The following quantity is taken as a feature.

$$a_{sm}^k = \text{Var}[S_{smooth}^k] \quad (15)$$

3.4.6 Axial Feature Pooling and Feature Vector

To conform to the Axis Constraint, as defined in section 3.3.2, the axial feature vectors F^x , F^y , and F^z are combined (pooling) to obtain an axis independent feature vector. The variance features, $\{a_{kp}^k, a_{binnedkp}^k, a_{sm}^k\}$ are pooled by taking the combined variance. We take arithmetic mean of feature $\{a_{auc}^k, a_{tss}^k\}$. Here, k denotes the three orthogonal axes x , y , and z as mentioned in section 3.4.5.

3.4.7 Classification using Supervised Learning

Various supervised learning schemes were utilised for the motion classification. The classifiers were trained using the labelled feature vectors and the resulting classifier was then used for cross validation and prediction.

3.4.8 Realtime Activity Prediction

To implement the realtime activity classification, the triaxial accelerometer is sampled at a sampling rate of 50 Hz. These samples are stored in a buffer array and then the feature vector is calculated for this buffer array. The predicted activity label for this feature vector is obtained by fitting the vector into the trained classifiers.

3.5 Feature Classification

Since the HAR task requires that multiple activities are classified, a multi-class classifier is required. We explored various multi-class supervised-learning methods to evaluate the classification performance of the features as discussed in section 3.4.5. The methods that were evaluated are: Decision

Tree Classifier (**DTC**), Random Forest Classifier (**RFC**), and Support Vector Machine (**SVM**) with **RBF** Kernel with varying parameters.

The performance as well as the relevance of the features are discussed in section 3.6.

3.6 Results and Discussions

3.6.1 Results

After calculating the features for the windows, and assigning the labels, we split the dataset into two segments with random cuts. First segment, having 75% the size of the dataset, was used for training the classifiers. The other segment, representing the 25% of the dataset, was used for testing purpose. The same data-subset was used for evaluating all the classifiers as mentioned in section 3.5.

Figure 7 contains the classification accuracy obtained by each classifier with respect to the chosen window length. Using the dataset as mentioned in section 3.6, we obtained a maximum classification accuracy of 96.32% using Random Forest Classifier closely followed by Support Vector Machine with 95.05% accuracy (Figure 8).

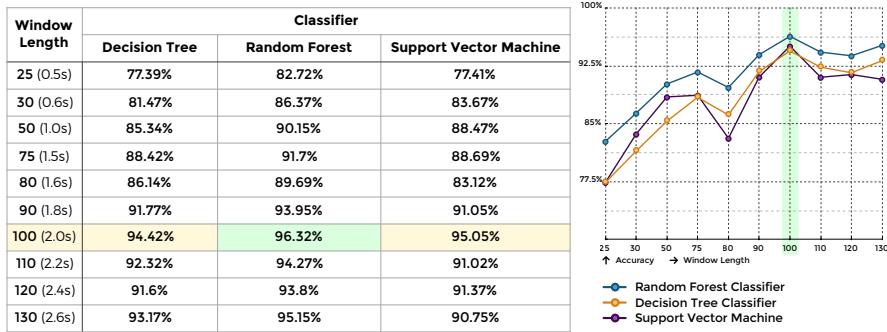


Figure 7: Comparative classification accuracy for varying window sizes.

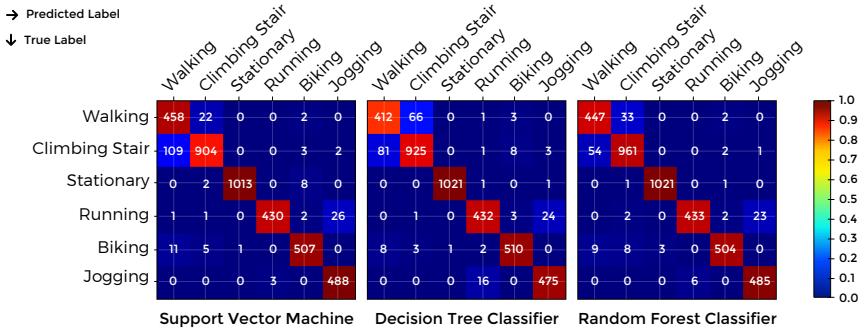


Figure 8: Confusion Matrix for window length $l = 100$.

3.6.2 Discussions

Human activities are performed during relatively long periods of time compared to the sensors sampling rate. Besides, a single sample on a specific time instant does not provide sufficient information to describe the performed activity. Thus, activities need to be recognized in a time window basis rather than in a sample basis. Now, the question is: how do we compare two given time windows? It would be nearly impossible for the signals to be exactly identical, even if they come from the same subject performing the same activity. This is the main motivation for applying feature extraction methodologies [10] [11] to each time window: filtering relevant information and obtaining quantitative measures that allow signals to be compared. In general, two approaches have been proposed to extract features from time series data: statistical and structural. In our study, we have used a combination of both statistical and structural features.

In addition to efficient feature extraction, selection of window length that is dividing the measured time series in time windows is also crucial in increasing the classification accuracy. A key factor is, therefore, the selection of the window length because the computational complexity of any feature extraction method depends on the number of samples. Having rather short windows may enhance the feature extraction performance, but would entail higher overhead due to the recognition algorithm being triggered more frequently. Besides, short time windows may not provide sufficient information to fully describe the performed activity. Conversely, if the windows are too long, there might be more than one activity within a single time window. Different window lengths have been used in the study but the maximum classification accuracy is observed with window length of 2s. Of course, this decision is conditioned to the activities to be recognized and the measured attributes.

Figure 9 is the parallel coordinate visualisation [12] of 50 feature vectors, chosen randomly, for each activity class as described in section 3.4.2. We used parallel coordinates to plot the components in order as presented in figure. The independence between individual features is clearly distinguished for the activity classes. The feature vectors are plotted in both Linear and Logarithmic scale to improve visibility of features that attain value close to zero, since the features are not normalised.

Similarly, figure 10 is the RadViz Visualisation [13][14] of 400 feature vectors chosen randomly for each activity classes. Distinct clusters of activity classes further support the conclusion made from the Parallel Coordinate plot of feature vector. Confusion between similar activities, such as Jogging and Running, and Walking and Climbing is also visible in the plot wherever the cluster of classes overlap. However, it should be noted here that only a small percentage of feature vectors overlap, as evident from section b in figure 10.

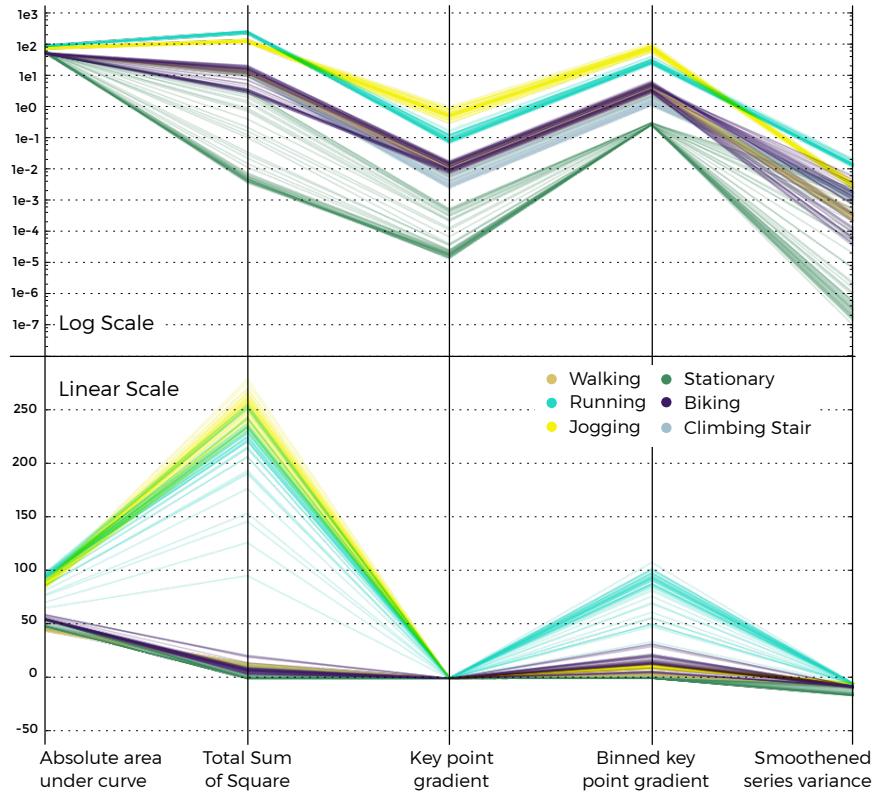


Figure 9: Parallel coordinates plot of feature vectors.

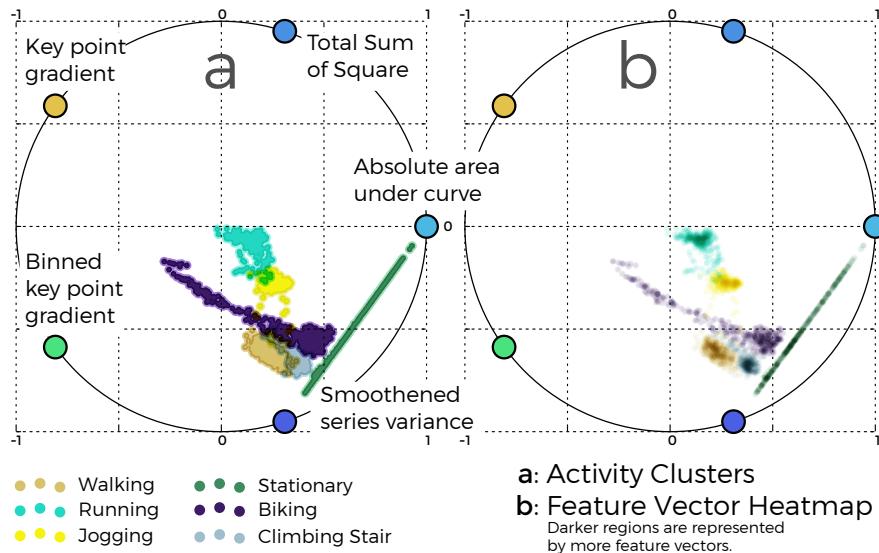


Figure 10: RadViz visualisation of feature vectors.

4 CONCLUSION

The focus of this research was to explore and analyze flexibility in various constraints, specifically the position, orientation and attachment of the mobile sensor device in a human activity recognition task. The results from our study provide evidence to the fact that activity recognition is possible

in spite of variation in the aforementioned constraints. Our findings also indicate that neither device orientation information nor orientation correction appear to be necessary, as we are able to correctly classify the activities across all subjects without orientation correction. Moreover, by using simple features, we can achieve high classification accuracies, above 95% irrespective of how the device is carried and placed by the subject. This method is suitable for a distributed system where users can report data, in real-time or in batches, and this information can be processed and used to act according to the individual context.

The results of the present research have been submitted for publication in International Journal of Human-Computer Studies.

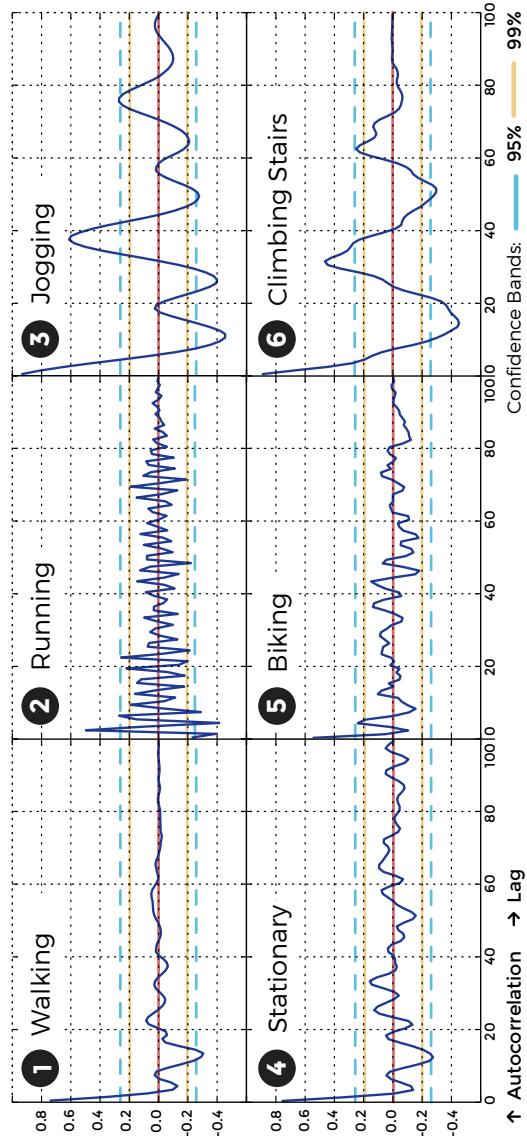
5 FUTURE WORK

- 3D Point Cloud generation using Time of Flight sensor.
- Development of intelligent Scene Description methodologies.
- Final assistive navigation device prototyping.
- Testing on live subjects.

REFERENCES

- [1] S. O. H. Madgwick, A. J. L. Harrison, and A. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm.," *IEEE ... International Conference on Rehabilitation Robotics : [proceedings]*, vol. 2011, p. 5975346, jan 2011.
- [2] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, "Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic.," *J. UCS*, 2013.
- [3] P. Gupta and T. Dallas, "Feature selection and activity recognition system using a single triaxial accelerometer," *Biomedical Engineering, IEEE Transactions*, 2014.
- [4] O. Banos, J. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, 2014.
- [5] J. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-Aware Human Activity Recognition Using Smartphones," *Neurocomputing*, 2015.
- [6] Z. He and L. Jin, "Activity recognition from acceleration data based on discrete cosine transform and SVM," *Systems, Man and Cybernetics, 2009. SMC 2009*, 2009.
- [7] M. Shoaib, J. Scholten, and P. Havinga, "Towards physical activity recognition using smartphone sensors," dec 2013.
- [8] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of smartphone motion sensors for physical activity recognition.," *Sensors (Basel, Switzerland)*, vol. 14, pp. 10146–76, jan 2014.
- [9] Bradie, *A Friendly Introduction to Numerical Analysis*. Pearson Education.
- [10] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, 2003.
- [11] Y. Chen and C. Lin, "Combining SVMs with various feature selection strategies," *Feature extraction*, 2006.
- [12] E. Wegman, "Hyperdimensional data analysis using parallel coordinates," *Journal of the American Statistical Association*, 1990.
- [13] J. Sharko, G. Grinstein, and K. Marx, "Vectorized radviz and its application to multiple cluster datasets," *Visualization and Computer Graphics*, 2008.
- [14] E. Bertini, "Springview: Cooperation of radviz and parallel coordinates for view optimization and clutter reduction," *Coordinated and Multiple Views in Exploratory Visualization*, 2005.

APPENDIX 1



Example autocorrelation plot.

