3D Accelerometer based Gesture Device for the Recognition of Digits

Himani Arora
Delhi Technological University
New Delhi
India

ABSTRACT

Gesture recognition helps in the development of a more natural and intuitive human computer interaction. It has several applications in virtual reality and can be used to control robots as well as home appliances. In this paper, the design and working of a compact handheld device that works with a computer to recognize hand gestures has been presented. A single 3-D accelerometer has been used for sensing the motion and Support Vector Machine has been employed for recognizing the gesture. The data processing has been implemented in MATLAB and a graphical user interface has also been developed to make the application user friendly. All digits from 0-9 have been recognized with a high accuracy for both user dependent and independent gesture recognition. In contrast to the previous models for digit recognition, a simpler approach that uses frame based temporal features has been presented to give a high recognition rate..

General Terms

Gesture recognition, Accelerometer, Support vector machine.

Keywords

Human Computer Interaction, Digit recognition

1. INTRODUCTION

A gesture is a form of non-verbal communication involving the movement of body parts, mainly hands and face in order to convey a specific message. Gesture recognition refers to the mathematical interpretation of gestures which can originate from any part of the body, however, the present technology focuses on emotion recognition from the face, lip movement recognition, eye tracking and hand gesture recognition. Gesture recognition can enable humans to interact with machines in a more natural manner. Gesture recognition can be used by people with walking disabilities and/or poor vision to operate any appliance or device with ease, by lecturers during presentations and has many applications in gaming and virtual reality.

There are mainly two approaches to gesture recognition, the first being vision based and the latter controller based. Vision based techniques use cameras either single or stereo cameras and depth aware cameras. Controller based techniques use a controller equipped with sensors such as accelerometer, gyroscope and/or flex sensors to sense movements. Gesture recognition has been used for controlling robots [1] by using a webcam to capture real-time video stream of hand gestures to generate commands for the robot. Gesture recognition has also been used to develop a system for automatic translation of static gestures of alphabets in the American Sign Language [2]. It used feature extraction and Artificial Neural Network (ANN) and achieved an accuracy of 92.33%. Three techniques for the recognition of static gestures using vision based approach have been proposed that used FD and 7 Hu moments as a feature descriptor, a finger counting algorithm and a convex hull method along with a 3D Kinect camera [3]. The major limitation observed by the authors was the requirement of static background and/or constant distance from the camera. Challenges faced by vision based gesture recognition include the inconsistency of images and videos in different lighting and the presence of background objects [4]. Moreover, the use of cameras lead to spatial limitation where the user is restricted within the field of view of the camera.

These limitations are overcome by using devices fitted with sensors such as VHand, a wireless data glove with 5 flex sensors which used a simple K-NN clustering algorithm based on Euclidean distance formula to recognize gestures [5]. A smart watch embedded with different sensors such as RFID reader, accelerometer, gyroscope, magnetometer and force sensors has been developed for the recognition of objects and gestures of the forearm as well as grasps, finger movements and wrist rotation [6].

Wiimote, the controller of Nintendo Wii equipped with a 3D accelerometer was used to acquire gesture acceleration data [7-9]. A handheld device for gesture recognition fitted with a 3D accelerometer was designed by Jianfeng Liu and Zhigeng Pan et al. [10-12]. A frame based descriptor and multi-class Support Vector Machine (SVM) approach with frames having 50% overlap has been used previously [7], that utilized both the temporal as well as spectral features. It achieved a recognition rate of 95.21% in user dependent case and 89.29% in user independent case for the 12 gestures recognized. Discrete Hidden Markov Model has also been used for pattern recognition [8,10]. An average rate of 90% was achieved for the recognition of the five gestures, namely, square, circle, roll, Z and tennis serve [8]. The gesture set in [10] consisted of both simple and complex gestures, and was able to achieve a recognition rate of more than 95% for the simple gestures and slightly less than 90% for the complex gestures. Neural Networks are also a popular choice for implementing gesture recognition and have been utilized for the recognition of gesture trajectories and handwritten digits [9,11]. K means algorithm was used to normalize the amount of values provided as input to the ANN classifier and it achieved an overall recognition rate of 92% [9]. The trajectory recognition algorithm in [11] extracted 2 features namely Range and ZCD and the recognition rate was 80% for the recognition of all digits from 0-9. An algorithm based on sign sequence and template matching has been developed which reduced each gesture to a gesture code of eight numbers [12]. Of these, seven gestures namely up, down, left, right, tick, circle, and cross were recognized with an accuracy of 95.6%. uWave, an efficient gesture recognition algorithm based on dynamic time warping using a single three-axis accelerometer has been presented by Jiayang Liu and Lin Zhong et al [13]. It required a single training sample and achieved an accuracy of 98.6% and 93.5% with and without template adaptation, respectively, for user dependent gesture recognition.

Recognition of digits is more complex than basic direction gestures and hence it is difficult to obtain high accuracy. In this paper, a gesture recognition technique using SVM has been presented for recognizing digits. The accelerometer data was first filtered and each gesture was divided into frames of variable size. Temporal features were extracted for each frame to form a feature vector which was then fed to the classifier. The processing and recognition algorithm was implemented in MATLAB. A compact handheld wireless device fitted with a 3D accelerometer was designed for the acquisition of data. This method gave high recognition accuracy for both user independent and dependent group.

2. MATERIALS AND METHODS

2.1 Hardware Components Used

The gesture recognition device (see Figure 1) consisted of the following components:

2.1.1 MSP430G2553

This is a microcontroller (MCU) with a 16 bit RISC processor, a 10 bit analog to digital convertor (ADC) as well as a universal serial communication interface module which supports universal asynchronous receiver/transmitter mode (UART).

2.1.2 ADXL335

This is a 3 axis accelerometer which can measure static as well as dynamic acceleration with a minimum full scale range of ±3 *g*.

2.1.3 HC-06

This is a Bluetooth serial interface module which can transmit data at different baud rates.

2.1.4 Power Supply

A 9V rechargeable battery was used as an input power supply.

2.1.5 LM2950-50 and LM2950-33

These are low dropout DC regulators which provide an output of 5V and 3.3V from the 9V battery for powering the MCU, accelerometer and bluetooth module.

2.1.6 Discrete Components

These include resistors, capacitors, switches and LEDs.

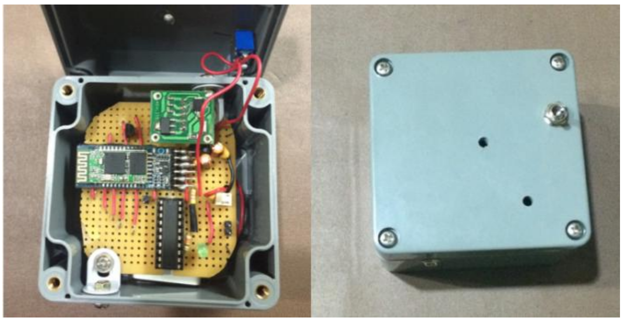

Figure 1: The assembled circuit (left) and top view of the gesture recognition device (right)

2.2 Working of the Hardware Device

The serial connectivity of the different components is elucidated in Figure 2. The X and Y-axis output of the accelerometer are connected to the ADC input channels. The MCU is programmed to read the sensor data on the ADC channels and transmit it to the PC wirelessly. Wireless transmission is accomplished by the Bluetooth module which is connected on the UART Tx and Rx pins of the MCU.

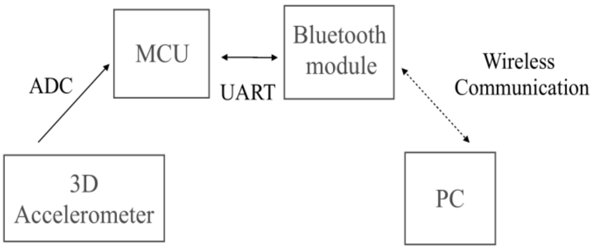

Figure 2: Serial connectivity of the hardware components

2.3 Software Used

2.3.1 MATLAB

It was used to process and analyze the data. It was also used to design a GUI which plots a graph of the X and Y axis accelerometer readings versus the sample number and displays the predicted digit.

2.3.2 CadSoft EAGLE

It was used to design the schematic and board layout.

2.3.3 Code Composer Studio

It was used to edit and debug the source code of the MSP430.

2.3.4 LIBSVM

It was used to implement SVM on MATLAB .

2.4 Code Methodology

2.4.1 Input Stage

When the user presses the start button, a beep occurs. He/she gets a time of 3 seconds to perform a gesture, that is, any number from 0-9 according to the convention shown in Figure 3. During this time period 250 samples of accelerometer data, each for X and Y axis, are recorded. A green LED is turned on the device which signifies that the sensor data is being read and simultaneously transmitted to the PC at the rate of 9600 baud.

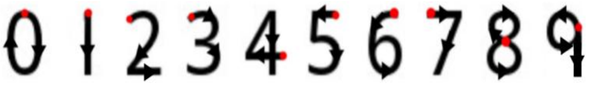

Figure 3: Convention of the gestures followed where the red dot signifies the starting point and the arrows indicate the direction of motion

2.4.2 Pre-processing Stage

The input data is pre-processed to make it suitable for feeding into the classifier . It is divided into frames and formed into a feature vector that contains all the relevant information needed to accurately classify it (see Figure 4).

Frame 1	Frame 2	Frame 3	Frame 4	Frame 5

Mean of X_i	Mean of Y_i	Variance of X_i	Variance of Y_i	Range of X_i	Range of Y_i	Variance from sensor mean for X_i	Variance from sensor mean for Y_i	Correlation between X_i and Y_i

Figure 4: The input data divided into 5 frames (top). The feature vector for the i^th^ frame (bottom).

2.4.2.1 Filter Data

Data is first filtered by using a moving average filter. The raw data from the accelerometer is noisy due to the vibration of the hand while making a gesture. Thus, the high frequency noise component is removed in this step (see Figure 5).

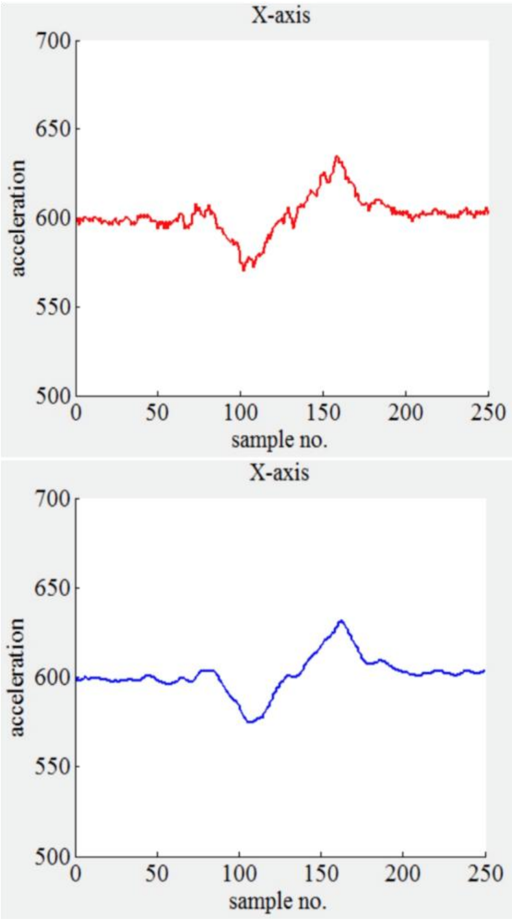

Figure 5: The raw X-axis accelerometer data(top) and the filtered data (bottom) for the digit 7.

2.4.2.2 Calculate Frame Length

This filtered data is then divided into 5 non-overlapping frames of variable length. The length is calculated dynamically. In the 3 second window, the user can make a number either very quickly or make it slowly. The reading will then be affected by the speed of motion since the peaks in the accelerometer data will appear at different points. This effect is somewhat nullified by having frames of variable length, where the length is calculated in a manner such that the peaks appear at nearly the same point. This is illustrated in Figure 6.

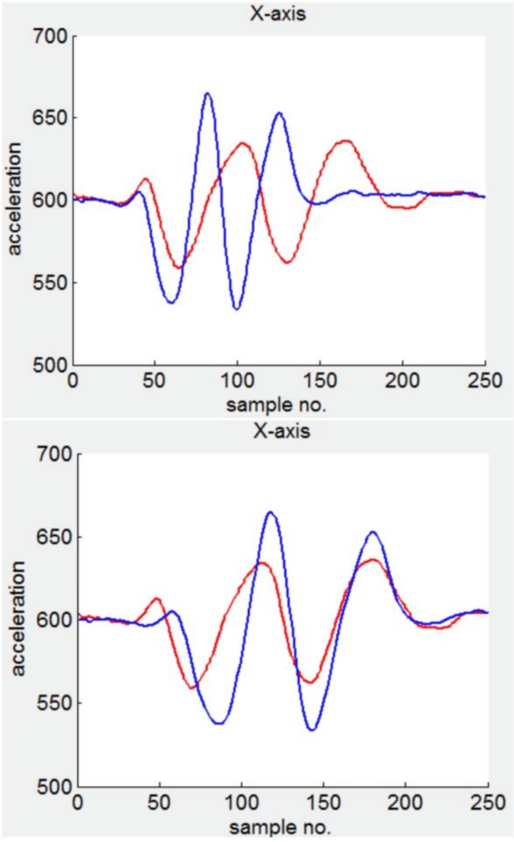

Figure 6: The original X-axis accelerometer data for the digit 3 from two different users with different speed of motion (top). The same data but shifted in the sample number domain such that the peaks nearly coincide (bottom).

2.4.2.3 Form Feature Vector

Features are calculated for each frame. Only temporal features are calculated and not the spectral features. This has further advantage since calculation of spectral features requires the computation of DFT which is more tedious. Let Xi represent the set of all values of the X axis output of accelerometer in the i^th^ frame. Similarly let Yi represent the set of all values of the Y axis output of accelerometer in the i^th^ frame.

The features calculated for each frame are mean, range, variance, correlation between Xi and Yi and variance from the mean of accelerometer reading (which is approximately 600 for x axis and 500 for y axis). Since there are five frames, this gives a feature vector of length 45 for each gesture.

2.4.3 Recognition Stage

The feature vector is fed to the multi-class SVM classifier which uses an RBF kernel. The SVM is trained to recognize the digits during the training phase. The accuracy of the classifier is then tested in the testing phase. The steps involved in the gesture recognition system are illustrated by means of a flowchart in Figure 7.

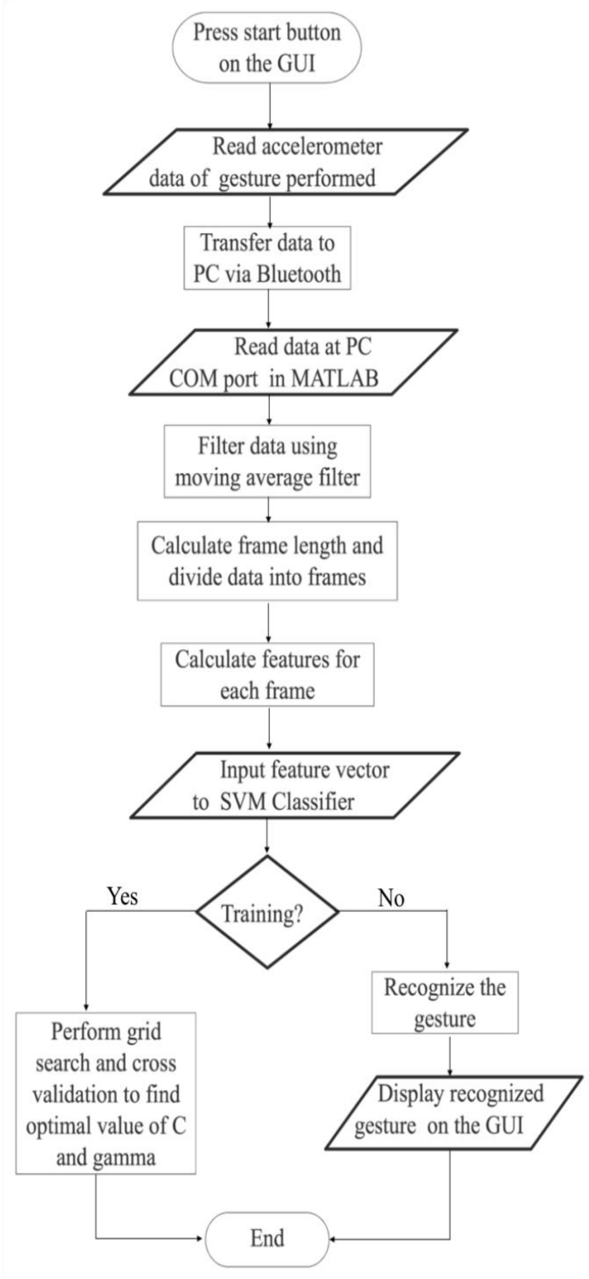
Press start button on the GUI
Read accelerometer data of gesture performed
Transfer data to PC via Bluetooth
Read data at PC COM port in MATLAB
Filter data using moving average filter
Calculate frame length and divide data into frames
Calculate features for each frame
Input feature vector to SVM Classifier
Training?
Yes
No
Perform grid search and cross validation to find optimal value of C and gamma
Recognize the gesture
Display recognized gesture on the GUI
End

Figure 7: Flowchart of gesture recognition system

2.5 Training Phase

Eight individuals were asked to perform each gesture daily for a time period of two weeks. The data set collected at the end of two weeks, that is, a total of 1120 samples were used to train the classifier. The optimal values of the parameters C and gamma were found by grid search and by performing four fold cross validation to avoid over-fitting.

2.6 Testing Phase

Testing was performed on 20 individuals both men and women aged between 17 years to 50 years (labelled from A to T for the purpose of this experiment). Out of the twenty individuals, eight were same as those who had been involved in the training phase (user dependent group) while the rest twelve had no previous involvement in the project (user independent group). Each individual was first made to practice the gesture once with the hardware to get acquainted with the process and then it was tested on the classifier. The classifier displayed the recognized gesture on the GUI which is shown in Figure 8.

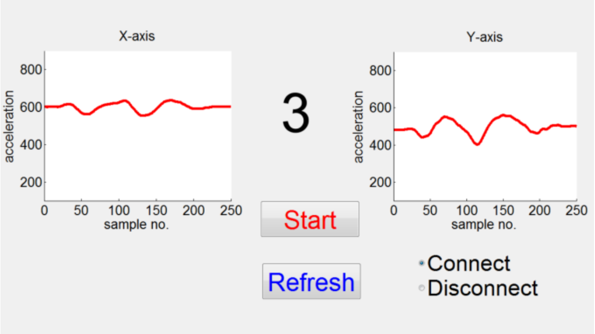

Figure 8: The predicted digit is displayed on the GUI. It also plots the filtered X and Y axis accelerometer readings.

3. RESULT

The outcome of the recognition algorithm is tabulated below (see Table 1 and 2). It achieved an accuracy of 92.5% and 97.5% for user independent and user dependent group respectively. The greater accuracy for user dependent group is understandable as the recognition algorithm has been trained using samples collected from the individuals belonging to this group. The recognition accuracy of each digit is plotted in Figure 9. The minimum accuracy was obtained for 0 while the digits 4, 6, 8 and 9 were recognized with an accuracy of 100% in both the groups.

Table 1. Outcome of classifier for user independent group

User	1	2	3	4	5	6	7	8	9	0
A	1	2	3	4	8	6	7	8	9	0
B	1	2	3	4	5	6	1	8	9	0
C	1	2	3	4	5	6	7	8	9	2
D	1	0	3	4	5	6	7	8	9	0
E	1	2	8	4	5	6	7	8	9	0
F	1	2	3	4	5	6	7	8	9	0
G	1	2	3	4	5	6	7	8	9	0
H	1	2	3	4	5	6	7	8	9	0
I	7	2	8	4	5	6	7	8	9	2
J	1	2	3	4	5	6	7	8	9	0
K	1	2	3	4	5	6	7	8	9	0
L	1	2	3	4	5	6	7	8	9	2

Table 2. Outcome of classifier for user dependent group

User	1	2	3	4	5	6	7	8	9	0
M	1	2	3	4	5	6	7	8	9	0
N	7	2	3	4	5	6	7	8	9	0
O	1	2	3	4	5	6	7	8	9	0
P	1	2	3	4	5	6	7	8	9	0
Q	1	2	3	4	5	6	7	8	9	0
R	1	2	3	4	5	6	7	8	9	0
S	1	2	3	4	5	6	7	8	9	2
T	1	2	3	4	5	6	7	8	9	0

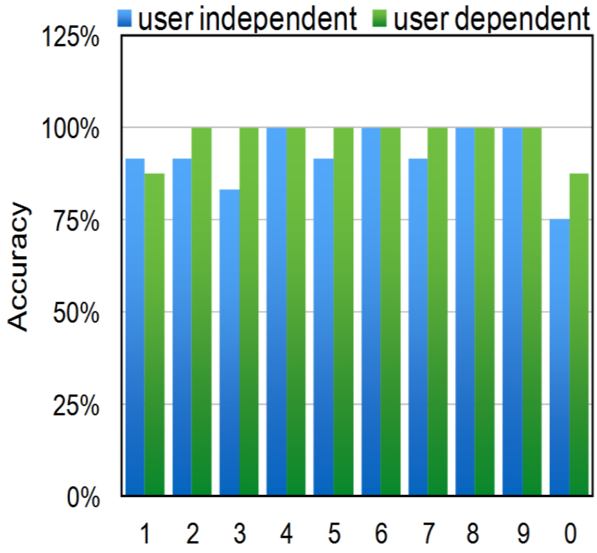

Figure 9: Comparison of the accuracy of recognition of each digit for user dependent and independent group

4. CONCLUSION

This paper presented a gesture recognition system for digits from 0-9. The wireless sensing device designed was fitted with a three dimensional accelerometer. The accelerometer data was filtered and divided into frames of variable lengths. Temporal features were extracted from each frame to form a feature vector which was given as input to the classifier. Multi-class SVM was used for pattern recognition. The SVM classifier achieved an accuracy that is comparable to the previous recognition techniques for user independent case. For user dependent case, it gave an accuracy that is much higher than the previous recognition techniques for the recognition of digits.

Since temporal features can be calculated easily with minimal computational cost, this method of gesture recognition can be employed in real time applications. Devices with digit recognition can be incorporated in ATM machines so that even people with vision disabilities can use them.

As observed, most of the inaccuracies that arose were due to the time limit to perform a gesture, since at times the users were not able to complete the gesture within the allotted window of 3 seconds. In the near future this will be overcome by incorporating a button on the recognition device to indicate start and stop of the gesture. Moreover the training dataset will be expanded to make the recognition system more robust especially for the user independent case. New features will also be investigated to incorporate in the feature vector to increase the recognition accuracy.

5. ACKNOWLEDGMENTS

The author would like to thank Dr. Rajiv Kapoor of Delhi Technological University for his support and guidance throughout the course of the research.

6. REFERENCES

[1] H. K. Kaura, V. Honrao, S. Patil, and P. Shetty, “Gesture Controlled Robot using Image Processing,” Int. J. Adv. Res. Artif. Intell., vol. 2, no. 5, pp. 69–77, 2013.

[2] V. S. Kulkarni and S. . Lokhande, “Appearance Based Recognition of American Sign Language Using Gesture Segmentation,” Int. J. Comput. Sci. Eng., vol. 02, no. 03, pp. 560–565, 2010.

[3] A. S. Ghotkar and G. K. Kharate, “Vision based Real Time Hand Gesture Recognition Techniques for Human Computer Interaction,” Int. J. Comput. Appl., vol. 70, no. 16, pp. 1–6, 2013.

[4] R. B. Dan and P. S. Mohod, “Survey on Hand Gesture Recognition Approaches,” Int. J. Comput. Sci. Inf. Technol., vol. 5, no. 2, pp. 2050–2052, 2014.

[5] P. Kumar, J. Verma, and S. Prasad, “Hand Data Glove : A Wearable Real-Time Device for Human- Computer Interaction,” Int. J. Adv. Sci. Technol., vol. 43, pp. 15–26, 2012.

[6] E. Morganti, L. Angelini, A. Adami, D. Lalanne, L. Lorenzelli, and E. Mugellini, “A Smart Watch with Embedded Sensors to Recognize Objects, Grasps and Forearm Gestures,” Procedia Eng., vol. 41, pp. 1169–1175, 2012.

[7] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, “Gesture Recognition with a 3-D Accelerometer,” in 6th International Conference on Ubiquitous Intelligence and Computing (UIC ’09), 2009, pp. 25–38.

[8] T. Schl, B. Poppinga, N. Henze, and S. Boll, “Gesture Recognition with a Wii Controller,” in 2nd international conference on Tangible and embedded interaction (TEI ’08), 2008, pp. 1–4.

[9] B. M. Lee-Cosio, C. Delgado-Mata, and J. Ibanez, “ANN for Gesture Recognition using Accelerometer Data,” Procedia Technol., vol. 3, pp. 109–120, 2012.

[10] J. Liu, Z. Pan, and L. Xiangcheng, “An accelerometer-based gesture recognition algorithm and its application for 3D interaction,” Comput. Sci. Inf. Syst., vol. 7, no. 1, pp. 177–188, 2010.

[11] K. J. Patil, A. H. Karode, and S. R. Suralkar, “Gesture Recognition of Handwritten Digit using Accelerometer based Digital Pen,” Int. J. Appl. or Innov. Eng. Manag., vol. 3, no. 4, pp. 353–357, 2014.

[12] R. Xu, S. Zhou, and W. J. Li, “MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition,” IEEE Sens. J., vol. 12, no. 5, pp. 1166–1173, 2012.

[13] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, “uWave: Accelerometer-based personalized gesture recognition and its applications,” Pervasive Mob. Comput., vol. 5, no. 6, pp. 657–675, Dec. 2009.

[14] Chang, C. & Lin, C. (2011). LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[15] C. Hsu, C. Chang, and C. Lin, “A Practical Guide to Support Vector Classification,” pp. 1–16, 2010.

[16] D. Figo, P. C. Diniz, D. R. Ferreira, and M. P. Cardoso, “Preprocessing Techniques for Context Recognition from Accelerometer Data,” Pers. Ubiquitous Comput., vol. 14, no. 7, pp. 645–662, 2010.