Real Time 3D-Handwritten Character and Gesture Recognition for Smartphone

Neelasagar K
Dept. of Electronics and Communication
College of Engineering
Trivandrum

K Suresh
Dept. of Electronics and Communication
Govt. Engg. College, Barton Hill,
Thiruvananthapuram

ABSTRACT

This Paper presents two different models for smartphone based 3D-handwritten character and gesture recognition. Smartphones available today are equipped with inbuilt sensors like accelerometer, gyroscope, and gravity sensor, which are able to provide data on motion of the device in 3D space. The sensor readings are easily accessible through native operating system provided by the smartphone. The accelerometer and gyroscope sensors data have been used in the models to make the systems more expressive and to disambiguate recognition. The acceleration generated by the hand motions are generated by the 3-axes accelerometer present in smartphone. Data from gyroscope is used to obtain the quaternion rotation matrix, which is used to remove the tilting and inclination offset. An automatic segmentation algorithm is implemented to identify individual gesture or character in a sequence. First model is based on training and evaluation mode, while second one is based on training-less algorithm. The recognition models as presented in this paper can be used for both user dependent and user independent 3d-character and gesture recognition. Results shows that, Model I gives an efficient recognition rate ranging from 90% to 94% obtained with minimal training sequence, while the Model II gives a recognition rate of 88%.

General Terms

Pattern Recognition, Smartphone

Keywords

Accelerometer, Dynamic time warping, Gyroscope, Mahalanobis distance, Quaternion Rotation

1. INTRODUCTION

Growth in miniaturization technologies in electronics field has greatly reduced the dimension and weight of consumer electronics product, such as smartphones and handheld computers. The increase in the usage of smartphones in our daily life has made user interface technology more reliable and important. Presence of touch screen interface in smartphone had been an innovative breakthrough in 2D-character or gesture recognition. However, 3D-character or gesture recognition is comparatively tougher because different users have different speeds and styles. But, by adding touch-free functionality to smartphones, there is a brand new dimension to mobile games and applications. the touch-free functionality to smartphones can be incorporated using the inbuilt sensors like accelerometer, gyroscope, and gravity sensors. These sensors have an advantage that when the phone is in motion, all these sensor's reading changes, and this can be exploited to recognize the motion trajectory.

There are multiple technical challenges in touch free interaction. Unlike many pattern recognition techniques, e.g. speech recognition, 3D-character/gesture recognition lacks a widely accepted vocabulary. Research in this area is focusing on new techniques and methods that would reduce the processing time and achives higher recognition accuracy.

Most of the existing 3D-character/gesture recognition algorithms are either *Camera-based* or *MEMS-sensor* based. Due to the camera based method limitations such as slower response and relatively large data collections/processing requirement. Hence, for 3D gesture recognition the prefer methods are based on MEMS sensors [1]-[7],[10]-[17].

Recently, some researchers have concentrated on decreasing the error of handwriting trajectory reconstruction by manipulating acceleration signals and angular velocities of inertial sensors [8]. However, the reconstructed trajectories suffer from various intrinsic errors of inertial sensors. Hence, new algorithms were developed for error compensation of inertial sensors to improve the recognition accuracy. Yang *et al.* [9] proposed a pen-type input device to track trajectories in 3D space by using accelerometers and gyroscopes. An efficient acceleration error compensation algorithm based on zero velocity compensation was developed to reduce acceleration errors for acquiring accurate reconstructed trajectory. Ruize Xu *et al.* [2] proposed three models for non-specific-user hand gesture recognition using MEMS accelerometer. In this approach the motion of the device is restricted to a single plane motion, i.e. motion involving just two axes, which is not always possible for smartphone. In [17], the velocity, and position features were obtained from raw acceleration signals, and then, the PCA was utilized to reduce the feature dimension size. They employed a hidden Markov model (HMM) with dynamic time warping algorithms to recognize 3-D handwritten character with a recognition rate of 90%.

In this paper, two models for recognizing the space drawn 3D-character\gestures with smartphone is proposed. In both the models accelerometer and gyroscope readings are used for recognition. The triaxial accelerometer present in smartphone gives the acceleration along the three axes, when smartphone is in motion. The triaxial gyroscope data is used to improve the performance of the system by providing rotational calibration for acceleration sequence with the help of quaternion rotation matrix. Model-I consists of training as well as detection mode. Dynamic Time Warping (DTW) and Mahalanobis distance measure is being employed in this model. Model-II is a training-less algorithm and is used for 3D-gesture recognition only. The presented algorithm is simple for both the models with minimum computation time so that it can be incorporated in smartphone as an android application.

The rest of this paper is organized as follows. In Section II, the data acquisition and data preprocessing methods are described. The proposed model I based on quaternion rotation matrix and DTW algorithm along with Mahalanobis distance measure is presented in Section III. Section IV, describes the model II which is based on the training-less algorithm. Implemented prototype of models is given in section V. In section VI, the performance analysis of the models is presented. Finally, the work is concluded in VII section.

2. DATA ACQUISITION AND PREPROCESSING

Here a smartphone which have motion sensors as a sensing system is used in the experiment for real time 3D-character\gesture recognition. A sensing application is developed with the help of native operating system of smartphone (Android) [23,24]. When the application developed is enabled, the acceleration and angular velocity along three axes are detected by the motion sensors in the smartphone. Once the raw data from the sensors are obtained using Android, the next stage in both the models are data preprocessing and then the segmentation stage which automatically identifies the start and end of each 3D-character\gesture.

A 3D-character or gesture sequence is considered to be a three dimensional vector having accelerometer reading along three axes. For this work Arabic numerals shown in Fig. 1 are used as 3D-handwritten character and Fig. 2 shows the 3D-gestures. Combining the accelerometer and gyroscope sensors it is intend to make the recognition more accurate and expressive. Motion sensors present in the smartphones have different sampling rate [5] and it can be varied using the native operating system. The sampling rate is phone specific and here it is sampled at the rate of 17 samples per second.

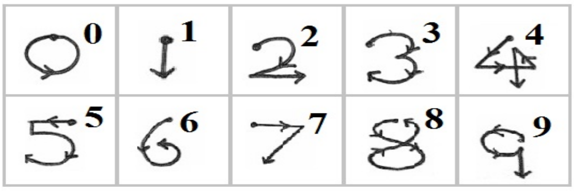

Fig. 1: Trajectories of Arabic numerals

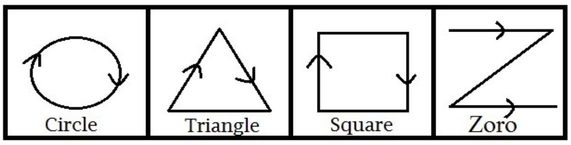

Fig. 2: 3D-Gestures used in the experiment

The sensitivity of the sensors in smartphones is relatively high. Data preprocessing is a technique that transforms raw sensor's data into an understandable format. Different preprocessing filters are present, in order to reduce the input data, without losing too much of relevant information. The data preprocessing eliminates noise and reduces user, device dependency for recognition.

2.1 Accelerometer Data Preprocessing

2.1.1 Segmentation. The purpose of the segmentation algorithm is to find the terminal points of each 3D-character gesture in a data set of character gesture sequence. The algorithm used in [3] checks various conditions on all data points and picks out the most likely data points as the character gesture termination points. To check that a sample X_i belongs to character gesture sequence,the average of the magnitude of N consecutive samples is calculated and compared with the threshold value V_{th} = 0.7 given in (1). The decision to take threshold value as 0.7 is based on the empirical tests. From the experimental results, the ideal value of the threshold is found to be 0.7 to achieve the best recognition result.

$$(\sum_{i=0}^{N-1} |X_{i+1}|/N) - X_i > 0.7 \tag{1}$$

2.1.2 Filtering. Like any other hardware system, smartphone is subject to undesirable noises. Furthermore, it is always vibrating due to human nature. An IIR (Infinite Impulse Response) low pass filter is used in [3] is to compensate the noise. For given input raw data sequence $X(1 : n)$, the smoothed output sample is obtained by using (2), where α is taken as 0.4, since the time constant (RC) of accelerometer is significantly larger than the sampling interval. Low pass filtered output is shown in Fig. 3(b).

$$Y_i = \begin{cases} X_i & i = 0 \\ Y_{i-1} + \alpha(X_i - Y_{i-1}) & 0 < i < N \end{cases} \tag{2}$$

Even when the smartphone is not in motion the accelerometer has an offset value of g (9.8 m/s^2) [5] along the axis perpendicular to the earth surface. To remove the effect of gravity on the accelerometer readings, an IIR digital high pass filter is used [3].

$$Y_i = \alpha(Y_{i-1} + X_i - X_{i-1}) \tag{3}$$

For given input sequence $X(1 : n)$ filtered output sample is obtained using (3), where α =0.8 is the ratio between time constant and sampling period [5]. High pass filtered output is shown in Fig. 3(c).

2.1.3 Normalization. The difference between a fast motion and slow motion has to be adjusted. Other than the length of the signal, amplitude of the acceleration vector seems to be a major difference. The normalization eliminates difference caused by speed variations. To normalize the acceleration vector, each value is divided by the magnitude of the vector. Fig. 3(d) shows the normalized plot of acceleration. After normalization, acceleration data values are found to be between the range (-1,1).

2.2 Gyro sensor Data Preprocessing

Gyro sensor provides the angular velocities in *rad/sec* along each of 3 axes. Similar to the acceleration data preprocessing, data processing for gyroscope readings is done to get valuable information, so that it can be used along with the acceleration sequence in the proposed recognition models to improve the performance by providing a rotational calibration. First the gyroscope readings was low pass

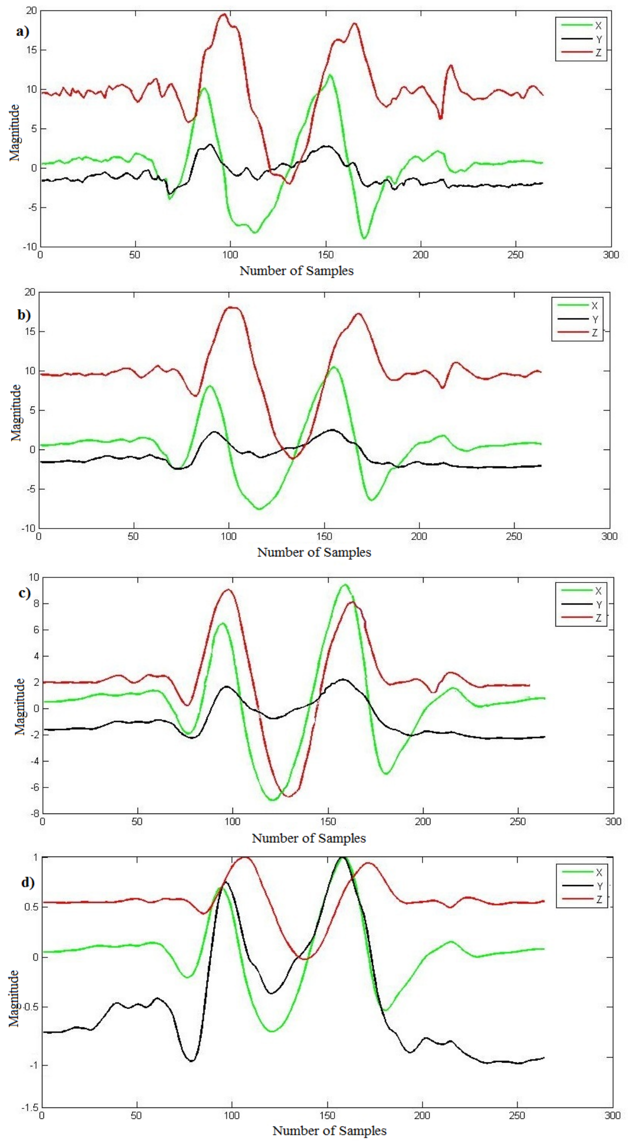

Fig. 3: Acceleration at each steps of the data pre-processing for the circle gesture. a)Raw values directly from accelerometer. b)Low pass filtered accelerations. c)High pass filtered accelerations. d)Normalized accelerations.

filtered using (2). Then the low pass filtered output is normalized, to get the unit quaternion vector [21,22]. To get the correct orientation of the device while on motion rotation matrix is used.
Quaternion is a method of specifying rotations and orientations of a device coordinate system. Quaternions encode rotations by four real numbers (or two complex numbers), whereas the linear representation of these transformations as 3 X 3 matrices requires nine. Interpolating the quaternionic representation of a sequence of rotations is more natural than doing so for the familiar Euler angles.

The quaternion representation of a rotation can be interpreted as an axis angle representation of the rotation according to (4)

$$p = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} cos(\frac{1}{2}\beta) \\ sin(\frac{1}{2}\beta)\begin{pmatrix} \hat{v}_x \\ \hat{v}_y \\ \hat{v}_z \end{pmatrix} \end{pmatrix} = \begin{pmatrix} cos(\frac{1}{2}\beta) \\ sin(\frac{1}{2}\beta)\hat{v} \end{pmatrix} \quad (4)$$

where β is a positive rotation around the axis of rotation v, assumed to be of unit length. If N represents the number of samples in a single character\gesture sequence, then there are N-1 samples of P matrices. Here it is considered Q as a collection of all p matrices given in (5).

$$Q = ([p_0], [p_1], [p_2], [p_3], [p_4]......[p_{N-1}]) \quad (5)$$

Q is a 4 x N-1 matrix. For every column of Q matrix, rotation matrix is obtained using (6). Using the gyroscope data, total of N-1 rotation matrix is obtained for a single character\gesture sequence. Unit length quaternion is converted to a rotation matrix R using (6)

$$\begin{pmatrix} 2q_0^2 - 1 + 2q_1^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 - 1 + 2q_2^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 - 1 + 2q_3^2 \end{pmatrix} \quad (6)$$

The rotation matrix, it is multiplied with the acceleration sequence to give a new set of acceleration for same character\gesture.

3. MODEL I: BASED ON QUATERNION ROTATION AND DYNAMIC TIME WARPING

The path of a character\gesture trajectory recorded by the smartphone sensors involves all three dimensions, giving multidimensional time series in which not only exceptional measurements are included but also their temporal relations to other dimensions, so that algorithm can be processed in real time. For recognition this means that it is not sufficient to examine the dimensions separately. There must also be synchronization between them. For this accelerometer, and gyroscope sensors present in smartphone are utilized.

3.1 Training phase

Training is the step in the recognition process and is depicted in Fig. 4.

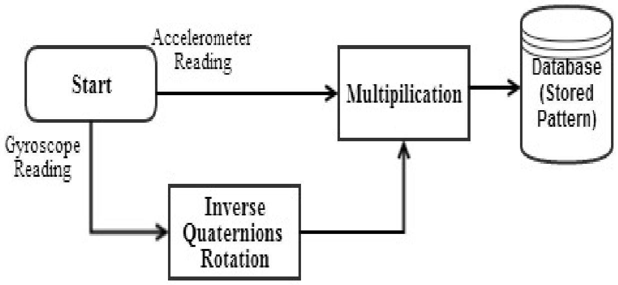

Fig. 4: Training Phase

It's a process of training characters\gestures, to be used as template sequence for detection process. For training both accelerometer and gyroscope readings are considered. The 3D characters\gestures are trained in such a way that the smartphone z axis is facing downwards as shown in Fig. 5

As its not always possible to hold and draw a character\gesture exactly as in Fig. 5. So to compensate such limitations, the acceleration sequence is multiplied with inverse quaternion rotation matrix to get the new pattren which is stored in the database.

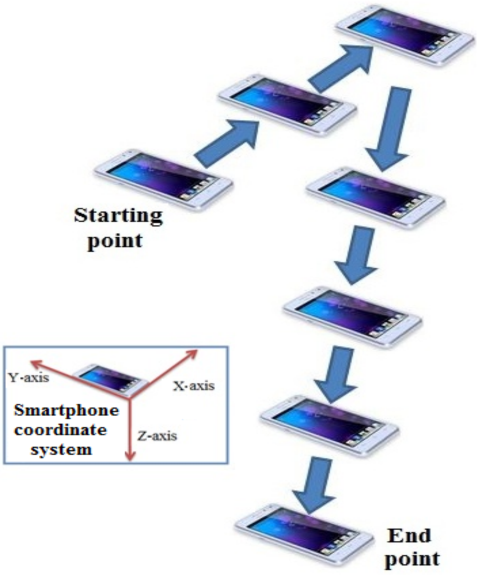

Fig. 5: Motion trajectory of Arabic Number ”7” using a smartphone.

The training set is created by randomly choosing M repetitions for each character\gesture. The value of M is varied from 1 to 10 in the experiment. The database consists of minimal repetitions and is built by acquiring 3d-handwritten character\gesture from 10 participants using a smartphone. 3D-handwritten character\gesture acceleration waveforms from the same person can differ drastically. The noise added by inevitable hand shaking or minor remote tilting is removed by taking the average of M repetitions to form a single sequence for a specific character or gesture.

3.2 Dectection phase

During the real time detection mode, the unknown character's accelerometer and gyroscope data is preprocessed separately. Using the gyroscope sensors reading the quaternion rotation matrix is obtained which is used to rotate the trained set of character in database to new roll θ, pitch ϕ, and yaw ψ angles [10]. The work flow of Model I is shown in Fig. 6.

The quaternion rotation matrix is obtained using (4), (5), and (6). There are N-1 rotation matrices for a single character\gesture sequence.The average of N-1 rotation matrix is calculated to reduce the computational complexity. After averaging, only one final rotation matrix exits per character\gesture sequence. Then the final rotation matrix obtained from unknown character\gesture sequence is multiplied with the stored pattern present in the database, i.e. the stored pattern is rotated to unknown character\gesture sequence orientation. The DTW (Dynamic time warping) [3] along with Mahalanobis distance measure [3] is applied to find the best match.

DTW is sequence aligning algorithm used widely in digital signal processing areas like speech recognition. The DTW algorithm matches two sequences irrespective of its length and rate at which the signal is produced. The algorithm computes best distance between two signals ignoring compressions and stretches in the signal. The learning data set sequence $X_{i...N-1} = [X_x, X_y, X_z]$, and unknown data sequence $Y_{j...M-1} = [Y_x, Y_y, Y_z]$, the matrix of size NxM is calculated using (7).

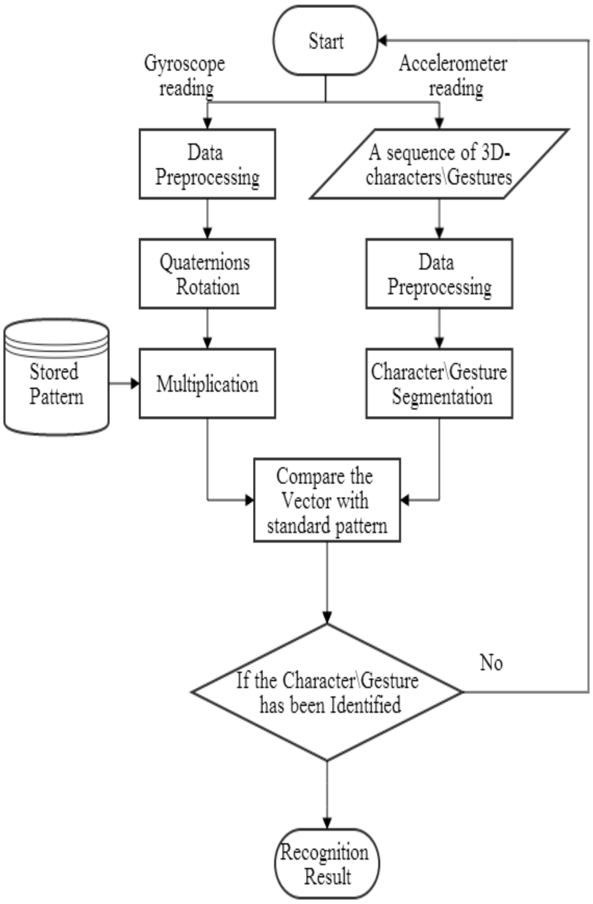
Start
Gyroscope reading
Accelerometer reading
Data Preprocessing
A sequence of 3D-characters\Gestures
Quaternions Rotation
Data Preprocessing
Stored Pattern
Multiplication
Character\Gesture Segmentation
Compare the Vector with standard pattern
If the Character\Gesture has been Identified
No
Recognition Result

Fig. 6: Work Flow of Model I recognition system

$$d_{min} = min \begin{Bmatrix} DTW_{matrix}(i-1, j), \\ DTW_{matrix}(i, j-1), \\ DTW_{matrix}(i-1, j-1) \end{Bmatrix} \quad (7)$$

$$DTW_{matrix}(i, j) = d(X_i, Y_i) + d_{min} \quad (8)$$

Usually $d(X_i, Y_i)$ is the Euclidean distance, but the Mahalanobis distance is calculated [3] between the two 3D point datasets X_i and Y_i. During evaluation, Mahalanobis distance of unknown sequence Y is obtained using (9).

$$D_{mahalanobis} = \left\{ (\mu_x - \mu_y)^T * K^{-1} * (\mu_x - \mu_y) \right\} \quad (9)$$

Where μ_x and μ_y are mean vectors of X and Y data sequence respectively. The sequence with minimum distance is taken as the best matching 3D-character\gesture sequence.

4. MODEL II: BASED ON TRAINING-LESS ALGORITHM

Model II does not depend on the training phase, it depend on the evaluation mode. Since there is no need of training sequence, the use of database for storing the pattern is avoided. The work flow of the model II is shown in the Fig. 7.
Instead of using the feature of acceleration alone, the velocity increment or the area bounded by the acceleration curve is used to implement the recognition algorithm. This algorithm is succesful only for 3D-geometrical gesture recognition and not for 3D-character recognition. The processed acceleration sequence is passed through the segmentation stage where start and end of the gesture is identified. Using the gyroscope reading or acceleration reading rotational calibration is provided to the acceleration data sequence. The velocity increment of the acceleration data is taken using trapezoidal integration.

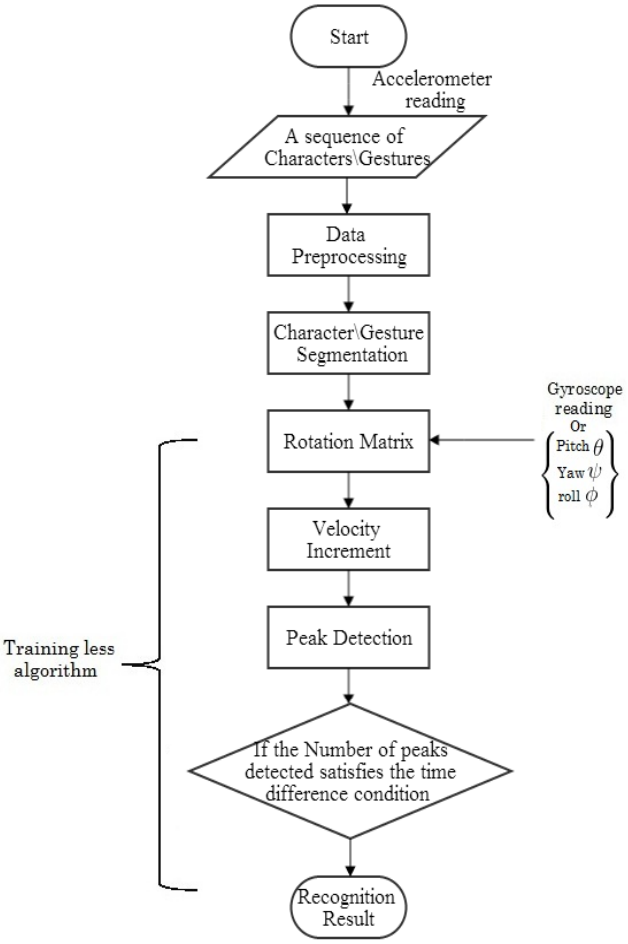

Fig. 7: Work Flow of Model II recognition system

The integration of acceleration on three directions gives velocity of the motion in three directions. The physical meaning of integrated data is that it represents the increase and decrease in velocity, and this concept has been exploited to obtain the training-less algorithm for 3D geometrical gesture recognition. Due to the intensity variance of each gesture, an integrated sequence should be normalized once again before it is passed through the next stage. Using (10) the normalized sequence of velocity [2] is obtained.

$$Norm_{sequence} = \frac{integrated_{sequence}}{max_{value}} \tag{10}$$

where max_{value} is the maximum value in the integrated sequence. Then the training-less algorithm is applied to normalized sequence. Peak detection is the most crucial action in the training less algorithm. For peak detection the three axes reading of accelerometer are processed separately to detect the sudden rise or fall in velocity value for that axis. The sudden fall of velocity is taken as negative peak and sudden rise as positive peak. A counter is provided to calculate the total number of negative and positive peaks in all the three axes. Considering time instances, t_a and t_b for two successive extrema (peak and valley), the following steps are performed to recognize the unknown character\ gesture.

—Calculate the time difference between two successive extrema $|t_a - t_b|$ where $t_a > t_b$

—If $|t_a - t_b|$ <threshold(T), then the peak detected at t_a is relevant peak.

—When one combination of positive and negative peaks are detected , reinitialize the next peak time value as the first time value and repeat the same steps as above until all the peaks detected satisfies the time difference condition in first step.

—Calculate the number of positive and negative peaks in the processed gesture sequence.

—A specific number of peaks recognize the unknown 3D gesture.

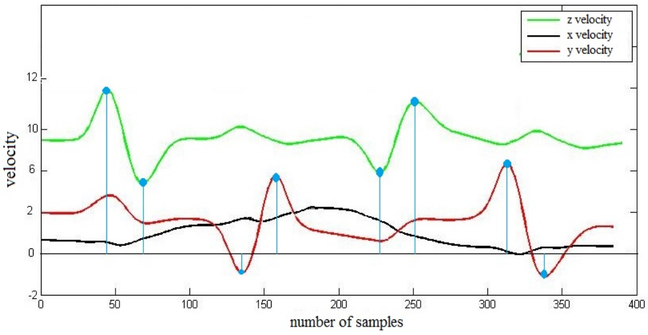

Fig. 8: Peaks detected for a square gesture

For example if the number of positive and negative peaks detected is four then the gesture is square, if it's three then the unknown gesture is triangle. From the empirical experiment it is found that whenever there is a sudden change of direction in motion of the smartphone, there is a sudden change in the acceleration and velocity values. Fig. 8 shows the velocity plot of square gesture with the sudden change of direction is found at each corner of square gesture.

5. IMPLEMENTATION DETAILS

The proposed models are implemented in Android and the data corresponding to the different 3D character\gesture is collected using a smartphone, which has a built-in 3-axes accelerometer and gyroscope sensors. The devices used for implementation are Samsung S3 and LG Nexus android phones. The system architecture consists of three different layers, a *sensor layer*, which acquires the information provided by the inertial sensors, a *fusion layer*, which includes the recognition algorithms to recognize the characters, and a *communication layer*, which offers the recognition result.

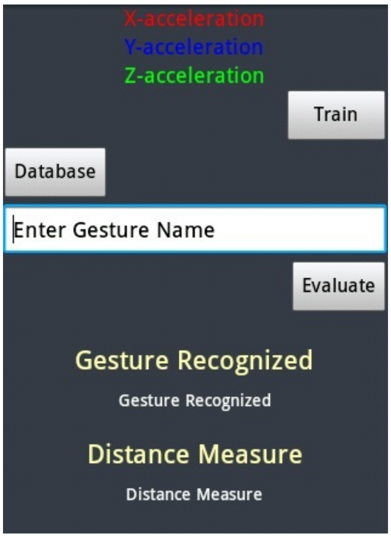

Fig. 9: Developed Android application screen for Model I

The developed application have recognition service which runs in background and the 3d-character\gesture data provider to insert, delete, update and retrieve character\gesture values from database. Using the sensor layer, the accelerometer and gyroscope readings are acquired using android from the smartphone when it's on motion. In Model I, during the training mode, only sensor layer works. In learning mode it receives a set of 3D-characters\gestures, preprocesses it and stores in database through 3D-character\gesture data provider in sensor layer. Fig. 9 shows the developed prototype android application screen for model I.
In evaluation mode the recognition services in fusion layer listens for unknown characters\gestures data sequence from sensor layer and when motion is detected, fusion layer preprocesses the input and pass it to evaluation algorithm which is also present in fusion layer. Communication layer gives the recognition result when unknown sequence is matched to a specific 3d-character from the database. For Model II, fusion layer and communication layer have been used for evaluation mode, as it does not depend on training mode.

6. PERFORMANCE ANALYSIS OF THE ALGORITHMS

To test the performance of the proposed recognition algorithms, few experiments have been carried out includes both user-dependent and user-independent cases.
For model I, the first experiment is done on 3D-character recognition, with Arabic numerals. Here 40 instances of input motion samples are recorded as shown in Fig. 1. Black points represent the beginning locations and the arrows represent the end locations and the directions of trajectories. Table 1 shows the confusion matrix, for the experiment on user-independent case for model I recognition system.

Table 1. : Confusion Matrix for the Model I system, 40 testing samples per arabic numerals are used

		Input Digit									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Recognized Digit	**0**	35						1		2	1
	1		38						1		
	2	1		37			1				
	3			1	36					1	
	4		1			38	1		1		
	5	1			1		35	1			
	6				1			36		1	
	7		1			1	1		37		1
	8	1		1	2		1	1		35	1
	9	2		1		1	1	1	1	1	37

The test result of model I shown in Table 2 is based on 5 training sequence per gesture from five different users and was tested for 30 samples for single gesture, totaling 120 testing gesture.

Table 2. : Comparison of gesture recognition accuracy of model I, for 3D gesture

Gesture	**Observed Recognition Results**		
	Total Number	**Correct Detection**	**Recognition Rate(%)**
Circle	30	28	93
Triangle	30	26	86
Square	30	27	90
Zoro	30	25	83

From Table 1 and Table 2, it is clear that the recognition result for model I is better for both 3D- character recognition as well as for 3D-gesture. The performance of proposed model I for 3D-character\gesture recognition is better than the models described in [2],[3],[20].

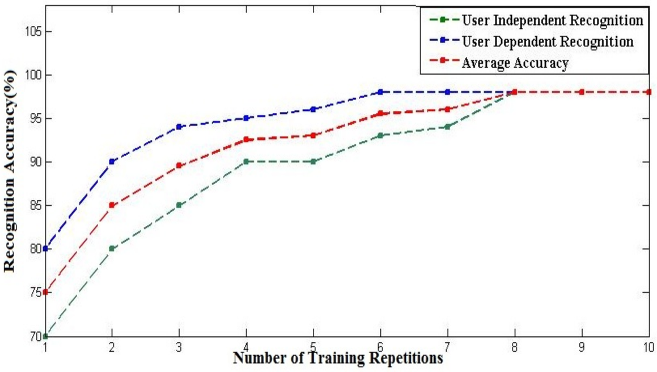

Fig. 10: Average accuracy against the number of training repetitions for Model I

The final experiment on model I is done by varying the number of training sample (M= 1 to 10). In this experiment, the ten participants were asked to hold the smartphone to write ten digits, which are shown in Fig.1 without any ambit restriction in a 3D space. Each participant was asked to repeat each digit for ten times. Therefore, a total of 1000 (=10 x 10 x 10) data are generated. Fig. 10 shows a graph of the average recognition rate against the number of training repetitions per character for both user dependent and user independent recognition. The graph shows that with as minimum as four repetitions, the recognition system is capable of achieving an average accuracy greater than 90%.

Table 3. : Comparison of gesture recognition accuracy of model II, for 3d-gesture

Gesture	Observed Recognition Results		
	Total Number	Correct Detection	Recognition Rate(%)
Circle	30	26	86
Triangle	30	27	90
Square	30	29	96
Zoro	30	25	83

For Model II, the experiment was done only for 3D-Gesture shown in Fig. 2. Table 3 shows the recognition accuracy of model II. Since it does not depend on training mode, the computation time is found to be faster than the first proposed model but the accuracy of the model II is found to be inferior to that of model I, but is better than the recognition system based on DTW alone.

7. CONCLUSION

In this paper, two algorithms for 3D-character\gesture recognition system based on smartphone are proposed. The system utilizes a 3-axes accelerometer and gyroscope sensors present in smartphone and thus can be readily implemented. The recognition system consists of sensor data collection, segmentation and recognition. After receiving data from the sensors, a segmentation algorithm is applied to determine the starting and end points of every input character automatically. Then the character sequence is preprocessed, quaternion rotation is applied to the stored 3d- character sequence and finally, the character sequence is compared with the unknown 3d-character patterns to determine the most likely character. The system is tested on Arabic numerals and some geometrical 3D gestures whose database contains minimal repetitions collected from 10 users. The proposed model I achieves recognition rate of more than 95% when carrying out user-dependent recognition. As for user-independent recognition, accuracy is obtained that is competitive with many other systems. Model II recognition system is based on training-less algorithm, so the computational time will be less than the model I, but the system fails to recognize the 3D-gesture. The performance of the model II can be increased by adding additional parameters for recognition.

8. ACKNOWLEDGMENT

The authors would like to thank the International Center for Free and Open Source Software (ICFOSS), Trivandrum, for supporting this work.

9. REFERENCES

[1] Ahmad Akl, Shahrokh Valaee, “Accelerometer-based gesture recognition via dynamic-time warping, Affinity propagation, and compressive sensing,” in Proc. of IEEE ICASSP, pp. 2270–2273, march 2010.

[2] Ruize Xu, Shengli Zhou, Wen J. Li, “MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition,” in IEEE Sensors Journal, vol. 12, no. 5, pp. 1166–1173, may 2012.

[3] Sethu Janaki V M, Satish Babu and Sreekanth S S, “Real Time Recognition of 3D Gestures in Mobile Devices,” in Proc. of IEEE RAICS,2013

[4] Jiahui Wu, Gang Pan, Daqing Zhang, Guande Qi, Shijian Li, “Gesture Recognition with a 3-D Accelerometer,”in UIC 2009,LNCS 5585 pp. 25–38, 2009.

[5] Mark Pedley, “Tilt Sensing Using a Three-Axis Accelerometer,”Application Note of Freescale Semiconductor, Doc No. AN3461,Rev. 6, 03/2013.

[6] Duan Hong,Luo Yang, “A Gestures Trajectory Recognition Method Based on DTW,” in proc. of International Conference on Computer Science and Electronics Engineering (ICCSEE), 2013.

[7] Jiayang Liu, Zhen Wang, and Lin Zhong,Jehan Wickramasuriya,Venu Vasudevan, “uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications,”in Proc. of IEEE International Conference on Pervasive Computing and Communications, pp. 1–9, march 2009.

[8] Oliver J. Woodman, “An introduction to inertial navigation,” Technical Report, UCAM-CL-TR-696, ISSN 1476-2986, Aug 2007.

[9] J. Yang, W. Chang, W. C. Bang, E. S. Choi, K. H. Kang, S. J. Cho, and D. Y. Kim“Analysis and compensation of errors in the input device based on inertial sensors,”in Proc. IEEE Int. Conf. Inf. Technol.-Coding and Computing ,pp. 790–796, 2004.

[10] Talat Ozyagcilar, “Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors,” Application Note of Freescale Semiconductor , Document Number: AN4248, Rev. 3, 01/2012.

[11] Pylvänäinen T, “Accelerometer Based Gesture Recognition Using Continuous HMMs,”in Springer Berlin / Heidelberg, Pattern Recognition and Image Analysis, pp. 411–430, 2005.

[12] Niezen G and Hancke GP, “Gesture recognition as ubiquitous input for mobile phones,” in Proc. of the Workshop on Devices that Alter Perception, 2008.

[13] Joselli M and Clua E, “A Framework for Touch and Accelerometer Gesture Recognition for Mobile Games, ”in VIII IEEE Brazilian Symposium on Games and Digital Entertainment, pp. 141–150, 2009.

[14] Westeyn T, Brashear H, Atrash A and Starner T, “Georgia Tech gesture toolkit: supporting experiments in gesture recognition, ”in Proc. of the 5th Int. Conf. on Multimodal Interfaces ACM, pp. 85–92, 2003.

[15] T. Schlömer, B. Poppinga, N. Henze,and S. Boll., “Gesture recognition with a wii controller” In Proc.TEI ,pp. 11–15, New York, NY, USA, 2008.ACM.

[16] L. Kratz, M. Smith, and F.J. Lee. Wiizards, “3D gesture recognition for game play input” In Proc.Future Play ,pp. 209–212, New York, NY, USA, 2007. ACM.

[17] S. D. Choi, A. S. Lee, and S. Y. Lee, “On-line handwritten character recognition with 3D Accelerometer, ” in Proc.IEEE Int. Conf. Inf. Acquisition, pp. 845–850, 2006.

[18] Wu J, Pan G, Zhang D, Qi G, and Li S, “Gesture recognition with a 3D accelerometer. Ubiquitous Intelligence and Computing, ”in LNCS,Springer, pp. 25–38, 2009.

[19] T. Yang and Y. Xu’, “A Hidden Markov Model for Gesture Recognition, ” CMU-RI-TR-94 10, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, May 1994.

[20] S. Zhou, Q. Shan, F. Fei, W. J. Li, C. P. Kwong, and C. K. Wu et al., “Gesture recognition for interactive controllers using MEMS motion sensors,” in Proc. IEEE Int. Conf. Nano/Micro Engineered and Molecular Systems, pp. 935–940, Jan. 2009.

[21] John C Hart, George K Francis,and Louis H Kauffman “Visualizing Quaternion Rotation,” ACM Transactions on Graphics, vol. 13, no. 3, pp. 256–276,July 1994.

[22] Jack B.Kuipers, “Quaternions and Rotation Sequences,” Geometry,Intergrability and Quantization, Coral Press,pp. 127–143, 2000.

[23] Marko Gargenta, and Masumi Nakamura, “Learning Android,”O’reilly , Second Edition, Jan 2014.

[24] Android , http://developer.android.com.