# An Accelerometer-Based Approach to Evaluate 3D Unistroke Gestures

Tahir Madani[1,2], Muhammad Tahir[3], Sheikh Ziauddin[1], Syed Raza[1], Mirza Ahmed[1], Majid Khan[1], and Mahmood Ashraf[4,5]
[1]Department of Computer Science, COMSATS Institute of Information Technology, Pakistan
[2]Department of Information Technology, University Technology Petronas, Malaysian
[3]Faculty of Computing and Information Technology, King Abdulaziz University, KSA
[4]Department of Computer Science, Szabist University, Pakistan
[5]Department of Software Engineering, University Teknologi, Malaysia

**Abstract:** *This paper, presents an evaluation of Three Dimensional (3D) unistroke human arm gestures. Our scheme employs an accelerometer-based approach by using Nintendo™ Wiimote as a gesture device. The system takes acceleration signals from Wiimote in order to, classify different gestures. It deals with numeric gestures, i.e., digits from 0 to 9 and simple mathematical operator gestures for addition, subtraction, multiplication and division. Two techniques, Dynamic Time Warping (DTW) and 2D trajectories are used to recognize and classify gestures. Successful recognition rates indicate that performing 3D gestures using accelerometer-based devices is intuitive and provides an effective means of interaction with computers.*

**Keywords:** *Human computer interaction, 3D gestures, accelerometer, 3D calculator, DTW, 2D trajectories, wiimote.*

*Received May 29, 2012; accepted September 26, 2013; published online August 17, 2014*

## 1. Introduction

In a computing environment, different input devices such as keyboard, mouse, numeric keypad, pen, touchpad, graphic tablets, etc., have been designed to draw gestures and perform interaction. These devices make human computer interaction easy and comfortable. But, humans in their daily life normally use gestures (motion of hands or body parts) to interact with different physical objects, to describe certain concepts or even to represent emotions. These human gestures provide a rich form of input to our physical Three Dimensional (3D) environment.

Gestures in computing environment can roughly be classified into two broad categories: Unistroke and multistroke gestures. Unistroke gestures are performed as a single line drawn without releasing mouse (or other input device such as pen) from the surface.

An example of unistroke gesture could be: Press the left mouse button at any position on the desktop, keep it pressing and move the mouse to another location and then release the mouse button as shown in Figure 1-a.

On the other hand, multistroke gestures are roughly the multiple unistroke gestures which require lifting of mouse in between. For example, drawing ‘A’ requires two strokes, first to draw “∧” and then “_” as shown in Figure 1-b.


a) Unistroke (numeric zero) gesture. b) Multistroke (alphabet A) gesture.

Figure 1. Example of unistroke (numeric zero) and multistroke (alphabet A) gestures. 1 and 2 indicates first and second stroke respectively.

Gesture recognition is the mathematical analysis or interpretation of human body motion or of a physical device (such as wiimote: A controller used in Nintendo™ game console, Figure 3). These physical devices generally use motion sensors to detect their movements along multiple dimensions. For example, Wiimote motion sensors determine its motion, its directional acceleration, its position in 3D, its tilt and its orientation (i.e., roll, pitch and yaw).

Multiple gesture recognition techniques [1, 2, 3, 6, 15] used numerous devices and applications. For example, certain techniques used data gloves [1] stereo rather than live gestures [6], camera-based bare hand recognition [2] and personalized accelerometers [15]. These techniques generally require wearing specific devices on hands and on other body parts like head. Moreover, involving camera and other colored gloves requires sophisticated detection algorithms and high computing power thus, resulting in costly solutions.

Our work, on the other hand, provides an accelerometer-based approach by using wiimote.

Wiimote device does not require line-of-sight for its interaction with the concerned system. Moreover, due to its motion sensors, it provides better motion detection and accurate position sensing. The benefits of our technique over existing techniques are its availability, mobility and simplicity. Furthermore, our work addresses the issue of human hand gestures in 3D space (in air). Our 3D Unistroke Gesture Identification (UGI) application (discussed in section 3) is capable of recognizing numeric gestures from 0 to 9 and basic mathematical operators such as addition, subtraction, multiplication and division.

Rest of the paper is organized into the following sections. Section 2 describes the related work while section 3 illustrates our UGI application. The experiments for the UGI application are presented in section 4. Section 5 describes the experimental results while section 6 describes the future work.

## 2. Related Work

A 3-axis accelerometer is used in a data glove for recognizing human hand gestures [7]. It used 3D digital model for the motion tracking and the recognition of human hand. Researchers [8, 19] tested gesture recognition by using 3D accelerometer (Wii Controller device). Hidden Morkov Model (HMM) [17] quantifier has been applied for the training and the recognition of gestures. Their overall recognition results are around 90%.

Another technique [10] employed a customized wearable input device that uses visual (camera) sensors, 2-axis accelerometer and pressure sensors. They used Dynamic Time Warping (DTW) and HMM techniques to classify gestures. The experiment involved three groups of gestures: Planar, curved and twisted. The combined visual and body sensors provided 81% to 91% recognition rate for human hand gestures.

An interesting gesture recognition algorithm was proposed by Liu *et al*. [14] for a virtual environment. This technique obtained accelerometer values from a Wiimote and used HMM for the training and recognition of gestures. The results indicate that the recognition can be achieved within the range of 90% to 95%.

The $3 recognizer [9] is based on gesture template matching in which real time coordinates in x, y and z plane are matched with the predefined coordinates of template gestures. The $3 recognizer doesn't require any toolkit or framework for gesture recognition, but only relies on geometric and trigonometric calculations of the gesture being performed.

Leong *et al*. [12] presented a gesture recognition technique using DTW and Euclidean distance approach. This method is comparatively better than that of HMM and provides a better accuracy up to 97% but, they did not take into account the orientation property.

In this paper, we use 2D trajectories and DTW [16, 18] techniques for gesture recognition. DTW has been proved less complex [13] and provides training and recognition procedures simpler and faster than HMM [5].

## 3. 3D-Unistroke Gesture Identification

### 3.1. Gestures

Multistroke gestures involve complex movements and require more precision at user's part especially if the user is drawing a gesture in 3D (Figure 1-b). On the other hand, unistroke gestures are comparatively easy to perform because they involve a single continuous movement (Figure 1-a). Due to the simplicity in performing unistroke gestures, we decided to test them for numeric recognition and for simple mathematical calculations.

We have defined numeric gestures from 0 to 9 and gestures for basic mathematical operations i.e., addition subtraction, multiplication, division as shown in Figure 2. We used 'p', 's', 'm' and 'd' gestures for plus, subtraction, multiplication and division because their structure supports the unistroke property.

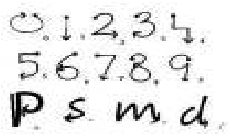


Figure 2. Numeric and mathematical operator gestures (the circle and arrow-head represent the start and the end of the gesture respectively).

### 3.2. Wiimote Acceleration

Our gesture recognition application recognizes simple human arm motions by using Wiimote device. Wiimote Figure 3 is a wireless handheld device that contains a built in 3-axis accelerometer for sensing arm motion [11]. It generates 3D acceleration (x, y, and z) data when performs translational and/or rotational movements. The Wiimote is equipped with multiple buttons, however we have used only two of them (button 'A' on the top and button 'B' down near power supply). The Wiimote establishes connection and exchanges information with UGI application via bluetooth.

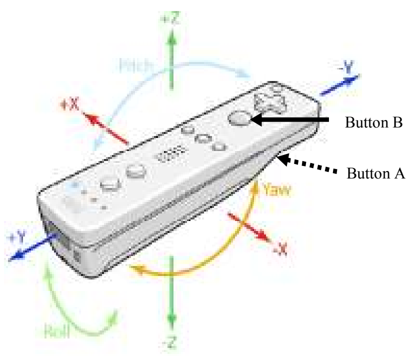
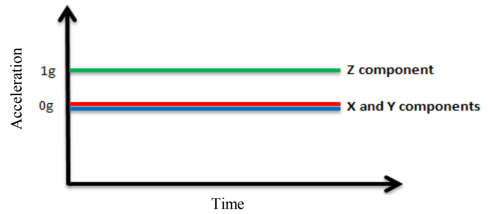


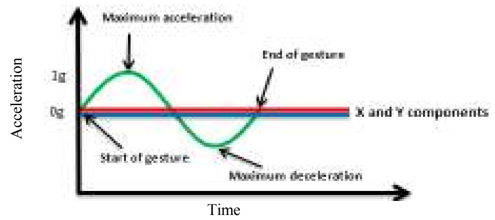

Figure 3. Illustration of acceleration of Wiimote and its three axis, adapted from [7] and modified (solid arrow represents button A while dotted arrow represents the location of button B).

### 3.3. Visualization of 3D Acceleration Data

When a Wiimote gesture is performed, we get accelerometer data and measure 3D inertial information acting on x, y and z axes [16]. This data is further used for the recognition of Wiimote motion (i.e., arm gesture). When the Wiimote is stable i.e., in rest state, the accelerometer values are 0g, 0g, 1g along x, y and z axes respectively where 'g' is the gravitational acceleration (9.8 $m/s^2$). When z component is 1g, it means equal and opposite forces are required to hold the wiimote. Figure 4 represents the movement of the arm (actually wiimote) as accelerometer signals in stable position and in upward direction.




a) Accelerometer signals when Wiimote is in stable position.




b) Accelerometer signals as sine wave when Wiimote is moved upwards (z-component is active).

Figure 4. Specific Wiimote accelerometer signals.

### 3.4. UGI Modes

When a user performs specific gesture, the UGI application receives the acceleration information from the device and matches it with pre-defined gestures. This is described below in two different modes: The training mode and the gesture recognition mode.

- The purpose of training mode is to record the gestures before performing the actual tests. In the training mode, user presses input button 'B' of Wiimote at the start of the gesture and releases it when the gesture is completed. This gesture gets stored in the gesture template library.
- The second mode is the gesture recognition mode in which user performs the specific gesture by pressing the input button 'A' of the Wiimote at the start of the gesture and releases it when the gesture ends.

Using the buttons 'A' and 'B' are intuitive to perform as a user can easily press 'A' button with his thumb and 'B' button as a trigger using index finger. Pressing the buttons 'A' and 'B' are analogous to pressing the mouse buttons and thus are easy to perform.

### 3.5. Gesture Recognition

The UGI recognizes a gesture by comparing the newly obtained accelerometer values with the gesture values already stored in the gesture template library. The gesture is recognized and then finally displays the best matched result on the output screen a Graphical User Interface (GUI), as shown in Figure 6. Gesture recognition phase is composed of further two components: Data analyzer and motion recognizer.

The data analyzer is responsible for detecting the start and the end of the gesture and passing the raw accelerometer values to the low pass filter. After passing through low pass filter, we obtain almost noise free accelerometer readings which are then transferred to the motion recognition component.

The motion recognition component uses two classifiers: 2D trajectories and DTW to recognize gestures. The 2D trajectories [16] technique is used for visualizing the accelerometer signals (arm movements). It selects 2D space as xy, yz or xz values based on the maximum displacement of the acceleration signals. For example, when a user makes a gesture of digit 3 (in air with wiimote) then xz axes get selected because change in the acceleration signals is maximum in x and z planes compared to xy and yz as shown in Figure 5.

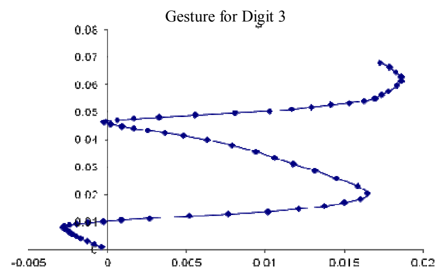



Figure 5. Identification of gesture for digit '3' using 2D trajectories.

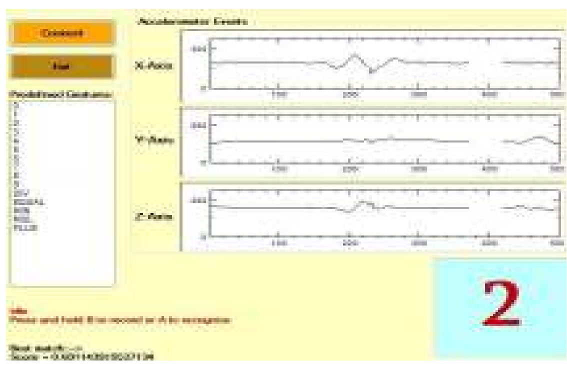



Figure 6. The x, y and z-axes representing accelerometer values when Wiimote moves in 3D.

The second classifier used in the motion recognition component is DTW [12, 13, 17]. It is used to compare the similarity between the gestures. It calculates the minimal distance between different points of the currently performed gesture with the points of each gesture already stored in template. This distance is calculated as the Euclidean distance. Gesture having least Euclidean distance is selected from the predefined gestures and is used by the UGI application.

### 3.6. Graphical User Interface

The "Connect" button GUI (Figure 6 top-left), establishes a connection between the Wiimote and UGI application. The gesture being performed is displayed on the GUI (at the bottom-right corner). To make sure that the Wiimote is always connected with our system and to check whether the Wiimote is idle or in active state, we have also implemented a graph that plots the accelerometer values in real time. The idle state is represented by straight lines for x, y and z-axes and the active state is represented by the corresponding curves of x, y and z-axes. Time is plotted on the horizontal axis while acceleration is plotted on the vertical axis.

## 4. Experimental Evaluation

Our experimental setup contains a UGI application running on a windows based platform and a Wiimote connected to UGI via bluetooth. Prior to the actual experiments, we trained participants with UGI application. It means, all the participants were briefed about how to perform different gestures by using button A and button B of the wiimote.

Total of 15 users (8 male, 7 female: Average age 22 years) participated in both the experiments. All of them were from different educational backgrounds where some users didn't have any previous experience with computers. Each user performed each experiment four times (for digit recognition).

Two experiments were conducted. In the first experiment each individual was asked to perform all numeric gestures (from 0 to 9) in random order.

In the second experiment, participants were told to perform simple mathematical calculations. A user can input any digit from 0 to 9 as the start of the gesture followed by one gesture of a mathematical operator and then the final gesture of any digit from 0 to 9. For example a user first performs the gesture of digit 3 and then the gesture for multiplication and then the gesture of digit 2. The result of the expression 3*2 is '6' which is shown in the Figure 7.

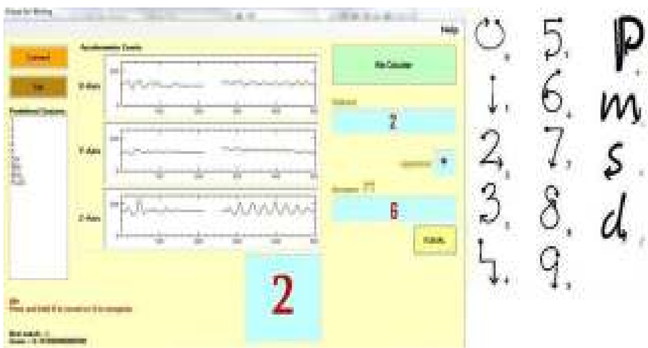


Figure 7. Wiimote calculator interface. UGI application shows the result of 3*2.

## 5. Results and Discussion

We trained and tested our system by obtaining data (gesture values) performed using wiimote. This allowed us to compare the recognition efficiency of different gestures. Bar graphs in Figures 8 and 9 represent recognition accuracy of all the gestures.

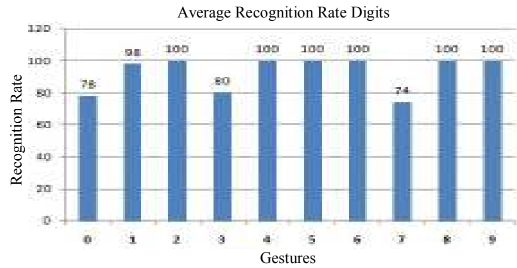



Figure 8. Average recognition rate for experiment 1 namely digits from 0 to 9.

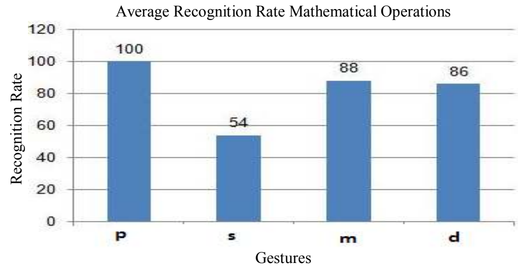



Figure 9. Average recognition rate for experiment 2 namely basic mathematical operators.

The bar graph in Figure 8 shows the average recognition rate is 93% for the digits from 0 to 9. Graph also, indicates that some gestures get recognized perfectly (100% recognition efficiency). While in Figure 9, the average recognition rate for the mathematical operators is 82%.

We have observed that some of the gestures like 6 and 0, 7 and 1, 5 and 's' were considered similar, sometimes, by UGI. This is most probably due to their similar orientation. It occurs because drawing a gesture in 3D lacks precision. It means the movement of a hand in 3D space is not perfect and gradually becomes unsteady due to the fatigue involved when Wiimote has been in hand for long time. This could result in drawing the gestures alike.

Based on the results above, we found that our method which uses the combination of DTW and 2D trajectories is practical and achieves encouraging results. The accuracy of our classifier can be increased further if we apply an orientation filter (by considering the 6DOF of the wiimote) to sort out the unsteady movements of hand.

The scaling of gestures is an important factor to consider. Our application clearly distinguishes and recognizes the same gestures drawn in small or large size. Furthermore, it has been observed that the unsteady movement of hands declines the recognition efficiency.

## 6. Conclusions and Future Work

In this paper, we have presented 3D UGI application that recognizes human arm motions using a Wiimote as an input device. We have utilized two accelerometer-based gesture recognition classifiers that are DTW and 2D trajectories. The classifiers recognize digits from 0 to 9. Furthermore, we developed a simple mathematical calculator that is capable of performing basic mathematical operations by recognizing the gestures performed in 3D space (i.e., taking Wiimote in hand and performing the gestures in air).

As a future work we would like to enhance the functionalities of our 3D mathematical calculator, for example, recognition of complex mathematical expressions, performing drawings and using Wiimote to manipulate Microsoft™ Powerpoint slides, etc. We would also like to focus on controlling the home appliances like TV, DVD player and some of the musical instruments by using customized accelerometer-based devices.

## References


[1] Amma C., Gehrig D., and Schultz T., "Airwriting Recognition using Wearable Motion Sensors," *in Proceedings of the 1st Augmented Human international Conference*, Megeve, France, pp. 1-8, 2010.

[2] Dhawale P., Masoodian M., and Rogers B., "Bare-hand 3D Gesture Input to Interactive Systems," *in Proceedings of the 6th ACM SIGCHI International Conference on Computer-Human Interaction*, New York, USA, pp. 25-32, 2006.

[3] El-Ghoul O. and Jemni M., "Multimedia Courses Generator for Deaf Children," *the International Arab Journal of Information Technology*, vol. 6, no. 4, pp. 424-430, 2009.

[4] Fitz-Walter Z., Jones S., and Tjondronegoro D., "Detecting Gesture Force Peaks for Intuitive Interaction," *in Proceedings of the 5th Australasian Conference on Interactive Entertainment*, Queensland, Australia, pp.1-8, 2008.

[5] Gerrit N., "The Optimization of Gesture Recognition Techniques for Resource-Constrained Devices" *Master Thesis*, University of Pretoria, South Africa, 2008.

[6] Keskin C., Erkan A., and A karun L., "Real Time Hand Tracking and 3D Gesture Recognition for Interactive Interfaces using HMM," availavle at: http://www.cs.nyu.edu/~naz/docs/icann.pdf, last visited 2003.

[7] Kim J., Thang N., and Kim T., "3-D Hand Motion Tracking and Gesture Recognition using a Data Glove," *in Proceedings of International Symposium on Industrial Electronics*, Seoul, Korea, pp.1013-1018, 2009.

[8] Kong J., Wang H., and Zhang G., "Gesture Recognition Model Based on 3D Accelerations," *in Proceedings of the 4th International Conference on Computer Science and Education*, Nanning, China, pp. 66-70, 2009.

[9] Kratz S. and Rohs M., "The $3 Recognizer: Simple 3D Gesture Recognition on Mobile Devices," *in Proceeding of the 14th International Conference on Intelligent User Interfaces*, New York, USA, pp. 419-420, 2010.

[10] Kwon D. and Gross M., "A Framework for 3D Spatial Gesture Design and Modeling using a Wearable Input Device," *in Proceeding of the 11th International Symposium on Wearable Computers*, Boston, USA, pp. 23-26, 2007.

[11] Lee J., "Hacking the Nintendo Wii Remote," *IEEE AI Magazine on Intelligent Systems*, vol. 7, no. 3, pp. 39-45, 2008.

[12] Leong T., Lai J., Panza J., Pong P., and Hong J., "Wii Want to Write: An Accelerometer Based Gesture Recognition System," *in Proceedings of International Conference on Recent and Emerging Advanced Technologies in Engineering*, Malaysia, pp. 1-31, 2009.

[13] Lévy C., Linarès G., and Nocera P., "Comparison of Several Acoustic Modeling Techniques and Decoding Algorithms for Embedded Speech Recognition Systems," *in Proceedings of IEEE Workshop on DSP in Mobile and Vehicular Systems*, Nagoya, Japan, pp. 1-4, 2003.

[14] Liu J., Pan Z., and Xiangcheng L., "An Accelerometer-Based Gesture Recognition Algorithm and its Application for 3D Interaction," *the Computer Science and Information Systems*, vol. 7, no. 1, pp. 177-188, 2010.

[15] Liu J., Wang Z., Zhong L., Wickramasuriya J., and Vasudevan V., "uWave: Accelerometer-based Personalized Gesture Recognition," *Technical Report TR0630-08*, Rice University and Motorola Labs, 2008.

[16] Quaedvlieg C., "Arm Gestures Applied to Remote Television Control: An Accelerometer Based Approach," *Master Thesis*, Radboud University, Nijmegen, Nederlands, 2007.

[17] Rabiner L., "A Tutorial on Hidden Markov Models and selected Applications in Speech Recognition," *IEEE Transactions on Speech Audio Processing*, vol. 77, no. 2, pp. 257-286, 1989.

[18] Sakoe H. and Chiba S., "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, pp. 43-49, 1978.

[19] Schlömer T., Poppinga B., Henze N., and Boll S., "Gesture Recognition with a Wii Controller," *in Proceedings of the 2nd International Conference on Tangible and Embedded interaction*, New York, USA, pp. 11-14, 2008.

**Tahir Madni** received BS computer science degree from COMSATS Institute of Information Technology, Islamabad and a MS degree in interactive system engineering from the Royal Institute of Technology, Stockholm, Sweden. Currently, he is pursuing his PhD degree from UTP Malaysia. His research interests include computer supported collaborative learning, human computer interaction, ubiquitous computing and multimodal interfaces.

**Muhammad Tahir** is currently an Assistant Professor in the Faculty of Computing and Information Technology, King Abdulaziz University, North Jeddah Branch, Kingdom of Saudi Arabia. He did his PhD degree in human computer interaction from Telecom ParisTech, France. His research interests include human computer interaction, tangible interaction, tactile interaction, augmented reality and the psychological aspects of human computer interaction.

**Sheikh Ziauddin** is working as an Assistant Professor in Departments of Computer Science at COMSATS Institute of Information Technology, Pakistan. He holds PhD degree from Asian Institute of Technology, Bangkok and completed a postdoctoral fellowship from I3S Lab, University of, France. His research interests include cryptography, network security, image processing, computer vision and pattern recognition.

**Syed Raza** is currently working as Project Engineer in UAE. He received his BS degree from COMSATS Institute of Information Technology and degrees in project management from Szabist University, Islamabad. His interests include interactive designs, mobile application development and programming.

**Mirza Ahmed** received his BS degree from COMSATS Institute of Information Technology, Pakistan. His interests include application development and programming and human computer interaction.

**Majid Khan** is working as an Assistant Professor in the Department of Computer Science at COMSATS Institute of Information Technology, Pakistan. He received his PhD degree from University of Vienna, Vienna, Austria. His research interests include wireless sensor systems, mobile ad-hoc networks and information security.

**Mahmood Ashraf** is working as an Assistant Professor in Department of Computer Science at Federal Urdu University of Arts, Science and Technology, Pakistan. He received his BS and MS degrees from International Islamic University, Pakistan and COMSATS Institute of Information Technology, Pakistan, respectively. Currently, he is a PhD scholar at University Teknology Malaysia, Malaysia. His research interests include human computer interaction, physicality and psychology.