

# Improving Accuracy and Practicality of Accelerometer-Based Hand Gesture Recognition

**David Mace**

Archbishop Mitty High School  
davidmace13@  
mittymonarch.com

**Wei Gao**

Boston University  
ECE Department  
davidgao@bu.edu

**Ayse K. Coskun**

Boston University  
ECE Department  
acoskun@bu.edu

## ABSTRACT

Wrist-watches are worn by a significant portion of the world's population, but their potential usefulness is not only limited to checking the time. Watches are located in a prime position to retrieve valuable position and acceleration data from a user's hand movements. In this paper, we explore the plausibility of using watches containing accelerometers to retrieve acceleration data from hand gesture motions for use in human-computer interaction tasks.

We compare two approaches for discerning gesture motions from accelerometer data: *naïve Bayesian classification with feature separability weighting* and *dynamic time warping*. We introduce our own gravity acceleration removal and gesture start identification techniques to improve the performance of these approaches. Algorithms based on these two approaches are introduced and achieve 97% and 95% accuracy, respectively. We also propose a novel planar adjustment algorithm to correctly recognize the same gestures drawn in different planes of motion and reduce spatial motion dissimilarities.

## Author Keywords

gesture recognition; accelerometer; watch gesture recognition; Bayesian classifier; feature separability weighting; dynamic time warping; plane adjustment

## ACM Classification Keywords

I.5.2[Pattern Recognition]: Design Methodology – Classifier design and evaluation

## INTRODUCTION

There is immense untapped potential for more natural human-computer interaction that lies within watches. Introducing computing power into watches by adding accelerometers and wireless transmission capabilities will

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IUI 2013 Workshop: Interacting with Smart Objects, March 18, 2013, Santa Monica, CA, USA

Copyright is held by the author/owner(s)

allow us to increase the diversity of the ways in which we use watches in our daily lives.

Gesture recognition is a growing area of interest because it provides a natural, expansive interface for humans to communicate with computers. The increased versatility and fluidity of hand gesture motions in comparison to key presses and finger swipes allows people to more seamlessly communicate with digital devices. Accelerometers implanted in wrist watches worn on users' hands can register unique acceleration signatures of motions that can be processed into simple motion types for use in various applications.

Using a watch with an accelerometer has lower complexity and cost compared to camera-based gesture recognition [1]. In addition, gesture recognition with accelerometers worn on the hands is simpler to set up than camera-based gesture recognition because a user does not need to face a particular direction or sit in front of a screen. For example, a user wearing a watch can control a stereo with a wave of the hand while sitting in a different room or scroll through a public display from a distant seat.

In this paper, we discuss two approaches, (1) Feature Weighted Naïve Bayesian Classifiers [3] and (2) Dynamic Time Warping [4], which require a smaller number of training samples but still provide high accuracy. We also introduce our own improvements to these algorithms that improve their usefulness in accelerometer-based gesture recognition.

Previous work has explored watch-based gesture recognition using dynamic time warping [9]. In this paper, we attempt to expand on previous research by testing the efficacy of rotationally normalizing gestures and applying feature weighted naïve Bayesian classification to gesture recognition.

## EQUIPMENT

Our implementation uses a TI eZ430-Chronos watch, which is cheap and simple to use, as the accelerometer data provider. The watch contains a VTI-CMA3000 3-axis accelerometer, with a measurement range of 2g, 8-bit resolution, and 100Hz sampling rate.

We use an ASUS TF300T Android tablet to run our algorithms (which are all implemented with Java); however, our implementation can be used with any Android device and can be ported to other mobile platforms. The tablet receives accelerometer data from the watch through an RF-receiver with USB interface, which is recognized as a serial port inside of Android.

Although we use a TI EZ430 Chronos Watch in our trials, any watch that can transmit data to a digital device could be used to achieve the same purpose.

## METHODS

The proposed gesture recognition methods can be split into three main phases. The *preprocessing* phase converts the acceleration measurements into a form that is more easily recognizable. The *plane adjustment* phase makes the acceleration data rotationally independent. The *gesture identification* stage uses either weighted feature classification or dynamic time warping to predict the most likely gesture given the acceleration measurements.

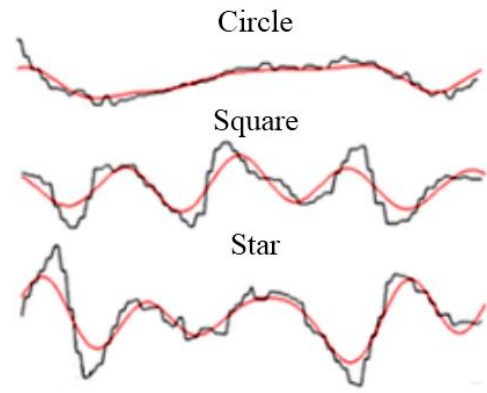
### Preprocessing:

The raw data set received from the accelerometer is noisy, contains still frames, and is skewed by gravity so the data must be adjusted before they can be properly classified.

The first step in the preprocessing phase is the removal of the acceleration caused by gravity from the watch's acceleration measures. Assuming the rotation of the watch is held reasonably constant throughout the gesture, the average of all of the acceleration measurements on each axis in practice approximately represents the constant value of gravity on that axis. To eliminate the effects of gravity, this average value is subtracted from each axis at each frame.

Still frames at the beginning and end of the data that are not part of the gesture are also removed. Still frames are detected by checking the average acceleration in each 0.5 second window. If the acceleration in a window is below a constant threshold, then that window is removed from the gesture.

The jolty nature of hand motions and the discrete sampling of the gestures contribute white noise to the data. A low-pass filter is used to extract the main gesture motion from the noisy accelerometer data. This common process is integral in recognizing gestures because it eliminates high-frequency noise while revealing underlying low-frequency patterns in the data. Figure 1 shows the difference between the acceleration data before and after the low pass filter is applied.



**Figure 1. Acceleration graphs of gesture trials. The black line is a graph of the acceleration magnitude from the watch vs time. The red line represents the acceleration graph after the low pass filter has been applied. These graphs only show the one dimensional x-axis acceleration.**

### Plane Adjustment:

One issue in gesture recognition that has not been explored in depth in prior work is recognizing gestures in different planes of motion as the same gesture. Sometimes when a user is told to trace a circle, he or she does so in the  $xy$  plane, but other times he or she might trace it in the  $yz$  plane, or in some plane in between.

Even if the user is trying to make all of the motions in a single plane, there are also usually slight discrepancies in the planes of motion among different gesture trials. To allow for more realistic and orientation-independent communication through the watch, a plane adjustment phase is included in our algorithm.

In this phase, first, the best-fit plane (shown in red in Figure 2) of the acceleration vectors is found. The rationale behind this is that if the motion lies in a single plane, then the acceleration vectors of a closed shape (e.g., a circle) should on average lie in that main plane. As there could be many motion vectors in the motion that do not lie in the main plane even after using a low-pass filter, all acceleration segments between points of inflection are added up to form one vector. In this way, we can identify the general direction of the user's motion, rather than identifying each individual motion segment.

If these gesture segments are represented as a set of vectors  $\{p_i = \langle x_i, y_i, z_i \rangle\}_{i=1}^n$  and the plane is represented by the equation  $z = Ax + By + C$  then the best fit plane is found by minimizing the error, which is

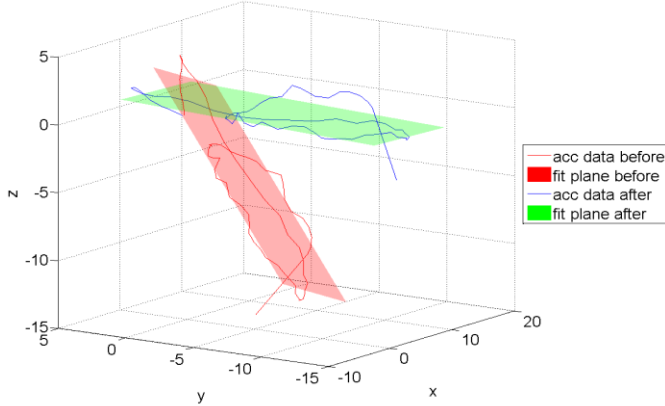
$$\sum_{i=1}^n (Ax_i + By_i + C - z_i)^2.$$

To find the best fit plane, the following matrix is solved using Gaussian Elimination [5].

$$\begin{bmatrix} \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m y_i & \sum_{i=1}^m 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m x_i z_i \\ \sum_{i=1}^m y_i z_i \\ \sum_{i=1}^m z_i \end{bmatrix}$$

After the best fit main plane is found, each vector is normalized relative to this plane (shown in Figure 2).

Comparison between Acceleration data before rotational normalization and after



**Figure 2.** The red curve represents a circle gesture performed in the yz plane and the blue curve represents the same gesture after its acceleration data has been reoriented relative to the xy plane.

The previous step takes into account rotation about the x and y axes, but does not account for rotation about the z axis. To fix this, the approximate best-fit line inside the best-fit plane is found. To approximate the best fit line, the lines extending at angles of  $\alpha = 22.5^\circ, 45^\circ, 67.5^\circ \dots 180^\circ$  from the origin are tested and the best fit line of these is chosen.

For the set of points  $\{p_i = \langle x_i, y_i, z_i \rangle\}_{i=1}^n$  the best fit  $\alpha$  value is obtained by minimizing  $\sum_{i=1}^n |\alpha - \tan^{-1}(y_i/x_i)|$ . This equation was chosen because it calculates the sum of differences between acceleration vector angles and the candidate best fit line. We want to find the line on which most acceleration vectors approximately fall, so using the differences between angles is logical.

Once  $\alpha$  is found, a new angle  $\beta_i = \tan^{-1}(y_i/x_i) - \alpha$  is calculated for each vector. A final vector  $u_i = \langle \sqrt{x_i^2 + y_i^2} * \cos \beta, \sqrt{x_i^2 + y_i^2} * \sin \beta, z_i \rangle$ , which is the original vector adjusted relative to the best fit line, replaces each original acceleration vector.

### Gesture Identification

We compared two approaches to identify gestures based on a user's acceleration data.

#### (a) Feature Weighted Naïve Bayesian Classification:

Naïve Bayesian Classification [3] is a promising technique in gesture recognition because it can make accurate predictions by using statistical measures to calculate

membership probabilities. In our implementation of this algorithm, twenty statistical features are extracted from the acceleration data. These include common statistical measures such as interquartile range, average energy, maximum of absolute value, and standard deviation.

Before a user operates the system, the user registers a set of training gestures. A weight between 0 and 1 is calculated for each feature type based on the similarity of feature measures of the different trained gestures of the same gesture type. A weight value close to 1 represents very precise measures and a value close to 0 represents imprecise measures.

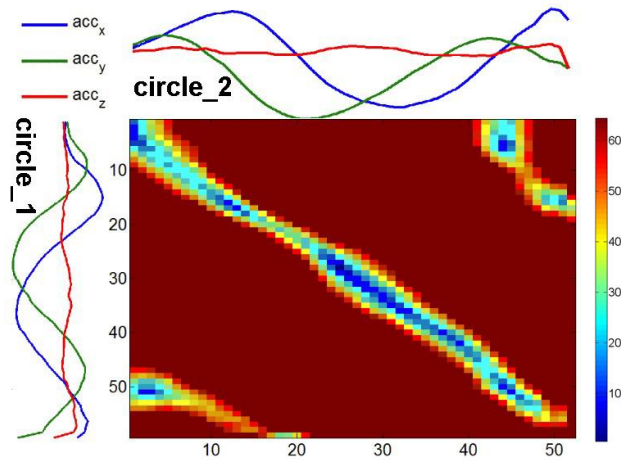
When the user is running the gesture recognition system, feature measures are extracted from the user's registered gesture. The proximity of each feature measure to the average trained feature measure of each gesture type is calculated by a normal distribution by the following equation:

$$proximity = e^{-(feature\ measure - trained\ avg)^2 / (trained\ \sigma)}$$

Then this proximity value is multiplied by the feature weight that was calculated in the training phase. All of these multiplied values are summed up and the system predicts the user's gesture to be the gesture type with the greatest calculated value.

#### (b) Dynamic Time Warping (DTW):

DTW is a widely used algorithm in gesture recognition that calculates the similarity between two time-series data sets. This algorithm is based on the idea that to find the time-independent similarity between a gesture and a template, the  $i^{th}$  point of the gesture can be aligned (warped) to the  $j^{th}$  point of template [4].



**Figure 3.** Each point in the grid represents the geometric distance between Circle\_1 at index y and Circle\_2 at index x. For example, to match up Circle\_1 at index 10 with Circle\_2 at index 3 requires a geometric distance of about 45.

Figure 3 provides a visual illustration of the process of DTW of two sets of data. In this algorithm, first a matrix **A** is calculated. Each element  $a(i,j)$  in the matrix represents

the geometrical distance between the sample data at time  $t(i)$  and template data (collected in training phase) at time  $t(j)$ . Any gesture that is “close” to the template data is likely to be of the same gesture type. Second, a path in the matrix  $A$  is found so that among all of the paths from  $a(0,0)$  to  $a(n,m)$ , the sum of all the elements on the path ( $P\_sum$ ) is minimized.

The above two steps give a value  $P\_sum$  representing the similarity between one sample data set and one template (training) data set. Then these steps are completed for all of the sample/template data pairs. The pair that has the smallest “path sum value” indicates the predicted gesture.

## RESULTS

### Naïve Bayesian and Dynamic Time Warping:

We tested both techniques using five gesture samples of four gesture types from five different people. The tested gesture types were circle, figure eight, square, and star. The average accuracy was 97% for the feature separability weighted Bayesian Classifier, and 95% for the dynamic time warping.

Both of the proposed methods have comparable accuracy with previously tested Hidden Markov Models and k-mean algorithms [6,7]. However, feature separability weighted naïve Bayesian classifiers and dynamic time warping run faster on large data sets and require a smaller number of training samples [2].

### Plane Adjustment:

When five training samples per gesture type are used, the average success of the feature separability weighted naïve Bayesian classification with plane adjustment is 83.75%, compared to 72.5% success without plane adjustment. When 10 training samples per gesture type are used in training, classification accuracy with plane adjustment improves to over 90%. Table 1a and 1b show the specific performance of plane adjustment for each gesture type when naïve Bayesian classification is used.

With Plane Adjustment					
		Predicted Class			
		Circle	Figure8	Square	Star
Actual Class	Circle	20	0	0	0
	Figure8	2	16	2	0
	Square	0	2	18	0
	Star	0	0	7	13

**Table 1a. Results when plane adjustment is used. Each gesture type on the top is how the algorithm classified the motion and each gesture type on the left is how the motion should have been classified.**

Without Plane Adjustment					
		Predicted Class			
		Circle	Figure8	Square	Star
Actual Class	Circle	20	0	0	0
	Figure8	0	14	2	4
	Square	2	0	12	6
	Star	0	6	2	12

**Table 1b. Results for the same gesture motions when plane adjustment is not used. Each gesture type on the top is how the algorithm classified the motion and each gesture type on the left is how the motion should have been classified.**

## APPLICATIONS

### Watch Gesture Recognition

The use of a common watch equipped with an accelerometer is sufficiently cheap and non-invasive to make it practical for real-world use in a variety of applications.

The most direct application of this technology is in more natural and flexible communication with digital devices such as tablets, televisions, and stereos. When sitting in front of a screen, a user could rotate a graphic on a presentation by moving his or her hand in a circular motion, automatically bookmark a webpage by making a star motion, or use a hand gesture as a shortcut to go to his or her emails. Of course, this form of interaction could not replace a keyboard and mouse for certain tasks, but it still opens the door for more diverse and seamless interaction with users.

This setup could also allow a user to remotely control devices when he or she is unable to or does not want to touch a device. A user could use the watch as a more natural universal remote to change the channel on the television, turn off a computer, or turn off the lights. Also in a public situation in which diseases can be spread by touch, users could interact with public displays like ATMs and airport kiosks through the watch instead of by touch.

Also there are many situations where people want to control a digital device but touch or keyboard control is impractical. When a user is cooking, wearing gloves, or driving, he or she may be unable to control a stereo, computer, or other device. Accelerometer-based gesture recognition through a watch is a feasible solution to this problem because a user could perform a hand gesture to control a device when they cannot access it directly.

Additionally there is tremendous potential for watch accelerometer based gesture recognition in immersive game technologies. This was recently evinced by the tremendous success of the Nintendo Wii. The Wii is primarily focused on recognizing short, linear motions and using a remote to track a cursor on the screen. On the other hand, our setup is

more concerned with recognizing gestures that can be added to in-game controls. These include using a circle to turn around and an up down motion to unlock a door.

We built an intelligent alarm clock Android application that uses the Chronos watch to detect if a user is asleep by checking for simple gestures [8]. We are also in the process of building Android applications that leverage the Chronos watch and gesture recognition in password detection and hand motion controlled features in media players.

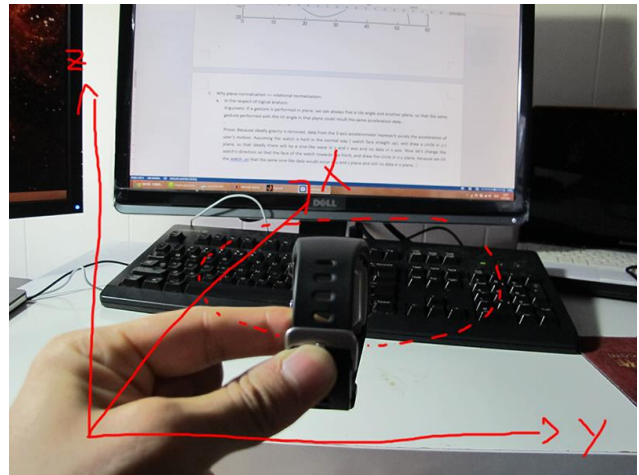
### Plane Adjustment

Normalizing the rotation of gestures can improve the accuracy and the flexibility of gesture recognition. An example of the usefulness of this technique is in public situations where devices communicate with different users. This form of user-independent communication is susceptible to different users using different rotations of the same gesture.

Interestingly, our plane adjustment algorithm improves gesture recognition not only in different planes (plane adjustment), but also when the watch is held in different orientations (rotation normalization). Figures 4 and 5 contain an example of the same gesture motion being performed with different watch orientations. Rotation normalization is useful because an accelerometer device is not always fixed in one direction each time the user holds it. Often a watch is fixed at an angle as long as it's worn on someone's wrist. Other accelerometer-containing devices that a user might hold instead of wearing, however, would not be fixed in one orientation, so the idea of rotation normalization could be extended to these devices.



**Figure 4.** A circle is drawn (the dotted line) in the yz plane when the watch is tilted up.



**Figure 5.** A circle is drawn in the xy plane when the watch is tilted to the side. This is the same motion as in Figure 4 because the watch is tilted at the same angle relative to the plane of motion.

### ACKNOWLEDGEMENT

Part of this research has been conducted during David Mace's internship in Boston University's Research Internships in Science and Engineering (RISE) program.

### REFERENCES

- [1] Patlolla, C.; Mahotra, S.; Kehtarnavaz, N.; , "Real-time hand-pair gesture recognition using a stereo webcam," *Emerging Signal Processing Applications (ESPA), 2012 IEEE International Conference on*, vol., no., pp.135-138, 12-14 Jan. 2012 doi: 10.1109/ESPA.2012.6152464
- [2] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," in *IEEE ASSP Magazine*, 1986, pp. 4-15.
- [3] P. Paalanen. "Bayesian Classification Using Gaussian Mixture Model and EM Estimation: Implementations and Comparisons." Lappeenranta University of Technology, Information Technology Project Report. <http://www2.it.lut.fi/project/gmmbayes/downloads/doc/report04.pdf>
- [4] E. Keogh, C. A. Ratanamahatana, "Exact Indexing of Dynamic Time Warping", In *Proceedings of International Conference on Very Large Data Bases (VLDB)*, pp. 406-417, 2002.
- [5] R. Sedgewick. "GaussianElimination.java." Web. 17 Jul. 2012. <http://introcs.cs.princeton.edu/java/95linear/GaussianElimination.java.html>
- [6] T. Schlomer, B. Poppinga, N. Henze and S. Boll, "Gesture Recognition with a Wii Controller", In *Proceedings of International Conference on Tangible and Embedded Interaction (TEI)*, pp. 11-14, 2008.
- [7] Klingmann, Marco. "Accelerometer-Based Gesture Recognition with the iPhone." <http://www.doc.gold.ac.uk/~mas02mb/Cognitive%20Computing/Dissertation/iPhone%20gestures.pdf>
- [8] Persistent alarm clock [http://processors.wiki.ti.com/index.php/Persistent\\_alarm\\_clock](http://processors.wiki.ti.com/index.php/Persistent_alarm_clock)
- [9] Jiayang Liu, Jehan Wickramasuriya, et al. "uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications." In *Pervasive Computing and Communications*, 2009.