ACCELEROMETER-BASED GESTURE RECOGNITION VIA DYNAMIC–TIME WARPING, AFFINITY PROPAGATION, & COMPRESSIVE SENSING

Ahmad Akl and Shahrokh Valaee

Department of Electrical and Computer Engineering, University of Toronto
Email: {ahmadakl, valaee}@comm.utoronto.ca

ABSTRACT

We propose a gesture recognition system based primarily on a single 3-axis accelerometer. The system employs dynamic time warping and affinity propagation algorithms for training and utilizes the sparse nature of the gesture sequence by implementing compressive sensing for gesture recognition. A dictionary of 18 gestures or classes is defined and a database of over 3,700 repetitions is created from 7 users. Our dictionary of gestures is the largest in published studies related to acceleration-based gesture recognition, to the best of our knowledge. The proposed system achieves almost perfect user-dependent recognition and a user-independent recognition accuracy that is competitive with the statistical methods that require significantly a large number of training samples and with the other accelerometer-based gesture recognition systems available in literature.

Index Terms— *gesture recognition, dynamic time warping, affinity propagation, compressive sensing*

1. INTRODUCTION

Gesture recognition has been one of the hottest fields of research for the past few decades since it serves as an intelligent and a natural interface between the human and the computer. The proliferation in technology and in microelectronics more specifically has inspired research in the field of accelerometer-based gesture recognition. 3-axis accelerometers are increasingly being incorporated and embedded into many personal electronic devices like the Apple iphone, Apple iPod touch, Wiimote, and Lenovo laptops to name a few [1][2].

The majority of the available literature on gesture or action recognition combines data from a 3-axis accelerometer with data from another sensing device like a biaxial gyroscope [3] or EMG sensors [4] in order to improve the system's performance and to increase the recognition accuracy. Accelerometer-based gesture recognition system using continuous Hidden Markov Models (HMMs) [5] has been developed. However, the computational complexity of statistical or generative models like HMMs is directly proportional to the number as well as the dimension of the feature vectors [5]. Therefore, one of the major challenges with HMMs is estimating the optimum number of states and thus determining the probability functions associated with the HMM. Besides, [5] assumes that variations in the gestures are Gaussian and this may not always be the optimum assumption.

The most recent gesture recognition system that is accelerometer-based is the uWave [6]. uWave supports personalized gesture recognition which means it is user-dependent. In [6], it is claimed that uWave requires only one single training sample for each gesture pattern which is stored in a template. The core of the uWave is dynamic time warping (DTW) and the system's database undergoes two types of adaptation: positive and negative adaptation. However, the way uWave's database gets adapted is basically continuous training and this is, definitely, unfavorable. Besides, removing the older template every other day might lead to replacing a very good representative of a gesture sequence which is again not efficient. Finally, being user-dependent greatly limits the applications of uWave since research on accelerometer-based gesture recognition is targeting a universal system that, given a dictionary of gestures, can recognize the different gestures with a competitive accuracy regardless of the user.

So, in this work, we propose an accelerometer-based gesture recognition system that uses only a 3-axis accelerometer to recognize gestures, where gestures here are hand movements. A dictionary of 18 gestures or classes is created for which a database of 3,780 runs or repetitions is built by collecting data from 7 participants. Some of the gestures defined in the dictionary are taken from a gesture vocabulary identified by Nokia [7]. Two tests are run: user-dependent and user-independent recognition. The core of recognizer's training phase is an integration of Affinity Propagation (AP) and DTW for both types of tests. For user-dependent, recognition involves comparing the unknown gesture repetition by DTW to the set of exemplars obtained during the training phase. On the other hand, for user-independent recognition, simple comparison by DTW doesn't suffice and here is where CS comes into picture. The system achieves an accuracy of 99.79% for user-dependent recognition with only three repetitions used for training and for a dictionary size of 18 gestures. As for user-independent recognition, the system achieves an accuracy of 96.89% for a dictionary size of 8 gestures which is very competitive with other statistical methods whereas uWave's accuracy for user-independent recognition is recorded to be 75.4%.

The rest of the paper is organized as follows: Section II presents the technical details of the proposed gesture recognition system. Section III describes an implementation of the gesture recognition system using a Wiimote. Section IV explains the simulations and discusses the results. Finally, section V concludes the paper.

2. GESTURE RECOGNITION SYSTEM

This section discusses the technical components of the proposed gesture recognition system: Temporal Compression, DTW, AP, and CS.

Hand gestures are well-known to suffer from inherent temporal variations. They differ from person to person and even the same person cannot replicate the same gesture exactly. This

entails that gesture sequences can be either compressed or stretched depending on the user and the speed of the hand movement. In other words, the recorded gesture sequences are of different lengths most of the time, if not always.

2.1. Temporal Compression:

The hand gestures defined are believed to have smooth acceleration waveforms since the hand follows a smooth trajectory while performing the gestures. However, acceleration data acquired suffers from abrupt changes due to hand shaking or accelerometer noise which needs to be eliminated.

The acceleration time series are temporally compressed by an averaging window of 70 ms and moves at a 30 ms step similar to [6]. Temporal compression, basically, filters out any variations not intrinsic to the gesture itself and further reduces the size of the acceleration signals which turns out to be crucial for the next step of DTW.

2.2. Dynamic Time Warping:

Dynamic time warping (DTW) is an algorithm that measures the similarity between two time sequences of different durations. In other words, DTW matches two time signals by computing a temporal transformation causing the signals to be aligned. The alignment is optimal in the sense that a cumulative distance measure between the aligned samples is minimized [8].

Assume we have two time sequences, X and Y, of length *n* and *m*, respectively, where $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_m\}$ and we want to compute the matching cost: DTW(X,Y). The matching cost is computed based on dynamic programming using the following formulation:

$$D(i,j) = d(x_i, y_j) + min\begin{Bmatrix} D(i,j-1) \\ D(i-1,j) \\ D(i-1,j-1) \end{Bmatrix} \quad (1)$$

where the distance function d(·,·) varies with the application. In our gesture recognition system, $d(x_i, y_j)$ is defined as,

$$d(x_i, y_j) = (x_i - y_j)^2 \quad (2)$$

For further details and thorough explanation of DTW, the reader is referred to [8].

2.3. Affinity Propagation:

Affinity propagation (AP) [9] is an algorithm which, unlike all other clustering techniques, simultaneously considers all data points as potential exemplars and recursively transmits real-valued messages until a good set of exemplars and clusters emerge.

AP operates primarily on a matrix **S** whose entities are real-valued similarities between data points, in which *S*(*i*,*j*) indicates the similarity between data points x_i and x_j. Clustering is based on the exchange of two type of messages: *responsibility* where *r(i,j)* indicates how well-suited point *j* is to serve as an exemplar for point *i*, and *availability* where *a(i,j)* indicates how appropriate it would be for point *i* to choose point *j* as its exemplar. In addition to **S**, AP takes as an input a real number *p* referred to a *preference* so that data points with larger values of *p* are more likely to be chosen as exemplars. Generally, *p* takes on the value of the median of the input similarities in case all data points are equally likely to be chosen as exemplars. As far as our gesture recognition system is concerned, the value of *p* is chosen to be *p* = 0.01*median of the input similarities in both cases: user-dependent and user-independent recognition. As for the similarity function, it is defined as the negative of the cost computed by DTW,

$$S(i,j) = -1 * \left(\left(DTW(i_x, j_x)\right)^2 + \left(DTW(i_y, j_y)\right)^2 + \left(DTW(i_z, i_z)\right)^2 \right) \forall i,j \in \{1, \dots, N\} \quad (3)$$

where *N* is the total number of repetitions in the training set.

2.4. Compressive Sensing:

The premise here is that hand gestures are believed to be sparse since the hand follows a smooth trajectory while performing a gesture and therefore, CS can be implemented to recognize a repetition of a gesture.

CS provides a novel framework for recovering sparse or compressible signals, with much fewer noisy measurements than that needed by nyquist rate [10][11]. If we denote the unknown gesture by $\mathcal{Y}$ and the repetitions to which the unknown gesture is to be compared to by the matrix **R** where each column represents a repetition. Assuming that $\mathcal{Y}$ is a replica of one of the repetitions in **R**, then $\mathcal{Y}$ can be related to **R** by,

$$\mathcal{Y} = \boldsymbol{R}\theta \quad (4)$$

and assuming that the number of repetitions in **R** is *P*, then θ is a 1-sparse *P* x 1 vector whose elements are all zeros except $\theta(n) = 1$, where n is the index of the repetition which the unknown gesture resembles, namely,

$$\theta = [0, \dots, 0, 1, 0, \dots, 0]^T \quad (5)$$

where the superscript *T* denotes transposition. Recall that the gesture repetitions are of different lengths and in order to solve (4), $\mathcal{Y}$, **R**, and θ have to be of compatible sizes. This problem is overcome by finding the maximum length among repetitions to construct **R** and the unknown repetition for each unknown repetition. The shorter repetitions are then padded with zeors until all repetitions are of the same length.

Recall also that gestures suffer from inherent temporal variations and therefore the above ideal example of having a unknown repetition replicating a template repetition doesn't exist in a real scenario. Yet, it should be of close resemblance. Therefore, the problem can be formulated as follows:

$$\mathcal{Y} = \boldsymbol{R}\theta + \varepsilon \quad (6)$$

where ε is the measurement noice.

CS requires two conditions for perfect signal recovery: (i) sparsity and (ii) incoherence. The first condition is satisfied by the way the vector θ is defined. As for the second condition, if **R** is the product of a pair of orthobases (Ψ, Φ), then it is obvious that Ψ and Φ are coherent. Therefore, using the same formulation as in [12], we introduce the preprocessor *W* (*i.e.* $Y = W\mathcal{Y}$), which is defined as,

$$W = QR^{\dagger} \quad (7)$$

where $Q = orth(R^T)^T$, and *orth*(**R**) is an orthogonal basis for the range of **R** and $R^{\dagger}$is the pseudo-inverse of the matrix **R**. By this transformation, the two conditions of CS are met and therefore, θ can be well recovered from *Y* with a high probability through the following ℓ_1 minimization problem,

$$\hat{\theta} = \arg\min ||\theta||_1, \; s.t. \; Y = Q\,\theta + \varepsilon' \qquad (8)$$

where $\varepsilon' = W\varepsilon$.

Recall again that the acceleration waveforms constitute of three signals: in the x-, y-, and z- directions. Therefore, when applying the above formulation to the gesture recognition problem, $\mathcal{Y}$ will be in fact three vectors, **R** will be three matrices and θ will be three vectors. Since θ is of three vectors, then

$$\theta_{eq} = \theta_x + \theta_y + \theta_z \qquad (9)$$

For user-independent recognition, θs that belong to the same user for the same class are added together as well, and then the the class with the highest θ is recognized as the correct class.

3. PROTOTYPE IMPLEMENTATION

The acceleration data corresponding to the different gestures is collected using a Wiimote, which has a built-in 3-axis accelerometer. A gesture repetition starts by pressing and holding the "trigger" button or "B" button on the bottom of the remote and it ends by releasing the button and thus the problem of gesture spotting is solved.

3.1. Gesture Vocabulary:

A dictionary of 18 gestures is created as is shown in figure 1. Our dictionary of gestures is the largest in published studies for accelerometer-based gesture recognition systems. The defined gestures are not limited only to one plane as is the case in [6][7], but span the two planes: XZ and YZ planes. Each gesture is referred to as a *class* and each run of a gesture is referred to as a *repetition*.

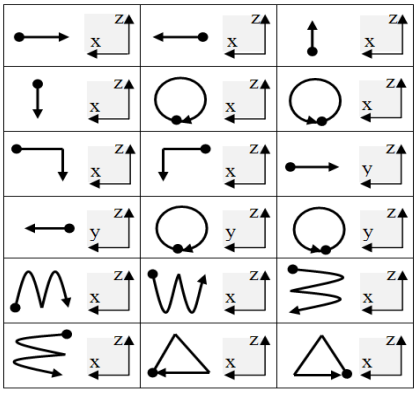

Figure 1. The dictionary of gestures created with 18 classes.

3.2. Gesture Database collection:

The database consists of 3,780 repetitions and is built by acquiring gesture sequences from 7 participants (2 females and 5 males) using the Wiimote. Each participant is asked to repeat each class 30 times resulting in a total of 540 repetitions for all classes per participant or a total of 210 repetitions from all participants per class.

All participants are asked to keep the remote straight because the remote and, in turn, the accelerometer can be tilted around any of the three axes. The acceleration waveforms from the same person corresponding to a class can differ drastically if the accelerometer happens to be tilted differently each time. We have already started looking into the issue of tilting however it is not yet taken into account in the current version of our gesture recognition system.

4. SIMULATION & RESULTS

4.1. Training Phase:

The training phase is the first main phase in our proposed gesture recognition system and is depicted in figure 2.

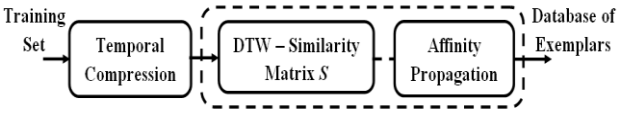

Figure 2. Training Phase.

User Dependent Training: The training set is created by randomly choosing *M* repetitions from each class. All the chosen repetitions are then temporally compressed in order to remove any noise that might have been added by inevitable hand shaking or remote tilting. DTW is then implemented to find the similarity or the cost between each two repetitions and thus forming the similarity matrix **S**. Next, AP works on **S** and partitions the training set into different clusters each represented by an exemplar. AP in the case of user-dependent recognition is forced to partition the data into *N* clusters where *N* is the number of classes in the dictionary. In other words, all repetitions pertaining to one class form one cluster. As a result, the output of the training phase is a set of *N* exemplars, one for each class.

User Independent Training: The training set is created by randomly choosing *M* repetitions from each class from *K* users only, resulting in a total of *KM* repetitions from each class. Again, all repetitions are temporally compressed for noise removal and DTW is implemented in order to produce the similarity matrix. Next, AP is run in order to partition the repetitions into different clusters. However, in this case, although AP is forced to create a cluster for each class, it doesn't succeed and consequently, repetitions from one class can fall into different clusters. Yet, repetitions from the same user for a class always stick together in one cluster. In other words, a cluster doesn't necessarily contain repetitions from the same class anymore but may contain repetitions from other classes and other users too. As a result, exemplars of all resulting clusters are stored and thus the output of the training phase is an arbitrary number of exemplars which are not distinctive to each class.

4.2. Testing Phase:

The testing phase is the second main phase in our proposed gesture recognition system and is depicted in figure 3.

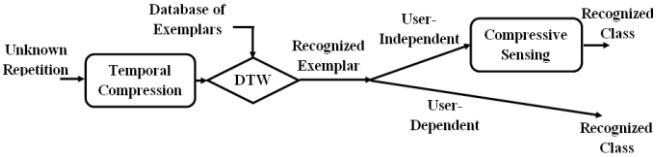

Figure 3. Testing Phase

User-Dependent Recognition: The testing set is formed by putting together all the repetitions that were not used in the training stage. Testing is done on each user separately and the average accuracy among all users is computed. So, an unknown repetition undergoes the same preprocessing of temporal compression and then compared by DTW to the N exemplars that were found in the training phase. The unknown gesture gets recognized as the class whose exemplar yields the lowest cost. For the seven participants, M is varied in order to examine the relationship between the number of training repetitions per class and the average accuracy of the system. Figure 4 shows a graph of the average recognition rate against the number of training repetitions per class. The graph shows that with as minimum as three repetitions, the recognizer was capable of achieving an accuracy of 99.79%.

User-Independent Recognition: The testing set is generated by putting together all the repetitions that were not used in the training stage specific to the K chosen users and adding to them all the repetitions from the four other users. So, an unknown repetition is first temporally compressed and then compared by DTW to the exemplars that were found in the training stage. All the repetitions that fall in the cluster whose exemplar gives the lowest cost are recovered and used to construct the matrix **R** in (6) and then the unknown gesture is recognized as explained in section 2.4. Figure 5 shows a graph of the average recognition rate against the number of classes for user-independent gesture recognition. Simulations were run with $K = 3$. Plots show the accuracy once based on the remaining repetitions belonging to the users whose data is used to train the system, and once based on all the repetitions from the other users, and the middle plot shows the average accuracy based on repetitions from both users.

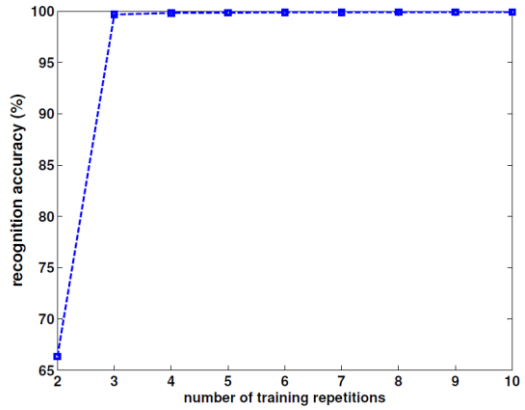

Figure 4. Average accuracy against the number of training repetitions for user-dependent recognition.

5. CONCLUSION

In conclusion, we have proposed a new gesture recognition system. The system utilizes a single 3-axis accelerometer and thus can be readily implemented on any commercially available consumer device that has a built-in accelerometer. The system employs dynamic time warping and affinity propagation algorithms for efficient training of the system and utilizes the sparse nature of the gesture sequence by implementing compressive sensing for user-independent gesture recognition. The system is tested on a dictionary of 18 classes whose database contains over 3,700 repetitions collected from 7 users. For some users, the proposed system achieves an accuracy of 100% when carrying out user-dependent recognition and a user-independent recognition accuracy that is competitive with many systems that are available in literature.

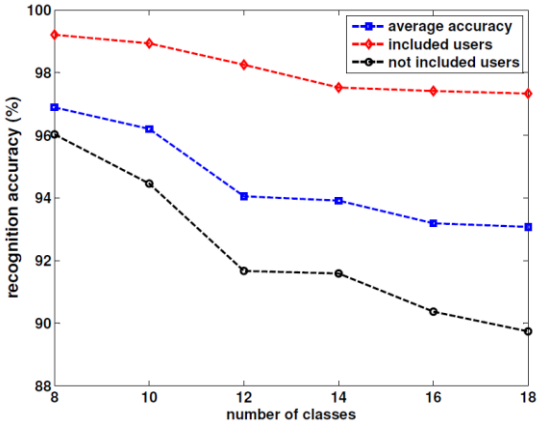

Figure 5. Average accuracy against the number of classes for user-independent recognition.

6. REFERENCES

[1] http://en.wikipedia.org/wiki/Accelerometer

[3] A. Y. Yang, S. Iyengar, S. Sastry, R. Bajcsy, P. Kuryloski, and R. Jafari, "Distributed segmentation and classification of human actions using a wearable motion sensor network," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop*, pp. 1-8, June 2008.

[4] Z. Xu, C. Xiang, W. Wen-hui, Y. Ji-hai, V. Lantz, W. Kong-qiao, "Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors," *International Conference on Intelligent User Interfaces*, pp. 401 - 406, February 2009.

[5] T. Pylyanainen, "Accelerometer Based Gesture Recognition Using Continuous HMMs," International Conference on Pattern Recognition and Image Analysis, pp. 639–646, 2005.

[6] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," in *IEEE PerCom*, 2009.

[7] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca, "Accelerometer-based gesture control for a design environment," *Personal Ubiquitous Computing*, vol. 10, pp. 285-299, 2006.

[8] E. Keogh, C. A. Ratanamahatana, "Exact Indexing of Dynamic Time Warping," Knowledge and Information Systems, Hong Kong, China, pp. 406-417, August 2002.

[9] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 1, pp. 972–976, February 2007.

[10] J. C. Emmanuel and B. W. Michael, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, pp. 21–30, March 2008.

[11] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Processing Magazine*, pp. 14–20, March 2008.

[12] C. Feng, W. S. A. Au, S. Valaee, Z. Tan, "Orientation-Aware Indoor Localization using Affinity Propagation and Compressive Sensing," *Submitted for publication*.