A Novel Accelerometer-Based Gesture Recognition System

Ahmad Akl, *Student Member, IEEE*, Chen Feng, *Student Member, IEEE*, and Shahrokh Valaee, *Senior Member, IEEE*

***Abstract*—In this paper, we address the problem of gesture recognition using the theory of random projection (RP) and by formulating the whole recognition problem as an ℓ_1-minimization problem. The gesture recognition system operates primarily on data from a single 3-axis accelerometer and comprises two main stages: a training stage and a testing stage. For training, the system employs dynamic time warping as well as affinity propagation to create exemplars for each gesture while for testing, the system projects all candidate traces and also the unknown trace onto the same lower dimensional subspace for recognition. A dictionary of 18 gestures is defined and a database of over 3700 traces is created from seven subjects on which the system is tested and evaluated. To the best of our knowledge, our dictionary of gestures is the largest in published studies related to acceleration-based gesture recognition. The system achieves almost perfect user-dependent recognition, and mixed-user and user-independent recognition accuracies that are highly competitive with systems based on statistical methods and with the other accelerometer-based gesture recognition systems available in the literature.**

***Index Terms*—Affinity propagation, compressive sensing, dynamic time warping, gesture recognition, random projection (RP).**

I. Introduction

GESTURE recognition refers to the process of understanding and classifying meaningful movements by a human's hands, arms, face, and sometimes head. It has become one of the hottest fields of research since it is of great significance in designing artificially intelligent human-computer interfaces for various applications which range from sign language through medical rehabilitation to virtual reality. The proliferation in technology, and in microelectronics more specifically, has inspired research in the field of accelerometer-based gesture recognition. Three-axis accelerometers are being increasingly embedded into many personal electronic devices like the Apple iPhone, Apple iPod touch, Apple iPad, wiimote, and Lenovo laptops, to name a few.

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 [1] or EMG sensors [2] 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) [3] 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 [3]. Therefore, one of the major challenges with HMMs is estimating the optimal number of states and thus determining the probability functions associated with the HMM. Besides, variations in gestures are not necessarily Gaussian and perhaps, other formulations may turn out to be a better fit.

The most recent gesture recognition system that is solely accelerometer-based is the uWave [4]. uWave is a user-dependent system that supports personalized gesture recognition. uWave functions by utilizing only one training sample, stored in a template, for each gesture pattern. 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, uWave's database adaptation resembles continuous training and in some cases, if thorough examination of templates is ignored, removing an older template every other day might lead to replacing a very good representative of a gesture sequence, which is best avoided. Although uWave demonstrates computational as well as recognition efficiency, being user-dependent limits the applications of uWave. Besides, judging from systems like Nintendo Wii, Apple iPhone, and other devices, researchers on accelerometer-based gesture recognition are envisaging a universal system that, given a dictionary of gestures, can recognize different gesture traces with a competitive accuracy and with the minimal dependence on the user.

In this paper, we propose an accelerometer-based gesture recognition system that uses only a single 3-axis accelerometer to recognize gestures, where gestures here are hand movements. The work of this paper is built upon a preliminary version of our gesture recognition system [5]. A dictionary of 18 gestures is created for which a database of 3780 traces is built by collecting data from 7 participants. Some of the gestures defined in the dictionary are taken from the gesture vocabulary identified by Nokia [6]. The core of the recognizer's training stage is an amalgamation of Dynamic Time Warping (DTW) and Affinity Propagation (AP). For recognition, as this paper shows, one-nearest neighbor DTW does not always suffice and therefore, the recognition problem is formulated as an ℓ_1-minimization problem after projecting all candidate gesture traces onto the same lower dimensional subspace. The system achieves an accuracy of 98.71% for mixed-user recognition for a dictionary of 18 gestures compared to 98.6% recorded accuracy by uWave for user-dependent recognition for a dictionary

Manuscript received September 28, 2010; revised February 02, 2011 and May 28, 2011; accepted July 31, 2011. Date of publication August 22, 2011; date of current version November 16, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Piotr Indyk.

The authors are with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4 Canada (e-mail: ahmad.akl@utoronto.ca; chenfeng@comm.utoronto.ca; valaee@comm.utoronto.ca).

Digital Object Identifier 10.1109/TSP.2011.2165707

1053-587X/$26.00 © 2011 IEEE

of 8 gestures. As for user-independent recognition, the system achieves accuracies of 96.84% and 94.6% for dictionaries of 8 gestures and 18 gestures, respectively, which is very competitive with other statistical models or other available techniques.

The rest of the paper is organized as follows: Section II sets up the problem and introduces a general overview of the proposed gesture recognition system. Section III describes the clustering algorithm employed to train the system. Section IV describes the recognition process and how random projection (RP) is utilized to classify an unknown gesture trace. Section V describes an implementation of the gesture recognition system and evaluates the system's performance through simulations. Finally, Section VI concludes the paper.

II. Problem Setup

Suppose that a system consists of N hand gestures and for each gesture M traces are stored in a database. Gesture complexity ranges from very simple ones, as simple as the hand moving either to the right or to the left or up or down, to more complex ones such as gestures representing letters or numbers. The acceleration of the hand is used as the data to represent a gesture rather than the hand position. The acceleration of the hand is measured at different times t using a single 3-axis accelerometer. Therefore, a trace of a gesture is basically a three column matrix representing the acceleration of the hand in the x-, y-, and z-directions. However, hand gestures inherently suffer from temporal variations. In other words, they differ from one person to another and even the same person cannot perfectly replicate the same gesture. This entails that gesture traces can be either compressed or stretched depending on the user and the speed of the hand movement. Consequently, traces of the same gesture are of different lengths which poses the first major challenge in developing a gesture recognition system.

Mathematically, the gesture recognition problem can be formulated as follows: The system consists of N gestures, each having M traces, tabulated as the following sets:

$$\begin{aligned} \mathcal{G}_1 &= \{\mathbf{G}_{1,1}, \mathbf{G}_{1,2}, \ldots, \mathbf{G}_{1,M}\}, \\ \mathcal{G}_2 &= \{\mathbf{G}_{2,1}, \mathbf{G}_{2,2}, \ldots, \mathbf{G}_{2,M}\} \\ &\vdots \\ \mathcal{G}_N &= \{\mathbf{G}_{N,1}, \mathbf{G}_{N,2}, \ldots, \mathbf{G}_{N,M}\}. \end{aligned} \tag{1}$$

Each $\mathbf{G}_{i,j}$ is a $l_{i,j} \times 3$ matrix, where each column represents the acceleration in the x-, y-, or z-direction. Notice that $l_{i,j}$ is different even for traces of the same gesture $\mathcal{G}_i$ since traces of the same gesture can have different durations and thus different number of rows.

Fig. 1 shows the acceleration waveforms for moving the hand in a clockwise circle. The accelerations have been acquired using a Wii Remote (wiimote for short) [7], [8], which has a built-in 3-axis linear accelerometer.

The gesture database (1) is produced in an off-line procedure and stored for later use, which constitutes the training stage. In a test stage, a user moves his/her accelerometer-equipped device, such as a smart phone or a wiimote, to signal a particular gesture from the above database in (1). The accelerometer readings are formed into an $l_y \times 3$ matrix $\mathbf{Y}$. Again, note that l_y may not be equal to any $l_{i,j}$. The objective of a gesture recognition system is to find out which gesture is intended by the user.

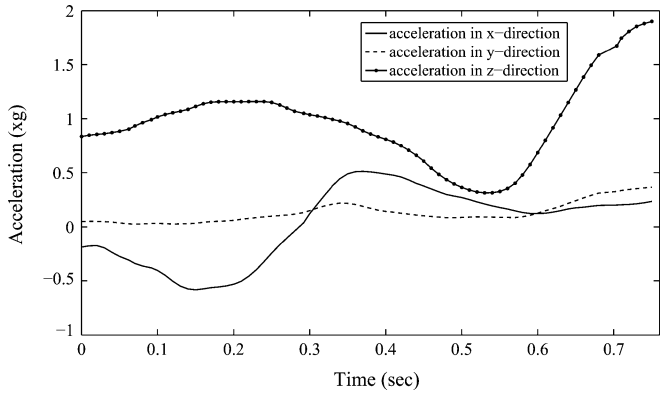

Fig. 1. Acceleration waveforms defining the gesture of moving the hand in a clockwise circle.

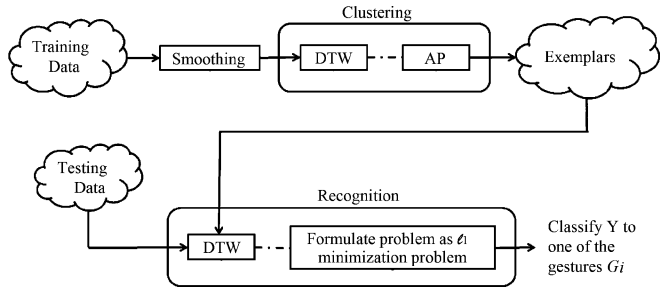

Fig. 2. General overview of the gesture recognition system.

Fig. 2 depicts the general overview of the proposed gesture recognition system. Notice that the block diagram implicitly represents a two-stage system: the first stage being the training stage is represented by the top part of the block diagram, whereas the second stage being the testing stage is represented by the bottom part of the block diagram.

The training stage comprises two parts. A sliding window, which acts as a moving average filter, is applied to the acquired data to remove any noise that might have been accumulated due to internal sampling, accelerometer calibration or sensitivity, or hand shaking during gesture acquisition. The smoothing step is followed by a clustering process which is broken into two subblocks. The first clustering subblock deals with the unequal durations of the gesture traces $\mathbf{G}_{i,j}$. This subblock uses DTW to compute a measure of similarity between vectors of unequal lengths. The measure of similarity is then used in the AP subblock to decompose the training data into multiple clusters. Clustering in essence represents the core of the training stage. Members of the same cluster should share the same characteristics; coming from the same gesture is the most desirable one. Each cluster is represented by one of its members called an "exemplar". So, the output of the clustering stage, and in turn the training stage, is a set of exemplars $\mathscr{E}$ each representing a cluster of gesture traces. The details of each subblock are discussed in the following section.

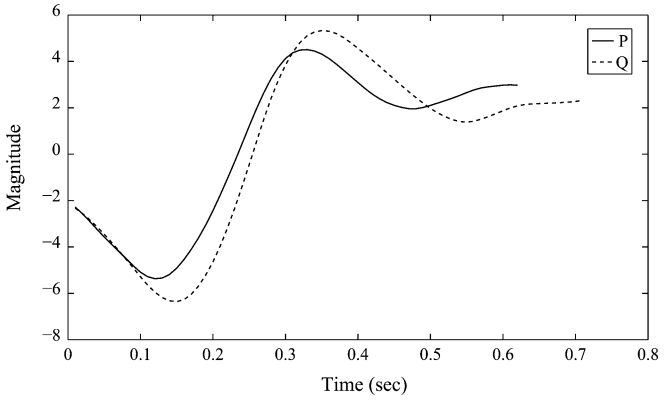

Fig. 3. Two time sequences P and Q that are similar but out of phase.

III. Clustering Algorithm

Recall that gesture traces suffer from inherent temporal variations, and therefore the conventional Euclidean distance is not applicable as a similarity measure between the gesture traces. Consequently, in our gesture recognition system, we resort to dynamic time warping to compute the similarities between the different gesture traces. In this sequel, we represent vectors by bold lower case letters, e.g., $\boldsymbol{r}$, matrices by bold upper case letters, e.g., $\mathbf{R}$, and sets by calligraphic upper case letters, $\mathcal{R}$.

A. Dynamic Time Warping

Dynamic time warping (DTW) matches two time signals, possibly of different durations, 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 [9].

Assume that two time sequences, $\boldsymbol{p}$ and $\boldsymbol{q}$, are similar but are out of phase and are of length n and m, respectively, where $\boldsymbol{p} = [p_1, \ldots, p_n]$ and $\boldsymbol{q} = [q_1, \ldots, q_m]$ as shown in Fig. 3. The objective is to compute the matching cost: $\mathrm{DTW}(\boldsymbol{p}, \boldsymbol{q})$. The matching cost is computed based on dynamic programming using the following formulation:

$$D_{i,j} = d(p_i, q_j) + \min\{D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}\} \quad (2)$$

where the distance function $d(\cdot,\cdot)$ varies with the application. In our gesture recognition system, $d(p_i, q_j)$ is defined as

$$d(p_i, q_j) = (p_i - q_j)^2 \quad (3)$$

and consequently

$$\mathrm{DTW}(\boldsymbol{p}, \boldsymbol{q}) = D_{n,m} \quad (4)$$

In the proposed 3-axis accelerometer gesture recognition system, since each gesture trace is defined by three acceleration waveforms, the similarity cost between gesture trace $\mathbf{G}_i$ of size $n \times 3$ and gesture trace $\mathbf{G}_j$ of size $m \times 3$ is computed as

$$\mathrm{DTW}(\mathbf{G}_i, \mathbf{G}_j) = \sqrt{D^2_{n,m}(x) + D^2_{n,m}(y) + D^2_{n,m}(z)} \quad (5)$$

where $D_{n,m}(x)$, $D_{n,m}(y)$, $D_{n,m}(z)$ are the DTW costs computed between the traces in the x, y, and z axes respectively.

Dynamic programming can potentially render the training stage a slow process since a similarity cost is computed between all gesture traces in the database. As far as our system is concerned, training is done off-line and therefore, the speed is not an issue of concern. However, for other systems designed with online training, speed can be crucial especially with large databases. Therefore, many generic data editing algorithms have been proposed to speed up DTW [10], [11] and alleviate its speed predicament.

B. Affinity Propagation

Affinity propagation (AP) [12] is an algorithm that simultaneously considers all data points as potential exemplars and recursively exchanges real-valued messages among data points until a good set of exemplars and clusters emerges. Clustering is based on the exchange of two types of messages: the "responsibility" message to decide which traces are exemplars, and the "availability" message to decide which cluster a trace belongs to. The responsibility message is given by

$$r(i,j) = s(i,j) - \max_{j' s.t. j' \neq j} \{a(i,j') + s(i,j')\} \quad (6)$$

where $i \neq j$, and $s(i,j)$ is the pairwise similarity that indicates how well the trace $\mathbf{G}_j$ is suited to be the exemplar for the trace $\mathbf{G}_i$ and is defined as

$$s(i,j) = \mathrm{DTW}(\mathbf{G}_i, \mathbf{G}_j)\ \forall i,j \in \{1,2,\ldots,L\} \quad (7)$$

where L is the total number of gesture traces, and the availability message is given by

$$a(i,j) = \min\left\{0, r(j,j) + \sum_{i' s.t. i' \neq i,j} \max\{0, r(i',j)\}\right\}. \quad (8)$$

In addition to the measure of similarity, AP takes as input a set of real numbers, known as self-similarity or *preference* (p) for each gesture trace, so that traces with larger values of p are more likely to be chosen as exemplars. For the proposed gesture recognition system, the self-similarity p is proportional to the median of the input similarities, that is

$$p = \beta * \mathrm{median}\{s(i,j),\ \forall i,j \in \{1,2,\ldots,L\}\} \quad (9)$$

where β is a constant that controls the number of clusters to be generated in an inversely proportional manner. In other words, as the value of β decreases, more clusters will be generated.

AP is chosen as the clustering technique because it does not operate on feature vectors or raw data but rather operates on a matrix of similarities between data points. The similarity costs are computed between the gesture traces which means that clustering is done based on the temporal characteristics of the traces. This configuration of AP utilizes the sparse nature of the gesture traces and eliminates the need for forcing all traces to be of the same length or generating feature vectors of equal lengths as is

the case in [1], [2], [6]. AP can generate better clusters, compared to other clustering techniques like K-means clustering, because of its initialization-independent property [12].

The output of AP is a set of exemplars $\mathscr{E}$ for the N gestures in our system, such that

$$\mathcal{E} = \{\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_H\} \tag{10}$$

where $H \geq N$. Notice that the number of exemplars H obtained is greater than or equal to the number of gestures N. The reason is due to the fact that the gesture traces are collected from different subjects, and as a result, there is a large variance in the data for one gesture. Consequently, a unique exemplar cannot be extracted per gesture. On the contrary, the number of exemplars per gesture, H_i, satisfies

$$1 \leq H_i \leq P \,\forall\, i \in \{1, 2, \ldots, N\} \tag{11}$$

where P is the number of subjects included in training the system.

IV. Recognition Using RP

In order to recognize an unknown gesture trace $\mathbf{Y}$, it is intuitive to compare it to the set of exemplars in $\mathcal{E}$ and classify $\mathbf{Y}$ to the gesture whose exemplar gives the lowest cost. However, since our clustering algorithm does not yield a unique exemplar for each gesture, this approach solely does not suffice in yielding a high recognition accuracy. We make the following observations on the clustering technique used above.

First, we note that, although not observed in our simulations, the affinity propagation technique does not guarantee that all members of a cluster and its exemplar are traces of the same gesture. The problem becomes more significant when gesture traces from different subjects are combined in the same database.

Second, although an exemplar is a representative of its cluster, it cannot be used to detect the corresponding gesture of a testing trace due to the fact that a unique exemplar cannot be extracted per each gesture. However, exemplars are useful in removing outliers, and reducing the size of the search space, hence reducing the computational complexity.

The proposed recognizer comprises two steps. In the first step, the set of exemplars that are closest to the observed data are selected. Then, in the second step, the best match among the members of the clusters chosen in the first step is selected.

To carry out the second step, we still need to address the different gesture trace duration problem. A very efficient solution is to project all the traces onto the same lower dimensional subspace and thus solve the problem of different durations and simultaneously reduce the computational cost. This proposition is motivated by the premise that, as seen in Fig. 1, the defined hand gestures appear to be sparse since the hand follows a smooth trajectory while performing a gesture. Therefore, gesture traces can be represented using fewer samples as per the theory of compressive sensing [13].

Compressive sensing (CS) is a method that allows to recover signals from far fewer measurements than the traditional sampling methods. Assume that the received signal can be represented as an $M \times 1$ vector $\mathbf{x} = \mathbf{\Psi s}$ where $\mathbf{\Psi}$ is an $M \times M$ basis matrix and $\mathbf{s}$ is an $M \times 1$ sparse vector that has only $L_s \ll M$ nonzero elements. Signal $\mathbf{x}$ is compressed using a $K \times M$ sensing matrix $\mathbf{\Phi}$, which yields the measurement vector $\mathbf{y}$ of dimension K as follows:

$$\mathbf{y} = \mathbf{\Phi} \boldsymbol{x} = \mathbf{\Phi\Psi} \boldsymbol{s}. \tag{12}$$

It has been shown that $\boldsymbol{s}$ can be recovered exactly if K satisfies the following:

$$K \geq cL_s \log\left(\frac{M}{L_s}\right) \tag{13}$$

where c is a constant and L_s is the sparsity level [13]. The signal can be reconstructed by solving the following ℓ_1 norm minimization problem:

$$\begin{aligned} \min_{\mathbf{s}} \quad & \|\mathbf{s}\|_1 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{\Phi\Psi s}. \end{aligned} \tag{14}$$

CS has been successfully used for image recovery. It has been shown that CS is a powerful technique in the field of face recognition and very robust in the presence of noise and occlusion [14]. Although the method discussed in [14] shows a great potential to be extended to other problems beyond face recognition, it does not entirely fit in the gesture recognition problem since the method does not address the issue of different gesture trace length.

Accordingly, one solution is to augment CS with a projection feature. In other words, one solution to overcome the different gesture trace sizes is to project all the traces onto the same lower dimensional subspace. An ideal dimensionality reduction technique has the capability of efficiently reducing the data into lower-dimensional subspace while preserving the properties of the original data. RP is the only technique that renders this solution feasible. Other dimensionality reduction techniques, like principal component analysis (PCA) or singular value decomposition (SVD) necessitate that gesture traces be of the same duration for eigenvector and eigenvalue decomposition [15]. RP, on the other hand, does not require traces to be of the same duration as the following sections demonstrate. Furthermore, it has been empirically shown that that RP outperforms the aforementioned dimensionality reduction techniques.

A. RP

RP has recently emerged as a powerful technique for dimensionality reduction [16], [17] . In RP, the original d-dimensional data is projected onto a k-dimensional $(k \ll d)$ subspace using a $k \times d$ random matrix $\mathbf{A}$ whose columns have unit lengths. Using matrix notation, let $\mathbf{X}_{d\times n}$ be the original set of n d-dimensional observations, then the projection problem can be formulated as

$$\mathbf{X}^{\mathrm{RP}}_{k\times n} = \mathbf{A}_{k\times d}\mathbf{X}_{d\times n} \tag{15}$$

where $\mathbf{X}^{\mathrm{RP}}_{k\times n}$ represents the projected data onto the lower k-dimensional subspace.The concept of RP is inspired by the Johnson-Lindenstrauss theorem [18].

Strictly speaking, (15) is not a projection because the projection matrix $\mathbf{A}$ is generally not orthogonal and such a linear mapping can result in significant distortion to the original data set. One solution is to orthogonalize $\mathbf{A}$ but this can be computationally very expensive. Alternatively, we can resort to the fact that in a high-dimensional space, the number of almost orthogonal directions is much larger than the number of orthogonal directions [19]. Therefore, vectors having random directions can be sufficiently close to orthogonal and thus can offer the necessary preservation of the original data set after projection.

Another way to view $\mathbf{A}$ is as a sampling operator for $\mathbf{X}$, and is invertible if each $\boldsymbol{x} \in \mathbf{X}$ is uniquely determined by its sampled or projected data $\mathbf{A}\boldsymbol{x}$; this means if for every $\boldsymbol{u}, \boldsymbol{v} \in \mathbf{X}$, $\mathbf{A}u = \mathbf{A}v$ then $\boldsymbol{u} = \boldsymbol{v}$. In other words, $\mathbf{A}$ is a one-to-one mapping between $\mathbf{X}^{\mathrm{RP}}$ and $\mathbf{X}$ and this allows a unique identification for each $x \in \mathbf{X}$ from $\mathbf{A}\boldsymbol{x}$. However, practically, we want that a small change in $\boldsymbol{x}$ only result in a small change in its sampled or projected data $\mathbf{A}\boldsymbol{x}$. Therefore, we consider a stricter condition given by

$$\begin{aligned} \alpha \parallel \boldsymbol{u}-\boldsymbol{v} \parallel_{\mathcal{H}}^2 \leq \parallel \mathbf{A}u-\mathbf{A}v \parallel_{l_2(k)}^2 &= \sum_{n\in k} |\langle \boldsymbol{u}-\boldsymbol{v}, \psi_n^2 \rangle|^2 \\ &\leq \beta \parallel \boldsymbol{u}-\boldsymbol{v} \parallel_{\mathcal{H}}^2 \end{aligned} \tag{16}$$

where α and β are a constants with $\alpha > 0$ and $\beta < \infty$, and $\mathcal{H}$ is an ambient Hilbert space [20].

The sampling condition (16) on $\mathbf{A}$ is related to the important concept of restricted isometry property (RIP) [21], and is interestingly the same as RIP if $\mathbf{X}$ has sparse columns and the columns come from the same subspace [20], [22] . The choice of the sampling matrix $\mathbf{A}$ is one of the key points of interest. Fundamentally, any distribution of zero mean and unit variance satisfies the sampling condition in (16). Most works use Gaussian distribution. However, simpler distributions such as

$$a_{ij} = \sqrt{3} \cdot \begin{cases} +1 \text{ withprobability} & \frac{1}{6} \\ 0 \text{ with probability} & \frac{2}{3} \\ -1 \text{ with probability} & \frac{1}{6}. \end{cases} \tag{17}$$

have also been reported that result in further computational savings since computations can be done using integer arithmetics [23]. The proposed gesture recognition system will be tested against both forms of the sampling or projection matrix $\mathbf{A}$: the Gaussian distribution and the distribution in (17).

For recognition, the following approach is followed. As per earlier notation, let $\mathbf{Y}$ represent an unknown gesture trace to be recognized, which is a matrix of size $l_y \times 3$, and let $\boldsymbol{y}$ denote one of the columns of $\mathbf{Y}$ and let $\mathcal{R}$ be the set of all traces that are of close resemblance to $\mathbf{Y}$. This resemblance is determined by computing the DTW cost between $\mathbf{Y}$ and every exemplar $\mathbf{E}_j \in \mathscr{E}$, and choosing the clusters whose DTW cost is below a certain threshold α. In other words

$$\mathcal{R} = \{\mathbf{C}_j | \forall j : \mathrm{E}_j \in \mathscr{E} \; and \; \mathrm{DTW}(\mathrm{E}_j, \mathrm{Y}) < \alpha\} \tag{18}$$

where $\mathbf{C}_j$ is any member of the jth cluster with the exemplar $\mathbf{E}_j$. By empirical examination, we have found that the threshold

$$\alpha = 2 \cdot \min \left\{ \mathrm{DTW}\left(\mathbf{E}_j, \mathbf{Y}\right), \; \forall \mathbf{E}_j \in \mathscr{E} \right\} \tag{19}$$

gives the best results.

In order to proceed with the recognition process, the set $\mathcal{R}$ is converted into three matrices by forcing all traces as well as the unknown gesture trace $\mathbf{Y}$ to be in the same space. This is done by finding the maximum length $l_{\max}$ to be

$$l_{\max} = \max\{l_y, l_1, \ldots, l_L\} \tag{20}$$

where L is the total number of traces in $\mathcal{R}$ and $l_1, \ldots, l_L$ represent their lengths, i.e., the number of rows in each trace. After $l_{\max}$ is found, all the traces including the unknown gesture trace $\mathbf{Y}$, which are shorter than $l_{\max}$ are padded by zeros forcing them to be of length $l_{\max}$. In other words, all traces are transformed to the largest subspace by assuming that the shorter traces have zero components in the higher subspaces. Zero padding can take any possible form, i.e., adding zeros to the beginning of the trace, adding zeros in between the trace samples, or adding zeros to the end of the trace. This is due to the fact that the RP matrix $\mathbf{A}$ satisfies the RIP condition and therefore, it makes no difference which columns of $\mathbf{A}$ are chosen to compress the trace [21]. However, for simplicity of application, we pad zeros to the end of the traces, and in this case

$$\mathbf{R}_x = [\mathbf{r}_1^x, \mathbf{r}_2^x, \ldots, \mathbf{r}_L^x] = \begin{bmatrix} r_{1,1}^x & r_{2,1}^x & \cdots & r_{L,1}^x \\ r_{1,2}^x & r_{2,2}^x & \cdots & r_{L,2}^x \\ \vdots & \vdots & \ddots & \vdots \\ r_{1,l_1}^x & r_{2,l_2}^x & \cdots & r_{L,l_L}^x \\ 0_1 & 0_2 & \cdots & 0_L \end{bmatrix} \tag{21}$$

and

$$\boldsymbol{y}_x = \begin{bmatrix} y_1^x \\ y_2^x \\ \vdots \\ y_{l_y} \\ 0_y \end{bmatrix} \tag{22}$$

where $\mathbf{R}_x$ is a matrix whose columns represent the x-components of the padded traces, $\boldsymbol{y}_x$ is the x-component of the padded unknown gesture trace, and 0_i and 0_y are zero vectors of length $(l_{\max} - l_i)$ and $(l_{\max} - l_y)$, respectively. $\mathbf{R}_y$, $\mathbf{R}_z$, $\boldsymbol{y}_y$, and $\boldsymbol{y}_z$ are constructed in a similar manner.

In order to project the data onto the lower dimensional subspace, the projection matrix $\mathbf{A}$ is constructed based on the distributions defined earlier and would be of size $l_k \times l_{\max}$, where l_k is the dimension of the new common lower dimensional subspace. According to Fig. 1, gesture waveforms are smooth curves, and one of the transformations which would give a sparse representation of the waveforms is the Fourier transform. So, for a sparse sequence $\boldsymbol{r}$, let $\tilde{\boldsymbol{r}}$ and $k_{\boldsymbol{r}}$ denote the Fourier transform and the sparsity level of the sequence $\boldsymbol{r}$, respectively. Moreover, let r_m denote the maximum magnitude in $\tilde{\boldsymbol{r}}$. The sparsity level $k_{\boldsymbol{r}}$ of $\boldsymbol{r}$ is defined as

$$k_{\boldsymbol{r}} = K \cdot B_\gamma \tag{23}$$

where K is a constant and B_γ is the number of samples in $\tilde{\boldsymbol{r}}$ that are greater than a threshold γ defined as

$$\gamma = c \cdot r_m \tag{24}$$

where c is a constant $\in (0,1)$ to preserve only the significant samples. In practice, K can be either 3 or 4 making the sparsity level $k_{\boldsymbol{r}}$ three or four times B_γ [13]. Accordingly, the Fourier transform of a trace $\mathbf{R}$ is defined as

$$\tilde{\boldsymbol{R}} = [\tilde{\boldsymbol{r}}_x \tilde{\boldsymbol{r}}_y \tilde{\boldsymbol{r}}_z]. \tag{25}$$

The sparsity of $\mathbf{R}$ is then given by

$$k_{\mathbf{R}} = \max\left\{k_{\boldsymbol{r}_x}, k_{\boldsymbol{r}_y}, k_{\boldsymbol{r}_z}\right\}. \tag{26}$$

The sparsity level $k_{\mathbf{R}}$ is computed for each trace in (1) and stored in the database as well. Consequently

$$l_k = \max\left\{k_{\mathbf{R}_i};\ \forall\, i \in \{1,2,\ldots,L\}\right\}. \tag{27}$$

After $\mathbf{A}$ is constructed, the data in the x-direction is projected as

$$\bar{\mathbf{R}}_x = \mathbf{A}\mathbf{R}_x = [\mathbf{A}\mathbf{r}_1^x, \mathbf{A}\mathbf{r}_2^x, \ldots, \mathbf{A}\mathbf{r}_L^x] \tag{28}$$

and

$$\bar{\boldsymbol{y}}_x = \mathbf{A}\boldsymbol{y}_x \tag{29}$$

where $\bar{\mathbf{R}}_x$, represents the projected data in the x-direction onto the new subspace and $\bar{\boldsymbol{y}}_x$ represents the projected x-component of the unknown gesture trace.

The relationship between $\bar{\mathbf{R}}_x$ and $\bar{\boldsymbol{y}}_x$ can be formulated as

$$\bar{\boldsymbol{y}}_x = \bar{\mathbf{R}}_x \boldsymbol{\theta}_x \tag{30}$$

where $\boldsymbol{\theta}_x$ is theoretically a 1-sparse $L \times 1$ vector whose elements are all zeros except $\boldsymbol{\theta}_x(n) = 1$, such that $\boldsymbol{r}_n^x$ best resembles $\boldsymbol{y}_x$. Namely,

$$\boldsymbol{\theta}_x = [0,\ldots,0,1,0,\ldots,0]^T \tag{31}$$

where T denotes transposition. However, gesture traces suffer from inherent temporal variations and therefore, the above ideal scenario of having a perfect match to the unknown gesture trace is impossible and therefore, the problem can be reformulated as

$$\bar{\boldsymbol{y}}_x = \bar{\mathbf{R}}_x \boldsymbol{\theta}_x + \boldsymbol{\varepsilon}_x \tag{32}$$

where $\boldsymbol{\varepsilon}_x$ is the measurement noise.

Using the same formulation as in [24], we introduce the preprocessor $\mathbf{W}$, which is defined as

$$\mathbf{W}_x = \mathbf{Q}_x \bar{\mathbf{R}}_x^{\dagger} \tag{33}$$

where $\mathbf{Q}_x = \mathrm{orth}(\bar{\mathbf{R}}_x^T)^T$, and $\mathrm{orth}(\bar{\mathbf{R}}_x)$ is an orthogonal basis for the range of $\bar{\mathbf{R}}_x$, and $\bar{\mathbf{R}}_x^{\dagger}$ is the pseudoinverse of the matrix $\bar{\mathbf{R}}_x$. The gesture recognition problem takes on a new formulation as,

$$\boldsymbol{h}_x = \mathbf{W}_x \bar{\boldsymbol{y}}_x = \mathbf{Q}_x \boldsymbol{\theta}_x + \boldsymbol{\varepsilon}'_x \tag{34}$$

where $\boldsymbol{\varepsilon}'_x = \mathbf{W}_x \boldsymbol{\varepsilon}_x$. $\boldsymbol{\theta}_x$ can be well recovered from $\boldsymbol{h}_x$ with a high probability through the following ℓ_1-minimization formulation:

$$\hat{\boldsymbol{\theta}}_x = \arg\min \|\, \boldsymbol{\theta}_x \,\|_1, s.t. \boldsymbol{h}_x = \mathbf{Q}_x \boldsymbol{\theta}_x + \boldsymbol{\varepsilon}'_x. \tag{35}$$

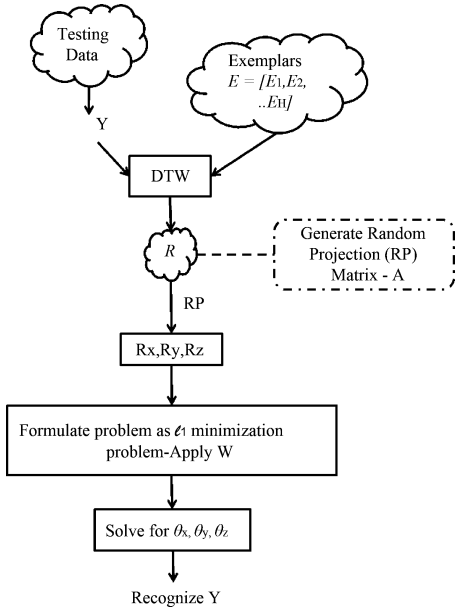

Fig. 4. Block diagram of testing stage.

This represents recognition of the unknown trace based on data in the x-direction only. $\boldsymbol{\theta}_y$ and $\boldsymbol{\theta}_z$ are solved for using the same approach.

In order to recognize the unknown gesture trace, the three $\hat{\boldsymbol{\theta}}_x, \hat{\boldsymbol{\theta}}_y, \hat{\boldsymbol{\theta}}_z$ vectors are combined together in the following manner,

$$\hat{\boldsymbol{\theta}}_{eq} = \hat{\boldsymbol{\theta}}_x^2 + \hat{\boldsymbol{\theta}}_y^2 + \hat{\boldsymbol{\theta}}_z^2. \tag{36}$$

The unknown gesture trace is then recognized as the gesture to which the trace $\mathbf{R}_i$ belongs such that $\hat{\boldsymbol{\theta}}_{eq}(i)$ is maximum. Fig. 4 shows a complete block diagram of the testing stage.

V. Implementation Results

The acceleration data corresponding to the different gestures is collected using a wiimote, which has a built-in 3-axis accelerometer. A gesture trace is segmented using the "trigger" button or "B" button on the bottom of the remote. In other words, a trace starts by pressing and holding the "B" button and ends by releasing it. This manner of acquiring the gesture traces overcomes the challenging problem of gesture spotting.

A dictionary of 18 gestures is created as shown in Fig. 5. To the best of our knowledge, our dictionary of gestures is the largest in published studies for accelerometer-based gesture recognition. The defined gestures are not limited to one plane only as is the case in other studies [4], [6], but span the two planes: XZ and YZ planes. The dictionary contains a variety of gestures ranging from the simple right, left, up, down gestures to more complex gestures resembling letters and numbers. In terms of sample length, gesture trace size ranges from about 35 samples up to 200 samples depending on the complexity of the gesture. This definition of gestures is to increase the robustness of the gesture recognition system.

The database consists of 3780 traces and is built by acquiring gestures from seven subjects (two females and five males) using

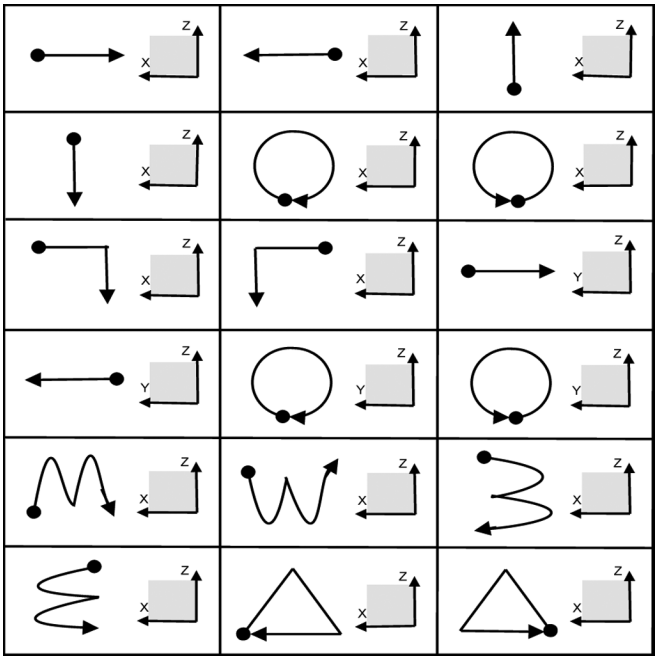

Fig. 5. The dictionary of 18 gestures.

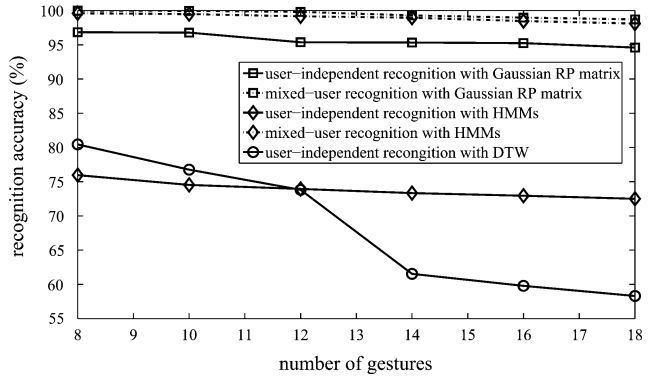

Fig. 6. System's performance against the number of gestures using a Gaussian RP matrix compared to the baseline model and HMM.

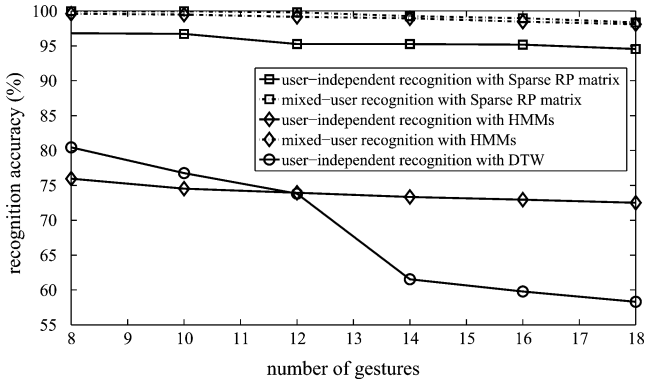

Fig. 7. System's performance against the number of gestures using a sparse RP matrix compared to the baseline model and HMM.

the wiimote. Each subject is asked to repeat each gesture 30 times resulting in a total of 540 traces for all gestures per participant or a total of 210 traces from all participants per gesture. A gesture acceleration waveform from the same person can differ drastically if the tilting angle of the accelerometer is large. Therefore, all participants are asked to try their best to perform the gestures without any, or with minimal, tilting of the remote.

For system evaluation, the database is split into two datasets: a training set and a testing set. The training dataset is generated by choosing traces from three users (two males and one female) out of the seven users, i.e., $P = 3$. Specifically, five traces are ***randomly*** chosen for each gesture from each of the three users resulting in a total of 15 traces per gesture, i.e., $M = 15$. The testing dataset comprises all the remaining traces from the three users plus the entire set of traces from the remaining four users (3 males and 1 female). Simulations are run for $N = \{8, 10, 12, 14, 16, 18\}$ gestures. A typical value of l_k is about 20 samples and the system uses ℓ_1 MAGIC toolbox, available online: http://www.acm.caltech.edu/l1magic/, to solve the ℓ_1 minimization problem 35 using basis pursuit.

The system's performance is compared to a baseline model where recognition is carried out using one-nearest-neighbor DTW. Furthermore, the system is compared to a system of continuous hidden Markov models (HMMs), the system in [4], and finally, to a system in literature developed using discrete HMMs [25]. Figs. 6 and 7 show the system's performance in terms of recognition accuracy against the number of gestures for a projection matrix $\mathbf{A}$, formed with a Gaussian distribution and the sparse distribution in (17), respectively. Each of Figs. 6 and 7 also show a comparison of performance between our system and the baseline model as well as a system of continuous HMMs. In order to develop the system of continuous HMMs, the system is set up in an identical manner to [3]: a left to right HMM with a continuous gaussian distribution is used to model each gesture. The output distributions are assumed to have diagonal covariance matrices. Consequently, each HMM can be described as an $8m$-parameter model, where m is the number of states. The eight parameters comprise the two state transition probabilities, and one Gaussian output distribution which constitutes a mean vector $\boldsymbol{\mu} \in \mathcal{R}^3$ and the three diagonal elements of the covariance matrix. For comparison, simulations are run with $m = 10$.

Figs. 6 and 7 show that the system's performance is almost identical for both definitions of the projection matrix $\mathbf{A}$ which confirms that the sparse distribution in (17) is a very good approximation of the Gaussian distribution. The system yields a very competitive performance for a system of 8 gestures, giving a recognition accuracy of 96.84%. The dashed lines show the system's performance only on traces in the testing dataset from the three subjects whose data is used in training the system. In other words, this type of recognition can be referred to as mixed-user recognition. The solid lines show the system's performance on the entire testing dataset which includes traces from all the seven subjects, and this type of recognition is referred to as user-independent recognition. As shown, the system greatly outperforms the baseline model and the continuous HMM-system for all dictionary sizes. As highlighted in [4], one-nearest neighbor DTW is very effective with small dictionary sizes and that is why the baseline model's performance drops sharply for a dictionary size of 12 gestures or more. Fig. 8 depicts the cumulative density functions (cdfs) of the system's performance using a

TABLE I
COMPARISON OF PERFORMANCE OF PROPOSED SYSTEM AND OTHER SYSTEMS IN THE LITERATURE

Technique	no. of gestures	Accuracy(%)		
		User-Dependent	Mixed-User	User-Independent
Proposed System	8 - 18	100 - 99.81	99.98 - 98.71	96.84 - 94.60
Continuous HMM-System	8 - 18	99.97 - 99.54	99.61 - 98.11	75.96 - 71.50
uWave	8	98.6	-	75.4
System of discrete HMMs in [25]	5	89.7	89.7	-

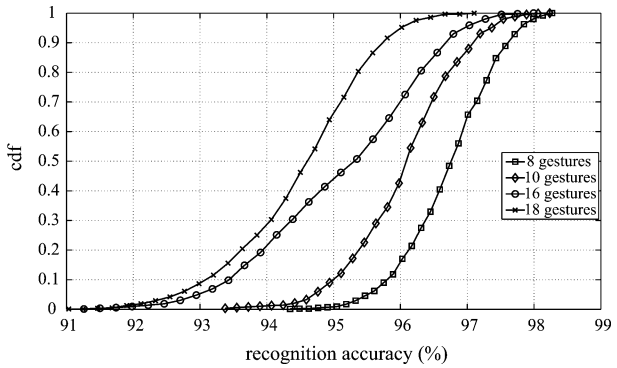

Fig. 8. Cdfs of the system's performance for $N = 8$, 10, 16, and 18 gestures using a Gaussian RP matrix.

Gaussian projection matrix for dictionary sizes of 8, 10, 16, and 18 gestures, respectively. The cdfs are generated by running the code 1000 times and recording the system's recognition accuracy for each run.

Table I shows a comparison of performance between our proposed system, continuous HMM-system, uWave, and the system of discrete HMMs in [25]. From Table I, we can deduce that our proposed system outperforms uWave system in [4] for user-dependent recognition. The system yields a perfect recognition accuracy for a dictionary of 8 gestures using the gestures defined compared to an accuracy of 98.4% by uWave. Finally, our system outperforms the system of discrete HMMs in [25] which yields an average accuracy of 90% for a dictionary size of five gestures only compared to our dictionary size of 18 gestures.

VI. CONCLUSION

In conclusion, we have proposed a novel gesture recognition system based solely on data from a single 3-axis accelerometer. The system employs dynamic time warping and affinity propagation algorithms for efficient training. In the testing phase, the unknown trace is compared to the exemplars induced by affinity propagation to select a subset of coordinate traces. The sparse nature of the gesture traces is exploited to project candidate traces and the unknown gesture trace onto the same lower-dimensional subspace. The system is tested on a dictionary of 18 gestures whose database contains over 3700 traces collected from 7 subjects. The system achieves almost perfect recognition for user-dependent recognition and extremely competitive accuracies for mixed-user and user-independent recognition when compared to the other systems in literature.

REFERENCES

[1] A. 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," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn. Workshops (CVPRW '08)*, Jun. 2008, pp. 1–8.

[2] X. Zhang, X. Chen, W. W. hui, J. Y. hai, V. Lantz, and K. W. qiao, "Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors," in *Proc. 13th Int. Conf. Intell. User Interfaces (IUI '09)*, New York, 2009, pp. 401–406.

[3] T. Pylvänäinen, "Accelerometer Based Gesture Recognition Using Continuous HMMs," in *Pattern Recognition and Image Analysis*. New York: Springer Berlin/Heidelberg, 2005, pp. 639–646.

[4] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "UWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive Mobile Comput.*, vol. 5, no. 6, pp. 657–675, 2009.

[5] A. Akl and S. Valaee, "accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 2010, pp. 2270–2273.

[6] J. Kela, P. Korpip, J. Mantyjarvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca, "Accelerometer-based gesture control for a design environment," *Pers. Ubiq. Comput.*, vol. 10, no. 5, pp. 285–299, 2006.

[7] The Global Wii Experience Website 2010 [Online]. Available: http://us.wii.com/iwata_asks/wii_remote/

[8] C. A. Wingrave, B. Williamson, P. D. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, and J. J. LaViola, "The wiimote and beyond: Spatially convenient devices for 3D user interfaces," *IEEE Comput. Graph. Appl.*, vol. 30, pp. 71–85, 2010.

[9] E. Keogh, "Exact indexing of dynamic time warping," in *Proc. 28th Int. Conf. Very Large Data Bases*, 2002, pp. 406–417.

[10] D. R. Wilson and T. R. Martinez, "Instance pruning techniques," in *Proc. Mach. Learn.: Proc. 14th Int. Conf. (ICML)'97*, 1997, pp. 404–411.

[11] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," *Proc. ICML'06*, pp. 1033–1040, 2006.

[12] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[13] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.* , vol. 25, no. 2, pp. 21–30, Mar. 2008.

[14] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.* , vol. 31, no. 2, pp. 210–227, Feb. 2009.

[15] J. Liu and M. Kavakli, "Hand Gesture Recognition Based on Segmented Singular Value Decomposition," in *Knowledge-Based and Intelligent Information and Engineering Systems, ser. Lecture Notes in Computer Science*, R. Setchi, I. Jordanov, R. Howlett, and L. Jain, Eds. New York: Springer Berlin/Heidelberg, 2010, vol. 6277, pp. 214–223.

[16] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discov. Data Mining*, 2001, pp. 245–250.

[17] J. Lin and D. Gunopulos, "Dimensionality reduction by random projection and latent semantic indexing," in *Proc. Text Mining Workshop, 3rd SIAM Int. Conf. Data Mining*, May 1–3, 2003.

[18] W. B. Johnson and J. Lindemstrauss, "Extensions of Lipshitz mapping into Hilbert space," in *Proc. Conf. Modern Analysis and Probabil. Contemporary Math.*, 1984, vol. 26 , pp. 189–206.

[19] R. Hecht-Nielsen, "Context vectors: General purpose approximate meaning representations self-organized from raw data," in *Computational Intelligence: Imitating Life*, J. M. Zurada, R. Marks, II, and C. J. Robinson, Eds. Piscataway, NJ: IEEE Press, 1994, pp. 43–56.
[20] Y. Lu and M. Do, "Sampling signals from a union of subspaces," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 41–47, Mar. 2008.
[21] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
[22] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
[23] D. Achlioptas, "Database-friendly random projections," in *Proc. ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Syst.*, 2001, pp. 274–281.
[24] C. Feng, S. W. A. Au, S. Valaee, and Z. H. Tan, "Orientation-aware localization using affinity propagation and compressive sensing," in *Proc. IEEE Int. Workshop on Computat. Adv. Multi-Sens. Adapt. Process. (CAMSAP)*, 2009.
[25] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a Wii controller," in *Proc. 2nd Int. Conf. Tangible and Embedded Interaction (TEI'08)*, 2008, pp. 11–14.

Ahmad Akl (S'10) received the B.Sc. degree from the American University of Sharjah, United Arab Emirates, in 2006, and the M.A.Sc. degree from the University of Toronto, Canada, in 2010, all in electrical engineering. His Master's degree work focused on developing a novel accelerometer-based gesture recognition system under the supervision of Dr. S. Valaee.

His research interests include gesture recognition, activity recognition, compressive sensing, artificial intelligence, smart systems, and machine learning.

Chen Feng (S'09) received the B. Engr. and Ph.D. degrees from the Department of Electronics and Information Engineering, Beijing Jiaotong University, China, in 2006 and 2011, respectively.

From 2008 to 2010, she was a visiting Ph.D. student with the WIRLab, University of Toronto, Canada. She is currently a Postdoctoral Fellow at the University of Toronto, working on indoor positioning systems.

Shahrokh Valaee (S'88–M'00–SM'02) received the B.Sc. and M.Sc. degrees from the University of Tehran, and the Ph.D. degree from McGill University, Canada, all in electrical engineering.

He is the Associate Chair for Undergraduate Studies and the holder of the Nortel Institute Junior Chair of Communication Networks, Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada. From 1994 to 1995, he was a Research Associate at INRS Telecom, University of Quebec, Montreal, Canada. From 1996 to 2001, he was an Assistant Professor with the Department of Electrical Engineering, Tarbiat Modares University, Tehran, Iran, and with the Department of Electrical Engineering, Sharif University of Technology, Tehran. During this period, he was also a consultant to Iran Telecommunications Research Center. Since 2001, he has been with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, and is the founder and the Director of the Wireless and Internet Research Laboratory (WIRLab). His current research interests are in wireless vehicular and sensor networks, location estimation, and cellular networks.

Dr. Valaee is an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and an Associate Editor of the IEEE *Signal Processing Magazine*. He is the Co-Chair of IEEE PIMRC 2011 and the Co-Chair for Wireless Communications Symposium of IEEE GLOBECOM 2006, a guest Editor for IEEE *Wireless Communications Magazine*, Wiley *Journal on Wireless Communications and Mobile Computing*, and the EURASIP *Journal on Signal Processing*.