This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2546942, IEEE Sensors Journal

Accelerometer-Based Hand Gesture Recognition by Neural Network and Similarity Matching

Renqiang Xie, and Juncheng Cao

***Abstract*—In this paper, we present an accelerometer-based pen-type sensing device and a user-independent hand gesture recognition algorithm. Users can hold the device to perform hand gestures with their preferred handheld styles. Gestures in our system are divided into two types: the basic gesture and the complex gesture which can be represented as a basic gesture sequence. A dictionary of 24 gestures including 8 basic gestures and 16 complex gestures is defined. An effective segmentation algorithm is developed to identify individual basic gesture motion intervals automatically. Through segmentation, each complex gesture is segmented into several basic gestures. Based on the kinematics characteristics of the basic gesture, 25 features are extracted to train the feedforward neural network model. For basic gesture recognition, the input gestures are classified directly by the feedforward neural network classifier. Nevertheless, the input complex gestures go through an additional similarity matching procedure to identify the most similar sequences. The proposed recognition algorithm achieves almost perfect user-dependent and user-independent recognition accuracies for both basic and complex gestures. Experimental results based on 5 subjects, totaling 1600 trajectories, have successfully validated the effectiveness of the feedforward neural network and similarity matching based gesture recognition algorithm.**

***Index Terms*—Accelerometer, gesture recognition, gesture segmentation, feedforward neural network, similarity matching.**

I. INTRODUCTION

GESTURE recognition refers to the process of understanding and classifying meaningful movements of a human's fingers, hands, arms or head [1]. Hand gesture as a natural, intuitive, and convenient way of human-computer interaction (HCI) will greatly ease the interaction process. For instance, in [2], a hand gesture recognition based motion control system of intelligent wheelchair is developed for those with physical accessibility problem; five gestures are employed to separately control the motion of the wheelchair: left turn, right turn, forward, backward, and stop. Other proposed applications of hand gesture recognition include robot-assisted living [3], automatic user state recognition for low-cost television control system [4], and smart ring [5].

Conventional computer vision-based hand gesture recognition can track and recognize gestures effectively without any contact to the user [6], [7]. However, vision-based techniques may be affected by lighting conditions, which will limit the application scenarios, particularly in mobile environment. With the rapid development of sensor technology, triaxial accelerometers are being increasingly embedded into consumer electronic products. A significant advantage of accelerometer-based sensing devices is that they can be operated without any external reference or limitation in working conditions [8]. Hand gesture recognition is relatively complicated since different persons have different speeds and styles to perform gestures. Thus, some researchers have tried to combine data from a triaxial accelerometer with data from electromyography (EMG) sensors [9], [10] or vision sensors [11], [12] in order to improve the system's performance and robustness. However, multi-sensor fusion increases additional cost as well as computational burden.

Concerning the recognition methodologies, Hidden Markov Model (HMM) and Dynamic Time Warping (DTW) as two important approaches are widely used to recognize hand gestures effectively [2], [3], [9], [13], [14]. Other proposed methods include Probabilistic Neural Network (PNN) [8], Most Probable Longest Common Subsequence (MPLCS) [15], Sign Sequence and Template Matching (SSTM) [16], and Stochastic Linear Formal Grammar (SLFG) [17]. Generally, the subjects from which the trajectories are collected to construct the classifier are not consistent with the end users of the system. To develop a user-independent algorithm, Akl *et al.* [1] proposed an accelerometer-based gesture recognition system, which employed Dynamic Time Warping and Affinity Propagation (DTW & AP) algorithms to create exemplars for each gesture during the training stage. A database of 3780 traces was created for a dictionary of 18 gestures. The system achieves accuracies of 99.81% and 94.60% for user-dependent and user-independent recognitions for the 18 gestures, respectively.

In this paper, an accelerometer-based pen-type sensing device as well as a Feedforward Neural Network and Similarity Matching (FNN & SM) based hand gesture recognition algorithm are presented. The work of this paper is built upon a preliminary version of our gesture recognition system [5]. The accelerations generated by hand movements are collected and transmitted to a personal computer (PC) via a USB cable. Users can hold the device to perform hand gestures with their preferred handheld styles in free space. Gestures in our system are divided into two types: the basic gesture and the complex gesture which can be represented as a basic gesture

This work was supported in part by the National Natural Science Foundation of China under Grant 61131006, in part by the 973 Program of China under Grant 2014CB339803, and in part by the 863 Program of China under Project 2011AA010205.

R. Xie is with the Key Laboratory of Terahertz Solid-State Technology, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China, and also with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: xierq@shanghaitech.edu.cn).

J. C. Cao is with the Key Laboratory of Terahertz Solid-State Technology, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China (e-mail: jccao@mail.sim.ac.cn).

sequence. The gesture recognition algorithm is composed of data acquisition and signal preprocessing, gesture segmentation, feature extraction, classifier construction, basic gesture encoding, and similarity matching. For basic gesture recognition, the input gestures are classified directly by the FNN classifier. Nevertheless, the input complex gestures go through an additional similarity matching procedure to identify the most similar sequences. The main contributions of this paper include the following: 1) a segmentation scheme is proposed to identify the starting and end points of each basic gesture automatically; through segmentation, each complex gesture is segmented into several basic gesture motion intervals; 2) based on the kinematics characteristics of the basic gesture, 25 effective features are extracted; 3) the FNN and SM algorithms are successfully combined; the FNN classifier provides good recognition accuracy, while the SM approach enhances the extendibility of the system; 4) the contrast test for user-dependent and user-independent classifications is conducted to validate the user dependency of the proposed system.

The remainder of this paper is organized as follows. We first introduce the accelerometer-based pen-type sensing device in Section II. The FNN and SM based hand gesture recognition algorithm is elaborated detailedly in Section III. In Section IV, experimental results are presented and discussed to validate the effectiveness of the proposed approach. Finally, conclusions are given in the last section.

II. Sensing System Overview

The pen-type gesture capturing device, which is shown in Fig. 1, consists of a triaxial MEMS (Micro-electromechanical Systems) accelerometer (MMA9551L) and a microcontroller (IAP15W4K58S4). The MMA9551L possesses a user-selectable acceleration full scale of ±2 g, ±4 g, and ±8 g, and its supply voltage range is 1.71 V – 1.89 V. The IAP15W4K58S4 is a new generation of 8051 MCU with high speed, wide voltage range (2.5 V – 5.5 V), low power consumption, and has a fully compatible instruction set with traditional 8051 series microcontroller. In our system, the supply voltages of MMA9551L and IAP15W4K58S4 are set to 1.8 V and 3.3 V, respectively. Due to mismatching of levels, a voltage-level translator (PCA9306) is added between them. The schematic diagram of the pen-type sensing device is shown in Fig. 2.

The accelerometer measures the analog acceleration signals generated by a user's hand movements and converts the signals to digital ones via the internal 16-bits A/D converter. The microcontroller collects the digital acceleration signals from the accelerometer through IIC interface and transmits the signals to a PC via a USB cable for further signal processing and analysis. The accelerometer's sensitivity is set from -8 g to $+8$ g in this study, and the output signals of the accelerometer are sampled at 100 Hz. Note that all signal processing procedures presented in Section III are performed on the PC. The size of the pen-type circuit board is 14 cm × 2.4 cm × 1.5 cm and the corresponding coordinate system of the accelerometer is illustrated in Fig. 1.

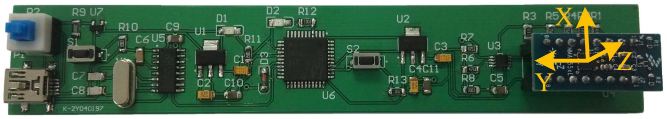


Fig. 1. Sensing device and its coordinate system.

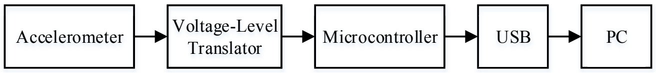


Fig. 2. Schematic diagram of the pen-type sensing device.

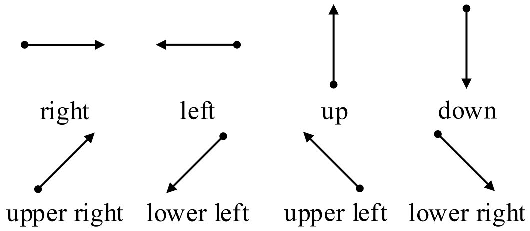


Fig. 3. Trajectories of eight basic gestures.

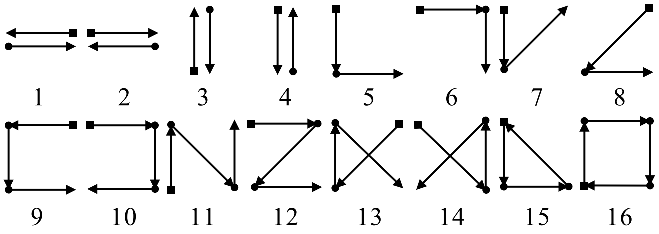


Fig. 4. Trajectories of sixteen complex gestures and their corresponding numbers.

III. Hand Gesture Recognition

The gesture recognition algorithm is composed of data acquisition and signal preprocessing, gesture segmentation, feature extraction, classifier construction, basic gesture encoding, and similarity matching. In this paper, a dictionary of 24 gestures including 8 basic gestures (see Fig. 3) and 16 complex gestures (see Fig. 4) is defined. Since the basic gesture recognition procedures are part of those of complex gesture recognition, we will focus on the whole procedures of the latter. The acceleration signals of hand movements are measured by a triaxial accelerometer and then preprocessed by a moving average filter. A segmentation algorithm is developed to identify the starting and end points of each basic gesture automatically. After feature extraction, the basic gesture samples are used to train the FNN model which subsequently is utilized to classify the basic gesture sequences. The recognized basic gesture sequence is then encoded with Johnson codes. Finally, the complex gesture is recognized by comparing the similarity between the predicted basic gesture sequence and the standard template sequences. Note that for basic gesture recognition, it does not include the encoding and matching procedures, and is classified directly by the trained FNN classifier. The block diagram of the proposed hand gesture recognition algorithm is shown in Fig. 5 and the detailed procedures of the algorithm are introduced as follows.

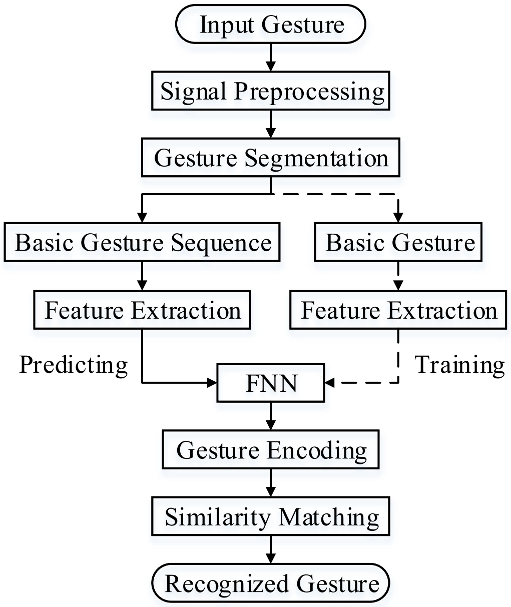


Fig. 5. Block diagram of the proposed hand gesture recognition algorithm.

A. Gesture Introduction

The 8 basic gestures including *right* (R), *left* (L), *up* (U), *down* (D), *upper right* (UR), *lower left* (LL), *upper left* (UL), and *lower right* (LR) are shown in Fig. 3, while the 16 complex gestures and their corresponding numbers (1–16) are shown in Fig. 4. The basic gesture movements can be extended by users as they can build their own profiles of sequences. For instance, the gesture ‘square’ (gesture number is 16) can be split into *up*, *right*, *down*, and *left*. Hence a basic gesture sequence of U-R-D-L is generated.

B. Data Acquisition and Signal Preprocessing

The raw acceleration signals of hand movements are generated by the triaxial accelerometer and collected by the microcontroller. In order to reduce the influence of unintended hand motions, a button is employed to trigger the sampling. Specifically, the button should be pressed before performing a gesture and released after the gesture is completed. The measured signals are always contaminated by the disturbances of the sensor as well as users’ unconscious trembles. To settle this problem, first, the gravitational acceleration is removed by subtracting the mean value of sampled accelerations from each data point to obtain accelerations generated by hand movement. The next step of the signal preprocessing is to reduce high frequency noise by using a moving average filter which is expressed as

$$a[n] = \frac{1}{2M+1} \sum_{m=-M}^{M} a_s[n+m] \tag{1}$$

where $a_s[n]$ is the acceleration signal without gravitational acceleration, and $a[n]$ is the filtered acceleration signal. In this study, we set $M = 5$, i.e., the number of points in the filter is 11.

C. Gesture Segmentation

Gesture segmentation aims to identify the starting and end points of each motion from the preprocessed accelerations. Reliable and accurate segmentation is essential to gesture recognition. Each complex gesture is segmented into two or more basic gesture motion intervals by the segmentation program. In [1], [8], a gesture trace is directly segmented by pressing and releasing a button. However, this is not always consistent with the real situation, i.e., the starting and end points of a motion are not the time pressing and releasing the button respectively. Here we propose a novel segmentation scheme. Given the preprocessed acceleration data sequence $AS = \{a[1], a[2], \cdots, a[L]\}$, where $a[n] = (a_x[n], a_y[n], a_z[n])$ is a three-dimensional vector. $d[n]$ is defined as the Euclidean distance between $a[n]$ and $a[n-1]$. The acceleration is relatively stable when there is no hand movement. In contrast, it varies dramatically when in hand motion state. This means $d[n]$ in motion state is much higher than in status of no movement, hence it can be used to segment a gesture motion. In order to avoid the disturbances of the signal, a moving average filter is applied,

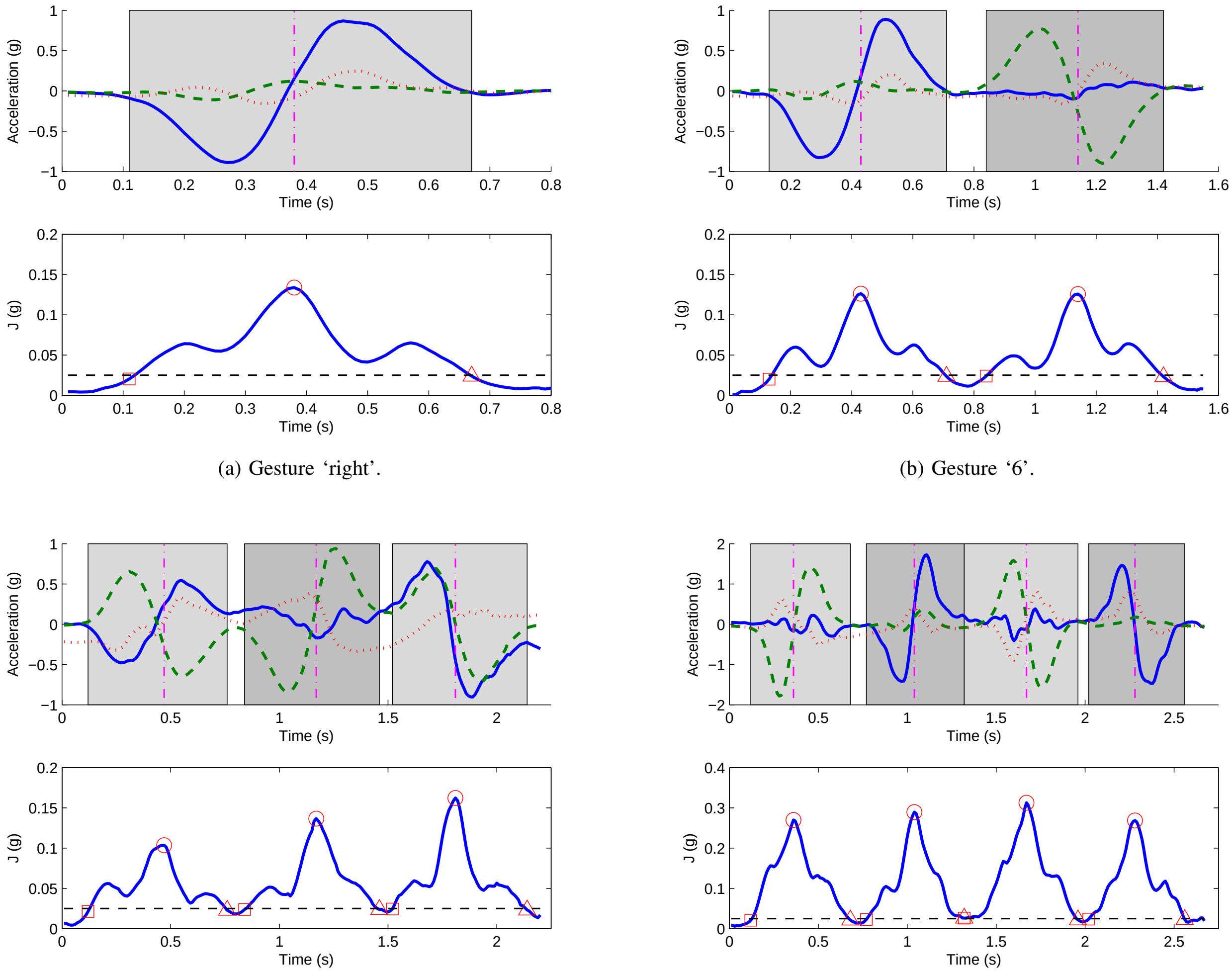
$$J[n] = \frac{1}{2N+1} \sum_{m=-N}^{N} d[n+m] \tag{2}$$

where $J[n]$ is the filtered Euclidean distance and $N = 5$.

The preprocessed accelerations and J of gesture ‘right’, ‘6’, ‘14’, and ‘16’ are shown in Fig. 6. The accelerations are mainly on x- and y-axis. Note that each basic gesture corresponds to a main peak (marked with a circle) of J. With a predefined threshold, which is set to 0.025 g and indicated by black dashed lines in the plots of J, the basic gesture motion intervals are identified. The starting point and end point of each basic gesture are marked with a square and a triangle, respectively; and the corresponding motion interval is indicated by shading. Furthermore, the sampling time of the peak is also indicated by a vertical magenta dashdot line in the acceleration plot. Note that it is located around zero crossing point of the acceleration curve. In order to detect the main peak accurately, we judge whether the value of current point is larger than the values of the 20 points on its left and right, and the interval time between two consecutive peaks should be greater than 0.3 seconds. Note that the minimal J between two adjacent peaks may be larger than the threshold as indicated in Fig. 6d, to settle this, the minimal point is selected as the segmentation point. In addition, there may exist some disturbances around the threshold, hence producing multiple intersection points. In this case, the point closest to the peak is chosen as the segmentation point. The handling of the two special cases makes the segmentation algorithm have good anti-noise ability.

D. Feature Extraction

The key of gesture recognition is to extract effective features which reflect the motion characteristics of different gestures. Since a complex gesture is segmented into several basic gestures, here we just extract the features of basic gestures.



(a) Gesture ‘right’.

(b) Gesture ‘6’.

(c) Gesture ‘14’.

(d) Gesture ‘16’.

Fig. 6. Gesture segmentation. The blue solid, green dashed, and red dotted lines in the top half of each part are accelerations on x-, y-, and z-axis, respectively.

In this study, 25 features are extracted from the segmented hand motion interval, including MAV_x, MAV_y, $RMAV$, $|\Delta MAV|$, AL_x, AR_x, AL_y, AR_y, SL_x, SR_x, SL_y, SR_y, AJ_x, AJ_y, SAJ_x, SAJ_y, RAJ, $SRAJ$, ΔAL, ΔAR, $|\Delta AL|$, $|\Delta AR|$, $SDAL$, $SDAR$, and r. Note that the subscripts x and y indicate the features are extracted from the accelerations on x- and y-axis, respectively. The detailed definitions are as follows.

1) Mean absolute value

$$MAV = \frac{1}{W}\sum_{i=1}^{W}|a[i]| \tag{3}$$

where W is the length of the segmented motion interval.

2) The ratio of MAV_x to MAV_y

$$RMAV = \frac{MAV_x}{MAV_y} \tag{4}$$

3) The absolute value of the difference between MAV_x and MAV_y

$$|\Delta MAV| = |MAV_x - MAV_y| \tag{5}$$

4) The acceleration value of the maximum absolute acceleration point located on the left of the peak

$$AL = a[\arg\max_i |a[i]|] \tag{6}$$

where $i = 1, 2, \cdots, T$, and T is the sampling time of the peak. Similarly, the acceleration value of the maximum absolute acceleration point located on the right of the peak is AR, and $i = T+1, T+2, \cdots, W$.

5) Setting a threshold

$$TH = C\frac{1}{W}\sum_{i=1}^{W}\sqrt{a_x[i]^2 + a_y[i]^2}$$

we get the sign of the maximum absolute acceleration point located on the left of the peak

$$SL = sgn(p) \tag{7}$$

where

$$p = \begin{cases} AL, & \max(|a[i]|) > TH \\ 0, & \max(|a[i]|) \le TH \end{cases}$$

$i = 1, 2, \cdots, T$, and the constant $C = 0.7$. Note that $SL = 0$ when $\max(|a[i]|) \leq TH$. Similarly, we can get SR where $i = T+1, T+2, \cdots, W$.

6) Average jerk [5] between AL and AR

$$AJ = \frac{AR - AL}{\Delta t} \tag{8}$$

where Δt is the duration time between the two points.

7) The sign of AJ

$$SAJ = sgn(AJ) \tag{9}$$

8) The ratio of AJ_x to AJ_y

$$RAJ = \frac{AJ_x}{AJ_y} \tag{10}$$

9) The sign of RAJ

$$SRAJ = sgn(RAJ) \tag{11}$$

10) The difference between AL_x and AL_y

$$\Delta AL = AL_x - AL_y \tag{12}$$

and we can also get the $|\Delta AL|$ and sign of ΔAL

$$SDAL = sgn(\Delta AL) \tag{13}$$

Similarly, the difference between AR_x and AR_y

$$\Delta AR = AR_x - AR_y \tag{14}$$

and also $|\Delta AR|$, $SDAR$.

11) Correlation coefficient between x- and y-axis

$$r = \frac{cov(x, y)}{\sigma_x \sigma_y} \tag{15}$$

where $cov(x, y)$ is the covariance, σ_x and σ_y are the standard deviations of accelerations on x- and y-axis, respectively.

Note that these features are the remainder attributes of the feature selection algorithm. We also tried the features $Mean$, STD, VAR, MAD, and rms presented in [8] and some other features. However, they were not selected by the attribute selection algorithm. After feature extraction, these reduced features will be fed into the FNN classifier.

E. Classifier Construction

FNNs have been used in many applications, such as handwritten digit recognition, email spam detection, and financial prediction. One of the big advantages of FNN is that all learned knowledge is encoded in the weights, this makes the prediction very fast. An FNN consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. In this paper, a three-layer FNN was developed, which consists of an input layer, a hidden layer, and an output layer as shown in Fig. 7. Supposed that N_i, N_h, and N_o are the numbers of nodes in the input, hidden, and output layers, respectively. The output of the kth node in the output layer is expressed as

$$y_k = f(\theta_k + \sum_{h=1}^{N_h} w_{kh} f(\lambda_h + \sum_{i=1}^{N_i} v_{hi} x_i)) \tag{16}$$

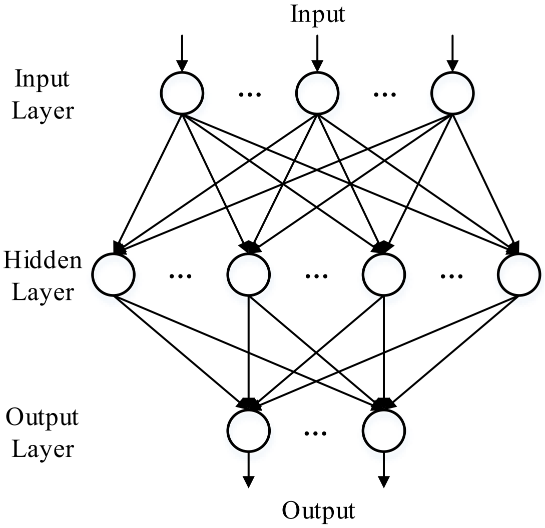


Fig. 7. Topology of a three-layer FNN classifier.

where x_i is the input of the ith node in the input layer, v_{hi} is the connective weight between nodes in the input and hidden layers, w_{kh} represents the weight between nodes in the hidden and output layers, λ_h and θ_k are bias terms, $k = 1, \cdots, N_o$ [18], [19]. Except for the input nodes, each node is a neuron with a nonlinear activation function which in this study is a logistic function, and is described by

$$f(z) = \frac{1}{1 + e^{-z}} \tag{17}$$

Here $f(z)$ is the output of the node which ranges from 0 to 1 and z is the weighted sum of the input synapses. There exists a computational efficient algorithm to train neural networks, the backpropagation (BP) algorithm. For details of the BP learning algorithm please refer to [20]. Once an FNN classifier is trained, it can be used to recognize different basic hand gesture movements.

F. Basic Gesture Encoding

In our system, each basic gesture segmented from a complex gesture is encoded with a 4-bit Johnson code so that it is possible to calculate the similarity between gestures. The corresponding codes of the eight basic gestures are listed in Table I. The gestures are encoded by four binary bits from R (C_0) to LR (C_7). The Hamming distance between two binary codes is defined as the number of different bits. Given two basic gesture codes $C_a = A_3A_2A_1A_0$ and $C_b = B_3B_2B_1B_0$, the Hamming distance between C_a and C_b can be expressed as follows [5]:

$$d(C_a, C_b) = \sum_{k=0}^{3} A_k \oplus B_k \tag{18}$$

where $d(C_a, C_b)$ ranges from 0 to 4. In this study, the Hamming distance is used to represent the dissimilarity between two basic gestures. As shown in Table I, two arbitrary adjacent gesture codes (including the pair of C_0 and C_7) differ in only one bit and incrementally two arbitrary opposite gestures

TABLE I
CORRESPONDING CODES OF THE EIGHT BASIC GESTURES

R	UR	U	UL	L	LL	D	LR
0000	0001	0011	0111	1111	1110	1100	1000

(for example: U and D) differ in four bits. Intuitively, this is consistent with the fact that two adjacent basic gestures are more similar than the opposite ones. The dissimilarity between C_a and C_b is transformed to the normalized similarity as

$$s(C_a, C_b) = 1 - \frac{1}{4} d(C_a, C_b) \tag{19}$$

where $s(C_a, C_b)$ locates in the range of $[0, 1]$, 1 means that the two gestures are identical, i.e., $C_a = C_b$, and 0 indicates completely different, i.e., the two gestures have opposite directions.

G. Similarity Matching

As mentioned above, a complex gesture is segmented into several basic gestures; through classification and encoding, a basic gesture sequence $X_1, \cdots, X_N$ is generated. The gesture sequence is then compared with the standard template sequences (see Table II) to identify the most similar gesture. Detailedly, calculate the similarities between the input sequence $X_1, \cdots, X_N$ and the ith template sequence $Y_{i1}, \cdots, Y_{iN}$, which is expressed as

$$S_i = \frac{1}{N} \sum_{j=1}^{N} s(X_j, Y_{ij}) \tag{20}$$

where S_i is the overall similarity between the input complex gesture and template gesture i. Note that only the gestures which have the same length with the input gesture participate in calculation. The recognized gesture (RG) is

$$RG = \arg\max_i S_i \tag{21}$$

i.e., the most similar gesture is selected as the output one.

IV. EXPERIMENTAL RESULTS

In this section, the effectiveness of the proposed hand gesture recognition algorithm is validated by the following two experiments: 1) basic gesture recognition and 2) complex gesture recognition. We conducted the user-dependent and user-independent experiments for both basic and complex gesture recognitions to test the user dependency of the system. The gesture recognition algorithm is composed of acceleration acquisition and signal preprocessing, gesture segmentation, feature extraction, classifier construction, basic gesture encoding, and similarity matching. Note that for basic gesture recognition, the procedures do not include basic gesture encoding and similarity matching, the two procedures are for complex gesture recognition only. We use Weka [21] software to train the FNN model and classify the basic gestures. The other parts of the gesture recognition algorithm are programmed by MATLAB and C++. The acceleration signals of the two experiments were collected from five subjects (four males and one female). Each participant was invited to repeat each basic gesture 20 times and each complex gesture 10 times, totaling 800 (8 × 5 × 20) basic gestures and 800 (16 × 5 × 10) complex gestures. The subjects were asked to hold the pen-type sensing device vertically to perform the gestures depicted in Fig. 3 and 4 in 3-D space, with positive x and y directions pointing to the left and down, respectively. Due to individual differences, gestures were made in different speeds, intensities, and sizes by different persons.

TABLE III
COMPARISON OF BASIC GESTURE RECOGNITION ACCURACIES (%)

	R	UR	U	UL	L	LL	D	LR
User-Dependent	99	100	100	100	100	100	100	100
User-Independent	97	100	100	100	98	100	99	97

A. Basic Gesture Recognition

1) User-Dependent Experiments: The user-dependent testing assesses the system performance using some of a user's signals while the other signals of the user are employed to train the classifier. We adopted the 5-fold cross-validation method to evaluate the performance of the FNN classifier. In the evaluation, 20 trajectories of each basic gesture from all participants were selected randomly to form the test set, and the rest of the trajectories were considered to be the training set. Therefore, the total training and test samples were 640 (8 × 80) and 160 (8 × 20), respectively. The detailed user-dependent basic gesture recognition results are listed in Table III, which shows the recognition accuracy corresponding to each gesture. Since every gesture has 100 samples, it is also the number of correctly recognized gestures. The overall average recognition rate is 99.88%.

2) User-Independent Experiments: The user-independent testing could simulate the most common scenario with training and test data from different users. 5-fold cross-validation was conducted: trajectories from one of the 5 participants were considered to be the test set, and trajectories from the other 4 participants were used to form the training set. Thus the total training and test samples were 640 (8 × 4 × 20) and 160 (8 × 1 × 20), respectively. The detailed user-independent basic gesture recognition results are listed in Table III, and the average recognition accuracy is 98.88%. Since the basic gesture recognition is the foundation of the complex gesture recognition, the high accuracies of both user-dependent and user-independent basic gesture recognitions provide a strong guarantee for the complex gesture recognition.

B. Complex Gesture Recognition

1) User-Dependent Experiments: For user-dependent complex gesture recognition, the whole 800 basic gesture samples were used to form the training set. Since a complex gesture is segmented into several basic gestures, totaling 2050 basic gesture samples are generated from the original 800 complex gestures and considered to be the test set. We call these

TABLE II
BASIC GESTURE SEQUENCES OF THE COMPLEX GESTURES

1	2	3	4	5	6	7	8
L-R	R-L	U-D	D-U	D-R	R-D	D-UR	LL-R
9	10	11	12	13	14	15	16
L-D-R	R-D-L	U-LR-U	R-LL-R	LL-U-LR	LR-U-LL	D-R-UL	U-R-D-L

TABLE IV
CONFUSION MATRIX FOR THE USER-DEPENDENT DERIVED BASIC GESTURE RECOGNITION

	R	UR	U	UL	L	LL	D	LR
R	547							3
UR		50						
U		6	344					
UL				50				
L				3	247			
LL					1	198	1	
D							449	1
LR							5	145

TABLE VI
CONFUSION MATRIX FOR THE USER-INDEPENDENT DERIVED BASIC GESTURE RECOGNITION

	R	UR	U	UL	L	LL	D	LR
R	534							16
UR		50						
U		8	342					
UL			1	49				
L				8	242			
LL					2	197	1	
D							449	1
LR							4	146

basic gestures as the *derived basic gestures*. The confusion matrix for the user-dependent derived basic gesture recognition is shown in Table IV. The average recognition accuracy is 99.02%. After classification by the FNN, the recognized derived basic gestures are then encoded by Johnson codes, and matched with the standard template sequences shown in Table II by comparing the similarity. The most similar complex gesture is regarded as the output one. The recognition accuracy of each complex gesture is listed in Table V. We note that only one is incorrectly recognized and the average recognition accuracy is 99.88%.

2) User-Independent Experiments: For user-independent complex gesture recognition, 5-fold cross-validation was conducted: complex gesture repetitions from one of the 5 participants were considered to be the test set, and the basic gesture repetitions from the other 4 participants were used to form the training set. Thus the total training samples were 640 (8 × 4 × 20) basic gestures, and the total test samples were 160 (16 × 1 × 10) complex gestures (i.e., 410 derived basic gestures). The recognition procedure is similar with *User-Dependent Experiments*. The confusion matrix for the user-independent derived basic gesture recognition is shown in Table VI and the average recognition accuracy is 98%. The user-independent recognition accuracy of each complex gesture is listed in Table V.

The accuracies for user-dependent and user-independent derived basic gesture recognition are shown in Fig. 8 and a comparison of average basic gesture recognition accuracies are listed in Table VII. For both user-dependent and user-independent classifications, the average recognition accuracies of derived basic gestures are lower than those of basic gestures, which is due to the interferences between derived basic gestures in a complex gesture. We can also see that the recognition accuracies in the user-independent classifications are slightly lower than the ones in the user-dependent classifications due to individual differences. Note that the incorrectly recognized derived basic gestures are all classified as adjacent gestures as shown in Table IV and VI. Through similarity matching, the complex gesture recognition accuracies for user-dependent and user-independent classifications (shown in Table V) are identical in our experiments, hence we conclude that our system is user-independent. Though some derived basic gestures are incorrectly recognized, the similarity matching obtains the effect of sequence error correction and improves the fault tolerance of the system.

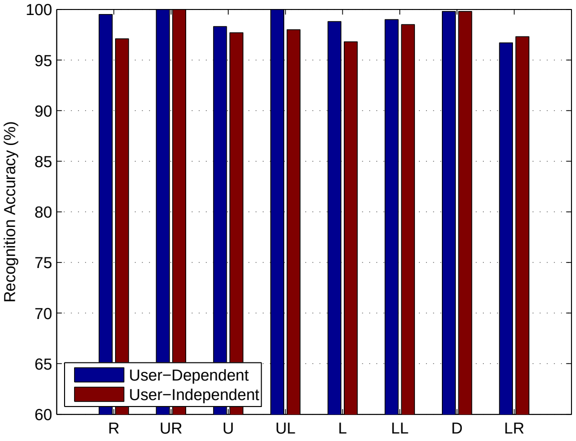


Fig. 8. Comparison of the derived basic gesture recognition accuracies.

Table VIII shows a comparison of performance between our proposed algorithm (Feedforward Neural Network and Similarity Matching, FNN & SM), Dynamic Time Warping (DTW) [13], Dynamic Time Warping and Affinity Propagation (DTW & AP) [1], Probabilistic Neural Network (PNN) [8],

TABLE V
COMPARISON OF COMPLEX GESTURE RECOGNITION ACCURACIES (%)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
User-Dependent	100	100	100	98	100	100	100	100	100	100	100	100	100	100	100	100
User-Independent	100	100	100	98	100	100	100	100	100	100	100	100	100	100	100	100

TABLE VII
COMPARISON OF BASIC GESTURE RECOGNITION ACCURACIES FOR USER-DEPENDENT AND USER-INDEPENDENT CLASSIFICATIONS

	User-Dependent	User-Independent
Basic Gesture	99.88	98.88
Derived Basic Gesture	99.02	98

and Sign Sequence and Template Matching (SSTM) [16]. These systems are all based on a triaxial accelerometer except for DTW, which employs an accelerometer, a gyroscope and a magnetometer. Note that though the performances of user-dependent classifications of these algorithms are very close, the user-independent recognition accuracy of our proposed algorithm outperforms the others. Furthermore, since a complex gesture is composed of several derived basic gestures, the users can easily define and add their own gestures as long as the gestures can be represented as sequences of basic gestures. Thus our system is of good extendibility. The experimental results successfully validated the effectiveness of the hand gesture recognition algorithm which is based on feedforward neural network and similarity matching. The gestures in the dictionary can be used as various commands for HCIs.

V. CONCLUSION

This paper has presented an accelerometer-based pen-type sensing device as well as an FNN and SM based hand gesture recognition algorithm. The proposed recognition algorithm consists of data acquisition and signal preprocessing, gesture segmentation, feature extraction, classifier construction, basic gesture encoding, and similarity matching. An effective segmentation algorithm is developed to identify individual basic gesture motion intervals automatically. Through segmentation, each complex gesture is segmented into several basic gestures. Based on the kinematics characteristics of the basic gesture, 25 features are extracted to construct the FNN model. For basic gesture recognition, the input gestures are classified directly by the FNN classifier. However, the input complex gestures go through an additional similarity matching procedure to identify the most similar gestures. The user-dependent and user-independent recognition accuracies for the basic gestures are 99.88% and 98.88%, respectively. In addition, the user-dependent and user-independent recognition accuracies for the complex gestures are both 98.88%. The system achieves perfect user-dependent and user-independent recognition accuracies for both basic and complex gestures.

REFERENCES

[1] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," *Signal Processing, IEEE Transactions on*, vol. 59, no. 12, pp. 6197–6205, Dec 2011.

[2] T. Lu, "A motion control method of intelligent wheelchair based on hand gesture recognition," in *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on*, June 2013, pp. 957–962.

[3] C. Zhu and W. Sheng, "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 41, no. 3, pp. 569–573, May 2011.

[4] S. Lian, W. Hu, and K. Wang, "Automatic user state recognition for hand gesture based low-cost television control system," *Consumer Electronics, IEEE Transactions on*, vol. 60, no. 1, pp. 107–115, February 2014.

[5] R. Xie, X. Sun, X. Xia, and J. Cao, "Similarity matching-based extensible hand gesture recognition," *Sensors Journal, IEEE*, vol. 15, no. 6, pp. 3475–3483, June 2015.

[6] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," *Multimedia, IEEE Transactions on*, vol. 17, no. 1, pp. 29–39, Jan 2015.

[7] X. Wu, X. Mao, L. Chen, and Y. Xue, "Trajectory-based view-invariant hand gesture recognition by fusing shape and orientation," *Computer Vision, IET*, vol. 9, no. 6, pp. 797–805, 2015.

[8] J.-S. Wang and F.-C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 7, pp. 2998–3007, July 2012.

[9] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and emg sensors," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 41, no. 6, pp. 1064–1076, Nov 2011.

[10] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *Human-Machine Systems, IEEE Transactions on*, vol. 44, no. 2, pp. 293–299, April 2014.

[11] S. Zhou, F. Fei, G. Zhang, J. Mai, Y. Liu, J. Liou, and W. Li, "2d human gesture tracking and recognition by the fusion of mems inertial and vision sensors," *Sensors Journal, IEEE*, vol. 14, no. 4, pp. 1160–1170, April 2014.

[12] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, "Fusion of inertial and depth sensor data for robust hand gesture recognition," *Sensors Journal, IEEE*, vol. 14, no. 6, pp. 1898–1903, June 2014.

[13] Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, "An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition," *Sensors Journal, IEEE*, vol. 15, no. 1, pp. 154–163, Jan 2015.

[14] S. Hussain and A. Rashid, "User independent hand gesture recognition by accelerated dtw," in *Informatics, Electronics Vision (ICIEV), 2012 International Conference on*, May 2012, pp. 1033–1037.

[15] D. Frolova, H. Stern, and S. Berman, "Most probable longest common subsequence for recognition of gesture character input," *Cybernetics, IEEE Transactions on*, vol. 43, no. 3, pp. 871–880, June 2013.

[16] R. Xu, S. Zhou, and W. Li, "Mems accelerometer based nonspecific-user hand gesture recognition," *Sensors Journal, IEEE*, vol. 12, no. 5, pp. 1166–1173, May 2012.

[17] M. Abid, E. Petriu, and E. Amjadian, "Dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar," *Instrumentation and Measurement, IEEE Transactions on*, vol. 64, no. 3, pp. 596–605, March 2015.

[18] C.-H. Chen, J.-C. Wu, and J.-H. Chen, "Prediction of flutter derivatives by artificial neural networks," *Journal of wind engineering and industrial aerodynamics*, vol. 96, no. 10, pp. 1925–1937, 2008.

[19] L. Zhang, H. Wang, J. Liang, and J. Wang, "Decision support in cancer base on fuzzy adaptive pso for feedforward neural network training," in

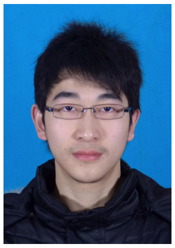
TABLE VIII
COMPARISON OF PERFORMANCE OF SEVERAL DIFFERENT ACCELEROMETER-BASED GESTURE RECOGNITION ALGORITHMS

Algorithm	No. of Gestures	Accuracy (%)	
		User-Dependent	User-Independent
Proposed Algorithm	16	99.88	99.88
DTW	8	99.8	98.1
DTW & AP	18	99.81	94.60
PNN	8	98.75	–
SSTM	7	–	95.6

Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on, vol. 1, Dec 2008, pp. 220–223.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, p. 3, 1988.

[21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *Acm Sigkdd Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.



Renqiang Xie received the B.Eng. degree in electronic information engineering from the School of Computer and Information, Hefei University of Technology, Hefei, China, in 2013. He is currently pursuing the M.E. degree in the Key Laboratory of Terahertz Solid-State Technology, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China, joint training with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China.

His research interests include hand gesture recognition, big data, and machine learning.

Juncheng Cao was born in Jiangxi, China, in 1967. He received the Ph.D. degree in electrical engineering from Southeast University, Nanjing, China, in 1994.

He is currently the Terahertz (THz) Group Leader and the Director of the Key Laboratory of Terahertz Solid-State Technology, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China. From 1999 to 2000, he was a Senior Visiting Scientist with the National Research Council, Ottawa, ON, Canada. He and his group members successfully developed a THz quantum-cascade laser and its applications in THz communication and imaging, a Monte Carlo simulation method for THz quantum-cascade lasers, a THz-radiation induced semiconductor impact ionized model, and also successfully explained the absorption of strong THz irradiation in low-dimensional semiconductors. His current research interests include THz semiconductor devices and their applications.

Dr. Cao was a recipient of the National Fund for Distinguished Young Scholars of China and the Natural Science Award of Shanghai (the Peony Award) in 2004. He received the Excellent Teacher Award from the Chinese Academy of Sciences twice, in 2006 and 2011.