An Accelerometer-Based Digital Pen With a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition

Jeen-Shing Wang, *Member, IEEE*, and Fang-Chen Chuang

***Abstract*—This paper presents an accelerometer-based digital pen for handwritten digit and gesture trajectory recognition applications. The digital pen consists of a triaxial accelerometer, a microcontroller, and an RF wireless transmission module for sensing and collecting accelerations of handwriting and gesture trajectories. The proposed trajectory recognition algorithm composes of the procedures of acceleration acquisition, signal preprocessing, feature generation, feature selection, and feature extraction. The algorithm is capable of translating time-series acceleration signals into important feature vectors. Users can use the pen to write digits or make hand gestures, and the accelerations of hand motions measured by the accelerometer are wirelessly transmitted to a computer for online trajectory recognition. The algorithm first extracts the time- and frequency-domain features from the acceleration signals and, then, further identifies the most important features by a hybrid method: kernel-based class separability for selecting significant features and linear discriminant analysis for reducing the dimension of features. The reduced features are sent to a trained probabilistic neural network for recognition. Our experimental results have successfully validated the effectiveness of the trajectory recognition algorithm for handwritten digit and gesture recognition using the proposed digital pen.**

***Index Terms*—Accelerometer, gesture, handwritten recognition, linear discriminant analysis (LDA), probabilistic neural network (PNN).**

I. Introduction

EXPLOSIVE growth of miniaturization technologies in electronic circuits and components has greatly decreased the dimension and weight of consumer electronic products, such as smart phones and handheld computers, and thus made them more handy and convenient. Due to the rapid development of computer technology, human–computer interaction (HCI) techniques [1]–[3] have become an indispensable component in our daily life. Recently, an attractive alternative, a portable device embedded with inertial sensors, has been proposed to sense the activities of human and to capture his/her motion trajectory information from accelerations for recognizing gestures or handwriting. A significant advantage of inertial sensors for general motion sensing is that they can be operated without any external reference and limitation in working conditions. However, motion trajectory recognition is relatively complicated because different users have different speeds and styles to generate various motion trajectories. Thus, many researchers have tried to narrow down the problem domain for increasing the accuracy of handwriting recognition systems.

Recently, some researchers have concentrated on reducing the error of handwriting trajectory reconstruction by manipulating acceleration signals and angular velocities of inertial sensors [4]–[8]. However, the reconstructed trajectories suffer from various intrinsic errors of inertial sensors. Hence, many researchers have focused on developing effective algorithms for error compensation of inertial sensors to improve the recognition accuracy. To name a few, Yang *et al.* [9] proposed a pen-type input device to track trajectories in 3-D space by using accelerometers and gyroscopes. An efficient acceleration error compensation algorithm based on zero velocity compensation was developed to reduce acceleration errors for acquiring accurate reconstructed trajectory. An extended Kalman filter with magnetometers (micro inertial measurement unit (μIMU) with magnetometers), proposed by Luo *et al.* [10], was employed to compensate the orientation of the proposed digital writing instrument. If the orientation of the instrument was estimated precisely, the motion trajectories of the instrument were reconstructed accurately. Dong *et al.* [11] proposed an optical tracking calibration method based on optical tracking system (OTS) to calibrate 3-D accelerations, angular velocities, and space attitude of handwriting motions. The OTS was developed for the following two goals: 1) to obtain accelerations of the proposed ubiquitous digital writing instrument (UDWI) by calibrating 2-D trajectories and 2) to obtain the accurate attitude angles by using the multiple camera calibration. However, in order to recognize or reconstruct motion trajectories accurately, the aforementioned approaches introduce other sensors such as gyroscopes or magnetometers to obtain precise orientation. This increases additional cost for motion trajectory recognition systems as well as computational burden of their algorithms.

In order to reduce the cost of systems and simplify the algorithms, much research effort has been devoted to extract important features from time-series inertial signals. To name a few, Lim *et al.* [12] computed correlation coefficients of the absolute value of acceleration and the absolute value of the first and second derivatives of acceleration to form feature vectors.

Manuscript received March 14, 2011; revised July 19, 2011; accepted August 31, 2011. Date of publication September 15, 2011; date of current version February 17, 2012. This research was supported by the National Science Council of Taiwan, under grants NSC 100-2628-E-006-206.

The authors are with the Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: jeenshin@mail.ncku.edu.tw; nonochuang@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2011.2167895

They then applied principal component analyis (PCA) and Fisher linear discriminant to reduce the dimension of the feature vectors. With the reduced features, a time-lagged feedforward network was trained to recognize 2-D handwriting gestures and the best performance with an overall accuracy of 95%. In [13], the acceleration, velocity, and position features were generated from raw acceleration signals, and then, the PCA was utilized to reduce the feature dimension size. They successfully employed a hidden Markov model (HMM) with dynamic time warping algorithms to recognize 3-D handwriting digits with a recognition rate of 90%. Krishnan *et al.* [14] calculated the time- and frequency-domain features (such as mean, variance (VAR), correlation, spectral entropy, and spectral energy) of the acceleration signals measured from the accelerometers worn simultaneously on different positions of a participant's hand. Subsequently, the AdaBoost, HMM, and k-NN classifiers were utilized to classify hand motions, and the AdaBoost classifier achieved the best performance with an overall accuracy of 86%.

In this paper, we developed a pen-type portable device and a trajectory recognition algorithm. The pen-type portable device consists of a triaxial accelerometer, a microcontroller, and an RF wireless transmission module. The acceleration signals measured from the triaxial accelerometer are transmitted to a computer via the wireless module. Users can utilize this digital pen to write digits and make hand gestures at normal speed. The measured acceleration signals of these motions can be recognized by the trajectory recognition algorithm. The recognition procedure is composed of acceleration acquisition, signal preprocessing, feature generation, feature selection, and feature extraction. The acceleration signals of hand motions are measured by the pen-type portable device. The signal preprocessing procedure consists of calibration, a moving average filter, a high-pass filter, and normalization. First, the accelerations are calibrated to remove drift errors and offsets from the raw signals. These two filters are applied to remove high-frequency noise and gravitational acceleration from the raw data, respectively. The features of the preprocessed acceleration signals of each axis include mean, correlation among axes, interquartile range (IQR), mean absolute deviation (MAD), root mean square (rms), VAR, standard deviation (STD), and energy. Before classifying the hand motion trajectories, we perform the procedures of feature selection and extraction methods. In general, feature selection aims at selecting a subset of size m from an original set of d features $(d > m)$. Therefore, the criterion of kernel-based class separability (KBCS) with best individual N (BIN) is to select significant features from the original features (i.e., to pick up some important features from d) and that of linear discriminate analysis (LDA) is to reduce the dimension of the feature space with a better recognition performance (i.e., to reduce the size of m). The objective of the feature selection and feature extraction methods is not only to ease the burden of computational load but also to increase the accuracy of classification. The reduced features are used as the inputs of classifiers. In this paper, we adopted a probabilistic neural network (PNN) as the classifier for handwritten digit and hand gesture recognition. The contributions of this paper include the following: 1) the development of a portable digital pen with a trajectory recognition algorithm, i.e., with the digital pen, users can deliver diverse commands by hand motions to control electronics devices anywhere without space limitations, and 2) an effective trajectory recognition algorithm, i.e., the proposed algorithm can efficiently select significant features from the time and frequency domains of acceleration signals and project the feature space into a smaller feature dimension for motion recognition with high recognition accuracy.

The rest of this paper is organized as follows. In Sections II and III, we survey related work on digital pen trajectory recognition and similar HCI applications and introduce the hardware components of the digital pen in detail, respectively. The proposed trajectory recognition algorithm consisting of acceleration acquisition, signal preprocessing, feature generation, feature selection, feature extraction, and a PNN is presented in Section V. In Section VI, simulation results are presented to validate the proposed approach. Finally, conclusions are given in the last section.

II. Related Work

Recently, some studies have focused on the development of digital pens for trajectory recognition and HCI applications. For instance, an alternative method of conventional tablet-based handwriting recognition has been proposed by Milner [15]. In his system, two dual-axis accelerometers are mounted on the side of a pen to generate time-varying x- and y-axis acceleration for handwriting motion. The author employed an HMM with a bandpass filtering and a down-sampling procedure for classification of seven handwritten words. The best recognition rate is 96.2% when the number of states of the HMM is equal to 60. Oh *et al.* [16] presented a wandlike input device embedding a triaxial accelerometer and a triaxial gyroscope for online 3-D character gesture recognition. Fisher discriminant analysis was adopted, and different combinations of sensor signals were used to test the recognition performance of their device. When all six axes raw signals were used as inputs of the recognition system, the recognition rate was 93.23%. In addition, they proposed an ensemble recognizer consisting of three subrecognizers with the following signals as inputs: acceleration, angular velocity, and estimated handwriting trajectory. The recognition rate of the recognizer was 95.04%. Similarly, a gesture recognition system consisting of a gesture input device, a trajectory estimation algorithm, and a recognition algorithm in 3-D space was proposed by Cho *et al.* [17]. The trajectory estimation algorithm based on an inertial navigation system was developed to reconstruct the trajectories of numerical digits and three hand gestures, and then, a Bayesian network was trained to recognize the reconstructed trajectories. The average recognition rate was 99.2%. Zhou *et al.* [18] proposed a μIMU for 2-D handwriting applications. They extracted the discrete cosine transform features from x- and y-axis acceleration signals and one angular velocity and used an unsupervised self-organizing map to classify 26 English alphabets and ten numerical digits. The recognition rate of 26 English alphabets and ten numerical digits achieved 64.38% and 80.8%, respectively.

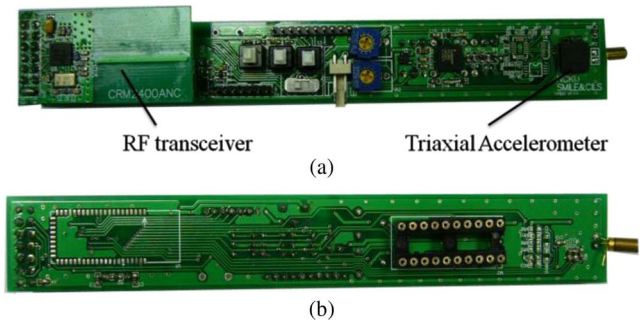

Fig. 1. Digital pen. (a) Front view of the circuit. (b) Back view of the circuit.

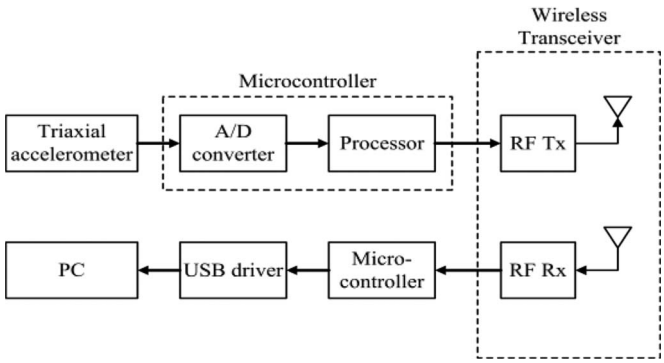

Fig. 2. Schematic diagram of the digital pen module.

III. Hardware Design of Digital Pen

Our digital pen consists of a triaxial accelerometer (LIS3L02AQ3, STMicroelectronics), a microcontroller (C8051F206 with a 12-b A/D converter), and a wireless transceiver (nRF2401, Nordic). The triaxial accelerometer measures the acceleration signals generated by a user's hand motions. The microcontroller collects the analog acceleration signals and converts the signals to digital ones via the A/D converter. The wireless transceiver transmits the acceleration signals wirelessly to a personal computer (PC). The dimension of the pen-type circuit board is 14 cm $\times$ 2 cm $\times$ 1.5 cm as shown in Fig. 1.

The LIS3L02AQ3 is a low-cost capacitive micromachined accelerometer with a temperature compensation function and a g-select function for a full-scale selection of ± 2 g and ± 6 g and is able to measure accelerations over the bandwidth of 1.5 kHz for all axes. The accelerometer's sensitivity is set from -2 g to $+2$ g in this study. The C8051F206 integrates a high-performance 12-b A/D converter and an optimized signal cycle 25-MHz 8-b microcontroller unit (MCU) (8051 instruction set compatible) on a signal chip. The output signals of the accelerometer are sampled at 100 Hz by the 12-b A/D converter. Then, all the data sensed by the accelerometer are transmitted wirelessly to a PC by an RF transceiver at 2.4-GHz transmission band with 1-Mb/s transmission rate. The overall power consumption of the digital pen circuit is 30 mA at 3.7 V. Therefore, if a typical AA battery (2000 mAh at 1.5 V) is used as the power of the system, the system requires three batteries simultaneously, and the lifetime is about 67 h. The schematic diagram of the pen-type portable device is shown in Fig. 2.

IV. Trajectory Recognition Algorithm

The block diagram of the proposed trajectory recognition algorithm consisting of acceleration acquisition, signal preprocessing, feature generation, feature selection, and feature extraction is shown in Fig. 3.

In this paper, the motions for recognition include Arabic numerals and eight hand gestures. The acceleration signals of the hand motions are measured by a triaxial accelerometer and then preprocessed by filtering and normalization. Consequently, the features are extracted from the preprocessed data to represent the characteristics of different motion signals, and the feature selection process based on KBCS picks p features out of the original 24 extracted features. To reduce the computational load and increase the recognition accuracy of the classifier, we utilize LDA to reduce the dimension of the selected features. The reduced feature vectors are fed into a PNN classifier to recognize the motion to which the feature vector belongs. We now introduce the detailed procedure of the proposed trajectory recognition algorithm as follows.

A. Signal Preprocessing

The raw acceleration signals of hand motions are generated by the accelerometer and collected by the microcontroller. Due to human nature, our hand always trembles slightly while moving, which causes certain amount of noise. The signal preprocessing consists of calibration, a moving average filter, a high-pass filter, and normalization. First, the accelerations are calibrated to remove drift errors and offsets from the raw signals. The second step of the signal preprocessing is to use a moving average filter to reduce the high-frequency noise of the calibrated accelerations, and the filter is expressed as

$$y[t] = \frac{1}{N}\sum_{i=1}^{N-1} x[t+i] \tag{1}$$

where $x[t]$ is the input signal, $y[t]$ is the output signal, and N is the number of points in the average filter. In this paper, we set $N = 8$. The decision of using an eight-point moving average filter is based on our empirical tests. From our experimental results, we found that the ideal value of the moving average filter to achieve the best recognition result is eight. Then, we utilize a high-pass filter to remove the gravitational acceleration from the filtered acceleration to obtain accelerations caused by hand movement. In general, the size of samples of each movement between fast and slow writers is different. Therefore, after filtering the data, we first segment each movement signal properly to extract the exact motion interval. Then, we normalize each segmented motion interval into equal sizes via interpolation. Once the preprocessing procedure is completed, the features can be extracted from the preprocessed acceleration signals.

B. Feature Generation

The characteristics of different hand movement signals can be obtained by extracting features from the preprocessed x-,

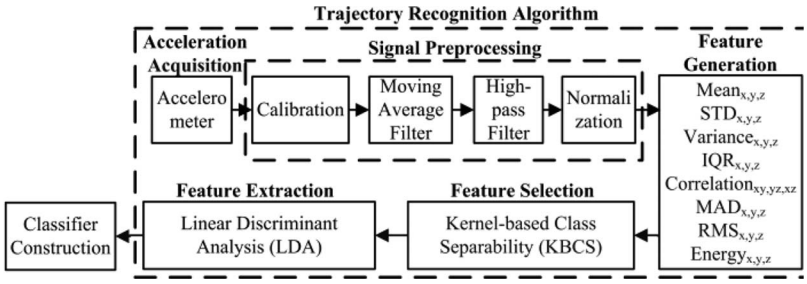

Fig. 3. Block diagram of the trajectory recognition algorithm.

y-, and z-axis signals, and we extract eight features from the triaxial acceleration signals, including mean, STD, VAR, IQR [19], correlation between axes [20], MAD, rms, and energy [21] in this study. They are explicated as follows.

1) Mean: The mean value of the acceleration signals of each hand motion is the dc component of the signal

$$Mean = \frac{1}{|W|}\sum_{i=1}^{|W|} x_i \tag{2}$$

where W is the length of each hand motion.

2) STD: STD is the square root of VAR

$$STD = \sqrt{\frac{1}{|W|-1}\sum_{i=1}^{|W|}(x_i - m)^2}. \tag{3}$$

3) VAR

$$VAR = \frac{1}{|W|-1}\sum_{i=1}^{|W|}(x_i - m)^2 \tag{4}$$

where x_i is the acceleration instance and m is the mean value of x_i in (3) and (4).

4) IQR: When different classes have similar mean values, the interquartile range represents the dispersion of the data and eliminates the influence of outliers in the data.

5) Correlation among axes: The correlation among axes is computed as the ratio of the covariance to the product of the STD for each pair of axes. For example, the correlation $(corr_{xy})$ between two variables x on x-axis and y on y-axis is defined as

$$corr = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y} = \frac{E\left((x-m_x)(y-m_y)\right)}{\sigma_x \sigma_y} \tag{5}$$

where E represents the expected value, $\boldsymbol{\sigma_x}$ and $\boldsymbol{\sigma_x}$ are STDs, and m_x and m_y are the expected values of x and y, respectively. Correlation is a useful feature in discriminating motions that involve translation in only one dimension.

6) MAD

$$MAD = \frac{1}{|W|}\sum_{i=1}^{|W|}|x_i - m|. \tag{6}$$

7) rms

$$rms = \sqrt{\frac{1}{|W|}\sum_{i=1}^{|W|} x_i^2} \tag{7}$$

where x_i is the acceleration instance and m is the mean value of x_i in (6) to (7).

8) Energy: Energy is calculated as the sum of the magnitudes of squared discrete fast Fourier transform (FFT) components of the signal in a window. The equation is defined as

$$Energy = \frac{1}{|W|}\sum_{i=1}^{|W|}|F_i|^2 \tag{8}$$

where F_i is the ith FFT component of the window and $|F_i|$ is the magnitude of F_i.

When the procedure of feature generation is done, 24 features are then generated. Because the amount of the extracted features is large, we adopt KBCS to select most useful features and then use LDA to reduce the dimensions of features.

C. Feature Selection

Feature selection comprises a selection criterion and a search strategy. The adopted selection criterion is the KBCS which is originally developed by Wang [22]. The KBCS can be computed as follows: Let $(\mathbf{x}, y) \in (\mathbb{R}^d \times \boldsymbol{Y})$ represents a sample, where $\mathbb{R}^d$ denotes a d-dimensional feature space, Y symbolizes the set of class labels, and the size of Y is the number of class c. This method projects the samples onto a kernel space, and $\mathbf{m}_i^{\emptyset}$ is defined as the mean vector for the ith class in the kernel space, n_i denotes the number of samples in the ith class, $\mathbf{m}^{\emptyset}$ denotes the mean vector for all classes in the kernel space, $\mathbf{S}_B^{\emptyset}$ denotes the between-class scatter matrix in the kernel space, and $\mathbf{S}_W^{\emptyset}$ denotes the within-class scatter matrix in the kernel space. Let $\emptyset(\cdot)$ be a possible nonlinear mapping from the feature space $\mathbb{R}^d$ to a kernel space κ and $\mathrm{tr}(\mathbf{A})$ represents the trace of a square matrix $\mathbf{A}$. The following two equations are used in the class separability measure:

$$\begin{aligned}\mathrm{tr}\left(\mathbf{S}_B^{\emptyset}\right) &= \mathrm{tr}\left[\sum_{i=1}^{c} n_i \left(\mathbf{m}_i^{\emptyset} - \mathbf{m}^{\emptyset}\right)\left(\mathbf{m}_i^{\emptyset} - \mathbf{m}^{\emptyset}\right)^T\right] \\ &= \sum_{i=1}^{c} n_i \left[\left(m_i^{\emptyset} - m^{\emptyset}\right)\left(m_i^{\emptyset} - m^{\emptyset}\right)^T\right]\end{aligned} \tag{9}$$

$$\operatorname{tr}\left(\mathbf{S}_W^{\emptyset}\right) = \operatorname{tr}\left[\sum_{i=1}^{c}\sum_{j=1}^{n_i}\left(\emptyset(\mathbf{x}_{ij})-\mathbf{m}_i^{\emptyset}\right)\left(\emptyset(\mathbf{x}_{ij})-\mathbf{m}_i^{\emptyset}\right)^T\right]$$
$$= \sum_{i=1}^{c}\sum_{j=1}^{n_i}\left[\left(\emptyset(x_{ij})-m_i^{\emptyset}\right)^T\left(\emptyset(x_{ij})-m_i^{\emptyset}\right)\right]. \quad (10)$$

The class separability in the kernel space can be measured as

$$J^{\emptyset} = \frac{\operatorname{tr}\left(\mathbf{S}_B^{\emptyset}\right)}{\operatorname{tr}\left(\mathbf{S}_W^{\emptyset}\right)}. \quad (11)$$

To maintain the numerical stability in the maximization of $J^{\emptyset}$, the denominator $\operatorname{tr}(\mathbf{S}_W^{\emptyset})$ has to be prevented from approaching zero. In order to maximize class separability, we adopt the BIN as the search strategy. In the BIN, a selection criterion is individually applied to each of the features. The features with larger values of the given criteria are selected.

D. Feature Extraction

For pattern recognition problems, LDA [23] is an effective feature extraction (or dimensionality reduction method) which uses a linear transformation to transform the original feature sets into a lower dimensional feature space. The purpose of LDA is to divide the data distribution in different classes and minimize the data distribution of the same class in a new space. First, two scatter matrices, a within-class scatter matrix S_W and a between-class scatter matrix S_B, are introduced as follows:

$$\mathbf{S}_{Wi} = \frac{1}{n_i}\sum_{j=1}^{n_i}\left(\mathbf{x}_j^{(i)}-\mathbf{m}_i\right)\left(\mathbf{x}_j^{(i)}-\mathbf{m}_i\right)^T \quad (12)$$

$$\mathbf{S}_W = \sum_{i=1}^{N} n_i \times \mathbf{S}_{Wi} = \sum_{i=1}^{N}\sum_{j=1}^{n_i}\left(\mathbf{x}_j^{(i)}-\mathbf{m}_{\boldsymbol{i}}\right)\left(\mathbf{x}_j^{(i)}-\mathbf{m}_{\boldsymbol{i}}\right)^T \quad (13)$$

$$\mathbf{S}_B = \sum_{i=1}^{N} n_i(\mathbf{m}_{\boldsymbol{i}}-\mathbf{m}_{\mathrm{all}})(\mathbf{m}_{\boldsymbol{i}}-\mathbf{m}_{\mathrm{all}})^T \quad (14)$$

where n_i is the number of samples in the ith class and $x_j^{(i)} \in \mathbb{R}^d$ represents the jth sample of the ith class. d is the dimension of the feature space, and n and N are the total numbers of the samples and classes, respectively. $\mathbf{m}_i$ is the mean of the ith class, and $\mathbf{m}_{\mathrm{all}}$ is the mean of all classes. Note that $\mathbf{S}_{Wi}$ is the covariance matrix of the ith class, $\mathbf{S}_W$ is the sum of the covariance matrices, and $\mathbf{S}_B$ is the sum of the squared distances between the mean of each class and the means of all classes.

The fundamental concept of LDA is to maximize the following Fisher criterion to search for the most efficient projection matrix $\mathbf{W}$

$$J(\mathbf{W}) = \frac{\mathbf{w}^T\mathbf{s}_B\mathbf{w}}{\mathbf{w}^T\mathbf{s}_w\mathbf{w}} \quad (15)$$

where $\mathbf{W}^T\mathbf{S}_W\mathbf{W}$ and $\mathbf{W}^T\mathbf{S}_B\mathbf{W}$ are the new within-class scatter and between-class scatter in the new feature space, respectively. That is, in order to achieve maximal discrimination in the new feature space, transformation matrix $\mathbf{W}$ is utilized to maximize the ratio of the between-class distance to the within-class distance. Note that $\mathbf{x}$ is the original feature vector, and the new feature vector $\mathbf{y}$ can be calculated by the equation $\mathbf{y} = \mathbf{W}^T\mathbf{x}$. In general, the optimal solution $\mathbf{W}$ in (16) satisfies the following equation:

$$\mathbf{S}_B\mathbf{W} = \lambda\mathbf{S}_W\mathbf{W}. \quad (16)$$

If the inversion of $\mathbf{S}_W$ exists, $\mathbf{W}$ can be obtained by applying an eigenvalue decomposition of $(\mathbf{S}_W^{-1}\mathbf{S}_B)$. Let k denotes the number of classes for recognition. Since the between-class scatter matrix is the summation of k rank-one matrices and only $k-1$ of them is independent, there exist at most $k-1$ nonzero eigenvalues. Consequently, the upper bound of the dimension of features after transforming is $k-1$, that is, the columns of $\mathbf{W}$ are equal to the eigenvectors corresponding to the first $k-1$ nonzero eigenvalues of (16). After feature extraction, these reduced features will be fed into the PNN classifier to recognize different hand movements.

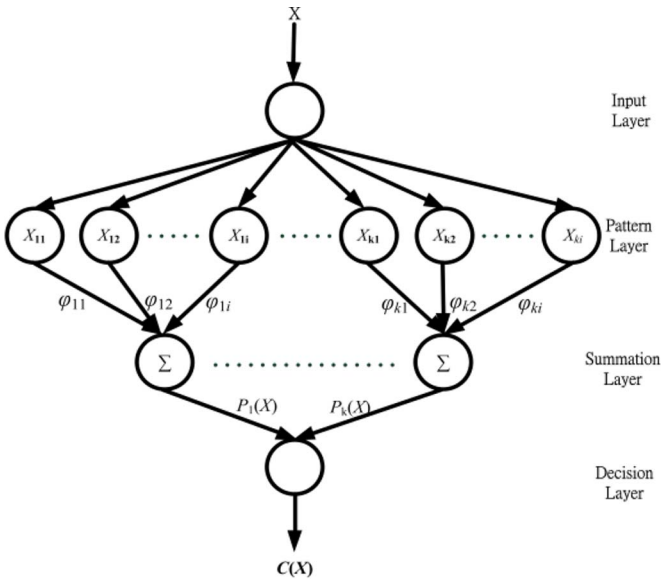

Fig. 4. Topology of a PNN classifier.

E. Classifier Construction

The PNN was first proposed by Specht [24]. With enough training data, the PNN is guaranteed to converge to a Bayesian classifier, and thus, it has a great potential for making classification decisions accurately and providing probability and reliability measures for each classification. In addition, the training procedure of the PNN only needs one epoch to adjust the weights and biases of the network architecture. Therefore, the most important advantage of using the PNN is its high speed of learning. Typically, the PNN consists of an input layer, a pattern layer, a summation layer, and a decision layer as shown in Fig. 4. The function of the neurons in each layer of the PNN is defined as follows.

1) **Layer 1**: The first layer is the input layer, and this layer performs no computation. The neurons of this layer

convey the input features $\mathbf{x}$ to the neurons of the second layer directly

$$\mathbf{x} = [x_1, x_2, \ldots, x_p]^{\mathrm{T}} \tag{17}$$

where p is the number of the extracted features.

2) **Layer 2**: The second layer is the pattern layer, and the number of neurons in this layer is equal to N_L. Once a pattern vector $\mathbf{x}$ from the input layer arrives, the output of the neurons of the pattern layer can be calculated as follows:

$$\varphi_{ki}(x) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp\left(-\frac{(x - x_{ki})^{\mathrm{T}}(x - x_{ki})}{2\sigma^2}\right) \tag{18}$$

where $\mathbf{x}_{ki}$ is the neuron vector, σ is a smoothing parameter, d is the dimension of the pattern vector $\mathbf{x}$, and φ_{ki} is the output of the pattern layer.

3) **Layer 3**: The third layer is the summation layer. The contributions for each class of inputs are summed in this layer to produce the output as the vector of probabilities. Each neuron in the summation layer represents the active status of one class. The output of the kth neuron is

$$p_k(x) = \frac{1}{2\pi^{d/2}\sigma^d}\frac{1}{N_i} \exp\left(-\frac{(x - x_{ki})^{\mathrm{T}}(x - x_{ki})}{2\sigma^2}\right) \tag{19}$$

where N_i is the total number of samples in the kth neuron.

4) **Layer 4**: The fourth layer is the decision layer

$$c(\mathbf{x}) = \arg\max\left\{p_k(\mathbf{x})\right\}, \quad k = 1, 2, \ldots, m \tag{20}$$

where m denotes the number of classes in the training samples and $c(\mathbf{x})$ is the estimated class of the pattern $\mathbf{x}$.

If the *a priori* probabilities and the losses of misclassification for each class are all the same, the pattern $\mathbf{x}$ can be classified according to the Bayes' strategy in the decision layer based on the output of all neurons in the summation layer.

In this paper, the output of the PNN is represented as the label of the desired outcome defined by users. For example, in our handwritten digit recognition, the labels "1," "2," "3," "4," "5," "6," "7," "8," "9," and "10" are used to represent handwriting digits 1, 2, . . ., 9, and 0, respectively.

F. Summary of the Trajectory Recognition Algorithm

We now summarize the proposed trajectory recognition algorithm in the following steps.

Step 1) Acquire the raw acceleration signals from the pen-type accelerometer module.

Step 2) Filter out the high-frequency noise of the raw accelerations by the moving average filter in (1) and then remove the gravity from the filtered accelerations by a high-pass filter. Finally, normalize each segmented motion interval into equal sizes via interpolation.

Step 3) Generate the time- and frequency-domain features from the preprocessed acceleration of each axis including $mean_{x,y,z}$, $STD_{x,y,z}$, $VAR_{x,y,z}$, $IQR_{x,y,z}$, $corr_{xy,yz,xz}$, $MAD_{x,y,z}$, $rms_{x,y,z}$, and $energy_{x,y,z}$.

Step 4) Select significant features by KBCS.

Step 5) Reduce the dimension of the selected features by LDA.

In our study, we assume that an additional button can be used to allow users to indicate the starting point and ending point of motion. That is, the limitation of the proposed trajectory recognition algorithm is that it can only recognize a letter or a number finished with a single stroke. We are currently developing algorithms for letters or words with multistrokes which involve more challenging problems. In the next section, we will focus on the experiment of motion with single stroke.

V. Experimental Results

In this section, the effectiveness of trajectory recognition algorithm is validated by the following two experiments: 1) handwritten digit recognition and 2) gesture recognition. The proposed trajectory recognition algorithm consists of the following procedures: acceleration acquisition, signal preprocessing, feature generation, feature selection, and feature extraction. We collected the acceleration signals of the two experiments from ten subjects (four females, six males; age 24.1 $\pm$ 2.13 years old) in a laboratory environment. We used different combinations of feature selection and extraction methods and employed PNN to recognize handwritten digits and hand gestures. In addition, we compared the recognition results of the PNN trained by the features from different feature engineering methods with those of feedforward neural networks (FNNs). Note that we tested several FNN topologies with different parameter settings. We only reported the best result of FNN in this paper.

A. Handwritten Digit Recognition

We recruited five males and five females (age: 24.3 $\pm$ 1.2 years old) to participate in this experiment. Each participant was asked to hold the digital pen to draw the trajectories of Arabic numerals as shown in Fig. 5, and the pen tip must touch a table. The acceleration signals after the signal preprocessing procedure of the proposed trajectory recognition algorithm for the digit 0 are shown in Fig. 5. The calibrated acceleration signals acquired from the pen-type accelerometer module are shown in Fig. 5(a). Subsequently, the acceleration signals shown in Fig. 5(b) were filtered via the moving average filter to reduce the high-frequency noise. Finally, the gravitational acceleration as shown in Fig. 5(c) was removed from the filtered acceleration signals via a high-pass filter to obtain the accelerations caused by hand movement. With the preprocessed accelerations, 24 features are generated by the feature generation procedure. Subsequently, the KBCS is adopted to choose characteristic features from the generated features. We choose digits 0 and 6 to illustrate the effectiveness of the KBCS, since their accelerations and handwritten trajectories (shown in Figs. 6 and 7) are pretty similar and difficult to classify.

In Fig. 8, we can see that the IQR features of these two digits are closely overlapped. Thus, the features are not effective for

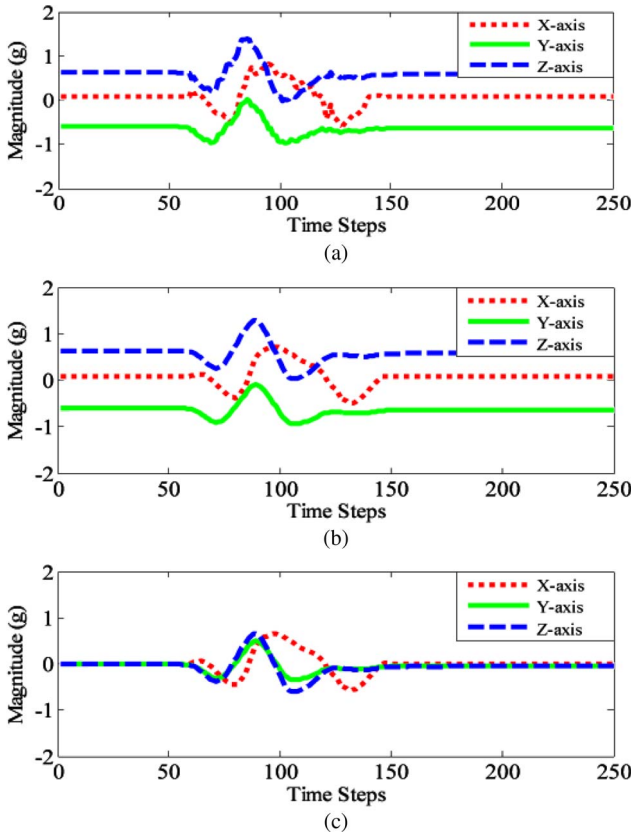

Fig. 5. Digit 0. (a) Calibrated accelerations. (b) Filtered accelerations by a moving average filter. (c) Filtered accelerations by a high-pass filter.

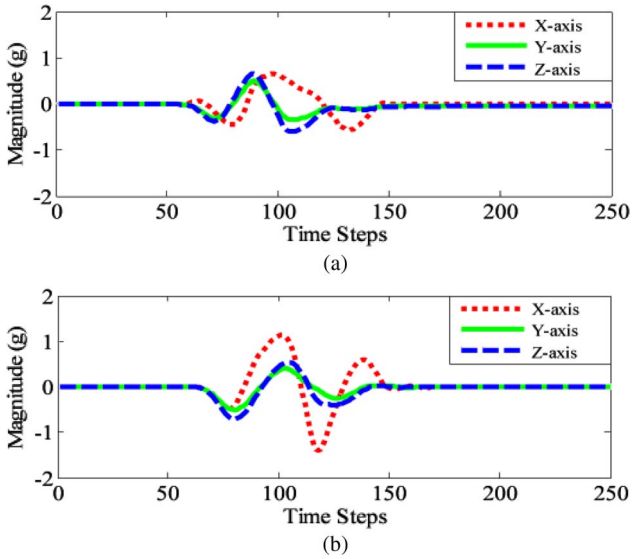

Fig. 6. Preprocessed accelerations. (a) Digit 0. (b) Digit 6.

classifying these two digits. On the contrary, the digits can be well separated by the mean features (shown in Fig. 9) according to their vector cluster distributions. In our experiments, the STD and VAR features are not selected in the final reduced feature set because they are similar in nature. If computational burden for feature selection is concerned, these can be excluded from the candidate features. There were 11 significant features including $corr_{xz}$, $mean_x$, $mean_y$, MAD_x, IQR_x, rms_x,

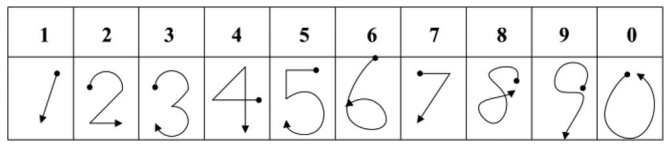

Fig. 7. Pictorial digit trajectories.

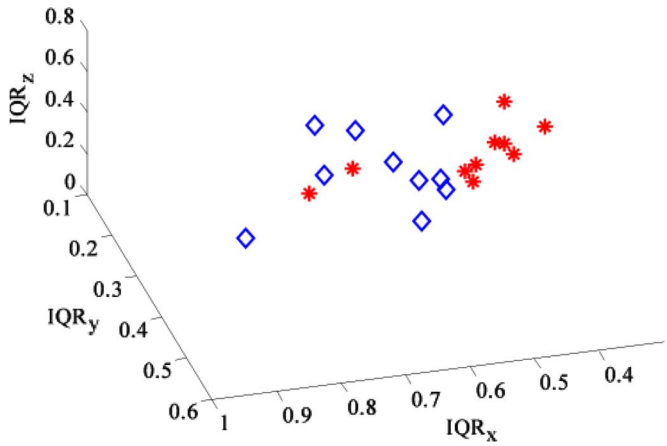

Fig. 8. IQR features of (red star) digit 0 and (blue diamond) digit 6.

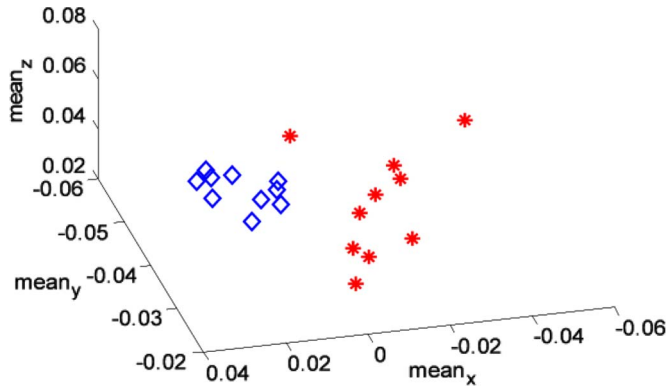

Fig. 9. Mean feature of (red star) digit 0 and digit (blue diamond) 6.

$corr_{xy}$, $mean_z$, $energy_x$, $energy_y$, and $energy_z$ selected from 24 features by the KBCS. Finally, the dimension of the selected features was further reduced to nine by the LDA not only to ease the burden of computational load but also to increase the accuracy of classification.

The participants were asked to repeat each digit for ten times. Therefore, a total of 1000 $(= 10 \times 10 \times 10)$ handwritten digit trajectories was generated. We adopted the leave-one-out cross-validation method to evaluate the recognition performance of the PNN. In the evaluation, we selected one of the trajectories of each digit from each participant randomly for the testing procedure, and the rest of the trajectories were used for the training procedure. Therefore, the total testing samples were 100 $(10 \times 10 \times 1)$ for the testing procedure, and the total training samples were 900 $(10 \times 10 \times 9)$ for the training procedure.

Because there are ten digits needed to be classified, the maximum of the dimension of the feature extraction by the LDA was nine. To see the performance variation caused by feature dimensions, we varied the dimensions of the LDA from one to nine. In Fig. 10, the best average recognition rate of

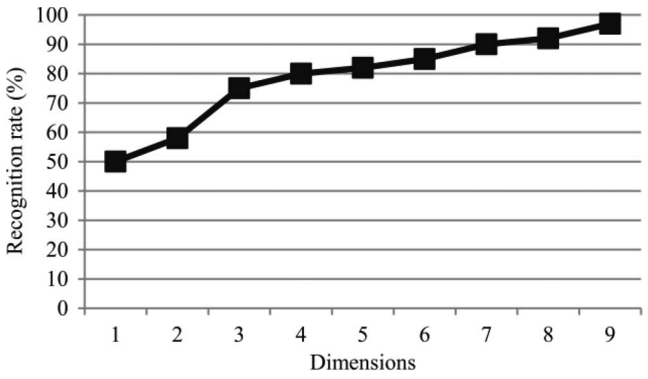

Fig. 10. Average recognition rates versus the feature dimensions of the PNN classifier by using the LDA.

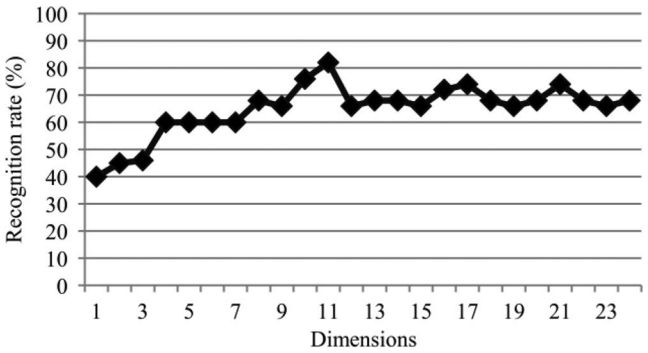

Fig. 11. Average recognition rates versus the feature dimensions of the PNN classifier by using the KBCS.

TABLE I
RECOGNITION RATES FOR DIFFERENT COMBINATIONS OF FEATURE SELECTION AND EXTRACTION METHODS WITH PNN

Feature selection method	KBCS	×	KBCS
Feature extraction method	×	LDA	LDA
Recognition rate (%)	82	97	98
Dimensions	11	9	9

the PNN classifier was 97% when the feature dimension was set to nine by using the LDA. The procedure to evaluate the recognition performance of the KBCS is the same as that of the LDA. The average recognition rates of the PNN classifier using different features selected by the KBCS are shown in Fig. 11. The best recognition rate was 82% when the feature dimension was set to 11, and there was no further improvement for larger numbers of dimensions.

Furthermore, we combined the KBCS with the LDA to reduce the feature dimension. Eleven out of 24 features were first selected by the KBCS. Subsequently, the selected features were used as the inputs of the LDA, and the feature dimension was further reduced by the LDA to nine. Table I shows the average recognition rates of different combinations of feature selection and extraction methods using a leave-one-out cross validation. From Table I, the feature selection method (KBCS+LDA) with the PNN classifier obtained the best recognition performance. The confusion matrix for the recognition algorithm by using the PNN classifier is in Table II. The total number of neurons

TABLE II
CONFUSION MATRIX OBTAINED BY PNN CLASSIFIER FOR HANDWRITING DIGIT RECOGNITION

Classified as	0	1	2	3	4	5	6	7	8	9
0	98	0	1	0	0	1	0	0	0	0
1	1	98	0	0	0	0	0	1	0	0
2	0	0	98	1	0	0	0	1	0	0
3	0	0	0	98	0	1	0	0	1	0
4	0	0	0	0	99	0	0	1	0	0
5	0	0	0	3	0	96	1	0	0	0
6	0	0	0	1	1	0	98	0	0	0
7	0	0	0	0	1	0	0	98	0	1
8	1	0	0	1	0	0	0	0	98	0
9	0	0	0	0	1	0	0	0	0	99

TABLE III
RECOGNITION RATES FOR DIFFERENT COMBINATIONS OF FEATURE SELECTION AND EXTRACTION METHODS WITH FNN

Feature selection method	KBCS	×	KBCS
Feature extraction method	×	LDA	LDA
Recognition rate (%)	77	94	96
Dimensions	11	9	9

TABLE IV
TOTAL CPU RUNTIMES (IN SECONDS) FOR TRAINING AND TESTING PHASES USING COMBINATIONS OF FEATURE SELECTION AND EXTRACTION METHODS WITH PNN

Method	PNN	KBCS+PNN	LDA+PNN	KBCS+ LDA+PNN
Feature dimensions	24	11	9	9
Total CPU time	16.156	16.078	15.156	15.125
Percentage of improvement over only PNN	--	0.48%	6.19%	6.38%

(CPU: Intel Core 2 Duo CPU 2.93 GHz, RAM: 2 G)

in the hidden layer of the PNN is 900, and the average training time is 0.055 s using a PC with an Intel Core 2 DUO CPU at 2.93 GHz and 2 G memory. We also compared the PNN with FNN classifiers for this experiment. The performance evaluations of the FNN are shown in Table III. The total number of neurons in the hidden layer of the FNN is 30; however, the average training time is 20.94 s using the backpropagation learning for the weight training. Obviously, the recognition rate of the proposed recognition algorithm (KBCS+LDA+PNN classifier) outperformed the FNN classifier. In order to demonstrate the efficiency of the proposed trajectory recognition algorithm, we use the total CPU runtimes for executing different combinations of feature methods in Tables IV and V that demonstrate that the feature selection and extraction methods can ease the burden of computational load and increase the accuracy of classification.

B. Gesture Recognition

In the second experiment, the ten participants were asked to hold the pen to perform eight hand gestures in a 3-D space.

TABLE V
TOTAL CPU RUNTIMES (IN SECONDS) FOR TRAINING AND TESTING PHASES USING COMBINATIONS OF FEATURE SELECTION AND EXTRACTION METHODS WITH FNN

Method	FNN	KBCS+FNN	LDA+FNN	KBCS+ LDA+FNN
Feature dimensions	24	11	9	9
Total CPU time	227.172	210.703	206.484	201.625
Percentage of improvement over only FNN	--	7.25%	9.11%	11.25%

(CPU: Intel Core 2 Duo CPU 2.93 GHz, RAM: 2 G)

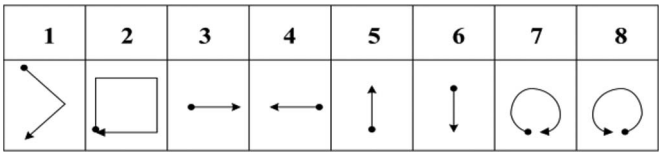

Fig. 12. Trajectories of eight hand gestures.

TABLE VI
RECOGNITION RATES BY THE PNN AND FNN CLASSIFIERS

Classifier	PNN	FNN
Recognition rate (%)	98.75	96.25

TABLE VII
CONFUSION MATRIX OBTAINED BY PNN CLASSIFIER FOR GESTURE RECOGNITION

Classified as	1	2	3	4	5	6	7	8
1	100	0	0	0	0	0	0	0
2	0	98	0	0	0	0	2	0
3	1	0	99	0	0	0	0	0
4	0	0	0	100	0	0	0	0
5	1	0	0	0	98	1	0	0
6	1	0	0	0	1	98	0	0
7	0	1	0	0	0	0	99	0
8	1	0	0	0	0	0	1	98

The gestures are shown in Fig. 12. The subjects were asked to repeat each of the hand gestures for ten times. Therefore, a total of 800 $(= 8 \times 10 \times 10)$ gesture trajectories was generated. The same validation procedure as that of the first experiment was conducted for the gesture trajectories.

Table VI exhibits the average recognition rates of the PNN and FNN classifiers using KBCS+LDA feature selection and extraction methods. The KBCS selected 12 out of 24 features, and the LDA further reduced the feature dimension to seven. In this experiment, 12 significant features including $mean_x$, $corr_{xz}$, $corr_{yz}$, $mean_z$, $mean_y$, MAD_z, MAD_y, IQR_z, IQR_x, MAD_x, rms_z, and rms_x were selected by the KBCS. Table VI shows that the average recognition rate of the PNN (98.75%) outperforms that of the FNN (96.25%). The confusion matrix for the recognition algorithm by using the PNN is shown in Table VII. From the aforementioned two experiments, the proposed recognition algorithm (KBCS+LDA+PNN classifier) can effectively recognize different hand trajectories that can be defined as various commands for HCIs, such as game controller, TV remote control, and presentation pointer with motion recognition capability.

VI. CONCLUSION

This paper has presented a systematic trajectory recognition algorithm framework that can construct effective classifiers for acceleration-based handwriting and gesture recognition. The proposed trajectory recognition algorithm consists of acceleration acquisition, signal preprocessing, feature generation, feature selection, and feature extraction. With the reduced features, a PNN can be quickly trained as an effective classifier. In the experiments, we used 2-D handwriting digits and 3-D hand gestures to validate the effectiveness of the proposed device and algorithm. The overall handwritten digit recognition rate was 98%, and the gesture recognition rate was also 98.75%. This result encourages us to further investigate the possibility of using our digital pen as an effective tool for HCI applications.

REFERENCES

[1] E. Sato, T. Yamaguchi, and F. Harashima, "Natural interface using pointing behavior for human–robot gestural interaction," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 1105–1112, Apr. 2007.
[2] Y. S. Kim, B. S. Soh, and S.-G. Lee, "A new wearable input device: SCURRY," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1490–1499, Dec. 2005.
[3] A. D. Cheok, Y. Qiu, K. Xu, and K. G. Kumar, "Combined wireless hardware and real-time computer vision interface for tangible mixed reality," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 2174–2189, Aug. 2007.
[4] Z. Dong, U. C. Wejinya, and W. J. Li, "An optical-tracking calibration method for MEMS-based digital writing instrument," *IEEE Sens. J.*, vol. 10, no. 10, pp. 1543–1551, Oct. 2010.
[5] J. S. Wang, Y. L. Hsu, and J. N. Liu, "An inertial-measurement-unit-based pen with a trajectory reconstruction algorithm and its applications," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3508–3521, Oct. 2010.
[6] S.-H. P. Won, W. W. Melek, and F. Golnaraghi, "A Kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1787–1798, May 2010.
[7] S.-H. P. Won, F. Golnaraghi, and W. W. Melek, "A fastening tool tracking system using an IMU and a position sensor with Kalman filters and a fuzzy expert system," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1782–1792, May 2009.
[8] Y. S. Suh, "Attitude estimation by multiple-mode Kalman filters," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1386–1389, Jun. 2006.
[9] J. Yang, W. Chang, W. C. Bang, E. S. Choi, K. H. Kang, S. J. Cho, and D. Y. Kim, "Analysis and compensation of errors in the input device based on inertial sensors," in *Proc. IEEE Int. Conf. Inf. Technol.—Coding and Computing*, 2004, pp. 790–796.
[10] Y. Luo, C. C. Tsang, G. Zhang, Z. Dong, G. Shi, S. Y. Kwok, W. J. Li, P. H. W. Leong, and M. Y. Wong, "An attitude compensation technique for a MEMS motion sensor based digital writing instrument," in *Proc. IEEE Int. Conf. Nano/Micro Eng. Mol. Syst.*, 2006, pp. 909–914.
[11] Z. Dong, G. Zhang, Y. Luo, C. C. Tsang, G. Shi, S. Y. Kwok, W. J. Li, P. H. W. Leong, and M. Y. Wong, "A calibration method for MEMS inertial sensors based on optical tracking," in *Proc. IEEE Int. Conf. Nano/Micro Eng. Mol. Syst.*, 2007, pp. 542–547.
[12] J. G. Lim, S. Y. Kim, and D. S. Kwon, "Pattern recognition-based real-time end point detection specialized for accelerometer signal," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2009, pp. 203–208.
[13] S. D. Choi, A. S. Lee, and S. Y. Lee, "On-line handwritten character recognition with 3D accelerometer," in *Proc. IEEE Int. Conf. Inf. Acquisition*, 2006, pp. 845–850.
[14] N. C. Krishnan, C. Juillard, D. Colbry, and S. Panchanathan, "Recognition of hand movements using wearable accelerometers," *J. Ambient Intell. Smart Environ.*, vol. 1, no. 2, pp. 143–155, Apr. 2009.
[15] B. Milner, "Probabilistic neural networks," in *Proc. Inst. Elect. Eng.—Colloq. Doc. Image Process. Multimedia*, 1999, pp. 5/1–5/6.
[16] J. K Oh, S. J. Cho, W. C. Bang, W. Chang, E. Choi, J. Yang, J. Cho, and D. Y. Kim, "Inertial sensor based recognition of 3-D characters gestures

with an ensemble of classifiers," in *Proc. IEEE 9th Int. Workshop Frontiers Handwriting Recognit.*, 2004, pp. 112–117.
[17] S. J. Cho, J. K. Oh, W. C. Bang, W. Chang, E. Choi, Y. Jing, J. Cho, and D. Y. Kim, "Magic wand: A hand-drawn gesture input device in 3-D space with inertial sensors," in *Proc. IEEE 9th Int. Workshop Frontiers Handwriting Recognit.*, 2004, pp. 106–111.
[18] S. Zhou, Z. Dong, W. J. Li, and C. P. Kwong, "Hand-written character recognition using MEMS motion sensing technology," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2008, pp. 1418–1423.
[19] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 3, pp. 871–879, Mar. 2009.
[20] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," *Pervasive*, *Lecture Notes in Computer Science*, no. 3001, pp. 1–17, 2004.
[21] Y. P. Chen, J. Y. Yang, S. N. Liou, G. Y. Lee, and J. S. Wang, "Online classifier construction algorithm for human activity detection using a tri-axial accelerometer," *Appl. Math. Comput.*, pp. 849–860, Nov. 2008.
[22] L. Wang, "Feature selection with kernel class separability," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1534–1546, Sep. 2008.
[23] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.
[24] D. F. Specht, "Probabilistic neural networks," *Neural Netw.*, vol. 3, no. 1, pp. 109–118, 1990.

Jeen-Shing Wang (S'94–M'02) received the B.S. and M.S. degrees in electrical engineering from the University of Missouri, Columbia, in 1996 and 1997, respectively, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 2001.

He is currently an Associate Professor with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. His research interests include computational intelligence, intelligent control, clustering analysis, and optimization.

Fang-Chen Chuang received the B.S. degree in automatic control engineering from the Feng Chia University, Taichung, Taiwan, in 2003 and the M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2005, where she is currently working toward the Ph.D. degree in electrical engineering in the Department of Electrical Engineering.

Her research interests include inertial sensing, handwritten recognition, activity classification, and energy expenditure estimation.