A Hand Gesture Recognition Framework and Wearable Gesture-Based Interaction Prototype for Mobile Devices

Zhiyuan Lu, Xiang Chen, *Member, IEEE*, Qiang Li, Xu Zhang, *Member, IEEE*, and Ping Zhou, *Member, IEEE*

***Abstract*—An algorithmic framework is proposed to process acceleration and surface electromyographic (SEMG) signals for gesture recognition. It includes a novel segmentation scheme, a score-based sensor fusion scheme, and two new features. A Bayes linear classifier and an improved dynamic time-warping algorithm are utilized in the framework. In addition, a prototype system, including a wearable gesture sensing device (embedded with a three-axis accelerometer and four SEMG sensors) and an application program with the proposed algorithmic framework for a mobile phone, is developed to realize gesture-based real-time interaction. With the device worn on the forearm, the user is able to manipulate a mobile phone using 19 predefined gestures or even personalized ones. Results suggest that the developed prototype responded to each gesture instruction within 300 ms on the mobile phone, with the average accuracy of 95.0% in user-dependent testing and 89.6% in user-independent testing. Such performance during the interaction testing, along with positive user experience questionnaire feedback, demonstrates the utility of the framework.**

***Index Terms*—Accelerometer, electromyograghy, gesture recognition, human–computer interaction.**

I. Introduction

Sensing and identifying gestures are two crucial issues to realize gestural user interfaces. The use of camera is an early developed technology to sense gestures, but it has not been applied in most mobile cases due to challenging problems such as changing light and background. Accelerometers and surface electromyography (SEMG) sensors provide another two potential technologies for gesture sensing. Accelerometers can measure accelerations (ACC) from vibrations and the gravity, therefore, they are good at capturing noticeable, large-scale gestures [3]–[6]. SEMG signals, which indicate the activities of related muscles during a gesture execution, have advantages in capturing fine motions such as wrist and finger movements and can be utilized to realize human–computer interfaces [7]–[11]. For example, a commercial gesture input device named MYO [1] is a wireless armband with several SEMG sensors designed for interactions. Various kinds of interaction solutions can be developed using its programming interface.

Manuscript received February 2, 2013; revised August 8, 2013, December 11, 2013 and January 14, 2014; accepted January 22, 2014. Date of publication February 26, 2014; date of current version March 12, 2014. This work was supported in part by Fundamental Research Funds for the Central Universities of China under Grant WK2100230002, the National Nature Science Foundation of China under Grant 61271138, and the Scientific Research Fund of Sichuan Provincial Education Department under Grant 12ZA185. This paper was recommended by Associate Editor C. Cao.

Z. Lu, X. Chen, and X. Zhang are with the Institute of Biomedical Engineering, University of Science and Technology of China, Hefei 230027, China (e-mail: luzhiy@mail.ustc.edu.cn; xch@ustc.edu.cn; xuzhang90@ustc.edu.cn).

Q. Li is with the School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China (e-mail: rightr90@mail.ustc.edu.cn).

P. Zhou is with the Institute of Biomedical Engineering, University of Science and Technology of China, Hefei 230027, China, the Sensory Motor Performance Program, Rehabilitation Institute of Chicago, Chicago, IL 60611 USA, and also with the Department of Physical Medicine & Rehabilitation, Northwestern University, Chicago, IL 60611 USA (e-mail: pzhou@ustc.edu.cn).

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/THMS.2014.2302794

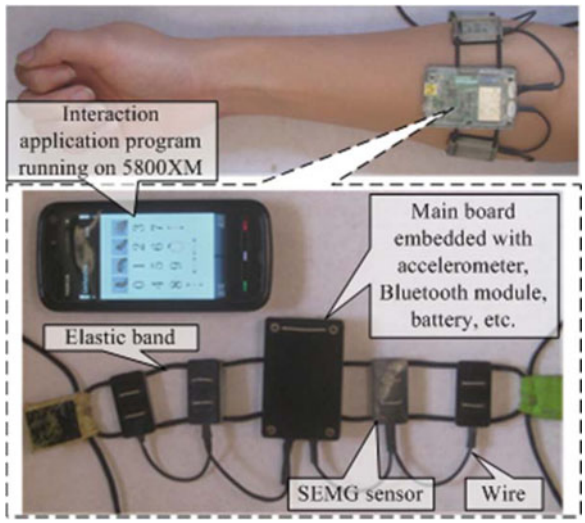


Fig. 1. Gesture-based interaction prototype with the gesture-capturing device designed to be worn around the forearm.

Since both accelerometers and SEMG sensors have their own advantages in capturing hand gestures, the combination of both sensing approaches may improve the performance of hand gesture recognition. Although studies that utilized both SEMG and ACC signals [12]–[14], few combined them to realize a gesture-based interaction system. In our pilot studies [15], [16], a series of promising applications with gestural interfaces relying on portable ACC and SEMG sensors were developed, including sign language recognition and human–computer interaction. We further designed a wearable gesture-capturing device and then realized a gesture-based interface for a mobile phone to demonstrate the feasibility of gesture-based interaction in the mobile application [2]. In that preliminary work, SEMG and ACC signals were not actually fused together in that interface, and only nine gestures were supported.

In this paper, a wearable gesture-based real-time interaction prototype for mobile devices using the fusion of ACC and SEMG signals is presented. As an extension to [2], there are four main contributions.

1) A small, lightweight, and power-efficient wireless wearable device to capture gestures records three-channel ACC and four-channel SEMG signals from forearm.
2) A novel real-time recognition scheme that is based on the fusion of SEMG and ACC signals is proposed. The algorithms are designed to be computationally tractable with high recognition accuracy.
3) An active segmentation scheme, overcoming the difficulties in ACC signal segmentation and the synchronization of active segments in SEMG and ACC signals, is presented.
4) An evaluation with a gesture-based interaction application on a mobile phone demonstrates the feasibility of the proposed interface.

II. System Architecture

This gesture-based interaction prototype enables operating a mobile phone without touching it. It consists of a custom-wearable gesture-capturing device and an interaction application program running on a smart phone (see Fig. 1). Worn on user's forearm, the gesture-capturing device records SEMG and ACC signals, and sends them to the phone through a wireless connection. The interaction application program processes these signals, translates each gesture into instructions, and then provides feedback.

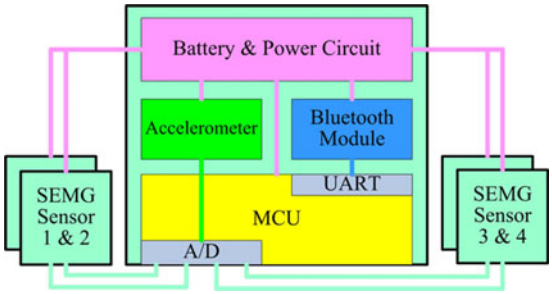


Fig. 2. Architecture of the gesture-capturing device.

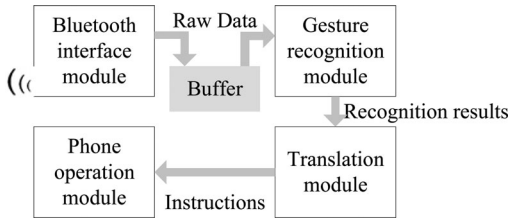


Fig. 3. Architecture of the interaction application program.

TABLE I
DEFINITION OF SMALL-SCALE GESTURES

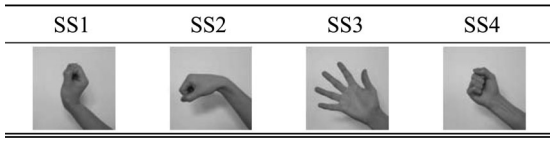
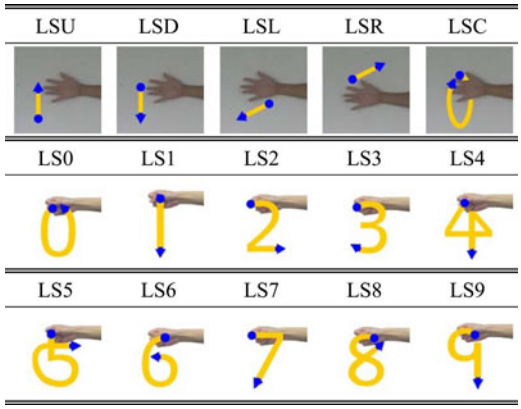


TABLE II
DEFINITION OF LARGE-SCALE GESTURES



A. Gesture-Capturing Device

A gesture-capturing device is designed to record SEMG and ACC signals synchronously. It weighs about 60 g, and consists of four dry SEMG sensors (30 mm × 16 mm × 8 mm) and a main board (56 mm × 36 mm × 14 mm) embedded with an accelerometer. These four SEMG sensors are connected to the main board by wires to share the battery and the controller. A 1000-mAh lithium battery, a charging circuit, and a power circuit are embedded on the main board. All five modules are strung with two elastic bands and can be worn around the user's forearm.

In the device (see Fig. 2), each SEMG sensor acquires one channel of SEMG signals, amplifies them by 500 times, and filters them within 20–300 Hz bandpass. Dry sensors are used because they can be attached to the skin without adhesives or conductive paste. The tri-axis accelerometer (MMA3761 L) is embedded with the main board. It measures the accelerations along three axes (x, y, z), and outputs three-channel ACC signals. Both the measured ACC and SEMG signals are digitized simultaneously by a 12-bit A/D convertor that is embedded with the microcontroller (MCU, C8051F411) at a sampling rate of 600 Hz, and then sent out via Bluetooth 2.0 using a Bluetooth serial port module that is produced by Ommitek Electronics Co.

B. Interaction Application Program

A Nokia 5800XM (with a 434-MHz ARM11 CPU, 128M RAM, Bluetooth 2.0 support, and running Symbian S60 v5.0) is used to demonstrate the feasibility of the gesture-based interaction. An interaction application program that is implemented in Symbian C++ includes Bluetooth interface, gesture recognition, translation, and phone operation modules (see Fig. 3).

The Bluetooth interface module receives data using Bluetooth API (Application Program Interface) and stores them into a buffer. The gesture recognition module reads data from the buffer and provides recognition results. The translation module maps gestures to instructions. The number of supported gestures is less than the number of interaction tasks and users are allowed to modify the mapping relationships by doing specific gestures. System events such as receiving a phone call can change the mapping relationships too. The phone operation module executes instructions coming from the translation module by calling system APIs or sending keyboard messages, which are used by the operating system to notify programs of key press events. Although 5800XM is a touch-enabled phone with only three keys, it supports all of the keyboard messages. Consequently, the phone operation module is able to manipulate most of the phone functions by mapping gestures to keyboard messages.

III. HAND GESTURE RECOGNITION

A. Hand Gesture Vocabulary

A dictionary of 19 gestures including four small-scale gestures (see Table I) and 15 large-scale gestures (see Table II) was created. To assess our signal fusion algorithms, two gestures ("LSD" and "LS1") that share the same trajectory were used, and the large-scale gestures share only two different hand shapes. When doing small-scale gestures, the user should move his wrist or fingers with no arm movement; while doing large-scale gestures, the user should grasp or open his hand, wave his arm along the predefined trajectories in the vertical plane, and keep the hand shape till the end of the gesture. Users can define personalized gestures by repeating them 24 times in the training mode of the interaction application program.

B. Algorithm Framework

The algorithms described here are implemented in the gesture recognition module of the interaction application program. Accurate recognition and fast response times are the basic requirements for algorithms running on mobile devices with limited computational resources. Because SEMG signals and ACC signals have their own advantages and disadvantages, small-scale and large-scale gestures are separated and processed using different schemes (see Fig. 4). Small-scale gestures are classified based only on SEMG signals, and large-scale gestures based on the fusion of SEMG and ACC signals. A novel segmentation scheme supporting unaligned active segments between SEMG and ACC signal

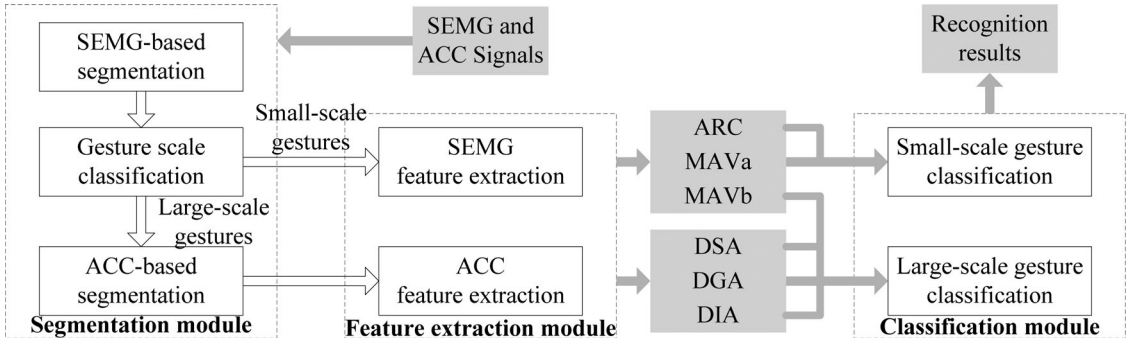


Fig. 4. Framework of the hand gesture recognition algorithms.

streams is proposed. Three new ACC features and a score-based sensor fusion scheme are designed to improve accuracy.

C. Segmentation

Segmentation aims to find the starting and end points of each motion from the signal stream. The recorded signals between these points are named the active segment. In [2] and [15], we demonstrated the feasibility of SEMG-based segmentation. ACC signals were segmented synchronously with the SEMG signal. However, ACC and SEMG signals are completely different on waveform and physical meaning. In mobile practice, SEMG active segments (SASs) are seldom aligned with corresponding ACC active segments (AASs), because it is difficult for a user (especially a novice) to ensure synchronization between arm waving and hand grasping. Therefore, a new method to segment ACC signals and SEMG signals separately is proposed to achieve better performance. Each movement corresponds to an AAS and an SAS, without strict synchronization. For ACC signal segmentation, each SAS is utilized to estimate a candidate ACC active segment (CAAS). The ACC segmentation algorithm only needs to process signals around CAAS to locate an AAS. ACC segmentation, therefore, becomes much easier because most artifacts are ruled out. This solution addresses the flexibility concerns of [17] and the lack of requirements for external messages in [3], [5], and [18].

ACC signals should be preprocessed before segmentation to minimize the noise. The preprocessing consists of smoothing and calibration. It is straightforward to apply moving average filter to smooth ACC signals [5]. Calibration aims to normalize ACC signals in each channel using subsection linear transformation: normalized ACC data are set to 0 when acceleration is 0, and to a constant (written as G) when acceleration is a gravitational acceleration. Parameters of the transformation are only related to the accelerometer and should be determined by experiments before the first use.

Let $SI_c\ (t)$ be the value of the $t\mathrm{th}$ sampling point in the $c\mathrm{th}$ channel of acquired SEMG signals or preprocessed ACC signals. Segmentation is based on the value that is defined in (1). Two thresholds are denoted as $\mathrm{Th_{on}}$ and $\mathrm{Th_{off}}$, respectively. An active segment starts at the $p\mathrm{th}$ point if $Sp\,(p+l,L)$ at its $l\mathrm{th}$ consecutive point is larger than $\mathrm{Th_{on}}$, and ends at the $q\mathrm{th}$ point if $Sp\,(q+l,L)$ at its $l\mathrm{th}$ consecutive point is smaller than $\mathrm{Th_{off}}$. The value of L and l are determined by experiments, and should be scaled linearly with the sampling rate. Here, L is set to 100 for SEMG signals and 50 for ACC signals, and l is chosen as 50 for both. The two thresholds are also determined by experiments. They should be tuned by the user to approach the optimum value. The length of each active segment should be in a reasonable range: 0.3–1.0 s for small-scale gestures and 0.5–2.5 s for large-scale gestures. An active segment will be omitted if its length exceeds the range

$$Sp(t,L)=\begin{cases}\dfrac{1}{L}\displaystyle\sum_{i=t-L+1}^{t}\left[\sum_{c=1}^{4}SI_c(i)\right]^2, & \text{if } SI \text{ is SEMG}\\ \dfrac{1}{3}\displaystyle\sum_{c=1}^{3}|SI_c(t)-SI_c(t-L)|, & \text{if } SI \text{ is ACC.}\end{cases}\tag{1}$$

D. Feature Extraction

1) SEMG Features: Various features such as mean absolute value (MAV), zero crossing rate, waveform length, and autoregressive (AR) model coefficients with a typical order 3–6 are effective for EMG pattern recognition [19]–[21]. Here, the time-domain features are preferred because of their low computational complexity. The combination of MAV and fourth-order AR model coefficients is practical and efficient [2], [15]. Considering the classification performance and the computing power of mobile devices, MAV and third-order AR model coefficients of each channel in the whole SAS (written as MAVa and ARC) are employed as an appropriate feature set. For large-scale gestures, there are additional movements (such as arm movements) in addition to finger and wrist motions. Features that are based on the whole SAS are therefore improper to describe the hand shape in these cases, while features that are based only on signals at the beginning of the SAS seem to be more useful. Results show that MAV of each channel in the first 200 ms of the active segment (written as MAVb) is effective for our predefined gestures.

2) ACC Features: Raw ACC data are directly applied as feature vector in [3] and [18]. Some other features such as mean value and variance [5] are also effective for classification. The recognition based on down-sampled raw ACC signals yielded comparable performance with the one based on original raw ACC signals in [15]. Down sampling makes all feature vectors equal in length and reduces the size of feature vector so that it is can speed up classification. However, our previous algorithms are too complex for real-time mobile applications. Therefore, we designed algorithms and features to maintain performance while reducing the computation. An additional normalization approach is further applied on the down-sampled ACC signals (DSA). In addition, two new features (written as DGA and DIA) are employed.

Assuming that the AAS starts at $s\mathrm{th}$ point and stops at $t\mathrm{th}$ point, it is normalized and linearly extrapolated to Nd points to calculate DSA (2), which is a $Nd\times 3$ sequence. DGA (3), a $Nd\times 1$ sequence, is sensitive to movements, especially to vertical motions. It equals 0 when the user's arm is motionless. DIA (4) is a 1×3 vector that quantifies the difference in orientation between the starting and end points. Nd is a constant and should be set to at least 16; change will significantly affect the amount of computation but not accuracy. It is

typically set to 32 or 16, and here is equal to 16 for the algorithm speed. There will be too much distortion in DSA if Nd is too small

$$\mathbf{DSA}_{i,c} = \frac{SI_c(sp_i) - \overline{SI_c}}{\frac{1}{Nd}\sum_{j=1}^{Nd}\left|SI_c(sp_j) - \overline{SI_c}\right|}, c = 1,2,3$$

$$sp_i = s + \text{ floor } \left(\frac{i-0.5}{Nd} \times (t-s)\right), i = 1,2,...,Nd$$

$$\overline{SI_c} = \frac{1}{t-s+1}\sum_{k=s}^{t} SI_c(k) \tag{2}$$

$$\mathbf{DGA}_{i,1} = \sqrt{\sum_{c=1}^{3} SI_c(sp_i)^2 - G} \tag{3}$$

$$\mathbf{DIA}_{1,c} = SI_c(t) - SI_c(s), c = 1,2,3. \tag{4}$$

3) Feature Combination: ACC signals are useless for small-scale gestures since there are no arm movements during performance, while ACC and SEMG signals are both important to large-scale ones. Nevertheless, SEMG signals of large-scale gestures are not as distinguishable as small-scale ones because of the similar component caused by arm waving. Feature sets are therefore constructed for small-scale and large-scale gestures, respectively. The feature set for small-scale gestures (SFS) only contains SEMG features, and that for large-scale ones (LFS) contains both SEMG and ACC features

$$\text{SFS} = \{\mathbf{MAVa}, \mathbf{ARC}\} \tag{5}$$

$$\text{LFS} = \{\mathbf{MAVb}, \mathbf{DSA}, \mathbf{DGA}, \mathbf{DIA}\} \tag{6}$$

E. Classification

1) Gesture-Scale Classification: None of the gesture employs all the six features mentioned previously. For example, ACC segmentation and ACC feature extraction can be omitted if a motion is classified as a small-scale gesture. Therefore, small-scale motions are picked right after SEMG segmentation by a threshold classifier in order to speed up the processing (see Fig. 4). That is, if the amplitude of ACC signals exceeds the given threshold, it is a large-scale gesture, and vice versa. Considering that ACC signals are often very smooth, only 32 sampling points (written as $Se_c(n), c = 1,2,3, n = 1,2,\ldots,32$) picked from CAAS using uniformly sampling are enough to quantify the amplitude (written as *Am*). Then, small-scale and large-scale gestures are recognized using different algorithms

$$Am = \frac{1}{96}\sum_{n=1}^{32}\sum_{c=1}^{3}\left|Se_c(n) - \frac{1}{32}\sum_{i=1}^{32} Se_c(i)\right|. \tag{7}$$

2) Small-Scale Gesture Classification: A Bayes linear classifier, which is able to classify samples in a linear feature space, was employed in this study for small-scale gesture classification. It has also been reported by previous studies on SEMG-based gesture recognition [22] that the Bayes linear classifier can achieve high accuracy with low computational complexity. Thus, this classifier is appropriate for real-time systems. Here, the classifier should be trained before use because of the randomicity of SEMG signals. After some pretests, we found that 32 repeats of each gesture are enough to train a classifier to reach stable and satisfactory classification performance.

3) Large-Scale Gesture Classification: Hidden Markov models (HMMs) [23] and dynamic time warping (DTW) [24] are two widely used algorithms in ACC-based classification. DTW is employed here to fit mobile devices (HMM-based recognition algorithms may result in unacceptable latency for a real-time system). Although DTW-based classification often achieves lower accuracy than HMM-based especially on large gesture vocabularies, its performance can be significantly improved if assisted by some strategies [3], [18].

Here, we propose three ACC features (DSA, DGA, and DIA), as the combination of them may yield significant improvement. The DSA and DGA of all repeats are of equal length. A fast DTW algorithm is proposed, where most matching paths can be ruled out to reduce the computational complexity. Assume that there are two time sequences of DSA written as $\mathbf{S}$ and $\mathbf{T}$, each is a $N \times C$ matrix (e.g., N points in the 3-D space). Each point in $\mathbf{S}$ is able to match one point in $\mathbf{T}$ using the following recursive algorithm. Suppose that $\mathbf{S}_{i-1}$ matches $\mathbf{T}_{k_{i-1}}$, then $\mathbf{S}_i$ matches $\mathbf{T}_{k_i}$ only when k_i satisfies the constraint conditions in (8). St and Nm are two constants that determine the searching region in $\mathbf{T}$ for each point in $\mathbf{S}$. St equals 4 and Nm equals 2 here, and should be scaled with N

$$\begin{cases} k_i \in Z, Z = \left\{ \begin{array}{c} z | 1 \le z \le Nd \text{ and } i - St \le z \le i + St \\ \text{and } k_{i-1} \le z \le k_{i-1} + Nm \end{array} \right\} \\ \sum_{c=1}^{C} |\mathbf{S}_{i,c} - \mathbf{T}_{k_i,c}| \le \sum_{c=1}^{C} |\mathbf{S}_{i,c} - \mathbf{T}_{z,c}|, \forall z \in Z \end{cases}. \tag{8}$$

After each point in $\mathbf{S}$ ($\mathbf{S}_i$ as an example) finds a matching point in $\mathbf{T}$ (written as $\mathbf{T}_{k_i}$), distance between $\mathbf{S}$ and $\mathbf{T}$ (written as $\mathrm{DTW}(\mathbf{S}, \mathbf{T})$) can be calculated according to (9). In our algorithms $\mathbf{S}$ represents DSA or DGA of a repeat to be classified, and $\mathbf{T}$ represents that of a template

$$\mathrm{DTW}(\mathbf{S}, \mathbf{T}) = \frac{\sum_{i=1}^{N}\sum_{c=1}^{C}|\mathbf{S}_{i,c} - \mathbf{T}_{k_i,c}|}{N \times C}. \tag{9}$$

Fusion of SEMG and ACC signals is necessary for large-scale gesture recognition. Four types of features are combined (6) to provide information from different aspects. The meanings, ranges, and forms of these features are different. For example, the ranges of MAVb and DIA are independent because MAVb is SEMG-based, while DIA is ACC-based. In addition, DSA is a time sequence, while DIA is a vector. A score-based classifier is therefore proposed to fuse these features. In the proposed classification method, an unknown repeat is given four scores by each type of gesture first. The four scores are calculated based on the four features, respectively; each contains contribution from all the templates of this gesture. Then, the product of these four scores is defined as the total score of each gesture, which indicates the similarity between the unknown repeat and this gesture. The unknown repeat is categorized as the gesture with the highest total score. The algorithm is discussed in detail below.

Assume that there are Ng gestures with Nr repeats each in the training set, and there is an unknown repeat with its feature

$$Fs = \left\{\mathbf{MAVb}^S, \mathbf{DSA}^S, \mathbf{DGA}^S, \mathbf{DIA}^S\right\}. \tag{10}$$

Let G^i $(i = 1,2,\ldots,Ng)$ be the training repeats of ith gesture, $R^{i,j}$ $(i = 1,2,\ldots,Nr)$ be the jth repeat in G^i (therefore, $G^i = \left\{R^{i,1}, R^{i,2}, \ldots, R^{i,Nr}\right\}$), $\mathbf{T}^{i,j}$ be one of the features of $R^{i,j}$, and $\mathbf{X}$ be the corresponding feature in Fs. The distance between $\mathbf{X}$ and each $\mathbf{T}^{i,j}$ is calculated according to

$$\mathrm{dis}(\mathbf{X}, \mathbf{T}^{i,j}) = \begin{cases} \sqrt{\sum_{c=1}^{4}\left(\mathbf{X}_{1,c} - \mathbf{T}^{i,j}_{1,c}\right)^2}, & \text{for } \mathbf{MAVb} \\ \mathrm{DTW}(\mathbf{X}, \mathbf{T}^{i,j}), & \text{for } \mathbf{DSA} \text{ or } \mathbf{DGA} \\ \sqrt{\sum_{c=1}^{3}\left(\mathbf{X}_{1,c} - \mathbf{T}^{i,j}_{1,c}\right)^2}, & \text{for } \mathbf{DIA}. \end{cases} \tag{11}$$

Therefore, $Ng \times Nr$ distances are calculated based on a single feature, and they are then sorted from small to large. $r^{i,j}$ is the ranking of

TABLE III
TYPICAL VALUES OF $Rb_{X,i}$

Feature	$Rb_{X,0}$	$Rb_{X,1}$	$Rb_{X,2}$	$Rb_{X,3}$	$Rb_{X,4}$
MAVb	1	8	5	1	0
DSA	2	6	3	1	0
DGA	2	6	3	1	0
DIA	4	4	2	1	0

the value $\mathrm{dis}\,(\mathbf{X}, \mathbf{T}^{i,j})$ in these $Ng \times Nr$ values. Each $r^{i,j}$ is mapped to a corresponding score by the function

$$s2(\mathbf{X}, r) = \begin{cases} Rb_{\mathbf{X},1}, & \text{if } 0 \le r < Nr/8 \\ Rb_{\mathbf{X},2}, & \text{if } Nr/8 \le r < 3Nr/8 \\ Rb_{\mathbf{X},3}, & \text{if } 3Nr/8 \le r < Nr \\ Rb_{\mathbf{X},4}, & \text{else} \end{cases} \tag{12}$$

and G^i can get

$$\mathrm{s1}(\mathbf{X}, G^i) = Rb_{\mathbf{X},0} + \sum_{j=1}^{Nr} s2(\mathbf{X}, r^{i,j}) \tag{13}$$

points only based on the feature $\mathbf{X}$. The total score of G^i is the product of the four scores that are based on four features, respectively,

$$\begin{aligned} \mathrm{score}(\mathbf{Fs}, G^i) = {} & s1(\mathbf{MAVb}^S, G^i) \times s1(\mathbf{DSA}^S, G^i) \\ & \times s1(\mathbf{DGA}^S, G^i) \times s1(\mathbf{DIA}^S, G^i). \end{aligned} \tag{14}$$

$Rb_{X,i}$ $(\mathrm{i} = 0, 1, \ldots, 4)$ in (12) are all constants, and determine the weight of each feature and each training repeat. They should satisfy $Rb_{X,i} > Rb_{X,i+1}$ $(i = 1, 2, 3)$ to ensure that the nearer a training repeat is, the higher weight it has. $Rb_{X,4}$ is recommended to be set to 0. This will speed up the classification significantly because the computation of DTW and sort can be terminated once the distance between $\mathbf{X}$ and $\mathbf{T}^{i,j}$ is too large. Furthermore, there are three ACC features while only one SEMG feature, and they are distinguished differently (see Section IV). Therefore, we give MAVb a higher weight by decreasing $Rb_{X,0}$. The values of $Rb_{X,i}$ in our algorithms are determined by experiments and are listed in Table III.

IV. RESULTS

Experiments were conducted to assess the performance of the proposed hand gesture recognition algorithm framework and the gesture-based interaction prototype. The 20 participants were college students (13 male, 7 female) aged 22–27 who had used mobile phones for at least four years so that they were familiar with the operations. Thirty two repeats of each small-scale gesture and ten repeats of each large-scale gesture were acquired from each participant. $(32 \times 4 + 10 \times 15) \times 20 = 5560$ repeats were included in our database.

A. Algorithms Performance

1) User-Dependent Testing: The user-dependent testing assesses the system performance when a user trains the classifier using his or her signals. Here, four repeats of each gesture from all participants were selected to build a training set. The remaining repeats from each participant formed a testing set. Thus, we used one training set and 20 testing sets. The average recognition accuracy of 20 participants across $\mathrm{C}_{10}^4 = 210$ possible combinations (picking four out of ten repeats) is shown in Table IV. The results show that the 19 gestures can be classified with the average accuracy of 95.0%. "LS0" and "LS6" are confusable, possibly because of the similarity of their SEMG and ACC signals. "LS1," "LS4," and "LS9" are also confusable as they share the same hand shape and similar traces in the second half.

2) User-Independent Testing: The user-independent testing strategy could simulate the most common use case with testing data and training data from different users. Cross-validation was conducted: repeats from one of the 20 participants were considered to be the testing set, and repeats from the other 19 participants were used to form the training set. Table V shows the classification accuracies of 19 gestures averaged across 20 participants. The average accuracy achieved 89.6%. The slight performance degradation that is compared with the user-dependent testing could be attributed to individual difference. Four of the relatively low-accurate gestures are affected by individual differences in SEMG signals: "SS3" and "SS4" are confusable because they are both small-scale gestures, for which only SEMG signals are processed in classification; "LS1" and "LSD" share the same trace while differ only in hand shape, therefore, SEMG features determine the classification result. Another three gestures yield low accuracy because of individual differences in ACC signals. For example, participants were asked to perform "LS8" according to our definition. However, several participants write the number "8" differently and were uncomfortable when following our instructions. Consequently, hand writing differences are one kind of individual differences among large-scale gestures.

3) Contributions of Features: Large-scale gesture classification is based on the fusion of SEMG and ACC signals. Fig. 5, which illustrates the classification accuracies in user-dependent testing that is based on different combinations of features, shows the contribution of each feature. In our gesture vocabulary, there are only two different hand shapes (indicated by MAVb) of all the large-scale gestures, while there are only two gestures with the same trajectory (indicated mainly by DSA). DSA, therefore, performs well as the only feature here while MAVb cannot. DGA is not very sensitive to horizontal motions, and consequently performs a little worse than DSA. DIA contains too little information to classify a motion independently, but is still helpful and uses few computational resources. Here, MAVb and DSA are the most important features, and DGA performs better than DIA. Therefore, the weight of DIA was reduced by setting $Rb_{\mathrm{DIA},0}$ (see Table III) to a larger value.

4) Size of the Training Set: The calculative burden of small-scale gesture classification is less sensitive to the size of training set, while that of large-scale gesture classification is opposite. Fig. 6 shows that using a training set with a larger size can improve the accuracy, but will cause a longer latency. Therefore, there is a tradeoff between accuracy and delay for large-scale gesture classification. The number of sampling point Nd also affects them because there is less distortion and more data in DSA if Nd is larger. For a real-time system, data processing should be accomplished within 300 ms. So that Nd is set to 16 as mentioned previously, and the number of repeats in the training set Nr is set to 40. Actually, our training set consists of 16 repeats of each small-scale gesture and two repeats of each large-scale gesture from each participant, and the time delay for data processing is about 300 ms.

One of the advantages of our algorithm framework is that it uses fewer computing resources so that real-time interaction can be guaranteed on mobile devices. As compared with a Digital Pen [5], a custom wireless gesture capture device that can only acquire ACC signals, which uses about 200 ms for recognition when running on the computer with Intel Core 2 Duo CPU, our prototype uses only 300 ms on our mobile phone. Although their system achieves a little higher accuracy, it supports only ten gestures, while our prototype supports 19 gestures.

TABLE IV
EXPERIMENTAL RESULTS FOR USER-DEPENDENT TESTING

Gesture	SS1	SS2	SS3	SS4	LS0	LS1	LS2	LS3	LS4	LS5
Accuracy	97.3±7.0	96.8±7.3	91.2±9.2	96.9±4.7	90.1±14.4	88.1±11.9	98±6.5	96.8±7.6	94.4±10.2	97.7±5.5
Gesture	LS6	LS7	LS8	LS9	LSU	LSD	LSL	LSR	LSC	
Accuracy	94.9±9.2	98.1±6.9	96.5±9.7	94.9±11.6	95.3±11.9	92.5±10.3	95.0±9.4	96.7±11.3	94.1±10.3	

TABLE V
EXPERIMENTAL RESULTS FOR USER-INDEPENDENT TESTING

Gesture	SS1	SS2	SS3	SS4	LS0	LS1	LS2	LS3	LS4	LS5
Accuracy	96.6±5.8	95.8±5.5	73.9±13.9	89.2±12.6	88.5±13.1	80.3±14.7	92.3±11.7	96.2±8.0	93.6±11.9	91.8±13.7
Gesture	LS6	LS7	LS8	LS9	LSU	LSD	LSL	LSR	LSC	
Accuracy	85.0±16.1	96.5±5.9	79.6±24.6	91.0±14.5	96.0±6.8	82.3±14.8	90.1±15.5	92.2±16.7	91.0±15.9	

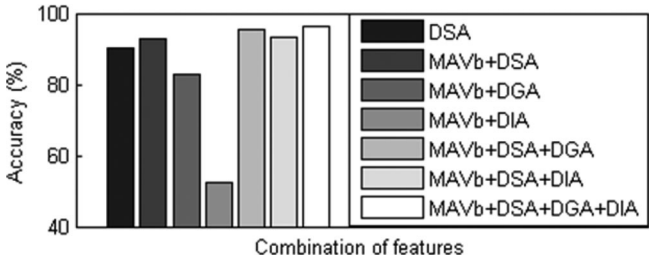


Fig. 5. Recognition accuracies using different combinations of features.

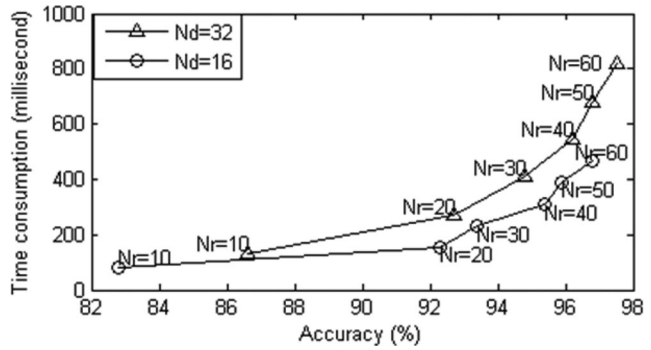


Fig. 6. Algorithm performance using different sizes of the training set.

TABLE VI
MAPPING TABLE OF GESTURES AND OPERATIONS

Gesture	Operation	Gesture	Operation
SS1	Exit	LSU	Move up
SS2	System	LSD	Move down
SS3	Cancel	LSL	Move left
SS4	OK	LSR	Move right
LS0-LS9	0-9	LSC	Open menu

B. Interaction Performance

1) Efficiency of the Gesture-Based Interaction: Participants were requested to accomplish a given interaction task only using gestures mapped to keyboard messages (see Table VI). They were to correct any mistakes using gestures. The number of gestures and completion time were recorded. Each participant could practice this task three times before testing. The interaction task was to operate the system menu and the media player. Typically, 11 motions (six small-scale motions and five large-scale motions) or eight taps on touch screen were necessary to accomplish this task.

TABLE VII
INTERACTION PERFORMANCE

	Experienced Group	Novices
Number	12.1±1.0	13.3±1.6
Time(s)	26.8±4.1	33.4±4.8

TABLE VIII
RESULTS OF THE SURVEY ON USER EXPERIENCE

Group	Experienced	Novices	p (Mann-Whitney U Test)
Accuracy	4.7±0.5	3.9±0.9	0.038
Practicability	4.4±0.5	3.9±0.8	0.352
Enjoyment	4.6±0.8	4.1±0.8	0.610
Nature	4.8±0.4	4.4±0.5	0.171
Comfort	3.8±0.8	3.0±1.0	0.067

Twenty participants were divided into an experienced group (four male and three female) and the novices (9 male and 4 female). Members in the experienced group had experienced gesture-based interaction before, while the novices had not. The results in Table VII indicate that most users, even novices can master gesture-based interaction. Although doing gestures takes more time than tapping on a touch screen (10–15 s typically), the gesture-based interface provides a new option and is useful in some use cases. The differences between novices and the experienced group indicate that more practice may make the interaction system more effective.

2) User Experience: A questionnaire was conducted to quantify user experience. All 20 participants participated. They assessed our system using a five-point scale from 1 (unacceptable) to 5 (excellent): 1) Accuracy: both the gesture recognition and interaction are accurate. 2) Practicability: this interaction system is practical in daily life or some use cases. 3) Enjoyment: the interaction is interesting or attractive. 4) Natural: the gestures are easy to learn and culturally acceptable, and the mappings between gestures and instructions are apparent. 5) Comfort: the interaction is easy and comfortable.

Table VIII shows the average scores given by each group. Participants in the experienced group provided better user experience scores in accuracy, perhaps because they are more familiar with gesture-based interaction. For both experienced group and the novices, practicability was mainly influenced by its efficiency. Enjoyment while good may be better if the classification runs faster. Comfort received the lowest score as completing large-scale gestures often made participants tired.

V. DISCUSSION

A wearable gesture-based interaction prototype demonstrates the feasibility of hand gesture interaction in mobile application that is based on the fusion of SEMG and ACC signals. A wireless wear-

able gesture capture device is designed to acquire ACC and SEMG signals, and an algorithm framework is proposed to realize gesture classification on mobile devices. An interaction program is developed for the mobile device to realize gesture recognition and to manipulate the mobile device taking recognition results as instructions. Our prototype supports 19 gestures, a large gesture vocabulary for mobile device-based systems. The experimental results from interaction testing show that gesture-based interaction is feasible and performs better with experienced users although its efficiency needs further improvement. A user experience questionnaire indicates that our prototype can be accepted by users. Because gesture-based interaction is intuitive and easy to learn, we expect this gesture-based interaction prototype to be accepted by mobile device users.

Battery autonomy is an important factor of the practicality. Our gesture-capturing device can work about 10 h, and our program and Bluetooth result in about 12% increase to the phone's basic power dissipation. We expect improvement by upgrading to lower energy using protocols.

Small-scale and large-scale gestures are separated according to the parameter defined in (7). The parameter of a large-scale gesture is dozens of times as much as that of a small-scale gesture. Therefore, threshold-based classification is efficient and effective, and there are few errors when users follow our instructions on doing gestures. However, errors may occur if a large-scale gesture is performed too slowly or a small-scale gesture is performed with additional waves or vibrations. Moreover, our segmentation method assumes active segments to be separated by resting states. The method is reliable and stable in most cases that are based on user-specific thresholds. However, it may be not so effective when the user performs unrelated forearm muscle activities. Therefore, the application of our interface is more or less restricted by the two challenges.

As a supplement to existing interfaces, our interface is not as efficient as touch screens or keyboards at present. However, just as the Digital Pen and the MYO, our interface brings a new interaction experience. It can recognize both small-scale and large-scale gestures, while the Digital Pen captures only one kind. Furthermore, a gesture-based interaction scheme for mobile device is realized in order to enable users to operate the phone without touching (or even seeing) the phone. However, MYO supports only a few simple interactions. The gesture vocabulary here is only an example, and can be redefined by users to achieve higher comfort and practicality. This technology will hopefully be applied in daily life for convenience, and in games for better user experience.

ACKNOWLEDGMENT

The authors would like to thank Dr. Z. Zhao for the hardware development, Dr. X. Zhang for useful discussion, and other volunteers in data acquisition.

REFERENCES

[1] Thalmic Labs. (2013). MYO—Gesture control armband by Thalmic Labs [Online]. Available: https://www.thalmic.com/myo/

[2] Z. Lu, X. Chen, Z. Zhao, and K. Wang, "A prototype of gesture-based interface," in *Proc. 13th Int. Conf. Human Comput. Interaction Mobile Devices Services*, 2011, pp. 33–36.

[3] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave—Accelerometer-based personalized gesture recognition and its applications," *Pervasive Mobile Comput.*, vol. 5, pp. 657–675, Dec. 2009.

[4] M. K. Chong, G. Marsden, and H. Gellersen, "GesturePIN: Using discrete gestures for associating mobile devices," in *Proc. 12th Int. Conf. Human Comput. Interaction Mobile Devices Services*, 2010, pp. 261–264.

[5] J. Wang and F. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 2998–3007, Jul. 2012.

[6] C. Zhu and W. Sheng, "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 3, pp. 569–573, May. 2011.

[7] T. S. Saponas, D. S. Tan, D. Morris, and R. Balakrishnan, "Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2008, pp. 515–524.

[8] S. Vernon and S. S. Joshi, "Brain-muscle-computer interface: Mobile-phone prototype development and testing," *IEEE Trans. Inform. Technol. Biomed.*, vol. 15, no. 4, pp. 531–538, Jul. 2011.

[9] T. S. Saponas, D. S. Tan, D. Morris, J. Turner, and J. A. Landay, "Making muscle-computer interfaces more practical," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2010, pp. 851–854.

[10] E. Costanza, S. A. Inverso, R. Allen, and P. Maes, "Enabling always-available input with muscle-computer interfaces," in *Proc. Comput. Human Interaction*, 2007, pp. 819–828.

[11] A. Hatano, K. Araki, and M. Matsuhara, "A Japanese input method for mobile terminals using surface EMG signals," in *Proc. 22nd Annu. Conf. Jpn.-Soc. Artif. Intell.*, 2009, pp. 5–14.

[12] S. M. Rissanen, M. Kankaanpää, M. P. Tarvainen, V. Novak, P. Novak, K. Hu, B. Manor, O. Airaksinen, and P. A. Karjalainen, "Analysis of EMG and acceleration signals for quantifying the effects of deep brain stimulation in Parkinson's disease," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2545–2553, Sep. 2011.

[13] R. A. Joundi, J. Brittain, N. Jenkinson, A. L. Green, and T. Aziz, "Rapid tremor frequency assessment with the iPhone accelerometer," *Parkinsonism Related Disorders*, vol. 17, pp. 288–290, May. 2011.

[14] A. Fougner, E. Scheme, A. D. C. Chan, K. Englehart, and Ø. Stavdahl, "A multi-modal approach for hand motion classification using surface EMG and accelerometers," in *Proc. IEEE Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2011, pp. 4247–4250.

[15] 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," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 6, pp. 1064–1076, Nov. 2011.

[16] Y. Li, X. Chen, X. Zhang, K. Wang, and J. Z. Wang, "A sign-component-based framework for Chinese sign language recognition using accelerometer and sEMG data," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 10, pp. 2695–2704, Oct. 2012.

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

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

[19] L. Xin, Z. Rui, Y. Licai, and L. Guanglin, "Performance of various EMG features in identifying arm movements for control of multifunctional prostheses," in *Proc. IEEE Youth Conf. Inf., Comput. Telecommun.*, 2009, pp. 287–290.

[20] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "Application of wavelet analysis in EMG feature extraction for pattern classification," *Meas. S ci. R ev.*, vol. 11, pp. 45–52, 2011.

[21] G. Yang, S. Wang, and Y. Chen, "SEMG analysis basing on AR model and Bayes taxonomy," *Appl. Mech. Mater.*, vol. 44–47, pp. 3355–3359, 2011.

[22] J. Kim, S. Mastnik, and E. André, "EMG-based hand gesture recognition for realtime biosignal interfacing," in *Proc. 13th Int. Conf. Intell. User Interfaces*, 2008, pp. 30–39.

[23] T. Pylvänäinen, "Accelerometer based gesture recognition using continuous HMMs," in *Pattern Recognition and Image Analysis*. Berlin, Germany: Springer, 2005, pp. 639–646.

[24] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proc. 23rd Int. Conf. Mach. Learning*, 2006, pp. 1033–1040.