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

RAMAKRISHNA KAMMARI, S. MAHABOOB BASHA

Abstract: An algorithmic framework is proposed to process acceleration and surface electromyography (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.

Keywords: Accelerometer, Electromyography, 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. 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 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.

Fig.1. Gesture-based interaction prototype with the gesture-capturing device.

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.

- A small, lightweight, and power-efficient wireless wearable device to capture gestures records three-channel ACC and four channel SEMG signals from forearm.
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.

An active segmentation scheme, overcoming the difficulties in ACC signal segmentation and the synchronization of active segments in SEMG and ACC signals, is presented.

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.

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 (30mm×16mm×8 mm) and a main board (56mm×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 band pass. 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 Omnitek 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

<table>
<thead>
<tr>
<th>TABLE I: Definition of small-scale gestures</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS1</td>
</tr>
<tr>
<td><img src="image1.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II: Definition of large scale gestures</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS0</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
</tr>
</tbody>
</table>

![Fig. 2. Architecture of the gesture-capturing device.](image10.png)

![Fig. 3. Architecture of the interaction application program.](image11.png)
A Hand Gesture Recognition Framework and Wearable Gesture-Based Interaction Prototype for Mobile Devices

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. Novel segmentation scheme supporting unaligned active segments between SEMG and ACC signal 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 a 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 reprocessed 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 $S_{ic}(t)$ be the value of the $i^{th}$ sampling point in the $c^{th}$ 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 $Th_{on}$ and $Th_{off}$, respectively. An active segment starts at the $p^{th}$ point if $Sp(p + 1, L)$ at its $l^{th}$ consecutive point is larger than $Th_{on}$, and ends at the $q^{th}$ point if $Sp(q + 1, L)$ at its $l^{th}$ consecutive point is smaller than $Th_{off}$. 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

$$Sp(t, L) = \begin{cases} \frac{1}{L} \sum_{i=1}^{L} \frac{1}{4} \sum_{c=1}^{4} S_{ic}(t), & \text{if } SI \text{ is SEMG} \\ \frac{1}{3} \sum_{c=1}^{3} |S_{ic}(t) - S_{ic}(t-L)|, & \text{if } SI \text{ is ACC.} \end{cases}$$

D. Feature Extraction
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

**International Journal of Innovative Technologies**
**Volume.03, Issue No.08, September-2015, Pages: 1412-1417**
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.

**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 the first point and stops at the t point, it is normalized and linearly extrapolated to Nd points to calculate DSA, which is a Nd x 3 sequence. DGA (3), a Nd x1 sequence, is sensitive to movements, especially to vertical motions. It equals 0 when the user’s arm is motional. DIA (4) is a 1 x3 vector that quantifies the difference in orientation between the starting and endpoints. 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.

\[
\text{DSA}_{i,c} = \frac{S_{i,c}(s_i) - S_{i,c}(s_Nd)}{Nd \sum_{j=1}^{Nd} |S_{i,c}(s_j) - S_{i,c}(s_Nd)|, c = 1, 2, 3}
\]

\[
s_i = \text{floor} \left( \frac{i - 0.5}{Nd} \times (t - s) \right), i = 1, 2, ..., Nd
\]

\[
S_{i,c}(s_i) = \frac{1}{t - s + 1} \sum_{k=i}^{t} S_{i,c}(k)
\]

\[
\text{DGA}_{i,1} = \sum_{c=1}^{3} S_{i,c}(s_i)^2 - G
\]

\[
\text{DIA}_{i,1} = S_{i,c}(t) - S_{i,c}(s), c = 1, 2, 3.
\]

**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.

\[
\text{SFS} = \{\text{MAVa}, \text{ARC}\}
\]

\[
\text{LFS} = \{\text{MAVb}, \text{DSA}, \text{DGA}, \text{DIA}\}
\]

**E. Classification**

**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 Sec (n), c = (1, 2, 3), n = (1, 2, ..., 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.

**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 random component 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.

\[
Am = \frac{1}{96} \sum_{n=1}^{32} \sum_{c=1}^{3} \left[ \text{Sec}(n) - \frac{1}{32} \sum_{i=1}^{32} \text{Sec}(i) \right]
\]

**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. \(32 \times 4 + 10 \times 15\) \(= 5560\) repeats were included in our database.

**A. Algorithms Performance**

**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 C4 10 = 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,” “LS3,” “LS4,” and “LS9” are also confusable as they share the same hand shape and similar traces in the second half.

**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 som one of the 20
participants were considered to be the testing set, and repeats
from the other 19 participants were used to form the raining
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.

**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 RbDIA,0 (see Table III) to a larger value.

**TABLE IV: Mapping Table of Geasture and Training Set**

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS1</td>
<td>Exit</td>
</tr>
<tr>
<td>SS2</td>
<td>System</td>
</tr>
<tr>
<td>SS3</td>
<td>Cancel</td>
</tr>
<tr>
<td>SS4</td>
<td>OK</td>
</tr>
<tr>
<td>LS0-L9</td>
<td>0-9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSU</td>
<td>Move up</td>
</tr>
<tr>
<td>LSD</td>
<td>Move down</td>
</tr>
<tr>
<td>LSL</td>
<td>Move left</td>
</tr>
<tr>
<td>LSR</td>
<td>Move right</td>
</tr>
<tr>
<td>LSC</td>
<td>Open menu</td>
</tr>
</tbody>
</table>

**B. Interaction Performance**

**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. 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 as shown in Fig.6. The differences
between novices and the experienced group indicate that
more practice may make the interaction system more
effective.

**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):

- Accuracy: both the gesture recognition and interaction
  are accurate.
- Practicability: this interaction system is practical in daily
  life or some use cases.
• Enjoyment: the interaction is interesting or attractive.
• Natural: the gestures are easy to learn and culturally acceptable, and the mappings between gestures and instructions are apparent.
• Comfort: the interaction is easy and comfortable.

V. REFERENCES