This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal

A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction using Accelerometer and Gyroscope sensors

Hari Prabhat Gupta, Haresh S Chudgar, Siddhartha Mukherjee, Tanima Dutta, and Kulwant Sharma

***Abstract*—Recent advances in smart devices have sustained them as a better alternative for the design of human-machine interaction because they are equipped with accelerometer sensor, gyroscope sensor, and an advanced operating system. This paper presents a continuous hand gestures recognition technique that is capable of continuous recognition of hand gestures using three-axis accelerometer and gyroscope sensors in a smart device. To reduce the influence of unstableness of a hand making the gesture and compress the data, a gesture coding algorithm is developed. An automatic gesture spotting algorithm is developed to detect the start and end points of meaningful gesture segments. Finally, a gesture is recognized by comparing the gesture code with gesture database using dynamic time warping algorithm. In addition, a prototype system is developed to recognize the continuous hand gestures based human-machine interaction. With the smartphone, the user is able to perform the predefined gestures and control smart appliances using the Samsung AllShare protocol.**

***Index Terms*—Accelerometer, gesture recognition, gyroscope, interactive controller, machine interaction.**

I. Introduction

The Human-Machine Interaction (HMI) describes how humans interact with machines. The HMI facilitates many applications in different domains, such as industrial, transportation, medical, and environmental systems [1]–[4]. A *hand gesture* is a meaningful hand movement expressed by the human. Hand gesture recognition technologies find their applications in many areas, such as gesture-based machine control and Sign Language Recognition (SLR). The hand gesture-based machine control translates gestures into commands as the input of machines [5]–[8]. The SLR aims to interpret sign languages automatically by a machine [9].

Based on the methods used to capture the gestures, the literature on hand gestures recognition techniques can be classified into the following categories: vision-based and sensor based techniques. The Vision-based Gesture Recognition (VGR) techniques are use vision or optical sensors [10]–[16]. The VGR techniques usually use 3D and appearance modeling to recognize the hand movement expressed by the human. A VGR technique is a challenging problem in the field of computer vision because it is sensitive to the user environment such as background texture, color, and lighting. The Sensor-based Gesture Recognition (SGR) techniques use a special device known as *hand data sensor glove* [1], [3], [4], [7]–[9], [17]–[20]. The device captures the motion of the hand. A SGR technique requires the human to wear the hand sensor glove for capturing hand movement.

A SGR technique consists of start and end points of a hand gesture action. A gesture recognition technique is said to be a *continuous hand gesture recognition* technique, if the start and end points of gestures are automatically detected. It does not require any human action for gesture spotting. The gesture spotting in a sequence of gestures is often regarded as one of the main difficulties in SGR techniques.

Recent advances in smart devices (*e.g.*, smartphone, tablet, and smartwear) have sustained them as a better alternative for the design of HMI because they are equipped with sensors and communication radios. The smart devices therefore not only change the way humans communicate with each other, but also the way they interact with the machine. Through the Samsung AllShare protocol, one can control any smart device [21].

In this paper, we address the problem: ***how to recognize continue hand gestures using the accelerometer and gyroscope sensors in a smart device?*** To address this problem, we use the accelerometer and gyroscope sensors of a smart device for recognizing the continuous hand gestures. Different from the existing work, our technique works on continuous gestures and does not use hand sensor gloves, cameras, kinect sensor, or any extra sensors.

Motivation and Major Contributions: To the best of our knowledge, this is the first work to recognize the continuous hand gestures for HMI using smart devices. The motivation and major contributions of our research are as follows:

- We present a Continuous Hand Gestures (CHG) recognition technique that is capable of continuous recognition of hand gestures. Our technique is different to existing work [3], [7]–[9], [16], [18], [19] in that it does not require a camera or dedicated sensors. It is therefore suitable for a low-cost and energy-efficient HMI. We use a smart device as a gesture recognition device. There is no assumptions for position, direction, or orientation of the smart device.
- As for intelligent HMI, it is important to automatically detect the start and end points of a gesture. However, some existing gesture recognition techniques [8], [18], [22], require a user action for gesture spotting. We

Hari and Tanima are with Indian Institute of Technology (BHU) Varanasi, India e-mail: {hariprabhat.cse,tanima.cse}@iitbhu.ac.in. Siddhartha is with Samsung R&D Institute Bangalore India, 560037 e-mail: siddhartha.m@samsung.com. Haresh is with University of Massachusetts Amherst e-mail: haresh.chudgar@outlook.com. Kulwant is with Banaras Hindu University, Varanasi, India e-mail: kulwantsharma.bhumca11@gmail.com.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal

develop an automatic gesture spotting and recognition algorithm to detect the start and end points of meaningful gesture segments without any human action.

- We develop a prototype system using three-axis accelerometer and gyroscope sensors in a smart device and implement Dynamic Time Warping (DTW) algorithm for gesture recognition. We use a smartphone as an input smart device. The proposed system can also be used on any smart device which has the accelerometer and gyroscope sensors and interacts with a machine. With the smartphone, the user is able to perform the gestures and transmits them to a television via Samsung AllShare service. The proposed system is wireless and operates accurately and efficiently in uncontrolled environments.

The rest of the paper is organized as follows: In the next section, we describe the existing work on gesture recognition. Section III states the assumptions and defines the terms used in this work. We present a CHG technique using a smartphone in Section IV. The implementation and experiment results are given in Section V and Section VI concludes this paper.

II. Related work

Researchers have made significant progress in the area of HMI in recent years. Recently, due to the advancement in the microelectromechanical system, wearable sensor-based gesture recognition has been gaining attention. Wearable sensors have less dependence on their surrounding compared to vision-based systems. In this section, we show some recent wearable sensor-based gesture recognition techniques.

Chun and Weihua [18] developed a system using an inertial sensor uIMU from MEMSense as input device and implemented a neural network and a hierarchical hidden Markov model for gesture recognition. The uIMU sensor connected to a personal digital assistant through a serial converter and sends the data to a desktop computer through Wi-Fi. A framework for activity awareness using surface electromyography and accelerometer signals was proposed in [19]. Xu *et al.* [9] developed a framework for hand gesture recognition, which can be utilized in both sign language recognition and gesture-based control. The framework combined information from accelerometer and EMG sensors to achieve gesture recognition.

Ruize *et al.* [3] presented gesture recognition models based on the input signals from microelectro mechanical systems (MEMS) 3-axes accelerometers. The authors developed a system using a MEMS 3-axes acceleration sensing chip integrated with data management and Bluetooth wireless data chips. A data fusion approach to hand gesture recognition based on the probabilistic HMM classification was introduced in [8]. It shown that fusing of sensory data from multiple sensors led to an overall gesture recognition rate of 93%. This recognition rate was higher when using each sensor individually on its own. Zhiyuan *et al.* [7] proposed a framework to process acceleration and surface electromyographic signals for gesture recognition, which utilized a Bayes linear classifier and an improved dynamic time-warping algorithm. The prototype system included accelerometer and SEMG sensors. An application program with the proposed algorithmic framework for a mobile phone, is developed to realize gesture-based real-time interaction. Shengli *et al.* in [15] presented an algorithm for hand gesture tracking and recognition based on the integration of image (Logitech QuickCam Pro 9000), accelerometer, and gyroscopes sensors.

III. Preliminary

In this section, we state the assumptions and define the terms used in this work. We also summarize the notation used in Table I. We use a smartphone for acquisition of sensory data. Let $A(t) = [a_x(t), a_y(t), a_z(t)]^\top$ and $G(t) = [g_x(t), g_y(t), g_z(t)]^\top$ denote the sensory data of accelerometer and gyroscope sensors at time instant t, respectively.

Definition 1. *A hand gesture is a meaningful hand movement expressed by the human. We use six gestures in this paper, denote by* $\{g_1, g_2, g_3, g_4, g_5, g_6\}$*, as shown in Fig. 1. The gesture vocabulary, denote by* $\mathbb{G}$*, is the set of gestures. We select these gestures because such gestures are easy to correlate with their actions.*

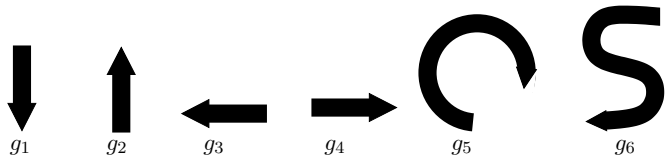


Fig. 1. Illustration of the gesture vocabulary $\mathbb{G}$.

Definition 2. *A hand gesture* $g_i \in \mathbb{G}$ *has start point and end point, denote by* g_i^s *and* g_i^e*, respectively,* $1 \leq i \leq 6$*. A point is known as inter-gesture transition point in continuous hand gestures, denoted by* g_{ij}*, where* $g_i^e = g_j^s = g_{ij}$*,* $g_i \in \mathbb{G}$*, and* $g_j \in \mathbb{G}$*. Fig. 2 illustrates the inter-gesture transition point. The process of segmenting the start and end points of continuous hand gestures is known as a gesture spotting.*

Definition 3. *Let* $P(t) = [p_x(t), p_y(t), p_z(t)]^\top$ *and* $P(t') = [p_x(t'), p_y(t'), p_z(t')]^\top$ *be the sensory data (accelerometer sensor or gyroscope sensor) at time instants* t *and* t'*, respectively. The distance between* $P(t)$ *and* $P(t')$*, denoted by* $|P(t) - P(t')|$*, is given by*

$$\sqrt{(p_x(t) - p_x(t'))^2 + (p_y(t) - p_y(t'))^2 + (p_z(t) - p_z(t'))^2}. \quad (1)$$

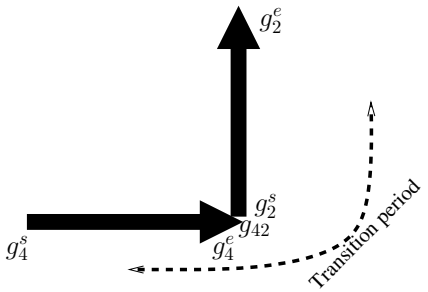


Fig. 2. Illustration of a inter-gesture transition point and transition period.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal

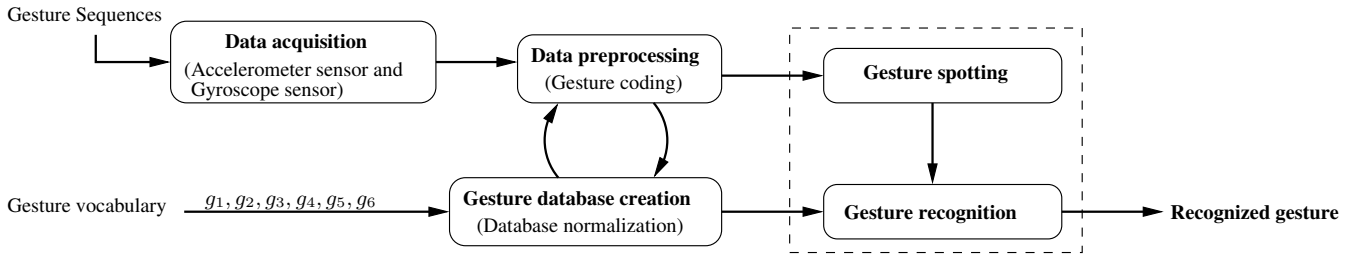


Fig. 3. Block diagram of the CHG technique using smart devices.

Definition 4. *Let $P(t) = [p_x(t), p_y(t), p_z(t)]^\top$ be the sensory data (accelerometer or gyroscope sensor), $1 \leq t \leq N$, where N is number of samples. The mean value of N samples, denoted by μ^N, is given by*

$$\mu^N = \left[\frac{\sum_{q=1}^{N} p_x(q)}{N}, \frac{\sum_{q=1}^{N} p_y(q)}{N}, \frac{\sum_{q=1}^{N} p_z(q)}{N} \right]^\top = \left[\mu_x^N, \mu_y^N, \mu_z^N \right]^\top. \tag{2}$$

The variance of M samples, denoted by σ^M, is given by

$$\sigma^M = [\sigma_x^M, \sigma_y^M \sigma_z^M]^\top,$$
$$= \left[\frac{\sum_{q=1}^{M} \left(p_x(q) - \mu_x^N\right)^2}{M}, \frac{\sum_{q=1}^{M} \left(p_y(q) - \mu_y^N\right)^2}{M}, \frac{\sum_{q=1}^{M} \left(p_z(q) - \mu_z^N\right)^2}{M} \right]^\top \tag{3}$$

The standard deviation of M samples, denoted by $\sqrt{\sigma^M}$, is given by

$$\sqrt{\sigma^M} = [\sqrt{\sigma_x^M}, \sqrt{\sigma_y^M} \sqrt{\sigma_z^M}]^\top. \tag{4}$$

TABLE I
NOTATION IN THIS WORK

Symbol	Description
$A(t)$	Sensory data of accelerometer at time instant t.
$G(t)$	Sensory data of gyroscope at time instant t.
$p_x(t)$	x-axis sensory data at time instant t.
$\mathbb{G}$	Gesture vocabulary.
$\lvert\mathbb{G}\rvert$	Size of the gesture vocabulary.
g_i	Gesture i in gesture vocabulary $\mathbb{G}$, $1 \leq i \leq \mathbb{G}$.
$\lvert g_i \rvert$	Length of a gesture i in $\mathbb{G}$, $1 \leq i \leq \mathbb{G}$.
μ^N	Mean value of N samples.
σ^M	Variance of M samples.
$\sqrt{\sigma^M}$	Standard deviation of M samples.
$\max\{a, b\}$	Maximum value between a and b.

IV. CHG: Continuous Hand Gestures Recognition

In this section, we propose a Continuous Hand Gestures recognition (CHG) technique. Fig. 3 illustrates the block diagram of CHG technique. In data acquisition step, a smart device collects the sensory data from the accelerometer and gyroscope sensors. The collected sensory data, then work as input of the data preprocessing step where the raw sensory data are compressed and coded. We create a gesture database for recognizing the gestures in gesture database creation step. The gesture spotting process automatically identifies the start and end points of each gesture. The coded sensory data finally process to the gesture reorganization step.

A. Data Acquisition

We use a smartphone (Samsung ATIV smartphone for experimental results) as a smart device for data acquisition in the CHG technique. When the system is switched on, the acceleration values are detected by the accelerometer sensor in the smartphone. Similarly, the gyroscope sensor in a smartphone detects the change in the orientations of the smartphone. We assume that the user should follow guidelines below during the data acquisition: Sensors in the smartphone are switched ON, allow to store the sensory data in the smartphone memory, and gestures should be performed as shown in Fig. 1.

B. Data Preprocessing

The accelerometer and gyroscope sensors in a smartphone are extremely sensitive and record minor hand shakes associated with hand motion. To reduce the influence of unstableness of a hand making the gestures, the data preprocessing algorithm uses the mean value of N samples of accelerometer and gyroscope sensors. We use Eq. 2 to estimate the mean value of N samples. To this end, we address the following challenge:

1) Estimation of N: The value of N reduces the size of the original sensory data up to the $1/N$ times. It is true that the large value of N reduces the size of the sensory data. However, a large value of N is not a good substitute of the original sensory data. To find an optimal value of N, we estimate the standard deviation (STD) of the sensory data using Eq. 4. Fig. 4 depicts the relationship between N and STD of the sensory data. It illustrates that $N = 4$ has a minimum STD of sensory data. The proposed CHG technique uses the value of N as 4. However, the value of N depends on the sensor specification.

2) Gesture coding: The obtained mean values of the sensory data are floating values. The floating value computation requires more time than integer values [22]. In this step, we encode the obtained mean values of sensory data into integer values. It helps to remove the floating point computation and reduce the size of the obtained mean values. The process of converting a float mean value to an integer value is called as *gesture coding*. We use the Probability Density Function (PDF) for gesture coding. Fig. 5 depicts the PDF of the obtained mean

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal

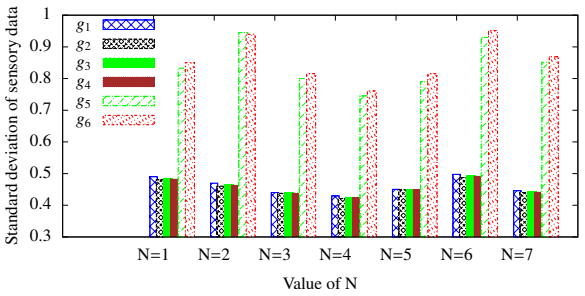


Fig. 4. Relation between standard deviation of the sensory data and N.

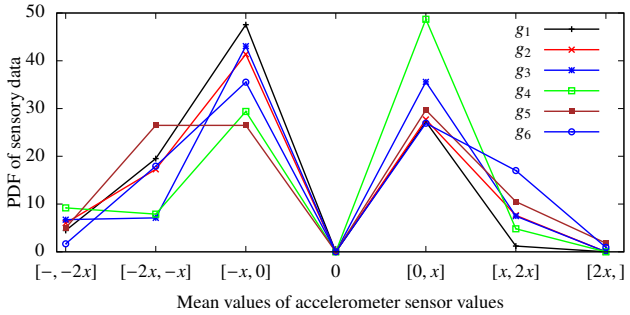


Fig. 5. Distribution of the PDF of sensory values, where $x = 9.8$.

values of sensory data of accelerometer sensor. It illustrates that the most of the values of accelerometer sensor are between -9.8 and 9.8. We also repeat the experiment for gyroscope sensor and the mean sensory data are 42, 52, 53, 55, 32, and 48 for g_1, g_2, g_3, g_4, g_5, and g_6 gestures, respectively.

Fig. 5 is used for creating the gesture code as shown in Fig. 6. Fig. 6 illustrates that the gesture code 15 is used for $[0, 0.98]$ mean value of sensory data. Finally, we use Fig. 6 for gesture coding of the obtained mean values of sensory data. The output of this step is called as *coded sensory data*. The sensory data of three-axis accelerometer sensor after data preprocessing are denoted by $\{A_x, A_y, A_z\}$ in Algorithm 1. The CHG technique repeats Algorithm 1 for gyroscope sensor and generates $\{G_x, G_y, G_z\}$.

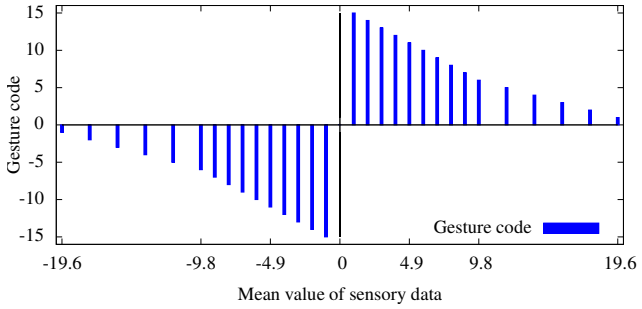


Fig. 6. Illustration of gesture code for the mean value of sensory data.

C. Gesture Database Creation

In this step, the CHG technique creates a gesture database. This is one time step. The gesture database is a standard

Algorithm 1: Data Preprocessing.

input : Sensory data;
output: Compressed and Coded sensory data;
while *Gesturing* **do**
- /* *Read N new samples from accelerometer sensor* */ $a_x(1)....a_x(N)$, $a_y(1)....a_y(N)$, $a_z(1)....a_z(N)$;
- /* *Estimate mean value using Eq. 2* */ μ_x^N, μ_y^N, μ_z^N;
- /* *Generate gesture code using* Fig. 6 */ $\mu_{\mathbf{x}}^{\mathbf{N}} \leftarrow \mu_x^N$, $\mu_{\mathbf{y}}^{\mathbf{N}} \leftarrow \mu_y^N$, $\mu_{\mathbf{z}}^{\mathbf{N}} \leftarrow \mu_z^N$;
- /* *Insert into three-axis accelerometer arrays* */ $A_x \leftarrow \mu_{\mathbf{x}}^{\mathbf{N}}$, $A_y \leftarrow \mu_{\mathbf{y}}^{\mathbf{N}}$, $A_z \leftarrow \mu_{\mathbf{z}}^{\mathbf{N}}$;

$\mathbb{A} \Leftarrow [A_x, A_y, A_z]^\top$;

gesture vocabulary which we will use in gesture recognition step (Section IV-D). Initially, a user performs a gesture g_i, where $g_i \in \mathbb{G}$ and $1 \leq i \leq |\mathbb{G}|$. The gesture code of g_i is created by Data Preprocessing step (Section IV-B) and saved in gesture database. The user repeats this step for all the gestures in the gesture vocabulary $\mathbb{G}$. At the end of this step, the gesture database has gesture codes of the gesture vocabulary $\mathbb{G}$.

• ***Gesture database normalization****:* In this paper, we consider a gesture vocabulary $\mathbb{G}$ as shown in Fig. 1. Due to the different shape of the gestures in $\mathbb{G}$, the length of all the gesture codes in gesture database may not be the same and therefore a gesture code of relatively bigger length can be recognized as multiple gesture codes of small lengths. Fig. 7 illustrates that gesture g_6 is a superset of g_3, g_1, and g_3 gestures. The gesture recognition step (Section IV-D) therefore may recognize gesture g_6 as gestures $\{g_3 \rightarrow g_1 \rightarrow g_3\}$. To remove this superset problem, we normalize the gesture codes of $\mathbb{G}$.

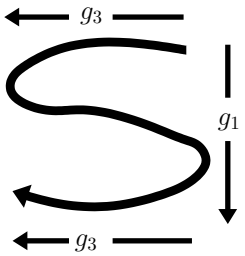


Fig. 7. Illustration of $g_6 \in \mathbb{G}$ and its subset gestures $\{g_3, g_1, g_3\} \in \mathbb{G}$.

After data preprocessing step, a gesture g_i has $\{A_{x_i}, A_{y_i}, A_{z_i}\}$ arrays for accelerometer sensor and $\{G_{x_i}, G_{y_i}, G_{z_i}\}$ arrays for gyroscope sensor, $1 \leq i \leq |\mathbb{G}|$. The Least Significant Code (LSC) and Most Significant Code (MSC) of an array A_{x_i} is the first and last elements of the array, respectively. Let $|g_i|$ denotes the length of a gesture code g_i. The maximum gesture code length in $\mathbb{G}$, denoted by g_{Max}, is given by $g_{Max} = \max\{|g_1|, |g_2|, \ldots, |g_6|\}$. For normalization of a gesture g_i, we add $\left\lceil \frac{g_{Max} - |g_i|}{2} \right\rceil$ number of LSC and MSC at the beginning and ending of the arrays of g_i, respectively. Fig. 8 illustrates an example of normalization. The normalization process is repeated for all the gesture codes in gesture database. At the end of this step, all gesture codes in gesture

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal

database are equal length, *i.e.*, $|g_1| = |g_2| = \cdots = |g_6|$. Algorithm 2 illustrates the gesture database creation step only for accelerometer sensor. In CHG, we also repeat Algorithm 2 for gyroscope sensor.

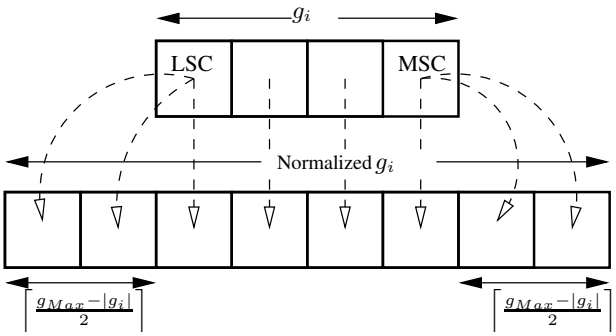


Fig. 8. Normalization of a gesture $g_i \in \mathbb{G}$.

Algorithm 2: Gesture database creation. (One time process performed by the user)

for $i \leftarrow 1$ **to** $|\mathbb{G}|$ **do**
- Perform a gesture g_i;
- Using Algorithm 1 on gesture g_i:

$$\mathbb{A}_i \Leftarrow [A_{x_i}, A_{y_i}, A_{z_i}]^\top ;$$
$$|g_i| \leftarrow \max\{|A_{x_i}|, |A_{y_i}|, |A_{z_i}|\} ;$$
$$g_{Max} \leftarrow \max_{1 \le i \le |\mathbb{G}|} |g_i| ;$$

for $i \leftarrow 1$ **to** $|\mathbb{G}|$ **do**
if $g_{Max} > |g_i|$ **then**
for *all arrays in* $\mathbb{A}_i$ **do**
for $i \leftarrow 1$ **to** $\left\lceil \frac{g_{Max}-|g_i|}{2} \right\rceil$ **do**
Add LSC of an array in $\mathbb{A}_i$ at the beginning of the array;
Add MSC of an array in $\mathbb{A}_i$ at the ending of the array;
Store $\mathbb{A}_i$ in Gesture database ;

D. Gesture Spotting and Recognition

An important problem in the gesture recognition techniques is to segment a gesture from nongesture movements, which is known a *gesture spotting*. In this section, we propose an algorithm for gesture spotting and gesture recognition.

1) Gesture Spotting: In this step, we propose a gesture spotting process to distinguish gestures from a daily hand movement. The input of the gesture spotting is the coded sensory data (*i.e.*, output of the Data Preprocessing step) and the gesture database. Initially, when a user performs continuous gestures, the sensory data of three-axis accelerometer sensor are coded using data preprocessing step and stored into the arrays A_x, A_y, and A_z. We estimate the variances of the first g_{Max} elements of the arrays A_x, A_y, and A_z using Eq. 3, where g_{Max} is estimated in Algorithm 2, N=M=g_{Max} and $i = 1$ in Eq. 2. Let $\sigma_{i,g_{Max}}$ is the maximum value of the variances of the first g_{Max} elements of the arrays A_x, A_y, and A_z. A hand movement is a gesture if $\sigma_{i,g_{Max}} \geq \sigma_{th}$, where σ_{th} is a threshold value. A small value of σ_{th} is suitable for small hand gestures. We remove first g_{Max} elements from A_x, A_y, and A_z in this case and repeat this step for next g_{Max} elements. A hand movement is a nongesture movement if $\sigma_{i,g_{Max}} \ngeq \sigma_{th}$. In this case, we remove first $p \times g_{Max}$ elements from A_x, A_y, and A_z and repeat this step for gesture spotting for next g_{Max} values, where p is a threshold value less than unity. The value of p is determined by the experimental results. In this paper, we used the value of p as 0.4 for evaluating the accuracy of the system. Algorithm 3 illustrates the gesture spotting process.

2) Gesture Recognition: The input of this step is the coded sensory data (*i.e.*, A_x, A_y, A_z) of size g_{Max}, denoted by $\mathbb{A}(g_{Max})$. In this step, the gesture $g_i \in \mathbb{G}$ of size g_{Max} is denoted by $\mathbb{A}_i(g_{Max})$, $1 \le i \le |\mathbb{G}|$. We use DTW algorithm for measuring the distance between $\mathbb{A}(g_{Max})$ and $\mathbb{A}_i(g_{Max})$, $1 \le i \le |\mathbb{G}|$. The distance between $\mathbb{A}(g_{Max})$ and $\mathbb{A}_i(g_{Max})$, denoted as $Cost_i$, is estimated by using Eq. 1. Here, the minimum distance between $\mathbb{A}(g_{Max})$ and $\mathbb{A}_i(g_{Max})$ indicates the good matching of the hand movement with the gesture $g_i \in \mathbb{G}$. We use arrays $L[\,]$ and $H[\,]$ for saving the minimum cost gesture and the cost, respectively. In continuous hand gesturing, some initial sensory data may not be a gesture. To find the closest matching between $\mathbb{A}(g_{Max})$ and $\mathbb{A}_i(g_{Max})$, we consider up to $k \times g_{Max}$ sensory data in this step.

As we have seen in Fig. 7 that the gesture g_6 is a superset of g_1, g_3, and g_1 gestures. To remove the uncertainty, we use gyroscope sensor. If the gesture g_6 is recognized as a minimum cost gesture, we repeat Algorithm 3 using sensory data of gyroscope sensor ($\mathbb{G}(g_{Max})$) and the gyroscope sensory database ($\mathbb{G}_i(g_{Max})$), $1 \le i \le |\mathbb{G}|$. Algorithm 3 illustrates the gesture recognition step. To this end, we address the following challenge:

- *Why we consider the mean value of the variances of A_x, A_y, and A_z:* The CHG technique uses a smart device for collecting the sensory data. There is no assumptions for position, direction or orientation of the device. It is true that the variance of the x-axis accelerometer sensory data is higher than y-axis and z-axis when a user moves the smart device along the x-axis. In this case, the maximum value of the variance, *i.e.*, g_{Max} is given by A_x. To remove the effect of the moving direction of the smart device, we consider the mean value of the variance of A_x, A_y, and A_z in Algorithm 3.
- *Estimation of σ_{th}:* Fig. 9 illustrates the variances of the three-axis sensory data. It shows that the mean value of the variances is more than 15 only if a gesture is performed. We repeat this process for all gestures in gesture vocabulary $\mathbb{G}$. Based on this result, we conclude that $\sigma_{th} \geq 15$ is possible only if a gesture is performed. In Algorithm 3, we therefore use $\sigma_{th} = 15$.
- *Estimation of k:* Fig. 10 illustrates the $Cost_i$ when a gesture $g_6 \in \mathbb{G}$ is performed. It shows that $k = 1$ and $k = 5$ have the equal values of the $Cost_i$ for the gestures in $\mathbb{G}$. For example, g_6 has a maximum value of $Cost_i$ at $k = 1$ and 5. Similarly, g_2 has a minimum value of $Cost_i$ at $k = 1$ and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal

Algorithm 3: Gesture Spotting and Recognition.

input : Gesture database $\mathbb{G}$, user gesture samples;
output: The recognized gesture is $L\left[\arg\min_{1\leq i\leq k} H[i]\right]$;
initialization: $j \leftarrow 1$, $k \leftarrow 1$;
initialization: $H[\,], L[\,]$;
while *Gesturing or* $k \leq 4$ **do**
  Gesture_Spotting(A_x, A_y, A_z);
for $i \leftarrow 1$ **to** k **do**
  if $L[i] = 6$ **then**
    /* Gesture g_6 is possible*/
    Gesture_Spotting(G_x, G_y, G_z);

Procedure Gesture_Spotting (P_x, P_y, P_z)
  • Generate $\mathbb{P}(g_{Max}) \Leftarrow [P_x, P_y, P_z]^\top$ using Algorithm 1
  • **if** $\max\{|P_x|, |P_y|, |P_z|\} \geq g_{Max}$ **then**
    Estimate variances of P_x, P_y, P_z for j to next g_{Max} samples using Eq. 3;
    Set $\sigma_{g_{Max}} \leftarrow$ Mean{variances of P_x, P_y, P_z };
    if $\sigma_{g_{Max}} \geq \sigma_{th}$ **then**
      /* Gesture Spotted.*/
      for $i \leftarrow 1$ **to** $|\mathbb{G}|$ **do**
        $\mathbb{P}_i(g_{Max}) \Leftarrow [P_{x_i}, P_{y_i}, P_{z_i}]^\top$;
        $Cost_i \leftarrow \text{DTW}\big(\mathbb{P}(g_{Max}), \mathbb{P}_i(g_{Max})\big)$;
      end
      $H[k] \leftarrow \min_{1\leq i\leq |G|} Cost_i$
      $L[k] \leftarrow \arg\min_{1\leq i\leq |G|} Cost_i$
      $k \leftarrow k + 1$;
      $j \leftarrow j + g_{Max}$;
    else
      /* No gesture Spotted.*/
      Set $j \leftarrow j + (p \times g_{Max})$;
    end
  end
Procedure DTW $(\mathbb{A}(g_{Max}), \mathbb{A}_i(g_{Max}))$
  $D_0 \leftarrow |\mathbb{A}(g_{Max}), \mathbb{A}_i(g_{Max})|$;
  $D_1 \leftarrow \text{DTW}\big(\mathbb{A}(g_{Max} - 1), \mathbb{A}_i(g_{Max})\big)$;
  $D_2 \leftarrow \text{DTW}\big(\mathbb{A}(g_{Max}), \mathbb{A}_i(g_{Max} - 1)\big)$;
  $D_3 \leftarrow \text{DTW}\big(\mathbb{A}(g_{Max} - 1), \mathbb{A}_i(g_{Max} - 1)\big)$;
  return $D_0 + \min\{D_1 + D_2 + D_3\}$;
return $L\left[\arg\min_{1\leq i\leq k} H[i]\right]$;

$k = 5$. We repeat this experiment for all the gestures in $\mathbb{G}$. We observe that a maximum of four iterations of the algorithm is required to recognize the gesture accurately, *i.e.*, the gesture corresponding to the lowest cost out of the four iterations is the correct gesture.

V. CHG Implementation and Results

In this section, we first discuss the implementation of CHG technique on Windows Phone. Next, we illustrate the user interface and the steps required for using CHG technique. Finally, we discuss the experimental results of CHG technique.

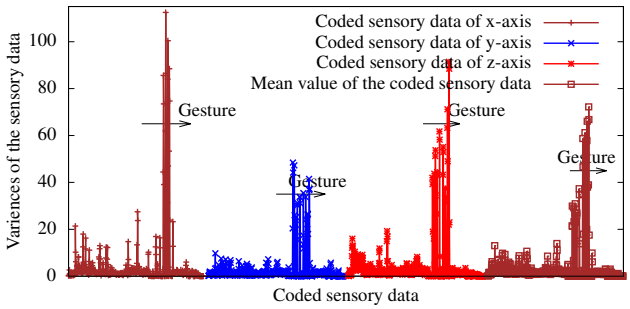


Fig. 9. Relationship between coded sensory data and variance of the data.

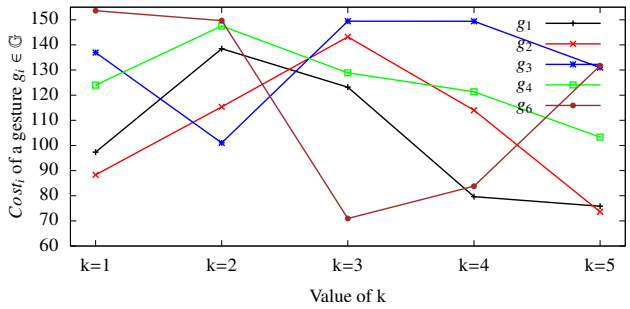


Fig. 10. Relationship between $Cost_i$ of a gesture g_i and the value of k.

A. Implementation of CHG technique

We implemented CHG technique on a Windows Phone operating system 8.0 and built an appliance controlling application to demonstrate the application and usefulness of the technique. Windows phone was chosen due to the ready availability of expertise and devices, a similar system can be implemented on any other operating system of smart devices.

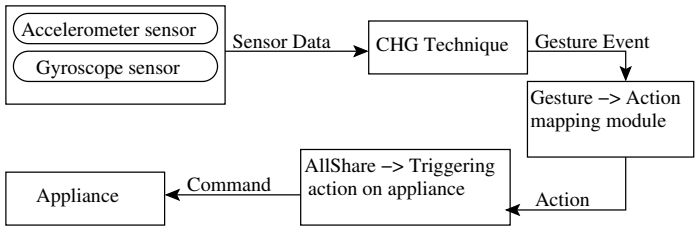


Fig. 11. A Block diagram of the implementation of CHG technique.

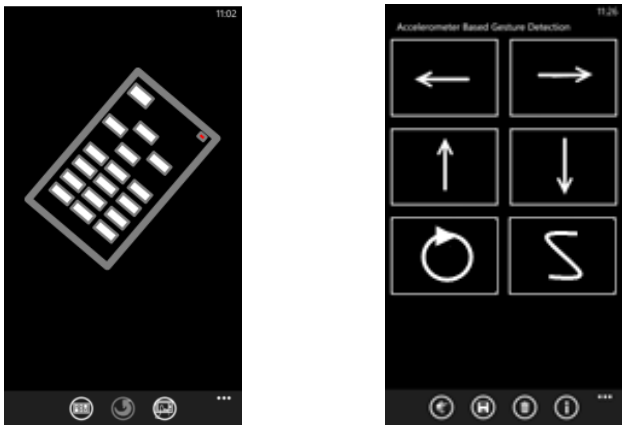
A block diagram of the implementation of CHG technique is shown in Fig. 11. It consists the following steps: Initially, CHG technique uses the procedures given in [23] to get the sensory data for windows phone. The procedures show that an application on windows phone can retrieve sensor data through the application program interface by the operating system. Next, the sensory data are sent to CHG technique which fires a gesture event when a gesture is detected. In the implementation, we consider the gesture vocabulary as shown in Fig. 1. Once an event is fired by the CHG module, the application module converts these gestures into actions as per the application requirement. We also developed an appliance control application by using the Samsung AllShare

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal

framework wherein the gestures are translated to controls on the particular appliance. Finally, Table II maps the gestures in $\mathbb{G}$ corresponding to their actions on a television and a music player.

B. User Interface of CHG technique

This subsection explains the procedure of using the proposed CHG technique. The user repeats the following steps for using the CHG technique: Initially, the user presses and holds one of the buttons displayed on the training screen for starting the program of CHG technique as shown in part (a) of Fig. 12. Part (b) of Fig. 12 illustrates GUI of six gestures as define in gesture vocabulary $\mathbb{G}$. The user performs the gesture displayed on the pressed button while holding it to create a gesture database. Next, the user goes to the control screen, and performs the gestures. Finally, the smart device uses the Samsung AllShare service for transferring the actions to the appliances. We use Samsung Smart television as an appliance in the experiment. The television performs the action based on the gesture. Table II maps the gestures in $\mathbb{G}$ corresponding to their actions on television, music player, and washing machine.



(a) Control screen. (b) Gesture vocabulary.

Fig. 12. Snapsorts of the prototype for CHG technique.

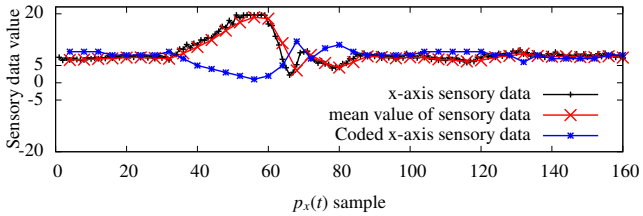
TABLE II
MAPPING BETWEEN GESTURES AND THEIR ACTIONS.

Gesture	Television	Music Player	Washing Machine
g_1	Previous channel	Previous song	Air dry level up
g_2	Next channel	Next song	Air dry level down
g_3	Volume up	Volume up	Water level up
g_4	Volume down	Volume down	Water level down
g_5	Play or pause	Play or pause	Start or Pause
g_6	Gesture start	Gesture start	Gesture start

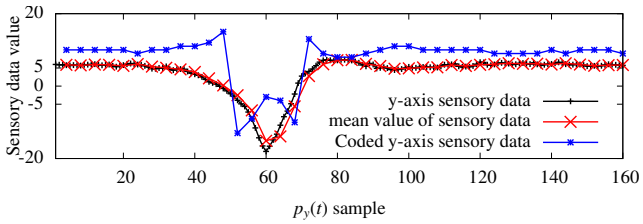
C. Experiment Results

Experiments were conducted to assess the performance of the proposed CHG technique and the gesture based interaction prototype. The ten participants were Samsung R&D Institute Bangalore India employees aged 22-50 who had used smartphones for at least five years so that they were familiar with the operations. Each gesture was acquired from each participant around 100 times. The total 1000 (100×10) repeats were included in our gesture database.

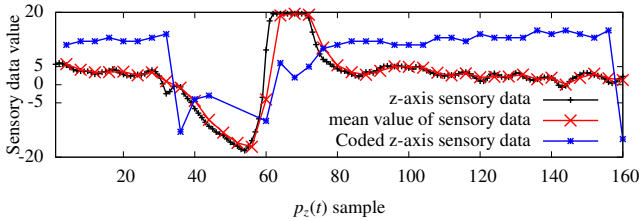
1) Gesture coding results: Fig. 13 illustrates the raw, mean, and coded sensory data of g_4 and g_5 gestures, respectively. The sensory data of three-axis accelerometer sensor is recorded when a user was continuously performing the gestures. The black color line in the figure denotes the raw values of sensory data. The figure illustrates that the sensory data is extremely sensitive. We used Algorithm 1 to estimate the mean and coded sensory data of the gestures. The coded values are integer values of the mean sensory data and therefore help to remove the floating point computation. Fig. 13 shows that g_4 gesture was performed between 40 to 80 samples of sensory data.



(a) Raw, mean, and coded x-axis sensory data of g_4 gesture.



(b) Raw, mean, and coded y-axis sensory data of g_4 gesture.



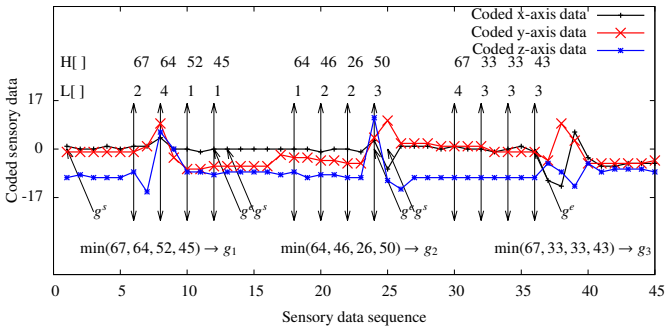
(c) Raw, mean, and coded z-axis sensory data of g_4 gesture.

Fig. 13. Illustration of the raw, mean, and coded sensory data of g_4 gesture.

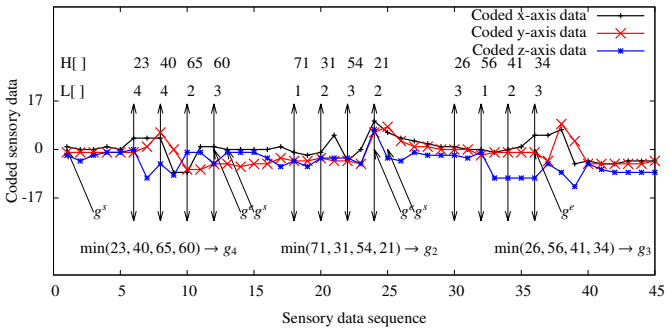
2) Gesture spotting and recognition results: Fig. 14 illustrates gesture spotting and recognition results. The sensory data of three-axis accelerometer and gyroscope sensors recorded when a user was continuously performing the gestures. The stream of coded sensory data rising above the threshold (σ_{th}) denotes the start point the end point of a segment. Using Algorithm 3, we estimated the arrays H[] and L[]. Part (a) of Fig. 14 illustrates that the minimum value of the array H[] is 45 and therefore gesture g_1 performed in the first segment. Similarly, g_2 and g_3 gestures performed in second and third segments in part (a) of Fig. 14, respectively. Part (b) of Fig. 14 illustrates the gesture spotting for $g_4g_2g_3$ sequence.

3) Accuracy and efficiency of CHG technique: In order to evaluate the accuracy and efficiency of CHG technique,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal



(a) Illustration of the gesture spotting for $g_1g_2g_3$ sequence.



(b) Illustration of the gesture spotting for $g_4g_2g_3$ sequence.

Fig. 14. Illustration of the gesture spotting.

TABLE III
CONFUSION MATRIX OF THE PROPOSED CHG TECHNIQUE (IN %).

Actual Gesture \ Predicted Gesture	g1	g2	g3	g4	g5	g6
g1	93.5	6.5	0	0	0	0
g2	6.25	93.75	0	0	0	0
g3	1.75	0	91.2	7.05	0	0
g4	2.5	0	2.5	92.5	0	2.5
g5	0	2.5	0	2.5	95.0	0
g6	2.5	0	0	2.5	0	95.0

we conduct an experiment. In this experiment, we use the accelerometer and gyroscope sensors of the smartphone. Table III is a confusion matrix of the experiment. The mean accuracy of this experiment is 94%. Table III illustrates that the gyroscope sensor almost removes the confusion of g_6 gesture. We also observed from this experiment that the following factors limiting the accuracy in the gestures: user behavior and the wireless communication link between the smartphone and the device (*e.g.*, television in our experiment). A user performing a gesture g_1 (down gesture), the hand first moves upwards and then downwards. In some cases, the participants perform the hand up action more accurately then the hand down and therefore CHG technique detects the hand action as gesture g_2 (up gesture). The other factor is a wireless communication link. Due to the wireless communication, the smartphone and the television some time does not connect perfectly or the gesture may not transmit successfully. We also observed that the proposed system predict wrong gesture in case of partially performed gestures.

Table IV is a confusion matrix for new users and experienced users. Here, a new user uses the proposed gesture technique first time. An experienced user already used the proposed gesture technique multiple times. Table IV shows that the accuracy of a new user is less because it uses the proposed gesture first time.

TABLE IV
CONFUSION MATRIX OF PROPOSED CHG IN PERCENTAGE FOR DIFFERENT USERS (SHOWN IN THE FORM OF [NEW USER, EXPERIENCED USER]).

Actual Gesture \ Predicted Gesture	g1	g2	g3	g4	g5	g6
g1	[90,96]					
g2		[90,97]				
g3			[88,94]			
g4				[86,95]		
g5					[89,98]	
g6						[89,98]

Portrait Orientation Landscape Orientation

Fig. 15. Orientation of a smart device.

Next, we illustrate the impact of the orientation of the smart device. In this result, we use portrait orientation and landscape orientation of a smart device as shown in Fig. 15. Table V illustrates the confusion matrix when a user uses the same orientation (portrait orientation) during training time and application time. Table VI illustrates a confusion matrices when a user uses the different orientations of the smart device, *i.e.*, landscape orientation and portrait orientation during training time and application time, respectively. The results illustrate that the accuracy of the proposed method is almost equal in all cases. This is because, the gyroscope sensor automatically detects the orientation of the device, and therefore removes the impact of the orientation on the results.

4) Comparison with other method: Next, we compare CHG technique proposed in this work with one of the existing techniques in the literature viz., uWave [22]. To the best of our knowledge, most of the existing gestures recognition techniques are not continuous, *i.e.*, they use some user action

TABLE V
CONFUSION MATRIX IN PERCENTAGE WHEN USER USES PORTRAIT ORIENTATION DURING TRAINING TIME AND APPLICATION TIME.

Actual Gesture \ Predicted Gesture	g1	g2	g3	g4	g5	g6
g1	[91,97]					
g2		[88,98]				
g3			[89,93]			
g4				[84,93]		
g5					[87,96]	
g6						[87,99]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal

TABLE VI
CONFUSION MATRIX OF PROPOSED CHG IN PERCENTAGE WHEN A USER USES LANDSCAPE ORIENTATION AND PORTRAIT ORIENTATION DURING TRAINING TIME AND APPLICATION TIME, RESPECTIVELY.

Actual Gesture \ Predicted Gesture	g_1	g_2	g_3	g_4	g_5	g_6
g_1	[92,96]					
g_2		[89,97]				
g_3			[88,94]			
g_4				[85,94]		
g_5					[86,97]	
g_6						[88,98]

TABLE VII
CONFUSION MATRIX OF UWAVE [22] (IN %).

Actual Gesture \ Predicted Gesture	g_1	g_2	g_3	g_4	g_5	g_6
g_1	85	12.5	2.5	0	0	0
g_2	20	80	0	0	0	0
g_3	10	0	85	5	0	0
g_4	5	0	7.5	75	12.5	0
g_5	0	15	0	30	55	0
g_6	32.5	0	5	22.5	0	40

TABLE VIII
CONFUSION MATRIX OF UWAVE [22] IN PERCENTAGE FOR DIFFERENT USERS (SHOWN IN THE FORM OF [NEW USER, EXPERIENCED USER]).

Actual Gesture \ Predicted Gesture	g_1	g_2	g_3	g_4	g_5	g_6
g_1	[82,88]					
g_2		[77,82]				
g_3			[83,86]			
g_4				[72,77]		
g_5					[52,57]	
g_6						[39,43]

TABLE IX
DIFFERENCE BETWEEN CHG TECHNIQUE AND UWAVE [22] (IN %).

Actual Gesture \ Predicted Gesture	g_1	g_2	g_3	g_4	g_5	g_6
g_1	+8.5					
g_2		+13.75				
g_3			+6.2			
g_4				+17.5		
g_5					+40	
g_6						+55

for determining the start and end points of the gestures. Some existing continuous gestures recognition techniques use advance and large number of extra sensors as we have seen in Section II.

The uWave is a three-axis accelerometer sensor based gesture recognition technique. Table VII is a confusion matrix of uWave. Table VIII is a confusion matrix of uWave for new users and experienced users. Table IV and Table VIII indicate that the accuracy of the uWave is less for different users. Table IX shows the difference between the proposed CHG technique and uWave. It shows the correctly recognized the continuous gestures in percentage. The mean accuracy of this experiment is 70% in uWave. The accuracy of the uWave is less because it does not use data normalization step and three-axis gyroscope sensor. The uWave also uses different coding algorithm and gesture recognition technique. The unequal size of the gestures in gesture database creates the confusion in the gesture recognition step in the uWave. The proposed CHG uses the data normalization and therefore perfectly handle this unambiguity. We can simply conclude from this experiment that the data normalization step is a necessary step in the continuous gestures.

5) Impact of the large size vocabulary: The size of the gesture vocabulary in this paper consists six gestures as shown in Fig. 1. In this section, we illustrate the accuracy of the proposed gesture recolonization technique on large size gesture vocabulary. We consider a gesture vocabulary of twelve gestures as shown in Fig. 16. Table X is a confusion matrix of the experiment. It illustrates that the proposed technique provides the sufficient accuracy of gesture recolonization in a large size vocabulary. An interesting observation from this result is that the accuracy of the mirror image gestures (*e.g.*, (g_5, g_{11}) and (g_6, g_{12})) is almost equal. This is because the proposed technique uses gyroscope sensor for monitoring the hand directions.

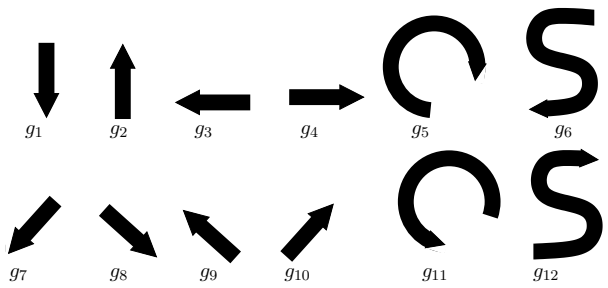


Fig. 16. Illustration of the gesture vocabulary with twelve gestures.

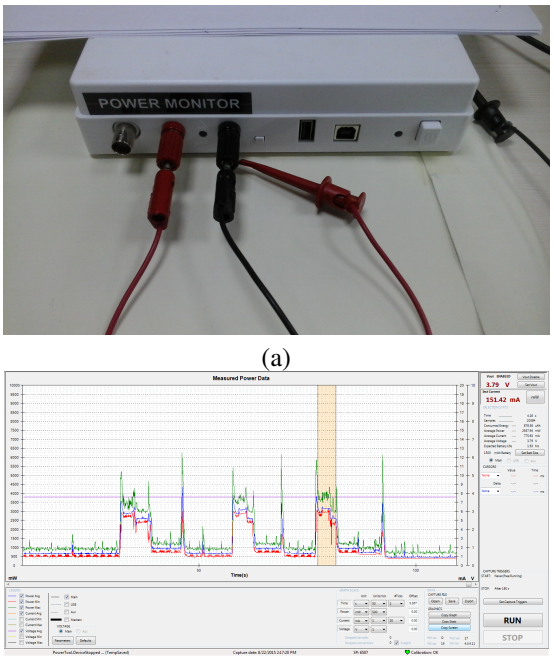
TABLE X
CONFUSION MATRIX OF THE PROPOSED CHG TECHNIQUE (IN %) WITH TWELVE GESTURES.

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}	g_{12}
g_1	93.2											
g_2		93.1										
g_3			94.2									
g_4				93.1								
g_5					93.2							
g_6						94.7						
g_7							92.1					
g_8								94.2				
g_9									94.5			
g_{10}										93.2		
g_{11}											93.1	
g_{12}												94.1

6) Estimation of Power Consumption: Finally, we measured the power consumption of the proposed technique. We use Monsoon power monitor as shown in Fig. 17(a). Fig. 17(b) illustrates the GUI of the application when a smart device

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2581023, IEEE Sensors Journal

(Samsung Note II smartphone) is connected with Monsoon power monitor. It shows that the average power consumption per unit time is 450mW and 2930mW when the smart device is ideal and during recognition of the continuous hand gestures, respectively.



(a)

(b)

Fig. 17. Parts (a) and (b) show a Mansoon power monitor device and the snapshot of the power consumption using the device, respectively.

VI. Conclusion

In this paper, we present a CHG technique that is capable of continuous recognition of hand gestures using a smart device. A key attribute of our CHG technique is that it does not require any active user intervention in terms of the start or end points of the continuous gestures. The proposed CHG technique employs a set of novel algorithms to resolve ambiguity in gestures during the influence of unstableness of the sensors, using gesture coding and database normalization. We demonstrated an application of the proposed CHG technique to develop a HMI system using Samsung AllShare wireless service. The prototype demonstrated that the CHG technique is able to perform accurate gesture detection and is suitable for HMI. We believe that the proposed CHG technique can also be used on any wearable device which has accelerometer and gyroscope sensors and interacts with a mobile device. The proposed algorithms in CHG technique may be utilized in pattern classification.

References

[1] P. Jung, G. Lim, S. Kim, and K. Kong, "A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 485–494, 2015.

[2] Y. Park, J. Lee, and J. Bae, "Development of a wearable sensing glove for measuring the motion of fingers using linear potentiometers and flexible wires," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 1, pp. 198–206, 2015.

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

[4] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, Aug 2013.

[5] B. Sayed, I. Traor, I. Woungang, and M. S. Obaidat, "Biometric authentication using mouse gesture dynamics," *IEEE Systems Journal*, vol. 7, no. 2, pp. 262–274, 2013.

[6] M. Khezri and M. Jahed, "A neuro-fuzzy inference system for semg-based identification of hand motion commands," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1952–1960, 2011.

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

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

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

[10] X. Ji and H. Liu, "Advances in view-invariant human motion analysis: A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 13–24, 2010.

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

[12] A. Varkonyi-Koczy and B. Tusor, "Human-computer interaction for smart environment applications using fuzzy hand posture and gesture models," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 5, pp. 1505–1514, May 2011.

[13] F. Erden and A. Cetin, "Hand gesture based remote control system using infrared sensors and a camera," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, pp. 675–680, Nov 2014.

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

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

[16] H. Liang, J. Yuan, and D. Thalmann, "Parsing the hand in depth images," *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1241–1253, 2014.

[17] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 4, pp. 461–482, 2008.

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

[19] J. Cheng, X. Chen, and M. Shen, "A framework for daily activity monitoring and fall detection based on surface electromyography and accelerometer signals," *IEEE J. Biomedical and Health Informatics*, vol. 17, no. 1, pp. 38–45, 2013.

[20] B. Nguyen, W. Tay, and C. Chui, "Robust biometric recognition from palm depth images for gloved hands," *IEEE Transactions on Human-Machine Systems*, vol. PP, no. 99, pp. 1–6, 2015.

[21] "Allshare," Available 2015, http://content.samsung.com/uk/contents/.

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

[23] "Windowsphone," Available 2015, https://dev.windows.com/en-us/windows-apps.