Multimed Tools Appl (2016) 75:9669–9684
DOI 10.1007/s11042-015-2782-3

Feature design scheme for Kinect-based DTW human gesture recognition

Ing-Jr Ding[1] · **Che-Wei Chang**[1]

Received: 26 September 2014 / Revised: 29 May 2015 / Accepted: 26 June 2015 /
Published online: 11 July 2015
© Springer Science+Business Media New York 2015

Abstract Feature selection is a crucial factor in Kinect-based pattern recognition, including common human gesture recognition. For Kinect-based human gesture recognition, the information contained in the feature extracted for gesture recognition is conventionally the (x,y,z) coordinates of the primary joints in the human body. However, such traditionally used feature information containing only joint positions is apparently insufficient for clearly describing the characteristics of human activity patterns. This paper proposes a feature design scheme involving hybridizations of joint positions and joint angles for human gesture recognition with the Kinect camera. The presented feature design method effectively hybridizes the 20 main human joint positions captured by the Kinect camera and the joint angle information of 12 critical joints, along with significant angle variations when a gesture is made. The method is employed in dynamic time warping (DTW) gesture recognition. When the proposed feature design method is used for Kinect-based DTW human gesture recognition, it derives an appropriately sized feature vector for each of the gesture categories in the DTW-referenced template database according to the activity characteristics of a certain category of gestures. Experiments on Kinect-based DTW gesture recognition involving 14 common categories of human gestures show that the feature determined using the proposed approach is superior to that obtained using the conventional approach, which considers only the joint position information.

Keywords Kinect camera · Human gesture recognition · Gesture feature · DTW · Recognition performance

1 Introduction

Human computer interaction (HCI) design has become a challenging technical problem in recent years. With the development of technology, machine manipulations or device control is

✉ Ing-Jr Ding
ingjr@nfu.edu.tw

[1] Department of Electrical Engineering, National Formosa University, Yunlin County, Taiwan

now achieved through advanced "noncontact" operations, in which biological characteristics of a person, such as acoustic data consisting of speech utterances and image data comprising human gestures, are acquired and used as control commands or analytic signals that are input into an identity verification system. Automatic speech recognition and automatic speaker recognition techniques, which involve the comparison of a person's voice pattern with a prerecorded voice pattern, are considered mature techniques [3, 6]. However, the presence of background voices can hamper speech recognition. A solution to this problem is to use human gesture recognition techniques for converting a person's gestures into control commands. Following the development of speech recognition, considerable attention has been paid to gesture recognition recently. The appearance of the Microsoft Kinect camera in the market has further accelerated the development of gesture recognition techniques [13].

The Kinect camera is a useful sensor that performs a fundamental analysis of the position data of a 3D object in acquired image data. Numerous Kinect-related applications have been developed, most of which involve human gesture recognition using Kinect-captured gesture data [4, 5, 9, 16, 17]. In [17], a finger-writing system that recognizes characters written in the air without requiring any additional handheld devices was presented. The system differs from well-known handwriting recognition systems in which text images are used for recognizing patterns, and it involves the use of Kinect-captured finger motion gesture images for recognizing patterns. In the study of Qian et al. [9], a gesture-based remote human-robot interaction system was developed. The developed system employs a gesture recognition technique to classify Kinect-captured gesture commands of an active user, and recognized gesture commands can then be used to control the action of a remote robot. In addition, a golf swing classification system involving the use of the Kinect camera was presented by Zhang et al. [16], in which the Kinect camera is used to capture a person's golf swing gesture and the gesture is then classified; a person learning golf could effectively learn the standard swing gesture by using the system. In the works of [4] and [5], Kinect-based gesture recognition with user adaptation is developed. The utilization of Kinect-based gesture recognition to the application of the humanoid robot imitation is further investigated in [5]. The objective of the current Kinect-related studies was to achieve Kinect-based human gesture recognition in a particular application or to improve the recognition performance of human gesture recognition by adjusting model parameters of gesture classification models using the test operator's action gesture data. Studies in the exploration of fine gesture feature designs for Kinect-based gesture recognition are rarely seen.

In this study, 14 common human gesture categories were employed for the command control of a smart home or a smart office. The dynamic time warping (DTW) method [10] was adopted for Kinect-based gesture recognition, and a feature design scheme was proposed for the recognition system. Figure 1 depicts the framework of the feature design scheme. In Kinect-based DTW gesture recognition involving the developed feature design scheme, the optimal gesture feature type is derived for each of 14 gesture categories. The most popular classification methods used for Kinect-based gesture recognition are DTW and the hidden Markov model (HMM) method. The HMM method is a model-based classification technique, and it is required for establishing a statistical classification model before gesture recognition. By contrast, DTW is categorized as a feature-based classification technique that is simple and computationally fast. A performance evaluation report of HMM and DTW gesture recognition was provided in [2]. Studies of Kinect-based gesture recognition involving the HMM method were presented in [7, 8, 15] and studies of Kinect-based gesture recognition

using DTW were discussed in [1, 11, 12, 14]. The studies that have used DTW for Kinect-based gesture recognition can be divided into two categories: 1) those using general DTW in a specific application for achieving a certain purpose such as a personal rehabilitation exercise assistant [11] and user identification and authentication [14], and 2) DTW template matching improvements on recognition calculations, such as probability-based DTW [1] and fuzzified DTW [12]. However, studies on feature analysis or feature design of gesture templates are rare. For feature-based DTW recognition, the feature of the template has the most significant effect on recognition performance, particularly for a DTW gesture recognition system using Kinect-captured data. As shown in Fig. 1, the current study developed a Kinect-based DTW gesture recognition system with a feature design scheme for the DTW-referenced template database and a corresponding DTW recognition scheme that is compatible with various feature type settings in DTW-referenced templates.

The proposed Kinect-based DTW gesture recognition system with a feature design scheme has several merits compared with that without a feature design scheme:

- It contains abundant and accurate 3D gesture feature information in DTW-referenced template databases.
- It is robust against the irregular gesture category since it derives the appropriate feature type.
- It shows superior performance and greater flexibility compared with conventional DTW gesture recognition systems with only invariable features.
- It is more competitive in recognition performance compared with the Kinect-based gesture recognition system involving the HMM method.

For the last advantage, the presented feature design scheme for Kinect-based DTW gesture recognition can enhance the recognition performance compared with conventional DTW gesture recognition, and therefore, the recognition performance is higher. Such a feature design scheme is highly suitable for the feature-based DTW method proposed in this study. Compared

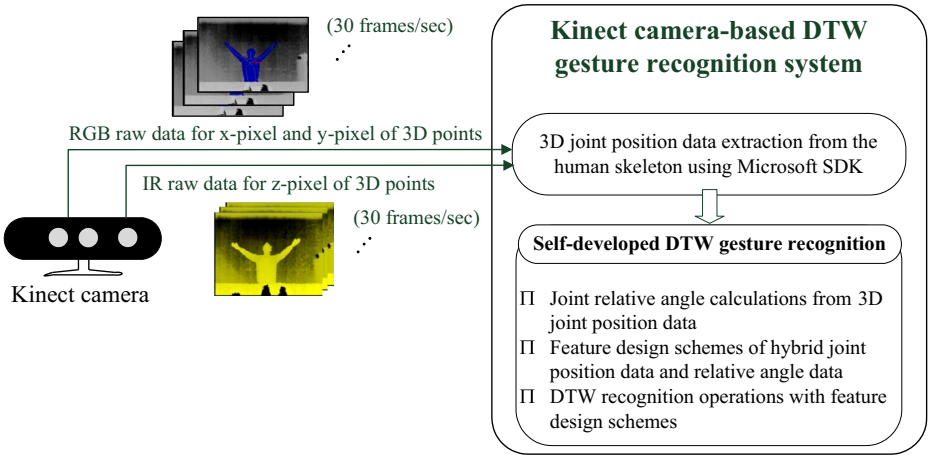

Fig. 1 The proposed feature design scheme for Kinect camera-based DTW gesture recognition using both Microsoft SDK and self-developed recognition techniques

with the model-based HMM recognition method in which a statistical model for gestures should be trained in advance, the feature design scheme developed in this study can be used to establish a model-like DTW-referenced template database, and therefore, feature-based DTW with the proposed method is superior to the model-based HMM method.

2 Kinect-based DTW human gesture recognition using joint positions

RGBD data containing information on both image data and depth data are captured by the Kinect camera [13]. For facilitating gesture recognition, the RGBD data can further be analyzed and transformed into human joint position data for a human skeleton. When a user makes a gesture, a series of continuous frames containing (x,y,z)-coordinate information of the joint positions is extracted. Figure 2 shows a human skeleton captured by the Kinect camera; the (x,y,z) coordinates of 20 joint points are calculated. In this study, the (x,y,z) coordinates of the 20 joint points were determined using the Microsoft software development kit (SDK). As shown in Fig. 2, 12 joints were defined as key joints, and they had relatively large motion variations. In the human skeleton defined by Microsoft Kinect, there are a total of 20 joints, which are distributed throughout the skeleton [13]. A frame rate of 30 frames/s was used in Kinect, and a frame had a dimension size of 60, corresponding to the three-dimensional (x,y,z)-coordinate positions of the 20 joint points. Equation (1) shows the joint position

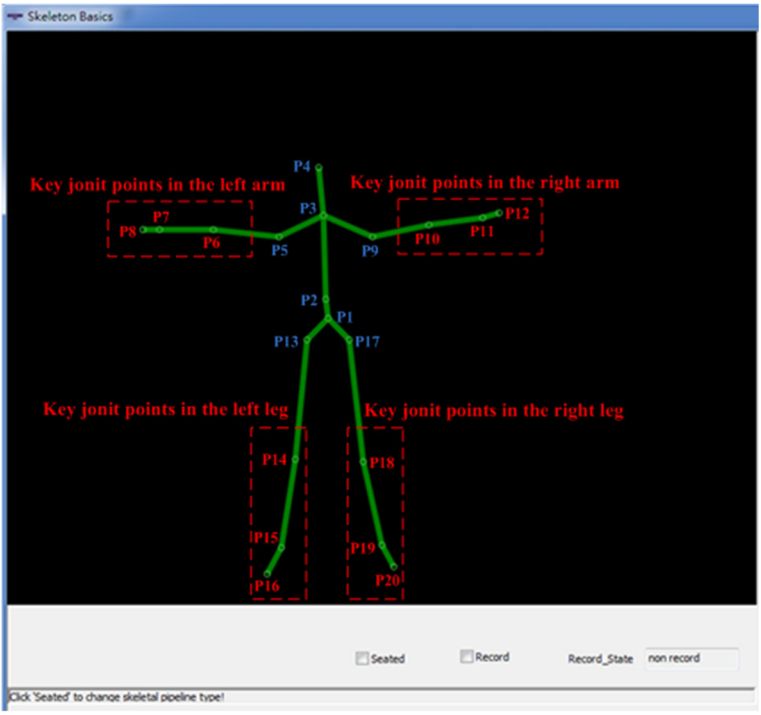

Fig. 2 The human skeleton captured by the Kinect camera where 20 joint point (x,y,z)-coordinate positions are calculated, and 12 joints are defined as key joints with large motion variations

features of n frames captured by the Kinect camera; each of the n gesture frames contained three-dimensional coordinate positions of the 20 joint points.

$$\textit{Positions of } 20 \textit{ joints of the frame } m = P_{ij}(m), \quad m = 1,2,\ldots,n, \quad i = 1,2,\ldots,20, \quad j = x,y,z \tag{1}$$

In this study, human gesture recognition using Kinect-captured joint positions (i.e., (x,y,z) coordinates of joint positions) was conducted using a DTW classification method. Owing to its simplicity, DTW is a popular classifier for pattern recognition using template matching. The main operation in DTW is matching the referenced gesture template and a test gesture template. This operation is explained in this section. DTW is a type of dynamic programming method, and it is highly effective in determining the degree of similarity between the referenced gesture template and the test gesture template in the time domain for gesture recognition [10]. The main operations of DTW gesture recognition by the Kinect camera would be introduced herein. In the current study, Kinect-based DTW gesture recognition involved two stages: the training stage for establishing the database of DTW-referenced gesture templates, and the test stage for performing matching the test template with each of the referenced templates established in the training stage. When performing DTW template matching in the test stage, a low distortion value between the test template and the reference template indicates a high degree of similarity between them.

In DTW template matching, an attempt is made to find an optimal comparison path between the test template feature vector and the referenced template feature vector [10]. Generally, in conventional Kinect-based DTW human gesture recognition, the feature of gesture signals used for DTW template matching is the joint positions. Figure 3 shows the template matching calculations of Kinect camera-based DTW gesture recognition, in which the feature vector of joint positions is used. The test gesture consists of T gesture

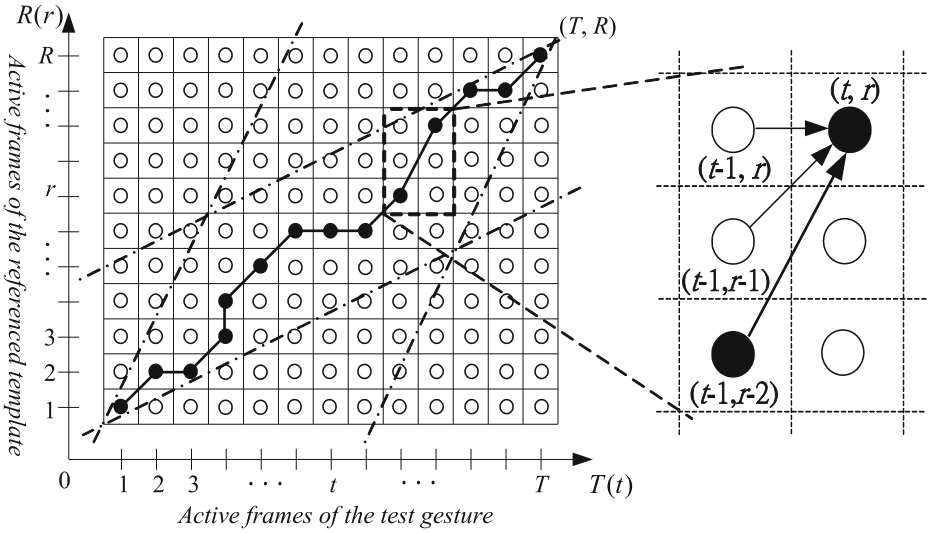

Fig. 3 Template matching calculations of Kinect camera-based DTW gesture recognition

frames, and an arbitrary frame (an arbitrary feature vector) is denoted by t. The gesture of the reference template consists of R gesture frames and an arbitrary frame is represented as r. The distortion between the T and R gesture frames can be represented as $d[T(t), R(r)]$. For frame t and frame r, the feature vectors are as follows:

$$T(t) = \left[P_{1x}(t), P_{1y}(t), P_{1z}(t),\ P_{2x}(t), P_{2y}(t), P_{2z}(t), \ldots, P_{20x}(t), P_{20y}(t), P_{20z}(t)\right], \quad t = 1, 2, \ldots, T \tag{2}$$

$$R(r) = \left[P_{1x}(r), P_{1y}(r), P_{1z}(r),\ P_{2x}(r), P_{2y}(r), P_{2z}(r), \ldots, P_{20x}(r), P_{20y}(r), P_{20z}(r)\right], \quad r = 1, 2, \ldots, R. \tag{3}$$

The starting point and the ending point of the comparison path for DTW template matching are $(T(1), R(1))=(1, 1)$ and $(T(M), R(M))=(T, R)$, respectively. If it is assumed that $(T(0), R(0))=(0, 0)$ and $d(0, 0)=0$, the accumulated shortest distance used for selecting the optimal source path can be represented as

$$min\ D(t,r) = d(t,r) + min\left\{\begin{array}{c} distance(t\text{-}1, r) \\ distance(t\text{-}1, r\text{-}1) \\ distance(t\text{-}1, r\text{-}2) \end{array}\right\}, \tag{4}$$

where $D(t, r)$ is the shortest distance from the starting position $(0, 0)$ to position (t, r). When DTW template matching is completed, the recognition outcome is the label of the referenced template with the smallest value of $D(T, R)$.

The conventional Kinect-based DTW human gesture recognition scheme involving joint positions is inefficient and ineffective with regard to gesture recognition performance because the gesture feature information acquired (consisting of only joint positions) is insufficient. For overcoming this problem and enhancing the recognition performance, a feature design scheme involving the determination of the appropriate hybridization of joint positions and joint angles is presented in the next section.

3 Proposed feature design scheme involving hybridization of joint positions and joint angles for Kinect-based DTW human gesture recognition

Figure 4 shows the training and testing phases of the proposed feature design method used for Kinect-based DTW human gesture recognition. This section describes the proposed feature design method, which involves the hybridization of joint positions and joint angles for Kinect-based DTW human gesture recognition. Joint angle information of 12 human key joints with significant activity variations is introduced first (the block of calculations for the 12 key joint angles in Fig. 4), and this is followed by a feature determination algorithm, developed in the current study, for each of the gesture categories in the referenced template database of DTW gesture recognition (the block of optimal hybridizations of joint positions and angles in Fig. 4). Kinect-based DTW human gesture recognition (based on various categories of designed features) for each gesture categorization is presented at the end of this section (the block of feature selections in Fig. 4).

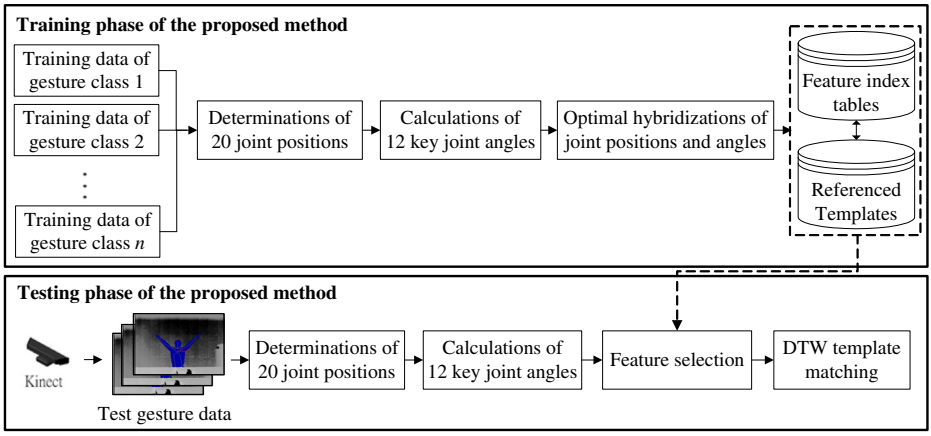

Fig. 4 Training and testing phases of the proposed feature design method for Kinect-based DTW human gesture recognition

3.1 Joint angle information for 12 human key joints with significant activity variations

In the conventional Kinect-based DTW human gesture recognition method, the feature used for representing the gesture is the joint position information. However, this gesture information is insufficient due to ineffective feature extraction, and therefore the recognition performance of the conventional method is unsatisfactory. In the proposed method, for feature extraction from Kinect-captured gesture data, joint angle information is considered in addition to joint position information.

Joint position information refers to the relative angle between two joints, with a third joint being considered the original point. As shown in Fig. 2, 20 human joints are defined in the Kinect-captured human skeleton: P1 to P20. Among these 20 joints, 12 joints with the apparent motion variation were defined as "key joints" in this study. In the captured human skeleton of Fig. 2, these 12 key joints are P6, P7, P8, P10, P11, P12, P14, P15, P16, P18, P19, and P20, and they correspond to the left elbow, left wrist, left hand, right elbow, right wrist, right hand, left knee, left ankle, left foot, right knee, right ankle, and right foot, respectively. The primary rationale for the determination of the 12 key joints is that they introduce an apparent change in the joint position information when specific gestures are performed; this apparent change also introduces a significant variation in the joint angle information. The remaining eight joints not included in the set of key joints provide considerably less information on the variation of the joint angle information available, and therefore, these joints are used as auxiliary joint points or the original joint points when it is necessary to calculate the joint angle. In this study, among the eight joints, three joints were considered to assist calculations of the joint angle of the key joint, with one original joint point and two auxiliary joint points, which is detailed as follows.

The method for obtaining the relative joint angle information for a key joint is based on the simple concept that a two-dimensional plane is basically composed of three different joint points. The joint points P4, P3, and P1 in Fig. 2 are defined as the original point, auxiliary point of the key joint points in the upper limb, and auxiliary point of the key joint points in the lower limb, respectively; these points are considered as invariable fixed points in the two-

dimensional plane for conveniently calculating the relative joint information of the 12 key joints. For each of the 12 key joints, the relative joint information mainly relating to angles between the indicated key joint point and the auxiliary joint point in the *x*-*y* plane and *y*-*z* plane is derived. Figure 5 illustrates an example of relative joint angle information: the joint angle information for key joint P8 when joint P4 and joint P3 are used as the original point and auxiliary point, respectively. Another example is presented in Fig. 6, which shows the relative joint angle information of key joint P16 when joint P4 is the original point and joint P1 is the auxiliary point. As shown in Figs. 5 and 6, key joint P8 contains two relative angles, the *x*-*y* plane angle $A_{xy}(P8)$ and the *y*-*z* plane angle $A_{yz}(P8)$; similarly, two relative angles, the *x*-*y* plane angle $A_{xy}(P16)$ and the *y*-*z* plane angle $A_{yz}(P16)$, are included in the relative angle information for key joint P16. The relative angle information for the remaining ten key joints—P6, P7, P10, P11, P12, P14, P15, P18, P19, and P20—with angles $A_{xy}(P6)$ and $A_{yz}(P6)$, $A_{xy}(P7)$ and $A_{yz}(P7)$, $A_{xy}(P10)$ and $A_{yz}(P10)$, $A_{xy}(P11)$ and $A_{yz}(P11)$, $A_{xy}(P12)$ and $A_{yz}(P12)$, $A_{xy}(P14)$ and $A_{yz}(P14)$, $A_{xy}(P15)$ and $A_{yz}(P15)$, $A_{xy}(P18)$ and $A_{yz}(P18)$, $A_{xy}(P19)$ and $A_{yz}(P19)$, and $A_{xy}(P20)$ and $A_{yz}(P20)$, respectively, is derived in a similar manner.

In addition to the 60 conventional joint position parameters (the *x*-coordinate, *y*-coordinate, and *z*-coordinate obtained for each of the 20 joints), 24 relative joint angle parameters (2 relative angles derived from each of the 12 key joints) can be additionally employed as gesture feature information for increasing the accuracy of gesture descriptions. The hybridization of joint positions and joint angles for use as the feature vector of a Kinect-captured gesture frame is designed as follows:

$$\textit{Gesture feature of a frame} = \left[P_{ij}\, A_{\mathrm{xy}}(P_k)\, A_{\mathrm{yz}}(P_k)\right],\ i = 1,2,\ldots,20,\ j = x,y,z,\ k = 1,2,\ldots 12, \tag{5}$$

where P_k denotes the *k*th key joint among the 12 key joints, $A_{\mathrm{xy}}(P_k)$ is the *x*-*y* plane angle of the *k*th key joint, and $A_{\mathrm{yz}}(P_k)$ represents the *y*-*z* plane angle of the *k*th key joint. Compared with the feature vector of a gesture frame in (1) that contains only 60 position parameters, the improved feature vector of a gesture frame in (5) consists of a maximum of 84 parameters, which is a

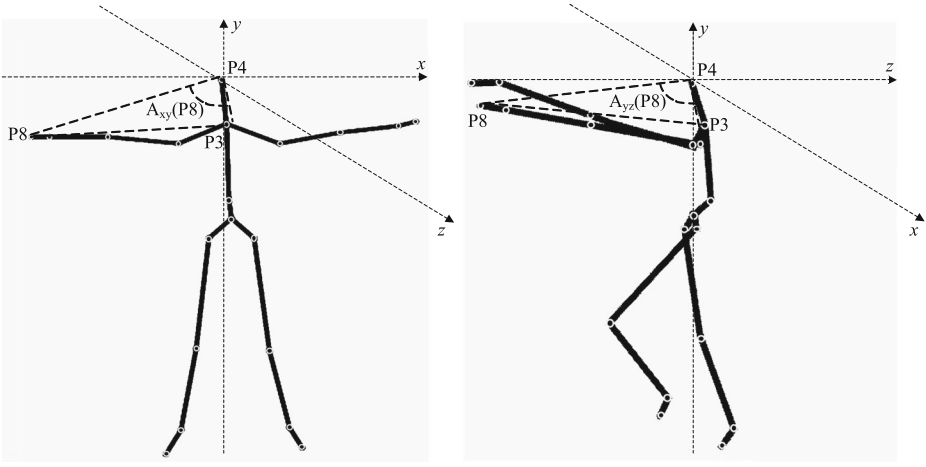

Fig. 5 The relative joint angle information of the key joint P8, including both the x-y plane angle $A_{xy}(P8)$ and the y-z plane angle $A_{yz}(P8)$, using the joint P4 as the original point and the joint P3 as the auxiliary point

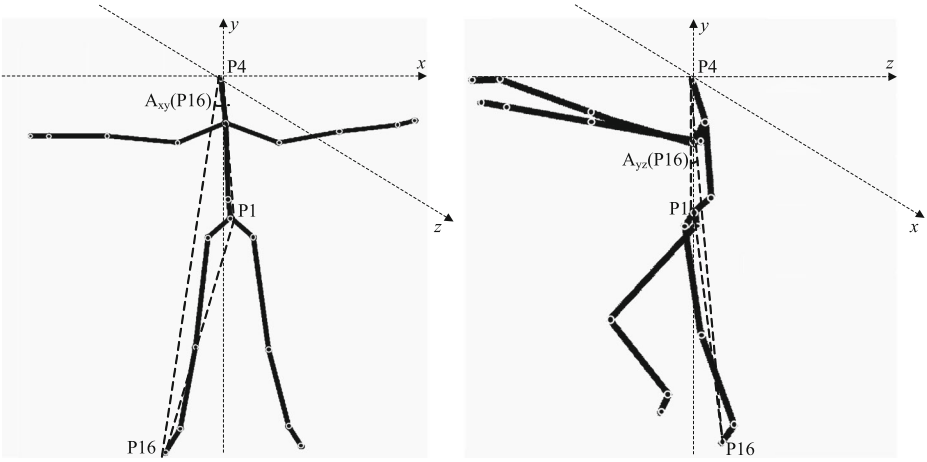

Fig. 6 The relative joint angle information of the key joint P16, including both the x-y plane angle $A_{xy}(P16)$ and the y-z plane angle $A_{yz}(P16)$, using the joint P4 as the original point and the joint P1 as the auxiliary point

result of combining both joint positions and joint angles; therefore, the improved feature vector contains a larger amount of information.

3.2 Feature design algorithms for determining an appropriate feature for each gesture category in DTW gesture recognition

As mentioned earlier, the hybridization of joint positions and joint angles for representing the feature vector of a Kinect-captured gesture frame as shown in (5) can help describe the characteristics of gestures accurately. However, the direct combination of joint position information and joint angle information is extremely inflexible and also redundant in some situations. Fuzzy thinking is a more appropriate choice for carefully considering an effective combination of joint position information and joint angle information. A soft computing-like thought on gesture feature design is that not all 24 relative joint angle parameters should be inserted in the feature vector; rather, an appropriate selection of some of the 24 relative joint angle parameters may be used as the gesture feature information, and the number of relative joint angle parameters selected for addition to the gesture feature vector should be determined according to the degree of motion variation of each gesture category in the gesture recognition system. The rationale behind the design is that categorization of a gesture with a high degree of motion variation requires a gesture feature vector with few relative joint angle parameters. By contrast, a small number of relative joint angle parameters are insufficient for accurately categorizing a gesture with a small degree of motion variation, and in this case, a large amount of relative joint angle information should be provided during the design of the gesture feature vector.

Key joints P8 and P12 in the upper limb and P16 and P20 in the lower limb were chosen for evaluating the degree of motion variation of a gesture for categorization. Figure 2 shows that these four key joints have considerably more apparent motion variations compared with the other 16 joints in the Kinect-captured human skeleton when an indicated gesture is performed. For efficiently evaluating the degree of motion variation of a gesture category by using each of these four key points, a complete gesture operation can be further divided into four active

gesture segments: the segment of the left upper limb, the segment of the right upper limb, the segment of the left lower limb, and the segment of the right lower limb; the decision of feature choices is made locally for each of these four segments. Consider the feature choice in the segment of the left upper limb as an example. As mentioned, the derived relative joint angles of key joint P8 are the *x*-*y* plane angle $A_{xy}(P8)$ and the *y*-*z* plane angle $A_{yz}(P8)$. The values of $A_{xy}(P8)$ and $A_{yz}(P8)$ are used to determine the size of the relative joint angle required in the active segment of the left upper limb. Such mentioned logical reasoning to explain the relationship between the indexes $A_{xy}(P8)$ and $A_{yz}(P8)$ of the key joint P8 and the required joint relative angle information size in the situation of the left upper limb segment is summarized in the following *If-Then* rules:

Rule 1: If $A_{xy}(P8)$ is large, then the x-y plane joint relative angle information size is small;
Rule 2: If $A_{xy}(P8)$ is medium, then the x-y plane joint relative angle information size is medium;
Rule 3: If $A_{xy}(P8)$ is small, then the x-y plane joint relative angle information size is large;
Rule 1-1: If $A_{yz}(P8)$ is large, then the y-z plane joint relative angle information size is small;
Rule 2-1: If $A_{yz}(P8)$ is medium, then the y-z plane joint relative angle information size is medium;
Rule 3-1: If $A_{yz}(P8)$ is small, then the y-z plane joint relative angle information size is large;

Accordingly, a feature design algorithm for considering all four local active sections is designed for determining an appropriate gesture feature for each of the gesture categories in DTW gesture recognition. Figure 7 shows the feature design algorithm used in this study. The algorithm can be used for appropriately designing a gesture feature for a gesture category according to the degree of motion variation of the gesture category. If there are a total of *N* gesture categories in the recognition system, there will be a maximum of *N* designed gesture feature vectors.

3.3 Kinect-based DTW human gesture recognition with gesture categories having different features

The proposed Kinect-based DTW gesture recognition method with a feature design scheme has an appropriate set of feature parameters for each active gesture categorization in the recognition system. When DTW template matching is performed using the proposed Kinect-based DTW human gesture recognition method and gesture categories (of the DTW-referenced template database) with different features, a feature selection process is required for the test gesture template for extracting the specific type of feature parameters identical to the feature parameter type of the encountering referenced template. Feature extraction is performed on the test gesture template according to the feature type determined during feature selection, and therefore, the test gesture template has the same type of gesture feature parameters as the matched referenced template. The feature selection process in the test phase of DTW gesture recognition effectively overcomes the problem of the test gesture template possibly encountering a referenced template such that the specific type of gesture feature information in both templates match. Figure 8 shows the Kinect-based DTW human gesture recognition algorithm for gesture categories with different features.

For the DTW method, the proposed feature design scheme does not have any effect on the main pattern recognition calculation of the method. The functionality of the DTW template matching operation remains unchanged, and, therefore, the computational complexity of DTW template matching does not increase for the proposed feature design method. Compared with conventional DTW with only one given feature type, in DTW with the proposed feature design

Proposed feature design algorithm for Kinect-based DTW gesture recognition
/* Initialize the feaure vector of each active gesture class to be the 60-D vector. */
For each n = 1 to N /* N is the number of active gesture classes in DTW referenced templates */
The feature vector of the n-th gesture class $V_n = [P_{ij}]$, $i = 1,2,\ldots 20$, $j = x, y, z$;
End For
Set counters, $A_{xy}(P8)\_Counter$, $A_{yz}(P8)\_Counter$, $A_{xy}(P12)\_Counter$, $A_{yz}(P12)\_Counter$, $A_{xy}(P16)\_Counter$, $A_{yz}(P16)\_Counter$, $A_{xy}(P20)\_Counter$, and $A_{yz}(P20)\_Counter$, to be zero;
/* Feature determinations for each gesture class in DTW recognition systems */
For each n = 1 to N /* N is the number of active gesture classes in DTW referenced templates */
For each m = 1 to M /*M is the number of the referenced template with the class label n */
T = Calculate the total frame number of the reference template m;
For each t = 1 to T /* T is the total frame number contained in the referenced template m */
Determine the (x, y, z)-coordinate positions of 20 joints in the gesture frame t;
Derive the joint relative angles of the x-y plane and the y-z plane of 12 key joints;
$A_{xy}(P8)\_Counter \mathrel{+}= A_{xy}(P8)$; $A_{yz}(P8)\_Counter \mathrel{+}= A_{yz}(P8)$;
$A_{xy}(P12)\_Counter \mathrel{+}= A_{xy}(P12)$; $A_{yz}(P12)\_Counter \mathrel{+}= A_{yz}(P12)$;
$A_{xy}(P16)\_Counter \mathrel{+}= A_{xy}(P16)$; $A_{yz}(P16)\_Counter \mathrel{+}= A_{yz}(P16)$;
$A_{xy}(P20)\_Counter \mathrel{+}= A_{xy}(P20)$; $A_{yz}(P20)\_Counter \mathrel{+}= A_{yz}(P20)$;
End For
End For
If ($A_{xy}(P8)\_Counter > Threshold\_High$) /* Relative angle parameters of the key joints in the left upper limb */
Add only $A_{xy}(P8)$ into the feature vector of the n-th gesture class, V_n;
Elseif ($Threshold\_Low \le A_{xy}(P8)\_Counter \le Threshold\_High$)
Add $A_{xy}(P8)$ and $A_{xy}(P7)$ into the feature vector of the n-th gesture class, V_n;
Elseif ($A_{xy}(P8)\_Counter < Threshold\_Low$)
Add $A_{xy}(P8)$, $A_{xy}(P7)$ and $A_{xy}(P6)$ into the feature vector of the n-th gesture class, V_n;
End If
If ($A_{yz}(P8)\_Counter > Threshold\_High$)
Add only $A_{yz}(P8)$ into the feature vector of the n-th gesture class, V_n;
Elseif ($Threshold\_Low \le A_{yz}(P8)\_Counter \le Threshold\_High$)
Add $A_{yz}(P8)$ and $A_{yz}(P7)$ into the feature vector of the n-th gesture class, V_n;
Elseif ($A_{yz}(P8)\_Counter < Threshold\_Low$)
Add $A_{yz}(P8)$, $A_{yz}(P7)$ and $A_{yz}(P6)$ into the feature vector of the n-th gesture class, V_n;
End If
If ($A_{xy}(P12)\_Counter > Threshold\_High$) /* Relative angle parameters of the key joints in the right upper limb */
The remainder is similar as the pesudo-code described in the case of the key joint P8;
...
If ($A_{xy}(P16)\_Counter > Threshold\_High$) /* Relative angle parameters of the key joints in the left lower limb */
The remainder is similar as the pesudo-code described in the case of the key joint P8;
...
If ($A_{xy}(P20)\_Counter > Threshold\_High$) /* Relative angle parameters of the key joints in the right lower limb */
The remainder is similar as the pesudo-code described in the case of the key joint P8;
...
End For
Establish the feature index table to record feature information of each gesture categorization;

Fig. 7 The developed feature design algorithm for determining the gesture feature of each active gesture class in a Kinect-based DTW gesture recognition system

scheme, there is more abundant and accurate gesture feature information in the DTW-referenced template database, where each referenced template has an appropriate corresponding feature type according to the provided gesture categorization of the referenced template.

```
Algorithm for Kinect-based DTW gesture recognition by presented feature designs
/* Initialize the joint position data and the key joint relative angle data of all gesture frames of test data to be zero. */
For each t = 1 to MAX  /* MAX is the defined max value for gesture frames in the test template. */
    For each j = 1 to 20  /* j denotes the joint index. */
        X_position[t][j] = 0;
        Y_position[t][j] = 0;
        Z_position[t][j] = 0;
    End For
End For
For each t = 1 to MAX  /* MAX is the defined max value for gesture frames in the test template. */
    For each k = 1 to 12  /* k denotes the key joint index. */
        XY_Angles[t][k] = 0;
        YZ_Angles[t][k] = 0;
    End For
End For
For each t = 1 to T  /* T is the total number of gesture frames in the test template */
    Acquisitions of 3D-positions of 20 joints (X_position[t], Y_position[t], Z_position[t]);
    Acquisitions of the x-y plane angle of 12 key joints (XY_Angles[t]);
    Acquisitions of the y-z plane angle of 12 key joints (YZ_Angles[t]);
End For
/* Initializations of the accumulated distance index of each referenced template to be matched */
For each r = 1 to R /* R is the number of the referenced templates in DTW recognition systems */
    distance[r] = 0;
End For
/* Performing feature selection for each referenced template and then doing template matching */
For each r = 1 to R /* R is the number of the referenced templates in DTW recognition systems */
    feature_type = feature selection (r-th_referenced_template, feature_index_table);
    test_template_features = feature extraction (X_position, Y_position, Z_position, XY_Angles, YZ_Angles,
                             feature_type);
    distance[r] = template matching (test_template_features, r-th_referenced_template_features);
End for
target_referenced_template = search the minimum distance element (distance[R]);
recognition_outcome = gesture label checking (target_referenced_template);
```

Fig. 8 The recognition algorithm for Kinect-based DTW gesture recognition by the presented feature design scheme

4 Experiments and results

Kinect-based DTW gesture recognition experiments were performed in the application of the gesture command recognition task. As voice command for target object control applications in speech recognition, a command made using the indicated human gesture could be efficiently used to control household electric equipment in a smart home, or to control the action of a robotic toy. Such Kinect-based DTW gesture recognition using gesture command operations could be conveniently integrated with a target hardware system with a specific purpose. A set of 14 categories of human gesture commands was designed in this study: "lifting the right foot with both hands held," "lifting the left foot with both hands held," "waving the right hand on the left side," "waving the left hand on the right side," "waving the right hand by positioning the hand above the head," "pulling a ceiling fan by using the right hand," "pulling a ceiling fan by using the left hand," "jumping in place," "maintaining a standing posture," "holding a phone in the right hand," "holding a phone in the left hand," "pushing a door with the right hand," "placing both hands on the hip with the right foot lifted to the right side," and "placing both hands on the hip with the left foot lifted to the left side"; these categories were labeled as Classes 1 to 14, respectively. When a series of active frames is captured using the Kinect camera, the Kinect camera has a default frame rate of 30 frames/s, is deployed at a height of 1.2 m, and is at a distance of 3 m from the active user. For effectively acquiring the (x,y,z) coordinates of the position of the 20 human joints in the human skeleton, the Microsoft Kinect SDK

(Version 1.8.0) was adopted. The Microsoft Kinect SDK was used only at the stage in which joint position information was acquired.

There were two main experimental phases in this study: the training phase of system establishments of Kinect-based DTW gesture recognition with the proposed feature design scheme and the test phase of performance evaluation of the proposed gesture recognition system established in the training stage. In the training phase, the training gesture data collected from active users are used to establish the DTW-referenced template database, in which each referenced template is a complete gesture representing a certain category of gestures performed by an active user. Five male users were requested to capture their gestures using the Kinect camera. The training data in the training stage consisted of 700 active gestures. Each of the five users was requested to perform the indicated 140 active gestures, 10 gestures for each of the 14 categories of gestures. In the proposed feature design scheme for Kinect-based DTW gesture recognition, the gesture feature for each of these 14 category gestures was determined using 50 captured gestures (10 gestures of each of the 5 active users). After the feature design for all these referenced template gestures was completed, a feature index table providing the corresponding feature type for each gesture category was constructed. For the conventional Kinect-based DTW gesture recognition with an invariable type of gesture features, the 700 active gestures collected were used only as the referenced templates for DTW template

Table 1 Estimated gesture feature types of 14 indicated gesture categories in Kinect-based DTW gesture recognition with the proposed feature design scheme

Gesture category	Gesture feature contents		Feature dimensions
	Position data of 20 joints	Angle data of the key joints	
Class-1	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{xy}(P_1), A_{yz}(P_1), A_{xy}(P_4), A_{yz}(P_4)$	80 (60+20)
Class-2	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{xy}(P_1), A_{yz}(P_1), A_{xy}(P_4), A_{yz}(P_4)$	80 (60+20)
Class-3	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{xy}(P_4), A_{yz}(P_4)$	82 (60+22)
Class-4	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{xy}(P_1), A_{yz}(P_1)$	82 (60+22)
Class-5	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{xy}(P_4), A_{yz}(P_4), A_{xy}(P_5), A_{yz}(P_5)$	80 (60+20)
Class-6	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{xy}(P_4), A_{yz}(P_4), A_{xy}(P_5), A_{yz}(P_5)$	80 (60+20)
Class-7	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{xy}(P_1), A_{yz}(P_1), A_{xy}(P_2), A_{yz}(P_2)$	80 (60+20)
Class-8	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k$=1,2,...12	84 (60+24)
Class-9	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k$=1,2,...12	84 (60+24)
Class-10	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{xy}(P_4), A_{yz}(P_4)$	82 (60+22)
Class-11	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{xy}(P_1), A_{yz}(P_1)$	82 (60+22)
Class-12	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k=1,2,\ldots12,$ $excluding\, A_{yz}(P_4)$	83 (60+23)
Class-13	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k$=1,2,...12	84 (60+24)
Class-14	P_{ij}, i=1,2,…,20, j=x,y,z	$A_{xy}(P_k), A_{yz}(P_k),\ k$=1,2,...12	84 (60+24)

matching calculations and not for designing the gesture feature. In the test phase, test experiments were conducted; gesture recognition was performed on gestures made by two new male users. These two male users were not among the active users in the training stage. Each of these two users was requested to capture ten gestures for each of the 14 category gestures. Therefore, a total of 280 active gestures were used as the test data for a performance comparison of the gesture recognition methods.

Table 1 shows the appropriate feature type derived using the proposed feature design scheme for each of 14 indicated categories of gestures. As can be seen in Table 1, after the developed gesture feature design algorithm was run, each gesture category in the DTW recognition system had an optimal gesture feature type. The gesture feature of the 14 gesture categories included the original 60 joint position parameters (each of the 20 joints containing the x, y, and z coordinate values) and a variable number of relative joint angle parameters of the key joints. Classes 1, 2, and 5 to 7 had the minimum number of feature parameters at 80, consisting of 60 joint position parameters and 20 joint angle parameters. Classes 8, 9, 13, and 14 had the largest number of feature parameters, and these consisted of all 60 joint position parameters and all 24 relative joint angle parameters. The conventional Kinect-based DTW gesture recognition method involves only the 60 fixed joint position parameters, and it is considerably less flexible than the variable-sized gesture feature design in this study.

A recognition performance comparison between the proposed Kinect-based DTW gesture recognition method with feature design and the conventional Kinect-based DTW gesture recognition method with only joint position data features is presented in Table 2. In the test experiment, an experiment for evaluating Kinect-based DTW gesture recognition using all 84 feature parameters (joint position parameters and joint angle parameters) was also conducted. The proposed recognition algorithm with a feature selection scheme discussed in Section 3.3 was specifically used for Kinect-based DTW gesture recognition with feature design. As shown in Table 2, in gesture recognition test experiments on the recognition of 14 designed gesture categorizations, the proposed feature design scheme for Kinect-based DTW gesture recognition achieved an exceptional recognition rate of 87.5 %, which was superior to the recognition rate of 82.14 % achieved using the conventional Kinect-based DTW gesture recognition method using only fixed-sized features of 60 joint position parameters. The recognition rate improvement of 5.36 % was considerable. In addition, the proposed feature design scheme using soft combinations of joint position data and joint angle data for deriving an optimal feature set for each gesture category in the recognition system was also superior to inflexible hard feature combinations of the 84 feature parameters, consisting of all 60 joint position and all 24 relative joint angle parameters. In Table 2, the comparable recognition experiment results of Kinect-based DTW gesture recognition in various feature setting situations show the effectiveness and competitiveness of the feature design scheme proposed in this paper.

Table 2 Recognition rates of Kinect-based DTW gesture recognition using conventional fixed 60 joint position parameters, 84 parameters of hard combinations of 60 joint position parameters and 24 joint relative parameters, and the proposed feature design scheme for each gesture category

Average recognition rates of Kinect-based DTW gesture recognition in different feature setting situations		
Conventional fixed 60 joint position parameters	Hard combinations of 60 joint position parameters and 24 joint relative parameters (84 parameters)	Proposed feature design schemes to derive the optimal feature set for each gesture category
82.14 %	77.86 %	87.50 %

5 Conclusions

In this paper, a feature design scheme involving appropriate hybridizations of joint position data and relative joint angle data is proposed for Kinect-based DTW human gesture recognition. The proposed scheme derives the optimal gesture feature parameters for each gesture category (defined in the DTW recognition system) according to the degree of motion variation of the gesture category, and it is expected to effectively increase the recognition accuracy in the template matching process in the DTW recognition test phase. Gesture recognition experiments involving classification calculations for 14 gesture categories showed that the proposed Kinect-based DTW gesture recognition method with a flexible feature design is superior to the conventional Kinect-based DTW gesture recognition method, in which only an inflexibly immutable joint position parameter determines the recognition performance.

Acknowledgments This research is partially supported by the Ministry of Science and Technology (MOST) in Taiwan under Grant MOST 103-2221-E-150-046.

References

1. Bautista MÁ, Hernández-Vela A, Ponce V, Perez-Sala X, Baró X, Pujol O, Angulo C, Escalera S (2013) Probability-based dynamic time warping for gesture recognition on RGB-D data. Lect Notes Comput Sci 7854:126–135
2. Carmona JM, Climent J (2012) A performance evaluation of HMM and DTW for gesture recognition. Lect Notes Comput Sci 7441:236–243
3. Ding IJ (2013) Speech recognition using variable-length frame overlaps by intelligent fuzzy control. J Intell Fuzzy Syst 25(1):49–56
4. Ding IJ, Chang CW An eigenspace-based method with a user adaptation scheme for human gesture recognition by using Kinect 3D data. Appl Math Model. doi:10.1016/j.apm.2014.12.054
5. Ding IJ, Chang CW An adaptive hidden Markov model-based gesture recognition approach using Kinect to simplify large-scale video data processing for humanoid robot imitation. Multimed Tools Appl. doi:10.1007/s11042-015-2505-9
6. Ding IJ, Yen CT (2013) Enhancing GMM speaker identification by incorporating SVM speaker verification for intelligent web-based speech applications. Multimed Tools Appl. doi:10.1007/s11042-013-1587-5
7. Han J, Shao L, Xu D, Shotton J (2013) Enhanced computer vision with microsoft Kinect sensor: a review. IEEE Trans Cybern 43(5):2168–2267
8. Nguyen-Duc-Thanh N, Lee S, Kim D (2012) Two-stage hidden Markov model in gesture recognition for human robot interaction. Int J Adv Robot Syst 9:1–10
9. Qian K, Niu J, Yang H (2013) Developing a gesture based remote human-robot interaction system using Kinect. Int J Smart Home 7(4):203–208
10. Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process 26(1):43–49
11. Su CJ (2013) Personal rehabilitation exercise assistant with Kinect and dynamic time warping. Int J Inform Educ Technol 3(4):448–454
12. Su CJ, Huang JY, Huang SF (2012) Ensuring home-based rehabilitation exercise by using Kinect and fuzzified dynamic time warping algorithm. Proc. the Asia Pacific Industrial Engineering & Management Systems Conference, pp. 884–895
13. Tashev I (2013) Kinect development kit: a toolkit for gesture- and speech based human-machine interaction. IEEE Signal Process Mag 30(5):129–131
14. Wu J, Konrad J, Ishwar P (2013) Dynamic time warping for gesture-based user identification and authentication with Kinect. Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2371–2375
15. Xu W, Lee EJ (2012) Continuous gesture recognition system using improved HMM algorithm based on 2D and 3D space. Int J Multimed Ubiquitous Eng 7(2):335–340

16. Zhang L, Hsieh JC, Wang J (2012) A Kinect-based golf swing classification system using HMM and neuro-fuzzy. Proc. IEEE International Conference on Computer Science and Information Processing (CSIP), pp. 1163–1166
17. Zhang X, Ye Z, Jin L, Feng Z, Xu S (2013) A new writing experience: finger writing in the air using a Kinect sensor. IEEE MultiMed 20(4):85–93

Ing-Jr Ding was born in Taipei, Taiwan, in 1975. He received the B.S. degree from Chang-Gung University in 1999, M.S. degree from National Central University in 2001, and Ph.D. degree from National Chiao-Tung University in 2008. He joined the Graduate Institute of Automation and Control at National Taiwan University of Science and Technology as a project assistant professor from March 2009 to July 2009. From August 2009 to July 2012, he served as an assistant professor in the Department of Electrical Engineering, National Formosa University. Since August 2012, he has been an associate professor in the Department of Electrical Engineering, National Formosa University. His research interests include speech processing, pattern recognition, machine learning, artificial intelligence, and multimedia techniques.

Che-Wei Chang received the B.S. degree from National Formosa University in 2012. He received his M.S. degree from the Department of Electrical Engineering, National Formosa University in 2014. Since September 2014, he has done his military service.