IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 25, 2022, accepted May 10, 2022, date of publication May 20, 2022, date of current version May 26, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3176717

Robust Hand Gesture Recognition Based on RGB-D Data for Natural Human–Computer Interaction

JUN XU[1], HANCHEN WANG[2], JIANRONG ZHANG[3], AND LINQIN CAI[2], (Member, IEEE)

[1]School of Computer and Data Engineering, Bengbu College of Technology and Business, Bengbu 233000, China
[2]School of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[3]China Information Technology Design and Consulting Institute Company Ltd., Chengdu 610042, China

Corresponding authors: Hanchen Wang (947541784@qq.com) and Jun Xu (xujunfff@126.com)

This work was supported in part by the 2021 Key Project of Natural Science of Anhui Colleges and Universities under Grant KJ2021A1236, in part by the 2020 Provincial Quality Engineering Project of Colleges and Universities in Anhui Province under Grant 2020kfkc312, and in part by the 2021 Provincial Quality Engineering Project of Colleges and Universities in Anhui Province under Grant 2021jxtd179 and Grant 2021sysxzx018.

ABSTRACT To naturally interact with virtual environment by hand gesture, this paper presents a robust RGB-D data based recognition method of static and dynamic hand gesture. Firstly, for static hand gesture recognition, starting from the hand gesture contour extraction, the palm center is identified by Distance Transform (DT) algorithm. The fingertips are localized by employing the K-Curvature-Convex Defects Detection algorithm (K-CCD). On the basis, the distances of the pixels on hand gesture contour to palm center and the angle between fingertips are considered as the auxiliary features to construct a multimodal feature vector, and then recognition algorithm is presented to robustly recognize the static hand gestures. Secondly, combining Euclidean distance between hand joints and shoulder center joint with the modulus ratios of skeleton features, this paper generates a unifying feature descriptor for each dynamic hand gesture and proposes an improved dynamic time warping (IDTW) algorithm to obtain recognition results of dynamic hand gestures. Finally, we conduct extensive experiments to test and verify the static and dynamic hand gesture recognition algorithm and realize a low-cost real-time application of natural interaction with virtual environment by hand gestures.

INDEX TERMS Hand gesture recognition, RGB-D, human computer interaction (HCI), dynamic time warping (DTW), virtual environment.

I. INTRODUCTION

Among different human body parts, the hand is the most effective interaction tool because of its dexterity. Adopting hand gesture as an interface in Human Computer Interaction (HCI) affords users the ability to interact with computers in more natural and intuitive ways, which allows deploying a wide range of applications such as virtual reality, computer games, and sign language recognition. Consequently, current hand gesture recognition is no surprise to become one of the active research areas in natural HCI [1], [2].

The principal components of hand gesture recognition are data acquisition, hand localization (e.g., segmentation and tracking), hand feature extraction, and gesture recognition based on identified features. Various approaches have been designed for hand gesture recognition including Dynamic Time Warping (DTW), Hidden Markov Models (HMM), Support Vector Machines (SVM), and so on [1]–[5]. Recently, current popular deep neural networks (DNN) such as Convolutional Neural Networks (CNN) have been applied to recognize human hand gestures, and achieved better recognition performance [4]–[6]. However, deep neural networks require enormous training data. Training a deep network requires carefully tuning the hyper-parameters and usually suffers from convergence to a local optimal solution. In addition, their implementation and high requirements for machine and equipment on the real-life applications also are the typical limitations. Therefore, traditional feature-modeling based approaches still acquire a lot of attentions and are widely used in real-life applications.

The associate editor coordinating the review of this manuscript and approving it for publication was Orazio Gambino.

Classic color cameras have already been employed for data acquisition of hand gesture recognition tasks [1]–[3]. These solutions are, however, sensitive to clutter, lighting conditions, and skin color. Video capture has the extra challenge related to the speed of movement. In terms of 3- dimensional (3D) motion capture at the level of the fingers, possible solutions include optical marker systems, accelerometers, magnetic trackers, and data gloves. These require extensive calibration, limit the natural movement of the fingers, and are generally very expensive. Recent development of depth sensors (e.g., Kinect sensor) provides a robust solution to hand gesture recognition [7], [8]. Data captured by Kinect in RGB-D (Red, Green, Blue, and Depth information) form, are often used as a source for hand gesture recognition. In spite of many recent successes in applying the Kinect sensor to face recognition, human body tracking and human action recognition, it is still an open problem to use Kinect for hand gesture recognition in natural HCI. Due to the low-resolution and inaccuracy of the Kinect depth map, it is difficult to detect and segment a hand gesture from an image with this resolution. In such a case, the segmentation of the hand is usually inaccurate, thus may significantly affect the recognition step [7].

In this paper, we aim to perform static and dynamic gesture cognition using RGB-D data from Kinect. For static hand gesture recognition, the depth and joint information collected by Kinect is proposed to locate and detect the hands. Starting from the hand gesture contour extraction, the center of the palm is identified by Distance Transform (DT) algorithm. The fingertips are localized by employing a K-Curvature-Convex Defects Detection algorithm (K-CCD). On the basis, the distances between the pixels on the gesture contour and palm center and the angle between fingertips are considered as the auxiliary features to construct a multimodal feature vector. To recognize dynamic hand gesture, we proposed an improved dynamic time warping algorithm (IDTW) in [8]. This paper extends the IDTW algorithm, especially, the restricted search path and the weight optimization is discussed in detail. On this basis, we demonstrate an application to interaction with a virtual environment for underground coalmine simulation (e.g. virtual coalmine) using static and dynamic gesture cognition.

The motivation of this work is to perform robust static and dynamic gesture cognition using RGB-D data with the aim of realizing a low-cost real-time application of natural interaction with virtual environment by hand gesture recognition. The main contributions of this paper are summarized as follows: (1) A K-Curvature-Convex Defects Detection algorithm (K-CCD) and multimodal feature vector are proposed for static hand gesture recognition. (2) A unifying feature descriptor is constructed for dynamic hand gesture. (3) A low-cost real-time application is presented for natural interaction with virtual coalmine by hand gesture recognition.

The remainder of this paper is organized into seven sections. Section II reviews related work of the state-of-the-art-of hand recognition. Section III describes the framework of static gesture recognition and the methods of hand detection, feature extraction, and gesture recognition. Section IV introduces the methods of hand tracking, feature extraction, and the improved DTW gesture recognition algorithm for dynamic hand gesture cognition. Section V illustrates experimental results of static and dynamic gesture recognition. Section VI demonstrates an application to interact with virtual coalmine environment. Section VII makes final conclusions and discusses the next work.

II. RELATED WORK

In recent years, somatosensory interaction technology has been widely concerned in every area. Hand gestures are essential parts of somatosensory interaction technology and has been applied to various scenes such as somatic game, military training, and virtual environment. The motivation of this paper is to perform hand gesture recognition based on RGB-D data and traditional feature-modeling approaches. A good survey for the gesture recognition is available in [1], [4], [5]. The following literature review mainly summarizes the related work of human hand gesture recognition with depth information and skeleton information for natural human-computer interaction with virtual environment applications.

Different approaches can be employed for hand gesture feature extraction. One of commonly used techniques are based on RGB-D data [2], [7], [9]. Specifically, depth data can be used to extract hand region as the area of body closer to the camera [9], and then to identify the fingers, palms and wrists by using geometric size, and finally extract a set of features descriptors which characterize the shape and the pose of hand gestures. Though this method is insensitive to lighting conditions and cluster background, it still has limitations, such as assumption of the hand is the closet object to the camera [7]. Another popular technique for hand features extraction is based on skeleton. This can be divided into two categories: position features and orientation features [5], [10], [11]. In contrast to the depth-data methods, the majority of the skeleton-based methods model temporal dynamics explicitly. Bhattacharya etc. [12] applied 20 joints to recognize three simple body gestures and used a Z-score normalization to deal with parameters of different units and scale of body-joint points. However, skeleton based feature extraction methods may increase computation amount and time complexity. Saponaro etc. [13] used geometric transformation to set hand coordinates to the reference system centered on the human torso, instead of the default sensor-centered reference frame. This transformation provides invariance to the starting point of a physical gesture. Using the sequences of joint coordinates, Du *et al.* [14] applied Kalman filter to estimate the hand position for the precise localization of hand movement in a human manipulation interface for robot tele-operation. Slama *et al.* [15] first placed the hip center joint at the origin of the coordinate system for the skeletons scale invariant, and

then took a skeleton template as reference to normalize all the other skeleton.

Orientation features based on the angular information between joint vectors can maximize the invariance of the skeletal representation. Angles between specific pairs of direction vectors are computed to obtain the corresponding joint angles in [16], [17]. Raptis *et al.* [18] used angular skeleton representation to map the skeleton motion data into a smaller set of features, which reduces the overall entropy of the signals and removes the dependence on camera position. The Euler angles have been largely used to describe the orientation of a rigid body in a 3D Euclidean space [19]. Another way to model orientation information is by means of the unit quaternions which represent a system of numbers that extends the complex numbers [20].

After the feature extraction, generating recognition model is another important step. The conventional recognition methods mainly include probability-based approach and distance-based method. The most common probability method is HMM [3], which is a statistical model. HMM has two hypotheses: output independence and Markov assumption. However, most of the sequence data in fact cannot be expressed as a series of independent events. In addition, defining states for gestures is not an easy task since hand gestures can be formed by a complex interaction of different joints. The distance-based method [10] is an earlier method applied in the classifier learning for real-time detection. Dynamic time warping (DTW) [3], [4], [10] is the most used technique to find the optimal alignment of two signals. The conventional DTW is basically a dynamic programming algorithm, which uses an iterative update of DTW cost by adding the distance between mapped elements of the two sequences at each iteration step. The distance between two elements is oftentimes the Euclidean distance, which gives equal weights to all dimensions of a sequence sample. However, a weighted distance might perform better in assessing the similarity between test sequences and reference sequences. In [10], a weighted DTW algorithm was proposed to maximize a discriminant ratio based on DTW costs. The weights were obtained from a parametric model which depends on how active a joint was in a gesture class. Chaaraoui *et al.* [21] constructed gesture sets by sequences of key poses and then defined a DTW distance between two sequences by combining the Euclidean distance between couples of key poses in all the possible alignments of the test and reference sequences. As far as static gesture recognition is concerned, SVM [8], [9] and multiclass SVM approaches are used broadly [17]. Cai *et al.* [8] used an adaptive square to extract the region of hand based on the depth information and applied SVM to classify the static hand gestures.

There are other prominent works reviewed in [1], [3], [5]. However, most of the existing solutions for hand gesture recognition are designed for hand properties (hand contour, hand palm center, fingers, and hand trajectory). Overall, there are only very few solutions for static and dynamic hand gestures recognition that work on hand, wrists, elbow, arm, and shoulder for natural HCI applications. The objective of this paper is to develop an improved, low-complexity, and real-time solution for the recognition of static and dynamic hand gestures from Kinect depth sensor. Experimental results show that our hand gesture recognition system not only operates accurately and efficiently, but also is robust to uncontrolled environments and hand gesture variations in orientation, scale, articulation, and shape distortions.

III. STATIC GESTURE RECOGITION

The framework for static hand gesture recognition is illustrated in Fig. 1, which mainly includes hand detection, feature extraction, and gesture recognition. We use Kinect to obtain joint positions. Raw data collected with the Kinect are used to recuperate the depth information on all the pixels of an image. The depth and joint information are then proposed to locate and detect the hands within the digital skeleton. Starting from the hand gesture contour extraction, the center of the palm is identified by DT algorithm. The fingertips are localized by employing the K-CCD algorithm, which is based on the change in the slope angle of the tangent line at selected points over the contour and the convex defects detection for filtering the noise points. On the basis, we consider the distances of the pixels on the gesture contour to palm center and the angle between fingertips as the auxiliary features to construct a multimodal feature vector. Finally, the gesture recognition algorithm is built to classify feature parameters.

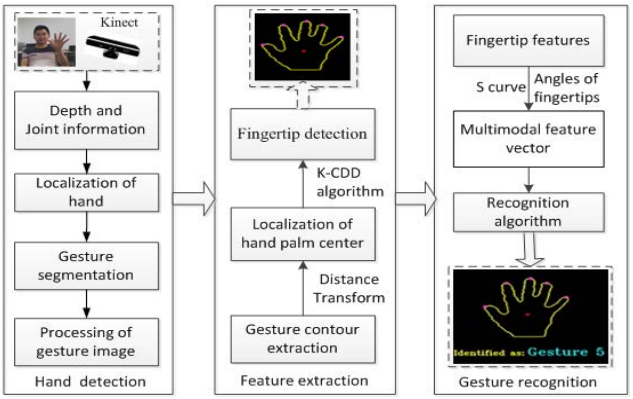


FIGURE 1. The proposed framework for static gesture recognition using RGB-D information.

A. HAND DETECTION

Hand detection mainly includes localization of hand, gesture segmentation, and processing of gesture image. Firstly, we use the digital skeleton provided by Kinect to identify the hand position and to locate joints of hand, elbow and wrist. Then combining the gray value distribution of the pixels in each frame depth image, we can segment the hand gestures, as shown in Fig. 2.

The hand gesture images in Fig.2 still contain an arm and some unwanted noise, such as rough edge, which will affect the accuracy of subsequent contour extraction, feature

FIGURE 2. The segmented image of hand region.

extraction, and gesture recognition. Due to wrist is the smallest part of whole arm except the fingers, the number of pixels contained in wrist is the least in the segmented hand image except the fingers. Therefore, we can count the number of pixels in the segmented hand gesture images to localize the wrist so as to remove the part section below the wrist. Referring to the method in [22], we firstly rotate gesture image to make its fingers horizontally to the right, as shown in Fig. 3 (b). And then, the corrosion operation of morphology is employed to erode the fingers so as to avoid the fingers' affection on the wrist, as shown in Fig. 3(c). Finally, we calculate the number of pixels in the vertical direction in Fig. 3(c) to generate the pixel waveform of hand gesture, as shown in Fig. 4. The blue curve is the pixel waveform of the original hand gesture. It can be found that there are some burrs in this curve. With the aid of least-square method, the original curve is fitted to generate a new depth pixel waveform, the green curve in Fig. 4. Analyzing the green fitting curve, we can find the first minimum point on the curve is the wrist position. And then we remove the wrist parts to get the hand gesture image without arm, as shown in Fig. 3 (d). Finally, we employ median filter to smooth the edge region of the hand gesture image. The final gesture image is shown in Fig. 3(e). As we can see, the edge of the hand gesture in Fig. 3 (e) is smoother and the hand gesture features are more clearly compared with that of the hand gesture in Fig. 3 (d).

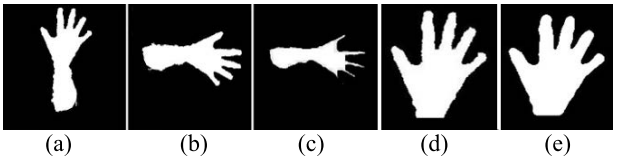


FIGURE 3. The removal process of wrist part. (a) The original image, (b) Horizontal rotated image, (c) Corroded image, (d) the hand gesture removed the arm, (e) the smoothed hand gesture image.

B. FEATURE EXTRACTION

The extraction of hand gesture features mainly includes the contour detection and tracking, the localization of hand palm center, and the detection of fingertips.

1) CONTOUR DETECTION AND TRACKING

To obtain a complete hand gesture contour, we firstly scan the segmented hand gesture image to identify the initial pixel. In order to optimize the search, Plouffe and Cretu proposed a method of detecting every other five pixels (that is, scan by 5×5 square search window) [2]. In this paper, we adjust

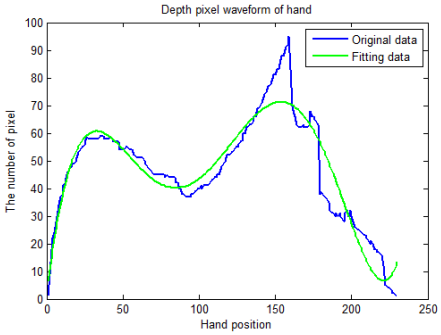


FIGURE 4. Depth pixel waveform of hand gesture.

the search range at 10×10 to improve the search algorithm. Similar to the approach in [2], if a pixel is valid and does not possess any valid neighboring pixel, the search continues pixel by pixel toward an invalid neighboring pixel until a contour pixel is found, as illustrated in Fig. 5. We verify if this pixel is part of an already found contour and, if not, a new potential hand is found. If a pixel is valid and none of the neighboring pixels are invalid, the search continues with the next block.

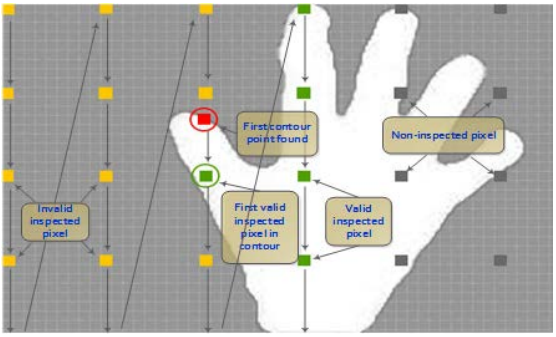


FIGURE 5. The detection of the first point of the contour inspecting each block of 10×10 pixels.

Once the initial contour pixel is found, a directional search is performed to identify the full contour of the hand. The considered directions in this paper are upper left, upper right, bottom left, and bottom right. A search direction is favored if it has been used for the previous pixel as well. For each potential hand contour, the algorithm traces in order each point of contour from the initial point found and stops when the next point is already present in the contour list. The algorithm also includes a solution to backtrack if an unknown valid configuration is encountered. To further improve this approach, a constraint is added to validate only closed contours. For the *Five* sign in Fig. 6 (a), the detected contour using this procedure is shown in Fig. 6 (b).

2) LOCALIZATION OF PALM CENTER

According to the characteristic of hand structure, if the distance from an interior pixel $n(x, y)$ in palm to the contour pixel on the edge of the hand has the maximal value, the

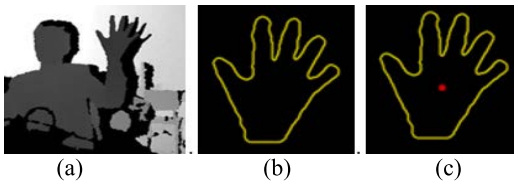


FIGURE 6. (a) Initial *Five* gesture, (b) Detected contour, (c) Identified palm center (red dot).

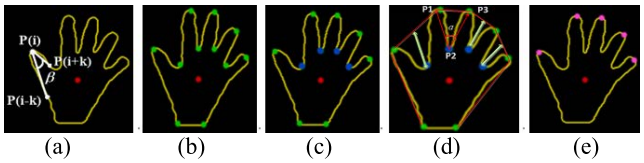


FIGURE 7. Results of fingertip detection by K-CDD method. (a) the K curvature algorithm, (b) the like-fingertips detection, (c) the concave points detection, (d) the convex defect detection, (e) the final result of the proposed K-CDD method.

pixel point $n(x, y)$ is considered to be the center of the palm. In this paper, Distance Transform (DT) is adopted to obtain the coordinate of palm center. The DT algorithm calculates the distance between non-zero pixel and the nearest zero pixel in a digital image so as to get the minimum distance from this pixel to the contour edge. According to the features of hand gesture image, the DT algorithm is defined as follows [23].

$$D(u) = min\{d(u, v)|P_d(v) \in O\} \tag{1}$$

where P_d is the detection result of gesture contour, D is the distance image of gesture image P, $d(u, v)$ is the distance measurement from the pixel v to the target pixel u, and O is the background target of gesture image. The transform results are also different due to different distance metric. In this paper, $d(u, v)$ is taken as Euclidean distance by (2).

$$d(u, v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2} \tag{2}$$

After DT transform, the image is the minimum distance from all target pixels to the image contour. The closer the pixel to target center, the larger the DT feature of the pixel. The result of localization of palm center for the *Five* gesture is shown in Fig.6(c).

3) DETECTION OF FINGERTIPS

Based on the results of contour detection and localization of palm center, the K-CDD algorithm is presented to identify the fingertips over the hand gesture contour. The proposed K-CDD method effectively combines K-Curvature method with convex defects detection. Firstly, the K-curvature algorithm is employed to detect the candidate fingertips of the gesture contour, which are also called like-fingertips [24]. And then, the concave points detection and convex defect detection are used to filter the concave points and noise points. K-CDD method can avoid the false fingertip detection of the traditional K-curvature method. In addition, it provides better results in terms of overall success rate, and supports the highest range of hand rotations within which it is capable of performing reliably.

As illustrated in Fig.7(a), the K-curvature algorithm takes each vector point $P(i)$ to its neighbor points $P(i - k)$ and $P(i + k)$ at distance of K. K is a constant value. According to our experiment, the final K value is 20 pixels, which is suitable for almost all situations. The angle β between the vector $\overrightarrow{P(i)P(i-k)}$ and $\overrightarrow{P(i+k)P(i)}$ is calculated over the contour of the hand. According to [2] and our tests, if the angle has a value between 25° and 55°, a fingertip is identified at that point. The detected fingertips by K-CDD for the sign *Five* are shown in green dots in Fig. 7(b).

In Fig.7 (b), there are still some valleys points between fingers. In this paper, the cross product of vectors is employed to remove the concave points in the valley between fingers. Randomly selecting a like-fingertip $P(i)$ on the contour, if the cross product of vectors $\overrightarrow{P(i)P(i-k)}$ and $\overrightarrow{P(i+k)P(i)}$ is negative, then $P(i)$ is the fingertip. The results of concave points for the Five sign are shown in blue dots in Fig. 7(c). After the concave points are filtered out, the remained noise points are mainly gathered around the wrist. Therefore, the convex defect detection method is employed to eliminate theses noise points. As shown in Fig.7(d), P_1 and P_3 are the convex hull points, and P_2 is concave point. The angle between the vector $\overrightarrow{P_2P_1}$ and $\overrightarrow{P_2P_3}$ is α. The sum of the modulus of $\overrightarrow{P_2P_1}$ and $\overrightarrow{P_2P_3}$ is M, namely, $M = |\overrightarrow{P_2P_1}| + |\overrightarrow{P_2P_3}|$. According to [15] and our tests by trial and error, if $\alpha < 90^{\circ}$ and $M > 10cm$, then P_1 and P_3 can be considered as the real fingertips. The final results of fingertip detection by the proposed K-CDD for the sign Five are shown in pink dots in Fig. 7(e).

C. GESTURE RECOGNITION

On the basis of the contour detection, the localization of palm center, and the detection of fingertips, a multimodal feature vector is constructed for the purpose of recognizing the hand gestures more robustly. In this paper, the distance S from part pixels on the gesture contour (every 4 pixels select 1 pixel) to palm center is considered as an auxiliary feature. Moreover, the angle α between two adjacent fingertips is also added to the auxiliary features.

According to [22] and the America sign language, we collected the 10 Chinese sign gesture data from *Zero* to *Nine*, as shown in Fig.8. The calculated distance S curves of the 10 digit gestures are illustrated in Fig. 9. As shown in the S curves, features of the 10 digit gestures have similarity. For instance, both gesture *Two* and gesture *Six* have two fingertips while their S curves are also increasingly blurred. The specific features of the 10 Chinese digit gestures are described in TABLE 1.

According to features of the digit gestures, the gesture recognition algorithm starts from localizing the hand region using the obtained depth data and skeleton data, and then calculates the number of fingertips and feature parameters.

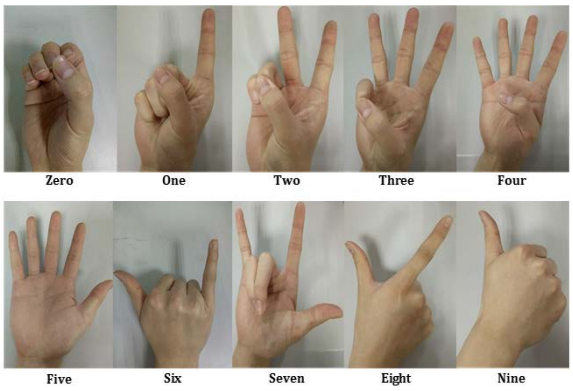


FIGURE 8. Chinese hand gesture of zero to nine.

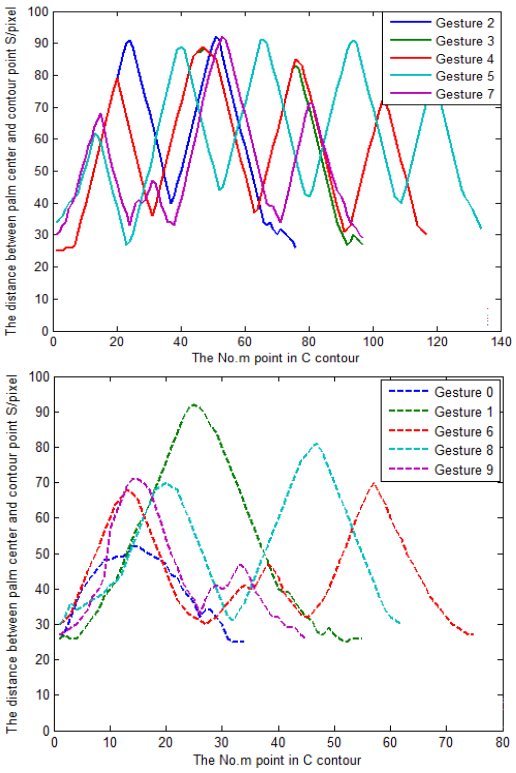


FIGURE 9. S curves of the 10 Chinese digit gestures.

IV. DYNAMIC GESTURE RECOGNITION

The dynamic gesture recognition includes hand tracking, feature extraction, and gesture recognition. We firstly apply Microsoft Kinect sensor to obtain the depth data and 3D coordinate information of hand joints (including hands, elbows, shoulders, etc.). The depth and joints information are used to generate a 3D motion trajectory of hand gestures to realize hand tracing and localization. The acquired joints information (3D coordinate sequence) of hand gestures is then used to extract the geometric feature of dynamic hand gestures by calculating the Euclidean distance between hand joints. Meanwhile, in order to further describe the relative position features of hand gesture to body, we create an auxiliary modulus ratio feature vector based on human skeleton structure. According to Euclidean distance between hand joints and the modulus ratio of feature vectors, we can generate a unifying feature vector descriptor to represent each dynamic hand gesture. Finally, the IDTW is built to obtain the final recognition results by calculating the similarity between test sequence and template sequence. The proposed approach allows the user to train a reference (template) sequence of dynamic hand gesture. In order to ensure real-time behavior, reference gesture sequence is limited to 40 images. When the training is finished, these images are saved in an xml file. During recognition, once a sequence of new images representing a dynamic hand gesture is made available by the Kinect, the IDTW algorithm is activated to recognize it based on the similarity between the observed gesture and each reference gesture.

TABLE 1. The features of the 10 Chinese digit gestures.

Gesture	Feature representation
Zero	0 fingertip
One	1 fingertip, the peak value of S curve greater than threshold μ ($\mu = 80$ after several calculations)
Two	2 fingertips and $\alpha < \omega$ (according to [16], $\omega = 60°$)
Three	3 fingertips, no convex point between two fingertips in S curve
Four	4 fingertips
Five	5 fingertips
Six	2 fingertips, $\alpha > 60°$, and exist convex point between two fingertips in S curve
Seven	3 fingertips, exist convex point between two fingertips in S curve
Eight	2 fingertips, $\alpha > 60°$, no convex point between two fingertips in S curve
Nine	1 fingertip, the peak value of S curve less than μ

A. 3D TRACING AND LOCATION OF HAND RECOGNITION

In order to ensure real-time, natural experiences in HCI system, hand tracing and positioning methods should be robust to the change of illumination, color and complex background. In this paper, hand tracking and localization algorithm takes fully use of the depth and joints information to describe the real-time coordinate of hand node and generate a 3D motion trajectory. To transform the coordinate system of the depth and skeleton image to that of the color image, some calibration parameters are adjusted so that the depth pixels can match the color pixels. The tracking results of a 3D dynamic hand gesture and its 3D trajectory of the acquired hand gesture are shown in Fig.10.

B. DYNAMIC FEATURE EXTRACTION

Dynamic hand gestures not only contain three-dimensional position information, but also involve time information. Therefore, the joint coordinate sequence of dynamic hand gesture should be transformed into a feature vector which can be used in training and recognition of classification model. In most of previous research, direction, position, and speed are the most commonly used gestures features in dynamic

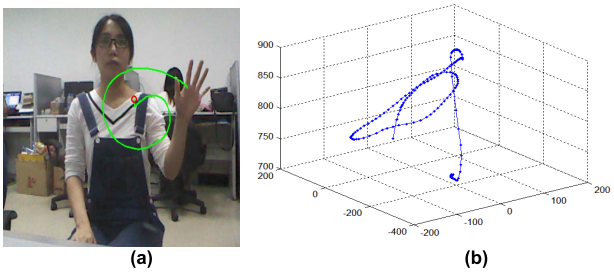


FIGURE 10. 3D tracing and localization of hand gesture. (a) The tracking results of a hand gesture; (b) 3D trajectory.

gesture recognition system. In the existing research on the dynamic gesture recognition based on Kinect, [2] proposed 15 skeleton nodes as feature vector. For a higher gesture recognition ratio, we adopt Euclidean distance of hand joints and the modulus ratio of human skeleton structure feature vector as the main feature of dynamic gesture recognition algorithm.

In order to ensure the translation invariance and scalability of the feature vector for dynamic hand gesture cognition, the difference of users, such as body, height, thin, position to Kinect and so on, should be effectively eliminated. In this paper, the 3D coordinate information of each joint is firstly normalized by calculating the Euclidean distance between hand joints. Meanwhile, the skeleton structure feature vector of human hand is also constructed using coordinate information of human hand joints. On this basis, the modulus ratio of skeleton structure feature vector is calculated. Finally, the unified feature vector for dynamic hand gesture recognition is built based on the Euclidean distance of hand joint points and the modulus ratio of skeleton structure feature vector.

The Euclidean distance between hand, elbow, shoulder joint point and shoulder center are concretely used to represent the geometric feature of dynamic hand gestures. Let three-dimensional coordinate of shoulder center point s and hand joint j in Kinect coordinate system at time t be $P_{st}(x_{st}, y_{st}, z_{st})$ and $P_{jt}(x_{jt}, y_{jt}, z_{jt})$, respectively. The Euclidean distance between them can be calculated as follows:

$$d_{jst} = \sqrt{(x_{jt} - x_{st})^2 + (y_{jt} - y_{st})^2 + (z_{jt} - z_{st})^2} \tag{3}$$

where $j = 1, 2, \ldots, 4$, P_j represents the joint of hand left, hand right, elbow left, and elbow right, respectively. According traits of human arms, the skeleton structure feature vector of hand gestures can be built with corresponding joint data.

In order to further describe the position features of hand gesture relative to the body, four auxiliary feature vectors are constructed as shown in Fig.11., i.e. $\overrightarrow{\lambda_{sh}}$, $\overrightarrow{v_{hr}}$, $\overrightarrow{v_{hl}}$, $\overrightarrow{v_{sr}}$, and $\overrightarrow{v_{sl}}$. Their modulus is $\left|\overrightarrow{\lambda_{sh}}\right|$, $\left|\overrightarrow{v_{hr}}\right|$, $\left|\overrightarrow{v_{sr}}\right|$, and $\left|\overrightarrow{v_{sl}}\right|$, respectively. Therefore, the modulus ratio of the auxiliary feature vector is calculated to create a modulus ratio feature vector $\overrightarrow{R}$

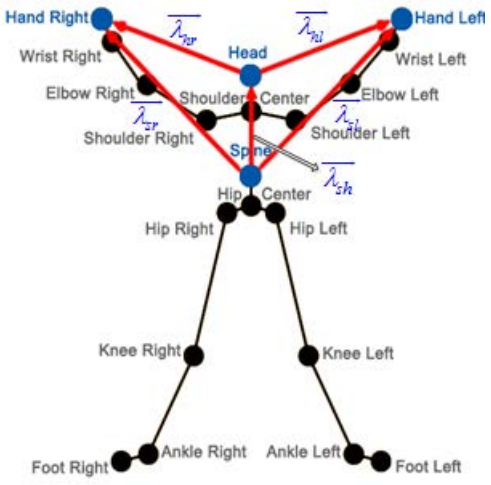


FIGURE 11. Auxiliary feature vector for modulus ratio information.

as follows:

$$r_1 = \frac{\left|\overrightarrow{v_{hr}}\right|}{\left|\overrightarrow{\lambda_{sh}}\right|}, \quad r_2 = \frac{\left|\overrightarrow{v_{hl}}\right|}{\left|\overrightarrow{\lambda_{sh}}\right|}, \quad r_3 = \frac{\left|\overrightarrow{v_{sr}}\right|}{\left|\overrightarrow{\lambda_{sh}}\right|}, \quad r_4 = \frac{\left|\overrightarrow{v_{sl}}\right|}{\left|\overrightarrow{\lambda_{sh}}\right|} \tag{4}$$

According to Euclidean distance between hand joints and the modulus ratio of feature vectors, we can analyze the normalization effect of three-dimension coordinate between hand joint points and the dynamic process of hand gesture. Once the Euclidean distance of hand joint points and the modulus ratio of auxiliary feature vectors are obtained, we can further construct a unifying gesture feature vector to represent the dynamic hand gesture by combining the two different gesture feature vectors. For the hand gesture feature of the i-th frame image in the dynamic hand gesture sequence, the unifying feature vector Z_i can be described in (5).

$$Z_i = < z_{i1}, z_{i2}, \ldots, z_{i8} > \tag{5}$$

where $z_{ij}(j = 1, 2, \ldots, 8)$ is the Euclidean distance between hand joints in the image or the modulus ratio of the auxiliary feature vector as above defined. Z_i is an eight-dimensional vector of hand gesture feature, including the Euclidean distance between four joints (i.e. hand left, hand right, elbow left, and elbow right) and shoulder center, and the four modulus ratios of auxiliary skeleton structure feature vectors. In order to ensure real-time behavior of the system, the length of a dynamic hand gesture sequence is limited to 40 images. Therefore, the unifying feature vector for a dynamic hand gesture are described in (6).

$$Z = < Z_1, Z_2, \ldots, Z_n > \tag{6}$$

where n is the number of total frames of a dynamic hand gesture sequence, $n = 1, 2, \ldots, 40$. Z_i is the gesture feature vector of i-th frame image defined in (5).

C. IMPROVED DTW ALGORITHM

In the previous research of dynamic gesture, the most frequently methods are HMM and DTW [3], [4], [10]. However, HMM not only needs a huge training data but also demands a cumbersome and complex computation. Therefore, we choose the DTW algorithm as the gesture recognition algorithm of the system.

DTW is a template matching algorithm to find the best match for a test pattern out of the reference patterns, where the patterns are represented as a time sequence of features. In our case, let template gesture sequence be $L = (l_1, l_2, l_3, \ldots, l_n), n \in \mathbb{N}$ and the test gesture sequence be $S = (s_1, s_2, s_3, \ldots, s_m), m \in \mathbb{N}, m \neq n$. l_i is i-th frame image of the template and s_j is j-th frame image of the test sequence. They have same internal dimension defined by (5). According to (6), l_i and s_j can be expressed as Z_{l_i} and Z_{s_j}, respectively.

The basic idea of DTW algorithm is to align the two sequences L and S in time via a best path to make the sum of cost minimum and this path must pass through all the points of sequence S. The computation and time complexity of conventional DTW algorithm will greatly increase with the length of gesture sequence in the iteration process. Moreover, in a typical hand gesture recognition problem, hand joints used in a hand gesture can vary from gesture class to gesture class. Hence, not all joints are equally important in recognizing a hand gesture. In this paper, we present an improved DTW algorithm (IDTW) by restricting the wrapping path and using a weighted distance in the cost computation [8].

Firstly, in order to reduce the DTW computational complexity and increase the reliability of DTW's dissimilarity measure, some global constraints have been imposed to the wrapping path [10], [25]. In this paper, we use a well-known global constraint parallelogram band to constrain search path [25], which can effectively limit the warping amount, i.e., slowing down or speeding up of a sequence in time. In the parallelogram, the maximum slope is 2 and the minimum slope is 0.5 shown in Fig.12.

According to the feature of the parallelogram in Fig.12, the length M and N of two hand gesture sequences can be limited in (7).

$$\begin{cases} 2M - N \geq 3 \\ 2N - M \geq 2 \end{cases} \tag{7}$$

If the length of template gesture sequence L and unknown gesture sequence S are not satisfied the constrains by (7), there is no need to compare each frame image in X axis and Y axis, then we only need to compare them in the interval $Q[y_{min}, y_{max}]$. y_{min} and y_{max} of interval Q can be computed by (8) and (9), respectively.

$$y_{\max} = \begin{cases} 2x & 0 \leq x \leq X_u \\ \frac{1}{2}x + (M - \frac{1}{2}N) & X_u < x \leq N \end{cases} \tag{8}$$

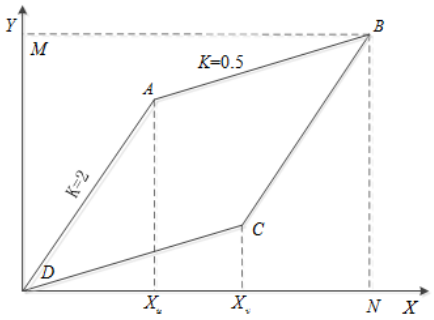


FIGURE 12. DTW constrain search path.

$$y_{\min} = \begin{cases} \frac{1}{2}x & 0 \leq x \leq X_v \\ 2x + (M - 2N) & X_v < x \leq N \end{cases} \tag{9}$$

On the other hand, the conventional DTW algorithm gives equal weights to all dimensions of a sequence sample. However, in typical dynamic hand gesture recognition, hand joints can vary from one gesture class to anther gesture class. Therefore, we propose a weighted DTW algorithm that uses a weighted distance in the cost computation. Different from the weighted DTW algorithms in [26], the weights in this paper are obtained based on a joint's displacement in a dynamic hand gesture. To infer a joint's weight in a trained template gesture, we compute its total displacement by (10).

$$D_j^w = \sum_{k=2}^{K} Dist^w(l_{k-1}^j, l_k^j) \tag{10}$$

where K is the total frame number of template gesture sequences, w is the gesture index, and j is the joint index. $Dist^w()$ calculate the moving distance of j-th joint's two consecutive coordinates in feature vectors l_{k-1}^j and l_k^j. Thus, j-th joint's weight value ω_j^w for hand gesture w is calculated in (11).

$$\omega_j^w = \frac{1 - e^{-\beta D_j^w}}{\sum_{k=1}^{N} (1 - e^{-\beta D_k^w})} \tag{11}$$

According to (11), if joint j remains static in performing hand gesture w, its weight ω_j^w is zero. On this basis, to incorporate these weights, the final DTW distance between template gesture sequence L and test gesture sequence S is transformed as:

$$D'[L, S] = \min_T \sum_{h=1}^{H} \sum_r [d_r(L_{i(h)}, S_{j(h)}) \omega_r^w] \tag{12}$$

where ω_r^w is r-th joint's weight value and r is the number of joints in hand gesture w.

The parameter β in (11) can be calculated by minimizing the within-class variation while between-class variation is maximized [21]. Defining the average weighted DTW distance cost between all samples of hand gesture

sequence N and gesture sequence M as $D_{nm}(\beta)$, the between-class dissimilarity $D_B(\beta)$ of two gesture sequence is the sum value of all $D_{nm}(\beta)$ by:

$$D_B(\beta) = \sum_{n} \sum_{m,n \neq m} D_{nm}(\beta) \tag{13}$$

Within-class dissimilarity $D_w(\beta)$ is the sum of within-class distance $D_{gg}(\beta)$ for all gesture g.

$$D_w(\beta) = \sum_{g} D_{gg}(\beta) \tag{14}$$

The discriminant ratio of a given β, $R(\beta)$, is then obtained by:

$$R(\beta) = \frac{D_B(\beta)}{D_w(\beta)} \tag{15}$$

The optimal value β^* is chosen as the one that maximizes $R(\beta)$ as follows:

$$\beta^* = \arg\max_{\beta}[R(\beta)] \tag{16}$$

V. EXPERIMENTAL RESULTS

Several experiments were performed in order to test the proposed methods for static and dynamic hand gestures recognition. All experiments were carried out on an Intel Core(TM) i7-4790 3.60 GHz CPU with 8 GB of RAM. Kinect for 3D sensor was used as data acquisition device. Visual Studio 2010, Kinect SDK-v1.8 and C# programming languages were employed as the programming tools.

A. RESULTS OF HAND DETECTION AND TRACING

We have tested the performance of hand detection and tracking by sign digits from *Zero* to *Nine* shown in Fig. 8. In normal scene, a few testing samples for the gesture segmentation, the detection of the contour, palm, and fingertips are shown in Fig.13. Statistic results of 10 sign digits are shown in TABLE 2. The average accuracy of each gesture in the Numbers scenario is calculated. For the gesture *Five* and *Four*, the accuracy is nearly 100%. The worst case is the *Nine* with accuracy of 74%. For *Three*, *Six*, *Seven*, *Eight*, and *Nine*, each of them consists of three fingers and presents some difficulty for the system to distinguish among them.

From TABLE 2, we are sure that the proposed solution is able to correctly locate the points of interest over the hand surface as well as its contour. The average accuracy of fingertip detection for 10 sign digits is 97.8%, and that of gesture segmentation is 98.4%. Compared with K curvature method [24], the proposed K-CDD method for fingertip detection is more robust and also can effectively eliminate noise near the wrist.

B. RESULTS OF STATIC HAND GESTURE RECOGNITION

In order to test the performance of the proposed solution for static hand gesture recognition, we use the sign digits from *Zero* to *Nine* and perform the experiments in normal lighting condition and complex scenes. Five volunteers are invited to perform each of the 10 gestures for 100 times: 50 times with left hand and 50 times with right hand. The distance between the camera and the hands is about 1000mm. The test user is given a sequence of images corresponding to different sign digits, and he/she is asked to reproduce them. Each test user is allowed to practice each gesture once or twice before tests. The recognition results can be seen on screen in real time. Some samples for the hand gesture cognition in normal illumination condition are shown in Fig.14. The confusion matrices of hand cognition for 10 sign digits are given in Fig.15 (a). The confusion matrices of 10 sign digits under weak illumination are shown in Fig.15 (b) for comparison.

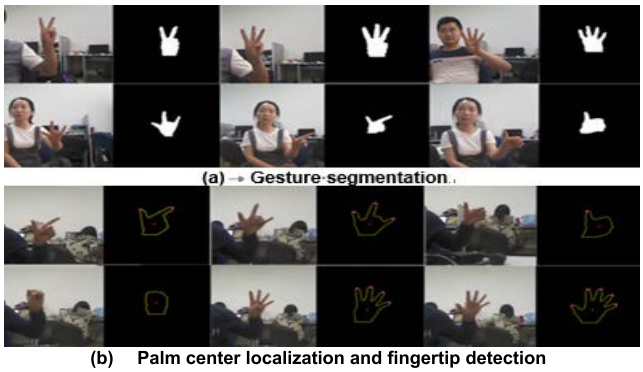


FIGURE 13. Test examples of gesture segmentation and fingertip detection.

TABLE 2. Statistic results of gesture segmentation and fingertip.

Gesture	Total times	Gesture segmentation			Fingertip detection		
		True	False	Accuracy	True	False	Accuracy
Zero	100	100	0	100%	100	0	100%
One	100	100	0	100%	99	1	99%
Two	100	99	1	99%	98	2	98%
Three	100	98	2	98%	96	4	96%
Four	100	97	3	97%	96	4	96%
Five	100	97	3	97%	95	5	95%
Six	100	96	4	96%	97	3	97%
Sever	100	98	2	98%	98	2	98%
Eight	100	99	1	99%	99	1	99%
Nine	100	100	0	100%	100	0	100%
Total	1000	984	16	98.4%	978	22	97.8%

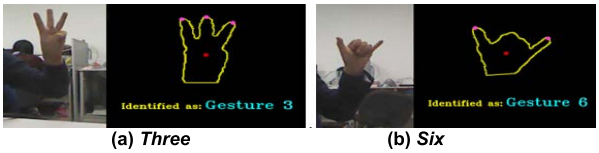
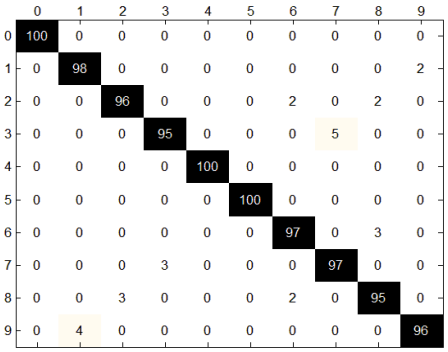


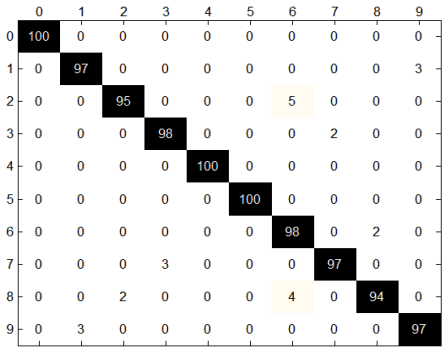
FIGURE 14. Recognition example in normal illumination scene.

The average recognition rates of 10 sign digits in two kinds of light conditions are 97.4% and 97.6%, respectively. Therefore, the illumination has no effect on the recognition results.

In traditional hand gesture cognition using RGB images, the complexity of the background is also one of crucial factors that affect the cognition performance. In order to further verify the performance of the proposed cognition methods in



(a) Normal lighting condition



(b) Weak lighting condition

FIGURE 15. The confusion matrix of static gesture recognition.

(a) (b) (c)

FIGURE 16. Recognition example in complex backgrounds, (a) the recognition result of gesture *Five* in multi-user scene, (b) The recognition result of gesture *Eight* in multi-user and weak light scene, (c) The recognition result of gesture *Zero* in strong light scene.

complex background conditions, we test the ten gestures in different scenes, i.e. multi-user scene, multi-user and weak light scene, and strong light scene. A few cognition samples are shown in Fig.16. Fig. 16(a) is the testing result of gesture *Five* in multi-user and normal light scene (Scene 1). Fig.16(b) is the testing result of gesture *Eight* in multi-user and weak light scene (Scene 2). The testing result of gesture *Zero* in strong light scene (Scene 3) is shown in Fig. 16(c). Statistic results of 10 sign digits in different scene are illustrated in Fig.17. The average recognition rates of 10 sign digits in complex background scenes is more than 95.5%.

Analyzing the experimental results of 10 sign digits in different scene, we can observe that the proposed method of static hand gesture recognition, which combines depth data, fingertip features and S curve features, can accurately identify the hand gesture and has a strong robustness to complex background such as illumination, multi-user interference, and so on.

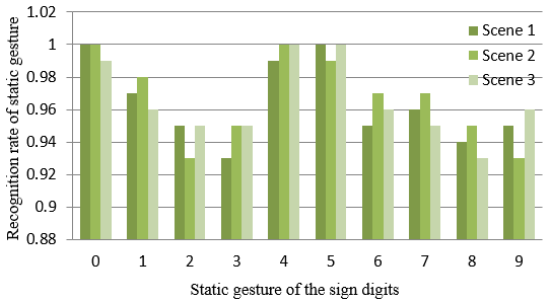


FIGURE 17. The statistic recognition rate of static gesture in complex backgrounds.

TABLE 3. Comparison with the state-of-the-art systems related to sign language for static gesture.

Hand Gesture	State-of-the-art Methods (%)					Ours (K-CCD)
	FEMD [7]	DTW [2]	K-Curvature [24]	Random Forest[28]	RBF-kernel[29]	
Zero	100	-	-	-	90	100
One	95	100	-	-	100	97
Two	86	100	-	-	-	95
Three	94	80	-	-	100	98
Four	87	100	-	-	80	100
Five	89	80	-	-	100	100
Six	95	90	-	-	100	98
Sever	96	95	-	-	-	97
Eight	92	100	-	-	-	94
Nine	98	100	-	-	80	97
Accuracy	93.2	93.9	73.7	96.3	94.58	97.4

Finally, we compared the static hand gesture recognition algorithm with the state-of-the-art systems related to the sign language, including the FEMD (Finger-EarthMover's Distance) [7], DTW [2], K-Curvature [24], Random Forest [28], and RBF-kernel (Radial Basis Function kernel) [29]. The results are demonstrated in TABLE 3. Due to the definitions of the 10 digital gestures are not exactly the same, we only list the recognition results of very similar individual gestures. On the other hand, K-Curvature in [24] and RBF-kernel in [28] only have the average recognition accuracy. In addition, the recognition accuracy in [29] is the average values of all sign language gestures.

C. RESULTS OF DYNAMIC HAND GESTURE RECOGNITION

1) DATASET

To validate the proposed algorithm for dynamic gesture recognition, we used the experimental hand gesture dataset trained and generated by our volunteers according to the UDLR-8 datasets in [27], as illustrated in Fig. 18, where arrow direction indicates the direction of hand gesture movement and $U_i(i = 1, 2, \ldots, 8)$ is the gesture number. For example, U1:RP represents the first hand gesture U1 that moves from left to right.

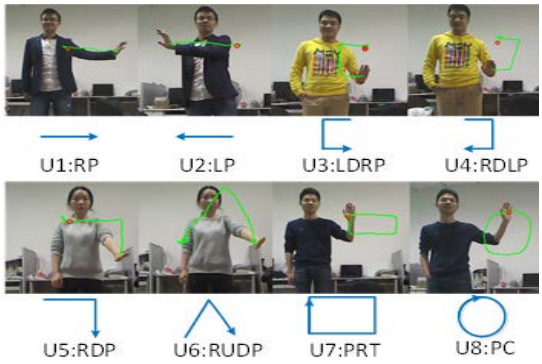


FIGURE 18. Experiment gesture dataset. UDLR-8 gesture dataset.

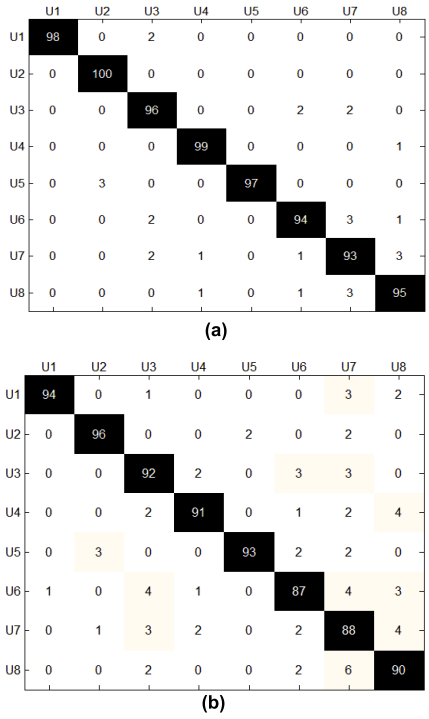


FIGURE 19. The confusion matrix for our UDLR-8 gestures. (a) IDTW recognition results, (b) DTW recognition results.

2) EXPERIMENTAL RESULTS OF DYNAMIC HAND GESTURE

For each gesture, we select ten samples from the fifty trained templates randomly and invite another ten volunteers to test the proposed cognition algorithm. Each gesture will be tested ten times by each of the ten volunteers. Therefore, the total test number of each gesture is one hundred. A gesture is considered unrecognizable if the IDTW algorithm displays a wrong recognized result within a predefined interval after test user finished his/her performance. We carried out test experiment under normal light and weak light condition to verify the performance of recognition algorithm.

The test results for our UDLR-8 gestures using IDTW algorithm and DTW are shown in confusion matrix in Fig.19 (a) and Fig.19 (b), respectively. We can further obtain that the average recognition rates of the proposed IDTW algorithm is 96.5%, and the average recognition rates of the DTW algorithm is 91.4%. According to our test results, the gesture recognition rates are almost same under different lighting conditions. In addition, the IDTW algorithm for dynamic gesture recognition mainly based on the human joint information and skeleton structure features obtained by Kinect, therefore, the recognition results also are independent of human body of different test users. More results and details can be referred in [8].

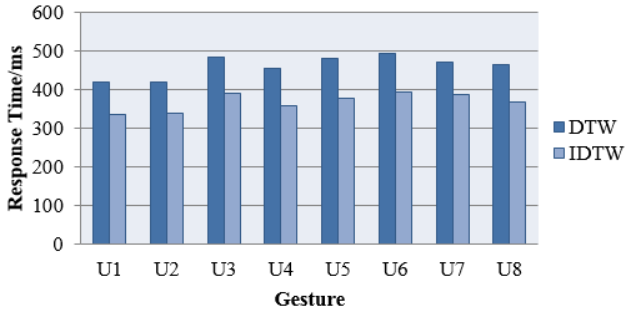


FIGURE 20. The response time comparison of DTW and IDTW.

Fig. 20 lists the comparison of IDTW algorithm with DTW algorithm on response time. As one can observe, the average response time is less than 500ms IDTW algorithm versus DTW algorithm. According to test results shown in Fig.19 and Fig.20, the proposed IDTW algorithm not only improves the total recognition rate but also decreases the recognition response time.

VI. APPLICATION TO INTERACTION WITH VIRTUAL COALMINE

To verify the proposed solutions of hand gesture recognition, we have developed an interactive application to control a virtual coalmine by hand gestures. The virtual coalmine is a specific application of virtual environment technology in coalmine to simulate the underground production environment, including devices, environments, and miners. The interactive system obtains the depth data and joint information of human hand using Kinect sensor and runs the proposed algorithms of static and dynamic hand gesture recognition. According to recognition results of the predefined hand gestures, the interactive system sends real time control command to virtual environment engine and realize the interaction with virtual coalmine. Considering principle of human ergonomics and daily habit of human communication, we defined 16 hand gestures illustrated in Fig. 21.

The corresponding interactive semantic of the predefined hand gestures is listed in TABLE 4. Using these predefined hand gestures, we can control virtual miner's motion, virtual device's status, and view angle of camera to realize the typical interaction with virtual mine.

Fig.22 shows the typical interactive scenario with virtual coalmine according to hand gesture recognition.

Fig.22 (a) and Fig.22 (d) are the interface to virtual mine, which displays user hand gesture, the depth image in Kinect, the recognized hand gesture, command semantic of hand

TABLE 4. The defined hand gestures.

No.	Descriptions	Functions	No.	Descriptions	Function
01	Both hands keep down	Initialization	09	Right hand swipes left and left hand keeps straight forward	Visual angle turns left
02	Left hand swipes up and right hand down	Virtual miner walks	10	Right hand swipes right and left hand keeps straight forward	Visual angle turns right
03	Right hand swipes left and left arm keeps down	Virtual miner turns left	11	Left hand swipes up and right hand keeps straight forward	Camera and visual angle move up
04	Right hand swipes right and left arm keeps down	Virtual miner turns right	12	Left hand swipes down and right hand keeps straight forward	Camera and visual angle move down
05	Zero	Push virtual mine car	13	Left hand swipes left and right hand keeps straight forward	Camera and visual angle move left
06	One	Press virtual devices bottom	14	Left hand swipes right and right hand keeps straight forward	Camera and visual angle move right
07	Right hand swipes up and left arm keeps straight forward	Visual angle turns up	15	Both hands straight forward, then swipe to left and right	Camera moves forward and view zooms in
08	Right hand swipes down and left hand keeps straight forward	Visual angle turns down	16	Both hands straight to left and right, then swipe forward	Camera moves backward and view zooms out

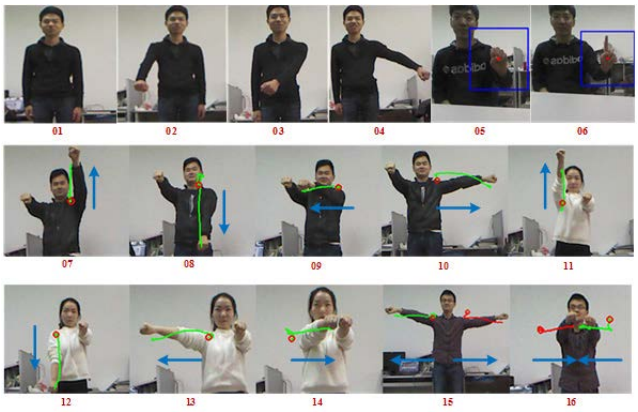


FIGURE 21. The defined gestures and corresponding control commands.

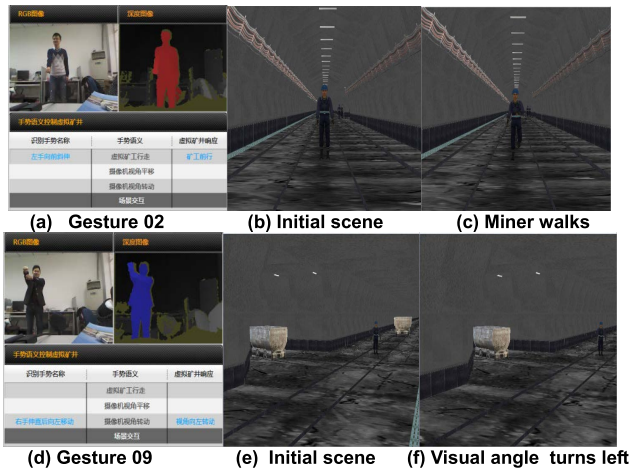


FIGURE 22. Natural interaction with virtual coalmine.

gesture, and the system response to user hand gestures. In Fig.22 (a), user performs the gesture 02, i.e. his left hand swipes up and his right hand down. Fig.22 (c) shows that virtual miner is walking. In Fig.22 (d), user performs the gesture 09, i.e. his right hand swipes left and his left hand keeps straight forward, and virtual coalmine's visual angle turns left, shown in Fig.22 (f).

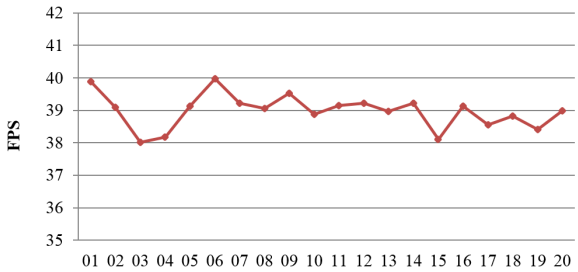


FIGURE 23. The FPS of interacting with virtual coalmine.

Fig.23 shows the real-time frame frequency test when users interactively control the virtual mine according to the defined interactive gestures. We have counted 20 times of interactive tests. It can be seen that when the system is in the initial state or triggers another state, the frame rate is high. When the user makes two gestures at the same time, the frame rate will decrease. Currently, system frame rate is basically stable at 38∼40 FPS and meets the real-time requirements of virtual environment system.

VII. CONCLUSION AND NEXT WORK

This paper proposed a RGB-D based recognition method of static and dynamic hand gesture for natural human-computer interaction application with virtual environment. To static hand gesture recognition, we built a novel K-CCD method, which adds the feature of S curves and angle α between two fingertips and palm center to reduce resemble fingertips and improve recognition rate. For dynamic hand recognition, we combined Euclidean distance between hand joints and

shoulder center with the modulus ratios of skeleton features to generate a unifying feature descriptor for each dynamic hand gesture and proposed an IDTW algorithm to obtain recognition results. In our experiment evaluation, the system achieved an average performance of 97.4%, 96% for static and dynamic gestures, respectively. Finally, we realized a low-cost real-time application of natural interaction with virtual coalmine by hand gesture recognition. However, although we defined 16 static and dynamic gestures to interact with the virtual coalmine, current interactive gestures mainly focus on the scene and vision of virtual environment, which are relatively simple. Moreover, the disabled people with hand deformities cannot used our methods. Some other HCI methods such as speech and brain-computer interface technology may be more suitable for them. In addition, this paper applied Kinect to obtained the RGB-D data for current application, other depth sensors such as Intel RealSense, Leap Motion Controller and ASUS Xtion, will be tested for the proposed algorithms of static and dynamic hand gesture recognition.

In future, we will improve the overall performance of hand gestures recognition by extracting more robust and discriminative features and optimizing the recognition algorithm. Moreover, we will further enrich the HCI functions for virtual coalmine by designing more effective interaction hand gestures so as to improve the practicability of virtual coalmine. In addition, current hand gesture recognition algorithm needs to be trained according to specific applications, we plan to explore the popular deep learning based approaches for hand gesture recognition with smaller datasets and lightweight algorithms to enhance its learning ability and further improve its adaptability and expansibility.

REFERENCES

[1] H. Cheng, L. Yang, and Z. Liu, "Survey on 3D hand gesture recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1659–1673, Sep. 2016, doi: 10.1109/TCSVT.2015.2469551.

[2] G. Plouffe and A.-M. Cretu, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 2, pp. 305–316, Feb. 2016, doi: 10.1109/TIM.2015.2498560.

[3] J. L. Raheja, M. Minhas, D. Prashanth, T. Shah, and A. Chaudhary, "Robust gesture recognition using Kinect: A comparison between DTW and HMM," *Optik, Int. J. Light Electron Opt.*, vol. 126, nos. 11–12, pp. 1098–1104, Jun. 2015, doi: 10.1016/j.ijleo.2015.02.043.

[4] M. Kowdiki and A. Khaparde, "Automatic hand gesture recognition using hybrid meta-heuristic-based feature selection and classification with dynamic time warping," *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100320, doi: 10.1016/j.cosrev.2020.100320.

[5] Y. Shi, Y. Li, X. Fu, K. Miao, and Q. Miao, "Review of dynamic gesture recognition," *Virtual Reality Intell. Hardw.*, vol. 3, no. 3, pp. 183–206, Jun. 2021.

[6] Y. S. Tan, K. M. Lim, and C. P. Lee, "Hand gesture recognition via enhanced densely connected convolutional neural network," *Expert Syst. Appl.*, vol. 175, Aug. 2021, Art. no. 114797, doi: 10.1016/j.eswa.2021.114797.

[7] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1110–1120, Aug. 2013, doi: 10.1109/TMM.2013.2246148.

[8] C. Linqin, C. Shuangjie, X. Min, Y. Jimin, and Z. Jianrong, "Dynamic hand gesture recognition using RGB-D data for natural human-computer interaction," *J. Intell. Fuzzy Syst.*, vol. 32, no. 5, pp. 3495–3507, Apr. 2017, doi: 10.3233/JIFS-169287.

[9] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with Kinect depth camera," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 29–39, Jan. 2015, doi: 10.1109/TMM.2014.2374357.

[10] T. Arici, S. Celebi, A. S. Aydin, and T. T. Temiz, "Robust gesture recognition using feature pre-processing and weighted dynamic time warping," *Multimedia Tools Appl.*, vol. 72, no. 3, pp. 3045–3062, Jul. 2013.

[11] X. Chen and M. Koskela, "Skeleton-based action recognition with extreme learning machines," *Neurocomputing*, vol. 149, pp. 387–396, Feb. 2015.

[12] S. Bhattacharya, B. Czejdo, and N. Perez, "Gesture classification with machine learning using Kinect sensor data," in *Proc. 3rd Int. Conf. Emerg. Appl. Inf. Technol.*, Kolkata, India, Nov. 2012, pp. 348–351.

[13] G. Saponaro, G. Salvi, and A. Bernardino, "Robot anticipation of human intentions through continuous gesture recognition," in *Proc. IEEE Int. Conf. Collaboration Technol. Syst. (CTS)*, San Diego, CA, USA, May 2013, pp. 218–225.

[14] G. Du, P. Zhang, and D. Li, "Human–manipulator interface based on multisensory process via Kalman filters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5411–5418, Oct. 2014, doi: 10.1109/TIE.2014.2301728.

[15] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava, "Accurate 3D action recognition using learning on the Grassmann manifold," *Pattern Recognit.*, vol. 48, no. 2, pp. 556–567, Feb. 2015.

[16] I. Almetwally and M. Mallem, "Real-time tele-operation and tele-walking of humanoid robot NAO using Kinect depth camera," in *Proc. 10th IEEE Int. Conf. Netw., Sens. CONTROL (ICNSC)*, Evry, France, Apr. 2013, pp. 463–466.

[17] Q. K. Le, C. H. Pham, and T. H. Le, "Road traffic control gesture recognition using depth images," *IEEE Trans. Smart Process. Comput.*, vol. 1, no. 1, pp. 1–7, Aug. 2012.

[18] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, New York, NY, USA, 2011, pp. 147–156.

[19] Y. Gu, H. Do, Y. Ou, and W. Sheng, "Human gesture recognition through a Kinect sensor," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Guangzhou, China, Dec. 2012, pp. 1379–1384.

[20] T. D'Orazio, C. Attolico, G. Cicirielli, and C. Guaragnella, "A neural network approach for human gesture recognition with a Kinect sensor," in *Proc. 3rd Int. Conf. Pattern Recognit. Appl. Methods*, Loire Valley, France, 2014, pp. 741–746.

[21] A. A. Chaaraoui, J. R. Padilla-López, P. Climent-Pérez, and F. Flórez-Revuelta, "Evolutionary joint selection to improve human action recognition with RGB-D devices," *Expert Syst. Appl.*, vol. 41, no. 3, pp. 786–794, Feb. 2014.

[22] A. Kulshreshth, C. Zorn, and J. J. LaViola, "Poster: Real-time markerless Kinect based finger tracking and hand gesture recognition for HCI," in *Proc. IEEE Symp. 3D User Interfaces (3DUI)*, Orlando, FL, USA, Mar. 2013, pp. 187–188.

[23] Y. Wang, Q. Gan, and L. Li, "A real time fingertip detection method combining curvature and paralleled-vector," *J. Graph.*, vol. 35, no. 2, pp. 285–289, Apr. 2014.

[24] M. Z. A. Bakar, R. Samad, D. Pebrianti, M. Mustafa, and N. R. H. Abdullah, "Finger application using *K*-curvature method and Kinect sensor in real-time," in *Proc. Int. Symp. Technol. Manage. Emerg. Technol. (ISTMET)*, Kedah, Malaysia, Aug. 2015, pp. 218–222.

[25] H. Kim, S. Lee, D. Lee, S. Choi, J. Ju, and H. Myung, "Real-time human pose estimation and gesture recognition from depth images using superpixels and SVM classifier," *Sensors*, vol. 15, no. 6, pp. 12410–12427, Mar. 2015.

[26] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, Sep. 2011.

[27] D. Xu, Y.-L. Chen, C. Lin, X. Kong, and X. Wu, "Real-time dynamic gesture recognition system based on depth perception for robot navigation," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Guangzhou, China, Dec. 2012, pp. 689–694.

[28] N. Saquib and A. Rahman, "Application of machine learning techniques for real-time sign language detection using wearable sensors," in *Proc. 11th ACM Multimedia Syst. Conf.*, İstanbul, Turkey, May 2020, pp. 178–189.

[29] Y. Zhang, Y. Huang, X. Sun, Y. Zhao, X. Guo, P. Liu, C. Liu, and Y. Zhang, "Static and dynamic human arm/hand gesture capturing and recognition via multiinformation fusion of flexible strain sensors," *IEEE Sensors J.*, vol. 20, no. 12, pp. 6450–6459, Jun. 2020, doi: 10.1109/JSEN.2020.2965580.

JUN XU was born in Anhui, China, in 1972. He received the master's degree in safety technology and engineering from the Anhui University of Science and Technology, Huainan, China, in 1999. He is currently an Associate Professor with the School of Computer and Data Engineering, Bengbu College of Technology and Business, Bengbu, China. He has published more than 60 papers in related journals and conferences. He has edited more than 20 textbooks. His current research interests include science and technology big data, computer networks, and information safety.

HANCHEN WANG was born in Liaoning, China, in 1997. He received the B.S. degree in electrical engineering and automation from Sichuan Agricultural University, Ya'an, China, in 2020. He is currently pursuing the M.S. degree with the School of Automation, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include computer vision, object detection and segmentation, and human–computer interaction.

JIANRONG ZHANG received the master's degree in control science and engineering from the School of Automation, Chongqing University of Posts and Telecommunications. He is currently an Engineer with China Information Technology Design and Consulting Institute Company Ltd., Chengdu, China. His research interests include human–computer interaction, artificial intelligence, and machine learning.

LINQIN CAI (Member, IEEE) was born in Sichuan, China, in 1973. He received the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2007. He is currently a Professor with the School of Automation, Chongqing University of Posts and Telecommunications, Chongqing, China. He has published more than 50 papers in related international journals and conferences. His current research interests include artificial intelligence, human–computer interaction, and pattern recognition. He has served as a reviewer for several international journals.