Dynamic Hand Gesture Recognition Based on 3D Hand Pose Estimation for Human-Robot Interaction

Qing Gao, *Member, IEEE*, Yongquan Chen, *Member, IEEE*, Zhaojie Ju, *Senior Member, IEEE*, Yi Liang

***Abstract*— Dynamic hand gesture recognition is a challenging problem in the area of hand-based human-robot interaction (HRI), such as issues of a complex environment and dynamic perception. In the context of this problem, we learn from the principle of the data-glove-based hand gesture recognition method and propose a dynamic hand gesture recognition method based on 3D hand pose estimation. This method uses 3D hand pose estimation, data fusion and deep neural network to improve the recognition accuracy of dynamic hand gestures. First, a 2D hand pose estimation method based on OpenPose is improved to obtain a fast 3D hand pose estimation method. Second, the weighted sum fusion method is utilized to combine the RGB, depth and 3D skeleton data of hand gestures. Finally, a 3DCNN + ConvLSTM framework is used to identify and classify the combined dynamic hand gesture data. In the experiment, the proposed method is verified on a developed dynamic hand gesture database for HRI and gets 92.4% accuracy. Comparative experiment results verify the reliability and efficiency of the proposed method.**

***Index Terms*— dynamic hand gestures, hand pose estimation, neural network**

I. Introduction

VISION-based hand gesture recognition includes static hand gesture recognition and dynamic hand gesture recognition [1]. Compared with static hand gestures, dynamic hand gestures are more reliable and more natural in human-robot interaction (HRI) or human-computer interaction (HCI). However, dynamic hand gestures include temporal features in addition to spatial features, thus making the recognition of dynamic hand gestures more difficult. In addition, for online recognition of dynamic hand gestures, it is necessary to locate and segment each dynamic hand gesture from streaming video. Therefore, developing effective dynamic hand gesture localization methods is an important research challenge. Looking at these problems, we study the online detection and recognition of dynamic hand gestures, develop a new recognition method, and apply the method to hand-based HRI.

The principal method of hand gesture recognition uses a feature extractor to extract hand gesture features and then a using classifier to classify the extracted features for the recognition of different hand gestures [2], [29]–[32]. At present, notable achievements have been made in image-based hand gesture recognition [2]–[5]; however, utilizing data-glove-based hand gesture recognition methods can yield high hand gesture recognition accuracies and are thus more reliable hand gesture recognition strategies [6]. The reasons for this superiority are as follows. (a) The data gloves interference is small when they are used to obtain information. This interference mainly comes from signal noise; the interference associated with hand images mainly comes from the complex background. Therefore, the interference caused by data gloves is smaller and easier to remove (such as by Kalman filtering) than the interference in images. (b) Data gloves can obtain hand joint information. Image-based hand gesture recognition mainly uses information such as the shape, colour, and 3D position of hand gestures which are common properties of human observation. This type of method is rough and superficial. The nature of hand gestures is linked to the spatial positions or spatial motion relationships of the hand joints. Data gloves can obtain the position information of the hand joints. By matching the position information for each hand joint with a hand joint distribution map (hand skeleton), data gloves can easily and accurately identify static hand gestures. However, data gloves do not have motion measurement modules, so they cannot recognize dynamic hand gestures. In view of this problem, inspired by the data-glove-based hand gesture recognition method, we propose a hand gesture recognition method based on hand pose estimation, as shown in Figure 1, where Figure 1(1) illustrates data-glove-based hand gesture recognition and Figure 1(2) shows hand gesture recognition based on hand pose estimation. This method first estimates the hand pose with a hand skeleton map. Then, the hand pose map is fused with a hand depth image. Detailed joint features and 3D space features of hand gestures can be used to improve

Corresponding author: Yongquan Chen (yqchen@cuhk.edu.cn).

This paper is partially supported by Shenzhen Fundamental Research grant (JCYJ20180508162406177, JCYJ20190813170601651) and the National Natural Science Foundation of China (62006204) from The Chinese University of Hong Kong, Shenzhen. This paper is also partially supported by funding from Shenzhen Institute of Artificial Intelligence and Robotics for Society.

Qing Gao, Yongquan Chen and Yi Liang are with the Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, 518035 China (e-mail: gaoqing@cuhk.edu.cn, yqchen@cuhk.edu.cn, Yi Liang@cuhk.edu.cn).

Qing Gao and Yongquan Chen are also with the Institute of Robotics and Intelligent Manufacturing & School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, 518172 China.

Zhaojie Ju is with the School of Computing, University of Portsmouth, Portsmouth, PO1 3HE U.K (e-mail: Zhaojie.Ju@port.ac.uk).

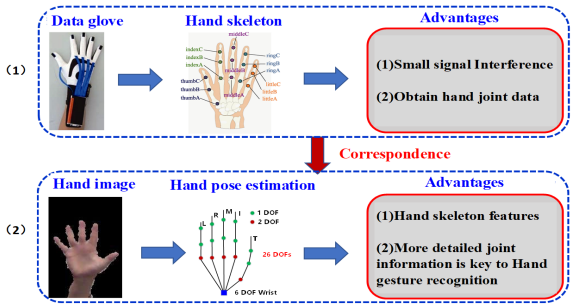

Fig. 1. Correspondence between data gloves and hand image.

the recognition accuracy of vision-based hand gestures.

Based on the above approach,a dynamic hand gesture recognition method based on 3D hand pose estimation and deep neural network is proposed, it can get 92.4% accuracy on a developed dynamic hand gesture database. The innovations and contributions of this method are shown as follows:

- An improved Faster-RCNN with a bi-stream attention (BAs) module is proposed to detect and cut dynamic hand gestures online. Using clipped hand images can improve the accuracy of hand pose estimation.
- By fusing the hand skeleton and depth images, 3D spatial position estimation of hand gestures can be obtained.
- A data-level fusion method using weighted sum is proposed, which can reduce the number of channels used for data fusion, thereby reducing the number of calculations.
- A 3DCNN + ConvLSTM framework for dynamic hand gesture feature extraction is proposed. The short-term spatiotemporal features of dynamic hand gestures are extracted by the 3DCNN, and the long-term spatiotemporal features of dynamic hand gestures are extracted by the ConvLSTM to improve the dynamic hand gesture recognition accuracy.
- A set of dynamic hand gestures for HRI is created, and a corresponding dynamic hand gesture database containing RGB, depth and skeleton information is produced. The proposed method can be validated based on this database.

The remainder of this paper is organized as follows. Chapter 2 introduces the work related to dynamic hand gesture recognition based on deep learning and data fusion. Chapter 3 introduces the 3D hand pose estimation method. Chapter 4 introduces the data fusion and 3DCNN+ConvLSTM framework for dynamic hand gesture recognition. Experiments are introduced in Chapter 5, and they include the hand gesture database and dynamic hand gesture recognition results, as well as the subsequent discussion. The conclusions and future work are introduced in Chapter 6.

II. Related Work

Different illumination intensities, complex backgrounds and environments, and the non-fixed and non-standard hand gestures of diverse people limit the efficient recognition of dynamic hand gestures. Dynamic hand gestures include a series of hand or arm poses. Therefore, learning the spatiotemporal features related to these gestures is essential for robust dynamic hand gesture recognition. According to [7], there are four typical characteristics of methods for effective spatiotemporal feature recognition: they should be (1) generic, (2) compact, (3) computationally efficient, and (4) simple to implement. Furthermore, 3D pose estimation of hand joints can increase the accuracy of dynamic hand gesture recognition. Because the hand often occupies only a small part of a collected image, the accurate detection of dynamic hand gestures is the key to dynamic hand gesture recognition.

A. Performance feature-based hand gesture recognition

At present, most of state-of-the-art hand gesture recognition methods use hand performance features, and deep neural network frameworks are designed to extract and classify these features. For example, [8], [9] proposed a multiscale method to detect and classify dynamic gestures. This approach uses a variety of data such as depth video, joint poses, and voice information to classify upper limb gestures and hand gestures. [10] proposed a CNN-based multisensor system (radar, colour camera and depth camera) to recognize dynamic hand gestures during driving. [11] proposed a ResC3D framework that used RGB, optical flow and depth images to recognize dynamic hand gestures. In addition, there have been some further studies of performance feature-based hand gesture recognition [12]–[15].

B. Skeleton-based hand gesture recognition

There have been also some studies of skeleton-based hand gesture recognition. For example, [16] designed a dynamic hand gesture recognition method based on 2D hand skeleton. [17] designed a 3D dynamic hand gesture recognition method based on skeleton features. This method uses a Gaussian mixture model to extract features, and then the extracted features are classified by a support vector machine (SVM) classifier. The above two methods use hand pose features, but the feature extractors both use traditional methods. Additionally, some scholars have introduced deep learning into pose-based hand gesture recognition. For example, [18] proposed an end-to-end STA-Res-TCN framework to recognize dynamic hand gestures based on skeletons. [19] combined CNN and LSTM methods to solve skeleton-based human motion and gesture recognition problems and proposed a data augmentation method for spatiotemporal 3D data sequences.

At present, most hand gesture recognition methods based on pose estimation still use hand skeleton information alone for recognition. Skeleton information helps to improve the accuracy of hand gesture recognition, but it is difficult to extract skeleton information. One method uses special sensors such as LeapMotion [20], but this method is not universally applicable, and the cost of the sensors is high. Another method involves performing hand pose estimation based on a colour image to obtain a two-dimensional skeletal structure of the hand, such as by using OpenPose [21]. However, this method has two drawbacks: (1) the 2D skeletal structure cannot express the spatial hand pose and motion well, and (2) hand pose estimation methods have limited accuracy. If the pose estimation is not accurate in the early stage, it will greatly affect the hand gesture recognition effect in later stages.

C. Database for skeleton-based hand gesture recognition

The current public databases of dynamic hand gestures that contain skeleton information are the DHG-14 /28 Dataset [17] and SHREC'17 Track Dataset [18]. The DHG-14 /28 Dataset contains 14 hand gestures. Some of these hand gestures involve only one-finger movement, and some involve movements of the entire hand. Each hand gesture acquisition process was repeated 5 times by 20 volunteers, and a total of 2800 sequences were gathered. Each frame includes a hand gestured depth image and 22 joint 2D and 3D hand skeletons. However, the skeleton information in the database was directly obtained by the sensors, and it is difficult to apply this information to a system using ordinary cameras. The SHREC'17 Track Dataset contains 14 dynamic hand gestures, all of which involve the movement of 5 fingers. There are a total of 2800 videos in the database, and each frame contains a 3D hand skeleton with 22 joints. However, this database contains only hand skeleton information.

D. Our method

For the above problems, this paper proposes a dynamic hand gesture recognition method based on the fusion of 3D hand skeletal features, colours and depth images. First, the 2D hand pose obtained from a colour image is mapped to the corresponding depth image to obtain a 3D hand pose. Then, 3D skeleton information is fused with colour and depth images of the corresponding hand gestures. This approach can easily and quickly obtain the 3D hand skeleton information, and it uses various features of hand gestures to reduce the impact of the hand pose estimation error. The explicit method is detailed below.

III. Hand Pose Estimation Based on Hand Detection

This chapter improves on the 2D hand pose estimation method based on OpenPose [22] and implements 3D hand pose estimation. This method is illustrated in Figure 2. First, a Faster-RCNN with BAs is utilized to detect and cut hands in the RGB original image. Then, OpenPose is utilized for the segmented RGB hand image to estimate the 2D pose of the hand. This approach can prevent the influence of a large number of complex backgrounds and thus improve the speed and accuracy of hand pose estimation. Next, the estimated 2D coordinates of the hand skeleton joints are mapped to the corresponding hand depth image. Sparse sampling and averaging methods are utilized to obtain the 3D coordinates of hand skeleton joints. This method can obtain 3D hand pose estimation in a simple and convenient way. Then, each obtained 3D hand pose is fused with the corresponding hand RGB and depth images. Fused data can be input into a deep neural network for feature extraction and classification. The hand pose features and hand movement features can be simultaneously used to improve the dynamic hand gesture recognition accuracy.

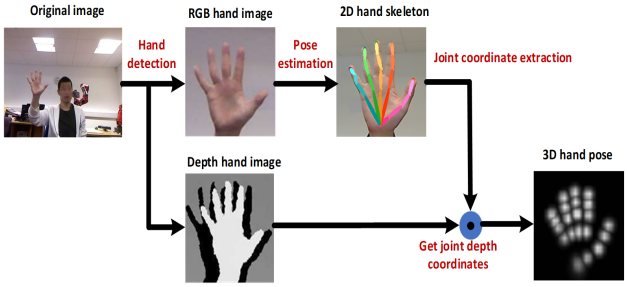

Fig. 2. 3D hand pose estimation pipeline.

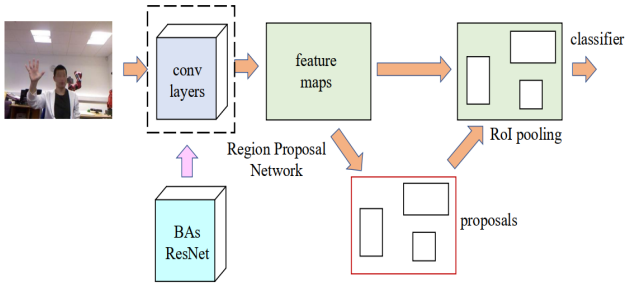

Fig. 3. The structure of Faster-RCNN with BAs.

A. Faster-RCNN with BAs

Hand detection consists of two important roles in this research. (1) Hand detection and the segmentation of each frame of dynamic hand gestures can remove most of the background and decrease the time of hand pose estimation. (2) Each dynamic hand gesture from a hand gesture video can be detected. In this process, the continuous detection of a hand in three frames is regarded as the starting point of a dynamic hand gesture, and the absence of a hand is three subsequent frames is regarded as the end point of the dynamic hand gesture.

Because hands usually appear as tiny objects in original images, some classic deep learning methods, such as SSD [23] and YOLO [24], are not appropriate for tiny object detection. The Faster-RCNN method [25] is good at tiny object detection. But its backbone is not good at extracting spatial and channel features of hand images. Therefore, we introduced a designed bi-stream attention (BAs) module which is inspired by CBAM [26] to the Faster-RCNN to improve the detection speed of the Faster-RCNN. The structure of the Faster-RCNN with BAs is illustrated in Figure 3.

Faster-RCNN with BAs uses BAs-ResNet to replace the original VGG-16 module, which is implemented in the convolutional layers of the Faster-RCNN for image feature extraction. The feature extraction, proposal extraction, bounding box regression, and classification tasks of this method are all integrated into one network. The overall performance of this method is significantly improved compared to that of traditional methods, especially in detection tasks.

Convolutional layers of the traditional Faster-RCNN use VGG-16 for feature extraction. The model can detect and locate tiny objects well, but its backbone is not good at extracting spatial and channel features of hand images. Compared with VGG-16, the BAs-ResNet use spatial attention and channel

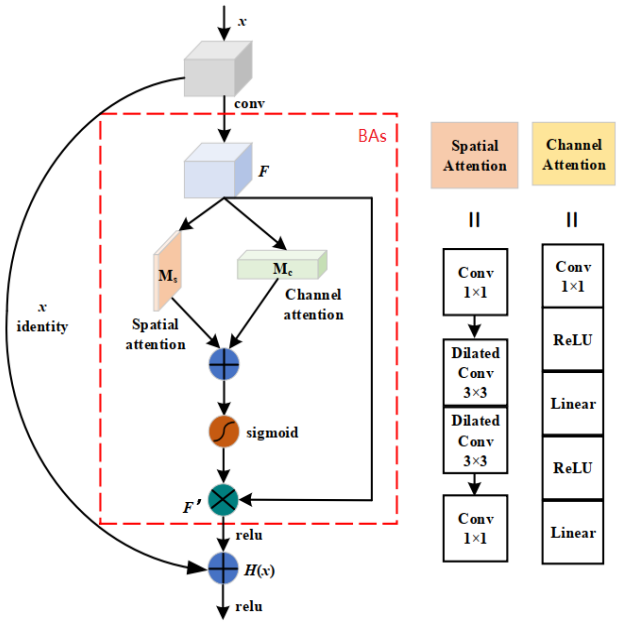

Fig. 4. The structure of BAs-ResNet.

attention modules to increase the ability of feature extraction. The structure of BAs-ResNet is illustrated in Figure 4. The role of channel attention is tantamount to extract the relationship between each channel in a feature map, as it teaches the network to look 'what'. Channel attention attempts to highlight local information by aggregating dimensions of the feature map, as it teaches the network look 'where'. Connecting the two attention modules in parallel can obtain feature maps with more hand semantic information and spatial context information, and the bi-stream structure can reduce the amount of calculation. In addition, the BAs module is introduced to the shortcut structure of the ResNet. It can further improve the effect of feature extraction.

B. OpenPose-based 2D hand pose estimation

The OpenPose-based hand pose estimation method can achieve 2D hand pose estimation for in RGB images, and it can ensure the real-time performance of hand pose estimation. This method uses the Part Conference Map (PCM) and Part Affinity Fields (PAFs) models to develop a bottom-up (obtain hand pose points and then obtain the hand skeleton) hand pose estimation method. First, the positions of the key points of hand gestures are detected. The detection result is obtained by generating a hot map of the key hand points. Each key point of the hand is associated with a Gaussian peak, which indicates that the neural network believes that the point is a key point. After obtaining the detection results, some key points are connected to form a hand. In other words, we must match specific key points with specific parts of the hand in a given picture. In this way, a 2D skeletal structure diagram of a hand can be obtained. The framework of this approach is illustrated in Figure 4.

In Figure 5, the hand pose estimation step first uses the VGG-19 network to obtain the RGB hand gesture features and then transmits the features to the PCM and PAFs. The output of Stage 1 is the corresponding PCM spectrum S^1 and PAFs spectrum L^1. The inputs of Stage 2 are the outputs S^1 and L^1 of Stage 1 and the feature map of the original image. The subsequent phases of the network are similar in Stage 2. Notably, the model loss function is the ordinary L2 norm, which describes the distance between the prediction result and ground truth:

$$f_S^t = \sum_{j=1}^{J} \sum_{P} W(P) \cdot \left\| S_j^t(P) - S_j^*(P) \right\|_2^2 \tag{1}$$

$$f_L^t = \sum_{c=1}^{C} \sum_{P} W(P) \cdot \left\| L_c^t(P) - L_c^*(P) \right\|_2^2 \tag{2}$$

where $S = (S_1, S_2, \cdots, S_J)$ has J confidence graphs and $L = (L_1, L_2, \cdots, L_J)$ has C vector fields. f_S^t and f_L^t are the error functions of the outputs S_t and f_L^t of Stage t, respectively. S_j^t is the output PCM spectrum of Stage t, L_c^t is the output PAFs spectrum of Stage t, S_j^* is the ground truth of the PCM, and L_c^* is the ground truth of the PAFs. W is a binary mask and P is an image position. Then the final error function of the network is:

$$f = \sum_{t=1}^{T} (f_S^t + f_L^t) \tag{3}$$

There are 21 key points in the hand structure, of which there are 4 key points associated with each finger joint and one key point at the centre of the wrist.

C. 3D hand pose estimation method

Through the above steps, a 2D hand pose based on a RGB image is obtained. For a variety of complex hand gesture recognition tasks, using 2D hand pose features often fails to yield good hand gesture recognition accuracy, especially for situations where some hand gestures can block certain finger joint points. Depth information associated with hand joints is very important, and the 3D pose information can fully describe the hand gesture features. Therefore, we designed a method to transform 2D hand pose estimation results to information for 3D hand pose estimation. The process uses the method of mapping hand pose key point coordinates from the RGB image to the corresponding depth image, and this process is divided into the following steps.

1) Alignment of the RGB image and depth image: The RGB images and depth images collected from Kinect are generally misaligned, which will affect the mapping process. Therefore, the acquired RGB image needs to be aligned with the depth image. The alignment formula is shown as follows:

$$Z_{rgb} * p_{rgb} = R * Z_{ir} * p_{ir} + T \tag{4}$$

where

$$R = K_{rgb} * R_{ir2rgb} * K_{ir}^{-1} \tag{5}$$

$$T = K_{rgb} * T_{ir2rgb} \tag{6}$$

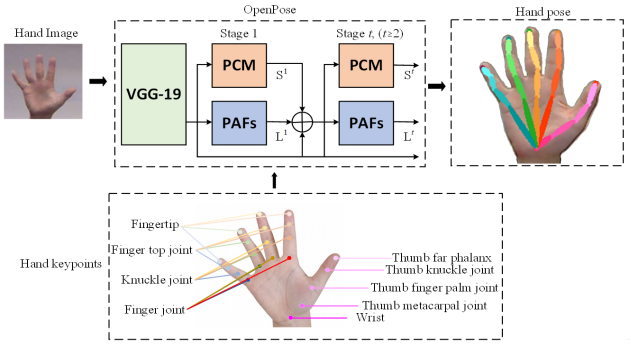

Fig. 5. Hand pose estimation framework.

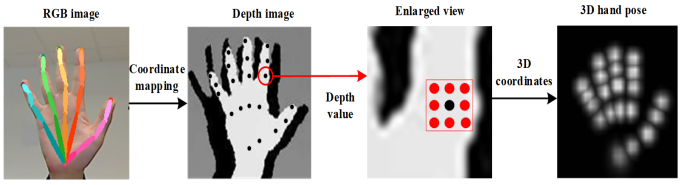

Fig. 6. Flow chart of 2D hand pose to 3D hand pose.

where R_{rgb} and T_{rgb} are external parameters of the colour camera and R_d and T_d are external parameters of the depth camera. K_{rgb} and K_d are internal parameters for the colour camera and depth camera. The two cameras have the following rigid body transformation formula:

$$R_{ir2rgb} = R_{rgb} * R_{ir}^{-1} \tag{7}$$

$$T = K_{rgb} * T_{ir2rgb} \tag{8}$$

where $z * p$ represents the mapping relationship between the homogeneous three-dimensional points ($P = [XYZ1]^T$) in the respective camera coordinate system and the pixel coordinates ($p = [uv1]^T$) in the respective pictures.

2) Coordinate mapping: After the RGB image and the depth image are aligned, the hand skeleton coordinate points estimated with the RGB image are mapped to the corresponding depth image, and the depth value of each key point is determined by the position of the key point on the depth map.

As shown in Figure 6, to reduce calculation redundancies and obtain smooth depth values for key hand point positions, we use sparse sampling and averaging methods. That is, the depth values of the coordinate points of two adjacent pixel values for each key point in the depth map are extracted, and then the average depth value of these points (9 coordinate points) is the depth value of the key point. Finally, the depth values of all key points are extracted and added to a 3D pose map of the hand, as shown on the right side of Figure 5.

IV. Dynamic Hand Gesture Recognition

A. Data fusion method based on weighted sums

The data fusion step can reasonably and effectively integrate the multi-source information representing the same dynamic hand gesture. So, it will extract the beneficial information in the respective channels to the greatest extent, and finally integrate them into high-quality gesture images or videos to improve the utilization of hand gesture image information and improve the accuracy of dynamic hand gesture recognition. Traditional image data fusion methods often use a mounting process [15], that is, an image with a size of $l \times w \times h_1$ is mounted on an image with a size of $l \times w \times h_2$ to generate an image with a size of $l \times w \times (h_1 + h_2)$. The advantages of this fusion method are that it does not lose information and uses all the information from the fused images. However, the disadvantage is that the image size increases after fusion, which results in increases in the deep neural network size and number of parameters and a reduction in the speed of the network model.

To ensure that the fused image does not affect the volume of the trained network model, we propose a weighted sum data fusion method. This approach involves pixel-level fusion, that is, the RGB and depth images and 3D hand pose images are multiplied by a weighting factor, and pixel-level addition is then performed. The corresponding formula is shown as follows:

$$P_F^i = \alpha_1 P_{RGB}^i + \alpha_2 P_D^i + \alpha_3 P_P^i + \alpha_4 \tag{9}$$

where P_{RGB}^i is the RGB hand image in the ith frame, P_D^i is the depth hand image in the ith frame, P_P^i is the 3D hand pose image in the ith frame, and P_F^i is the fusion image in the ith frame obtained by the weighted sum. This image contains information from the RGB hand image, depth hand image and 3D hand pose image without changing the size of the image. α_1 is the weighting coefficient of the RGB hand image, α_2 is the weighting coefficient of the depth hand image, α_3 is the weighting coefficient of the 3D hand pose image, and α_4 is the brightness adjustment coefficient. To balance the information from the RGB, depth and 3D hand pose images, we set the value of α_1 to 0.3, the value of α_2 to 0.3, the value of α_3 to 0.3, and the value of α_4 to 0.1. Finally, the obtained fusion image P_F^i is invoked as the data input of the hand gesture recognition model. A result example of this data fusion step is indicated by Figure 8.

B. Dynamic hand gesture recognition network framework

For the extraction of spatiotemporal features in dynamic hand gesture videos, we propose a network framework combining 3DCNN and ConvLSTM, and this framework is shown in Figure 7.

As shown in Figure 7, the fused dynamic hand gesture video is input into the network framework composed of the 3DCNN and ConvLSTM models for feature extraction and dynamic hand gesture recognition. The proposed framework is based on the following three factors: (1) the 3DCNN is a very effective and excellent module for learning short-term spatiotemporal features; (2) ConvLSTM networks are highly suitable for learning long-term spatiotemporal features; and (3) spatiotemporal correlation information plays a very important role in dynamic hand gesture recognition. Therefore, we propose using the 3DCNN and ConvLSTM models to learn the spatiotemporal features of dynamic hand gestures. In

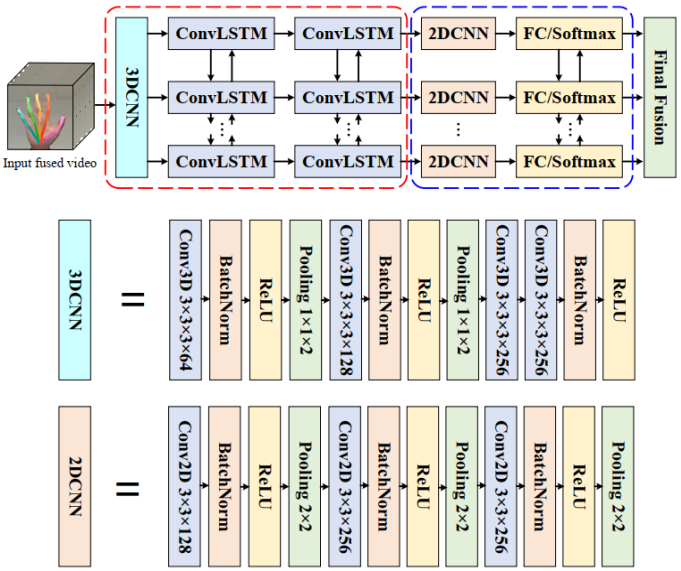

Fig. 7. Network framework.

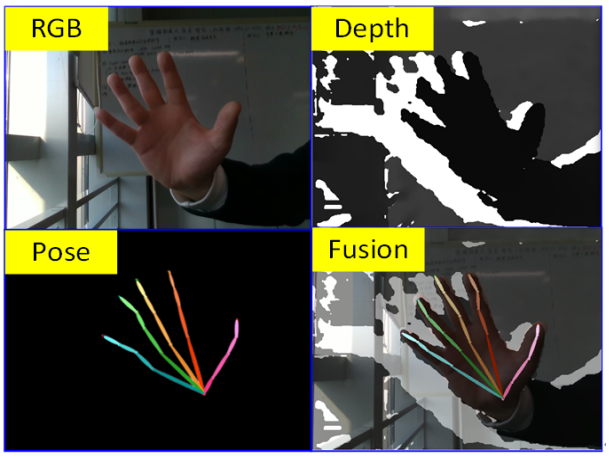

Fig. 8. The result sample of data fusion.

addition, 2DCNNs are used after ConvLSTM to learn high-level spatiotemporal features. The 2D feature maps output from ConvLSTM still have large space sizes and require dimensionality reduction. 2DCNNs can learn more high-level features while reducing dimensionality. The decision fusion step used in the final output part can get a better recognition result.

The structure of the 3DCNN is modelled based on a C3D method [27]. For video analysis problems, the motion information encoded between consecutive frames must be captured. Therefore, during the convolution operation phase of a CNN, a 3D convolution operation is carried out to capture features from both the temporal and spatial dimensions. The 3D convolution operation is implemented as follows: a cube formed by stacking multiple consecutive frames is convolved with a 3D kernel. Through this construction step, the feature map of the convolutional layer is connected to multiple consecutive frames of the previous layer to capture motion information. Formally, the values at point (x, y, z) in the ith and jth feature maps are given by the following formula:

$$v_{ij}^{xyz} = tanh(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} \omega_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}) \tag{10}$$

where R_i represents the size of the 3D kernel in the time dimension and ω_{ijm}^{pqr} is the point (p, q, r) value associated with the kernel of the mth feature map in the previous layer.

In general, for a fully connected LSTM model, vector features are used as the inputs to learn temporal features. The subsequent results lack spatial information, and for dynamic hand gesture recognition, the position changes of hands and fingers in the spatial range play an important role. Therefore, the ConvLSTM model is utilized to learn long-term spatiotemporal features. The convolution and recursive operations of the network can take full advantage of spatiotemporal information in the input-to-state and state-to-state transmission steps.

In the ConvLSTM model, $X_1, X_2, \cdots, X_t$ are the inputs; the neural unit states are $C_1, C_2, \cdots, C_t$; the hidden states are $H_1, H_2, \cdots, H_t$; and the gates i_t, f_t, o_t are all 3D tensors. Let "$*$" denotes the convolution operation, and "$\circ$" denotes the Hadamard product. The ConvLSTM model can be expressed by the formula:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i) \tag{11}$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f) \tag{12}$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o) \tag{13}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \tag{14}$$

$$H_t = o_t \circ \tanh(C_t) \tag{15}$$

where σ is the sigmoid equation and $W_{x\sim}$ and $W_{h\sim}$ are 2D convolutional kernels. The features output by ConvLSTM are input into the 2DCNNs for feature extraction and then classified with Softmax classifiers. Finally, the classification results are combined based on decision fusion to obtain the final results of dynamic hand gesture recognition result.

V. Experimental Results and Analysis

In this section, the proposed dynamic hand gesture recognition method based on pose estimation is verified based on an author-generated hand gesture database. First, we introduce the developed dynamic hand gesture database. Then, a detailed introduction to the training process of the network is provided. Finally, the experimental results are displayed, and the performance of the proposed method is analysed.

A. Dynamic hand gesture database

Because this experiment involves fusing RGB hand images and depth hand images with 3D hand pose images, there is currently no publicly available dynamic hand gesture recognition database that meets the experimental requirements. Thus, we must create a database that meets the needs of the experiment. The creation of the database is split into the following steps.

1) Design of HRI dynamic hand gestures: Because the dynamic hand gestures in this paper are used for HRI, a set of HRI hand gesture datasets containing 10 types of dynamic hand gestures is designed, as shown in Table 1.

TABLE I
HRI DYNAMIC HAND GESTURE DATASET.

Gesture number	Dynamic gesture diagram	Semantic meaning
Hand gesture 1		Start
Hand gesture 2		Stop
Hand gesture 3		Turn left
Hand gesture 4		Turn right
Hand gesture 5		Upward
Hand gesture 6		Downward
Hand gesture 7		Turn up
Hand gesture 8		Turn down
Hand gesture 9		Grap
Hand gesture 10		Release

Among the gestures in Table 1, hand gestures 1-2 control the start and stop of robots, respectively. Hand gestures 3-6 control the space motion directions of robots. Hand gestures 7-8 are utilized to change the parameters of the robots, such as the speed of motion. Hand gestures 9-10 are used to control the gripping and releasing of objects by robot grippers. By using these 10 types of dynamic gestures, the general operation control of robots can be achieved.

2) Creating the database: Dynamic hand gesture data were gathered by a Kinect v2 sensor. There were 20 volunteers for the experiment, and each volunteer repeated each dynamic hand gesture 10 times. Each dynamic hand gesture operation began by moving the hand into the image frame. After completing the action, the hand was removed from the frame. All the original RGB and depth videos were captured. From data collection to fusion, the following steps were carried out.

- Align the acquired RGB image and depth image using the alignment method described above.
- The Faster-RCNN-MobileNet model proposed above is used to detect and segment hands in RGB images. To ensure the consistency of the image size of each frame in the segmented video, each image is cropped with a bounding box centred on the center of the hand to achieve a 200×200 pixel image during spatial segmentation.
- The detection of a hand in three continuous frames denotes the start of a dynamic hand gesture, and the lack of a hand in three continuous frames denotes the end of each dynamic hand gesture. Each segmented video contains only one dynamic hand gesture.
- The spatially and temporally divided RGB video is mapped to a depth video to obtain a video with a consistent segmented depth.
- The OpenPose method is used to perform 2D hand pose estimation for each frame of the RGB video, and the abovementioned method is used to obtain a 3D hand pose map for each frame.
- Data fusion is performed on the corresponding RGB, depth image and hand pose map by using the weighted sum method proposed above.

Through the above steps, a total of 2000 videos of RGB hand gestures, 2000 videos of depth hand gestures, 2000 videos with 3D hand pose maps, and 2000 fused videos can be obtained. Each of these videos is 200×200 pixels in size. These data are brought together to form a database for dynamic hand gesture recognition in this experiment. The data for subject 1 - subject 12 are used as training data, and the data for subject 13 and subject 20 are used as testing data. In addition, in order to avoid over-fitting caused by the small amount of training data and increase generalization ability, a left-and-right-mirror operation used to increase the amount of training data is also added. The corresponding information is presented in Table 2.

TABLE II
DYNAMIC HAND GESTURE DATABASE INFORMATION.

subset	Training data	Testing data
RGB videos	2400	800
depth videos	2400	800
3D hand pose videos	2400	800
fusion videos	2400	800

B. Network training process

For sensors, dynamic hand gesture data were collected by a Kinect v2 sensor. And for network training and testing hardware, a GeForce TITAN RTX (24G) graphics processing unit (GPU) was used. The above proposed 3DCNN + ConvLSTM deep network framework was trained and tested with the TensorFlow platform, and the operating system was Ubuntu16.04. Two experiments were carried out separately.

Experiment 1: The 3DCNN structure used in our network framework is selected as C3D, so the C3D model trained on UCF101 was used as pre-trained model in our training

stage. And other modules like ConvLSTM and 2DCNN are connected follow the C3D and trained from the primitive models. Using a batch method can increase the ease and speed of training. Therefore, we use a high learning rate, which requires few epochs. We first train the original deep network model based on the RGB training videos. The initial value of the learning rate is set to 0.1, and the learning rate decreases by 0.1 every 20,000 steps. This approach can not only ensure the rapid convergence of errors in the early stage, but also avoid excessive fluctuations in the later stage. The initial value of weight decay is set to 0.004, and the weight decay value is reduced to 0.00004 after 40,000 iterations. The batch size is set to 16. The total number of training steps is set to 80,000 steps. In this way, we can obtain a model trained on RGB dynamic hand gesture data. Then, the model is used as a pretrained model, and adjustment is performed based on the depth videos of dynamic hand gestures, the 3D hand pose videos and the fused videos. The training parameters remain the same as those above. In this way, we can obtain models used for hand gesture recognition based on depth videos, hand pose videos and fusion videos.

Experiment 2: To verify the superiority of the proposed 3DCNN + ConvLSTM framework to other method, we trained some state-of-the-art methods such as C3D [27], two-stream convolutional network model [12], I3D [28] and MTUT [29] with the training data from the combined dynamic hand gesture videos. Each parameter used in training was consistent with those in the above training process. That is, the initial value of the learning rate was set to 0.1, and the learning rate was decreased by 0.1 every 20,000 steps. This approach can not only ensure the rapid convergence of errors in the early stage, but also avoid excessive fluctuations in the later stage. The initial value of weight decay was set to 0.004, and the weight decay value was reduced to 0.00004 after 40,000 iterations. The batch size was set to 16. The total number of training steps was set to 80,000 steps. After training, two dynamic hand gesture recognition models based on C3D and a two-stream convolutional network were obtained.

C. Testing and analysis

The three deep network models for dynamic hand gesture recognition trained in Experiment 1 were tested based on the corresponding dynamic hand gesture database, and the test results are shown in Table 3 and Figure 9.

TABLE III
EXPERIMENTS ON THE DIFFERENT KINDS OF DATASETS.

Network model	Data category	Accuracy(%)
3DCNN+ConvLSTM	RGB only	88.3
3DCNN+ConvLSTM	Depth only	88.0
3DCNN+ConvLSTM	3D hand pose only	91.1
3DCNN+ConvLSTM	Fusion	**92.4**

Table 3 shows the dynamic hand gesture recognition test results obtained by using the 3DCNN+ConvLSTM framework for RGB, depth, 3D hand pose and fusion videos from the testing database. According to Table 3, the dynamic hand

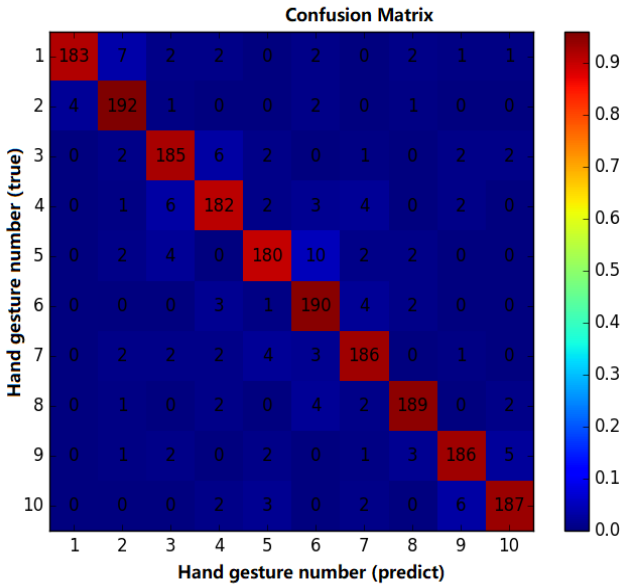

Fig. 9. Hand pose estimation framework.

gesture recognition accuracies for RGB data, depth data and3D hand pose data are 88.3%, 88.0%, and 91.1%, respectively. The dynamic hand gesture recognition accuracy for the combined data is 92.4%. This recognition accuracy is 4.1% higher than that of the RGB dynamic hand gesture recognition model, 4.4% higher than that of the depth dynamic hand gesture recognition model and 1.3% higher than that of the 3D hand pose recognition model. Thus, utilizing 3D hand poses can describe hand gesture features better than using RGB and depth features. The data obtained from combining RGB, depth and 3D hand pose information encompasses more hand gesture features than do individual data sets, and using this fused data can yield a higher hand gesture recognition accuracy than the individual methods. In addition, the hand gesture recognition confusion matrix for the combined testing data is shown in Figure 7. The recognition accuracy for each dynamic hand gesture is higher than 90%. This result verifies the effectiveness of the proposed method for HRI dynamic hand gesture recognition.

To study the import of each contribution of the proposed method, another comparisons of each part of the method are listed and displayed as follows.

From Table 4 we can see that when the fusion strategy is used, the accuracy of hand gestures recognition increased by 3.5% compared with that of RGB data using 3DCNN and increased by 4.1% compared with that of RGB data using our proposed network. It can prove that the proposed fusion strategy is useful to increase accuracy of gesture recognition because of including more useful gesture information. In addition, the accuracy of hand gesture recognition increased by 1.6% with our method compared with that of 3DCNN on RGB data and increased by 2.2% with our method compared with that of 3DCNN on fusion data. It can prove that the proposed network framework is helpful to increase accuracy of gesture recognition because of extracting more effective

gesture spatio-temporal features. In conclusion, each contribution of the proposed method is significant in recognition of dynamic hand gestures.

The dynamic hand gesture recognition model using fused data trained in Experiment 1 and the two models trained in Experiment 2 were tested with the combined dynamic hand gesture testing data respectively. The experimental results are shown in Table 5.

TABLE IV
ABLATION EXPERIMENT ON THE DYNAMIC HAND GESTURE DATABASE.

Method	Data category	Accuracy(%)
3DCNN	RGB	86.7
3DCNN	Fusion	90.2
3DCNN+ConvLSTM	RGB	88.3
3DCNN+ConvLSTM	Fusion	**92.4**

TABLE V
DYNAMIC HAND GESTURE RECOGNITION TEST RESULTS WITH DIFFERENT MODELS.

Network model	Data category	Accuracy(%)
3DCNN+ConvLSTM	Fusion	**92.4**
C3D [27]	Fusion	90.2
Two-Stream CNNs [12]	Fusion	88.8
I3D [28]	Fusion	91.3
MTUT [29]	Fusion	**92.4**

According to Table 5, in this test, the average recognition accuracy of dynamic hand gestures using the C3D, Two-Stream CNNs, I3D and MTUT models are 90.2%, 88.8%, 91.3% and 92.4%, respectively, and the recognition accuracy of dynamic hand gestures using the proposed 3DCNN + ConvLSTM model is 92.4%. This value is 2.2% higher than that the C3D, 4.0% higher than that of the two-stream CNN, 1.1% higher than that of the I3D and the same with that of the MTUT. Compared with other state-of-the-art dynamic hand gesture recognition methods, the proposed 3DCNN + ConvLSTM method can obtain better recognition results.

VI. CONCLUSION AND FUTURE WORK

This paper studies hand gesture recognition methods based on hand pose estimation. The innovations and contributions of this research are as shown follows. (1) A hand gesture recognition method based on hand pose estimation is proposed; it combines RGB and depth images with 3D hand pose maps to increase the dynamic hand gesture recognition accuracy. (2) A Faster-RCNN with BAs model is developed that utilises the advantages of the individual Faster-RCNN and BAs-ResNet to achieve the accurate and fast detection of tiny hand gestures in images. (3) A 3D hand pose estimation method using depth images is proposed. This method can accurately and quickly estimate 3D hand poses. (4) A set of dynamic hand gestures is designed for HRI, and a corresponding database is created. (5) A 3DCNN + ConvLSTM framework is proposed to effectively improve the recognition accuracy of dynamic hand gestures by at least 2.5% compared with that of other state-of-the-art methods. In the future, we will focus on developing more efficient data fusion methods utilizing hand pose features and more effective networks for dynamic hand gesture recognition. In addition, we will add on-line learning or incremental learning to our method for unknown hand gestures.

REFERENCES

[1] S. S. Rautaray, A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial intelligence review*., vol. 43, no. 1, pp. 1–54, 2015.
[2] O. K. Oyedotun, A. Khashman, "Deep learning in vision-based static hand gesture recognition" *Neural Computing and Applications*., vol. 28, no. 12, pp. 3941–3951, 2017.
[3] Q. Gao, J. Liu, Z. Ju, "Robust real-time hand detection and localization for space humanCrobot interaction based on deep learning" *Neurocomputing*., 2019.
[4] Q. Gao, J. Liu, Z. Ju, "Hand gesture recognition using multimodal data fusion and multiscale parallel convolutional neural network for humanCrobot interaction" *Expert Systems*., 2020.
[5] N. Neverova, C. Wolf, G. Taylor, "Moddrop: adaptive multi-modal gesture recognition" *IEEE Transactions on Pattern Analysis and Machine Intelligence*., vol. 38, no. 8, pp. 1692–1706, 2015.
[6] B. Fang, F. Sun, H. Liu, C. Liu, "3D human gesture capturing and recognition by the IMMU-based data glove" *Neurocomputing*., vol. 277, pp. 198–207, 2018.
[7] L. Zhang, G. Zhu, P. Shen, J. Song, "Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*., 2017, pp. 3120–3128.
[8] N. Neverova, C. Wolf, G. Paci, G. Sommavilla, "A multi-scale approach to gesture detection and recognition," *Proceedings of the IEEE International Conference on Computer Vision Workshops*., 2013, pp. 484–491.
[9] N. Neverova, C. Wolf, G. W. Taylor, F. Nebout, "Multi-scale deep learning for gesture detection and localization," *European Conference on Computer Vision (ECCV)*., 2014, pp. 474–490.
[10] P. Molchanov, X. Yang, S. Gupta, K. Kim, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*., 2016, pp. 4207–4215.
[11] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, "Multimodal gesture recognition based on the resc3d network," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*., 2017, pp. 3047–3055.
[12] K. Simonyan, A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in neural information processing systems*., 2014, pp. 568–576.
[13] Y. Zhu, Z. Lan, S. Newsam, A. Hauptmann, "Hidden two-stream convolutional networks for action recognition," *Asian Conference on Computer Vision*., 2018, pp. 363–378.
[14] N. Nishida, H. Nakayama, "Multimodal gesture recognition using multi-stream recurrent neural network," *Image and Video Technology*., 2015, pp. 682–694.
[15] O. Kopuklu, N. Kose, G. Rigoll, "Motion fused frames: Data level fusion strategy for hand gesture recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*., 2018, pp. 2103–2111.
[16] B. Ionescu, D. Coquin, P. Lambert, V. Buzuloiu, "Dynamic hand gesture recognition using the skeleton of the hand," *EURASIP Journal on Advances in Signal Processing*., vol. 2005, no. 13, pp. 190–236, 2005.
[17] Q. De. Smedt, H. Wannous, "Skeleton-based dynamic hand gesture recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*., 2016, pp. 1–9.
[18] J. Hou, G. Wang, X. Chen, J. H. Xue, "Spatial-temporal attention res-TCN for skeleton-based dynamic hand gesture recognition," *Proceedings of the European Conference on Computer Vision (ECCV)*., 2018, pp. 1–9.
[19] J. C. Nunez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognition*., vol. 76, pp. 80–94, 2018.
[20] F. Weichert, D. Bachmann, B. Rudak, D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*., vol. 13, no. 5, pp. 6380–6393, 2013.
[21] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," arXiv preprint arXiv:1812.08008, 2018.

[22] Z. Cao, T. Simon, S. E. Wei, "Spatial-temporal attention res-TCN for skeleton-based dynamic hand gesture recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*., 2017, pp. 7291–7299.
[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, "Ssd: Single shot multibox detector," *European conference on computer vision (ECCV)*., 2016, pp. 21–37.
[24] J. Redmon, S. Divvala, R. Girshick, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*., 2016, pp. 779–788.
[25] S. Ren, K. He, R. Girshick, J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*., 2015, pp. 91–99.
[26] S. Woo, J. Park, J. Y. Lee, et al, "Cbam: Convolutional block attention module," *PProceedings of the European conference on computer vision (ECCV)*., 2018, pp. 3–19.
[27] D. Tran, L. Bourdev, R. Fergus, L. Torresani, "Learning spatiotemporal features with 3d convolutional networks," *Proceedings of the IEEE international conference on computer vision (ICCV)*., 2015, pp. 4489–4497.
[28] J. Carreira, A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*., 2017, pp. 6299–6308.
[29] M. Abavisani, H. R. V. Joze, V. M.Patel, "Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*., 2019, pp. 1165–1174.
[30] Q. Huang, D. Yang, L. Jiang, et al, "A novel unsupervised adaptive learning method for long-term electromyography (EMG) pattern recognition," *Sensors*., vol. 17, no. 6, pp. 1370, 2017.
[31] M. Simao, N. Mendes, O. Gibaru, et al, "A review on electromyography decoding and pattern recognition for human-machine interaction," *IEEE Access*., vol. 7, pp. 39564-39582, 2019.
[32] P. Kaczmarek, T. Makowski, J. Tomczyski, "putEMGłA Surface Electromyography Hand Gesture Recognition Dataset," *Sensors*., vol. 19, no. 16, pp. 3548, 2019.

Zhaojie Ju (M'08–SM'16) received the B.S. degree in automatic control and the M.S. degree in intelligent robotics from the Huazhong University of Science and Technology, China, and the Ph.D. degree in intelligent robotics from the University of Portsmouth, U.K. He held research appointments at University College London, London, U.K., before he started his independent academic position at the University of Portsmouth, in 2012. He has authored or coauthored over 200 publications in journals, book chapters, and conference proceedings and received four Best Paper Awards and one Best AE Award in ICRA2018. His research interests include machine intelligence, pattern recognition and their applications on human motion analysis, multi-fingered robotic hand control, humanCrobot interaction and collaboration, and robot skill learning.

Dr. Ju is an Associate Editor of several journals, such as IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Neurocomputing.

Qing Gao (M'20) was born in Tangshan, China. He received his B.S. degree in automation from Electrical Engineering and Automation School, Liaoning Technology University, China in 2013. He received his Ph.D. degree in the State Key Laboratory of Robotics, Shenyang Institute of Automation (SIA), Chinese Academy of Sciences (CAS), Shenyang, China. Currently, he is working in Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), The Chinese University of Hong Kong, Shenzhen (CUHKSZ), Shenzhen, China. His research interests include robotics, artificial intelligence, machine vision and human-robot interaction.

Yi Liang was born in Zhaoqing, China. He received his B.Eng. degree in Electronic Engineering from University of Central Lancashire, the United Kingdom in 2019. Currently, he is working in Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen, China. His research interests include deep learning, computer vision, reinforcement learning and neural network acceleration circuit.

Yongquan Chen (M'19) received his B.S. degree in 2005, and M.S. degree in 2007, both in Electronics and Information Engineering, from Huazhong University of Science and Technology, and the Ph.D. degree from the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China in 2014. He is currently a researcher in Institute of Robotics and Intelligent Manufacturing, The Chinese University of Hong Kong, Shenzhen, and director of Research Center on Unmanned Systems (RCUS) of Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China. His current research interests include design, sensing and control of robotics system and multi-agent systems.