An Online Robot Teaching Method using Static Hand Gestures and Poses

Digang Sun, Ping Zhang, Mingxuan Chen, and Jiaxin Chen

Abstract—With an increasing number of robots are employed in manufacturing, a human-robot interaction method that can teach robots in a natural, accurate, and rapid manner is needed. In this paper, we propose a novel human-robot interface based on the combination of static hand gestures and hand poses. In our proposed interface, the pointing direction of the index finger and the orientation of the whole hand are extracted to indicate the moving direction and orientation of the robot in a fast-teaching mode. A set of hand gestures are designed according to their usage in humans' daily life and recognized to control the position and orientation of the robot in a fine-teaching mode. We employ the feature extraction ability of the hand pose estimation network via transfer learning and utilize attention mechanisms to improve the performance of the hand gesture recognition network. The inputs of hand pose estimation and hand gesture recognition networks are monocular RGB images, making our method independent of depth information input and applicable to more scenarios. In the regular shape reconstruction experiments on the UR3 robot, the mean error of the reconstructed shape is less than 1 mm, which demonstrates the effectiveness and efficiency of our method.

Index Terms—Deep learning, Hand gesture recognition, Hand pose estimation, Human-robot interaction, Robot teaching, Transfer learning.

I. INTRODUCTION

WITH the development of industrial production, an increasing number of robots are employed in manufacturing. Industrial robots, which face a great variety of tasks, put forward higher requirements on robot programming. Tasks such as assembling require precise position and orientation control of the robot at the start and end point of the moving trajectory, while in the task of welding, careful adjustment is necessary during the whole trajectory. In addition, some scenarios with a limited operation space also need fine-grained control for the position and orientation of the robotic End-Effector (EE). Generally, a human-robot interaction method that can teach robots in a natural, accurate,

and rapid manner is expected.

In the past decades, various human-robot interaction methods have been devised. Teaching pendants [1] [2] [3], which provide a commonly used traditional human-robot interface, is a real-time controller with the ability of accurate position and orientation control. Joystick-based methods are widely utilized in the teleoperation of remote or mobile robots [4], as well as other applications such as healthcare [5]. However, both teaching pendants and joystick-based methods are still in a machine-centered mode, and therefore not natural enough for humans to interact with the robot.

Speech is one of the most frequently utilized communication manners in humans' daily life. Speech recognition is widely employed to identify the operator [7] [10] and extract the emotions of human beings [8] [9] to communicate with social robots and service robots. Speech instructions are also recognized to program industrial robots [6]. However, the aforementioned methods either contain a limited number of speech instructions or have relatively low accuracy in speech recognition, which is not sufficient for fast and accurate control of both the position and orientation of the robotic EE.

Hand gestures and hand poses are extensively used in human-robot interaction. Generally, they can be divided into three main categories: hand gesture recognition, hand moving trajectory tracking, and hand pose estimation and mapping. Hand gestures are able to express discrete but deterministic intentions to indicate the robot to perform predefined tasks. Hand gestures can be utilized to control humanoid robots [15], service robots [11], unmanned aerial vehicles (UAVs) [14]. Some methods [12] [16] combine static and dynamic hand gestures together to indicate the robot. In addition, hand gestures are also used in human-robot co-operation scenarios [12] [13]. Hand moving trajectory tracking methods are mainly adopted to control a robot's position and orientation in a direct mapping manner [17] [18] [19]. Therefore, how to obtain the position and orientation of the hand exactly is very important. The position of the hand can be captured by sensors, such as Kinect, Leap Motion; the orientation can be collected using IMU and Leap Motion. However, because of the intrinsic noise of the sensors, the errors of the collected data can be accumulated over time. To alleviate this problem, some post-process methods, such as Kalman Filter (KF) and Particle Filter (PF) are needed. Hand poses are represented by 3D coordinates of the hand joints. Therefore, it can be used to directly map the pose from human hands to dexterous robot

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. This work was supported by the Science and Technology Planning Project of Guangdong Province (2019B090915002). (Corresponding author: Ping Zhang.)

The authors are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: cssundg@mail.scut.edu.cn; pzhang@scut.edu.cn; cscmx@mail.scut.edu.cn; csjiaxinchen@mail.scut.edu.cn).

hands [22] [23] to perform tasks, such as grasping. Moreover, the pointing direction of fingers and the orientation of the whole hand [23] can also be extracted, which are well corresponding to the moving direction and orientation of the robotic EE.

In this paper, we propose a novel human-robot interface that integrates static hand gestures and hand poses tightly to achieve a balance between naturalness, accuracy, and rapidness. In our proposal, the pointing direction of the index finger and the orientation of the whole hand are mapped to the moving direction and orientation of the robotic EE in a fast teaching manner. Static hand gestures, which indicate six basic moving directions (move- (forward, backward, left, right, up, and down)), three Euler angles (roll, pitch, and yaw) are also used to control the position and orientation of the robot in a fine-tuned fashion. Hand gestures and hand poses adopted are all static, therefore the problem of the hand moving out the effective sensing area of the sensor, which has to be dealt with in [21], can be avoided. In contrast to [19], which uses both hand gestures and speech to express intentions, only human hands are employed in our system, which means our system depends on fewer interaction manners and thus is simpler and more robust. Any wearable or hand-held devices, which might increase the burden of the operator, are not needed in our system. The inputs of the hand pose estimation and hand gesture recognition methods are all monocular RGB images; therefore, our proposed interface is independent of depth information input, which makes the interface applicable to more situations. The hand gestures also include necessary auxiliary operations, such as switching control mode, changing step distances and step angles, saving the position and orientation of the robotic EE; consequently, our method can teach the robot using hands without any interruption (e.g. using mouse or keyboard).

The main contributions of this article can be summarized as follows.

1) We systematically propose a novel human-robot interface that combines hand gestures with hand poses to implement a balance between naturalness, accuracy, and rapidness for interactions.

2) We build a training dataset that contains seventeen classes of static hand gestures that are designed according to their usage in humans' daily life; therefore, the burden of remembering and presenting the hand gestures can be reduced enormously.

3) We integrate transfer learning with attention mechanisms to improve the performance of the hand gesture recognition method.

4) We conducted a series of regular trajectory/shape reconstruction experiments to validate the effectiveness and efficiency of our proposed human-robot interface; the mean error of the reconstructed shape is less than 1 mm.

The remainder of this paper is organized as follows: Section II presents an overview of the proposed interface. Section III and IV describe hand pose estimation and hand gesture recognition methods, respectively. Section V introduces the realization of robot teaching. Section VI

Fig. 1. Configurations of the interface.

describes experiments and analyses, and finally, a conclusion is made in Section VII.

II. OVERVIEW

Fig. 1 shows the configuration of our presented human-robot interface. An RGB camera (we only use the monocular RGB sensor of the Intel RealSense Depth Camera D435i) is employed to capture color hand images, which will then be cropped to 256×256 pixels from the center and fed into deep neural networks to carry out hand pose estimation and hand gesture recognition. When teaching robots, the operator is supposed to stand in front of the camera and stretch their hands forward to present various static hand gestures and hand poses.

Seventeen static hand gestures (see Fig. 2) were elaborately designed according to their usage in the daily life of human beings, to increase naturalness and decrease the burden of remembering and presenting them during interactions. Generally, the hand gestures can be divided into three categories according to their functions (see Table I): (1) position or moving direction control gestures, (2) orientation control gestures, (3) auxiliary gestures, such as switching teaching modes, selecting step distances, and selecting step angles. In addition, we define a null gesture to represent that no effective hand gestures were recognized, and thus no effective operations will be performed. On the other hand, the pointing direction of the index finger and the orientation of the whole hand are used to indicate the moving direction and the orientation of the robotic EE, respectively.

To integrate hand gestures and hand poses effectively, four teaching modes are explicitly defined within our system: (1) basic moving direction control mode, (2) arbitrary moving direction mapping mode, (3) basic orientation angle (i.e. Euler angle) control mode, and (4) arbitrary orientation mapping mode.

In order to achieve a balance between rapidness and accuracy, the whole teaching process is implicitly divided into rapid teaching stage and accurate teaching stage. When the robotic EE is relatively far away from the target, the system is in the rapid teaching stage; the position of the EE can be controlled by indicating either six basic moving directions or an arbitrary direction represented by the index finger, coupled with a large step distance; the orientation of the EE can be controlled by either mapping the orientation of the whole hand or adjusting the Euler angles with a large step angle. When the

Fig. 2. Static hand gestures used in our proposed interface.

TABLE I
HAND GESTURES FOR INTERACTION

No.	Gesture (Code)	Meaning
0	Null (N)	Idle operation
1	Forward (F)	Move forward
2	Backward (B)	Move backward
3	Up (U)	Move up / Pitch up
4	Down (D)	Move down / Pitch down
5	Left (L)	Move left / Yaw left
6	Right (R)	Move right / Yaw right
7	Index finger (I)	Move according to the finger's pointing direction
8	Increase (Inc)	Increase the step distance or angle for one instruction
9	Decrease (Dec)	Decrease the step distance or angle for one instruction
10	Grasp (Gr)	Grasp an object
11	Release (Re)	Release an object
12	Rotate clockwise (RC)	Rotate clockwise
13	Rotate anticlockwise (RA)	Rotate anticlockwise
14	Pause (P)	Pause
15	Switch (S)	Switch teaching mode
16	O.K. (O)	Save current position and orientation
17	Double O.K. (DO)	Save line type (i.e. straight line or arc)

robotic EE is near the destination, the system is in the accurate teaching stage; the position of the EE can be controlled by indicating the six basic moving directions, coupled with a small step distance; the orientation can be controlled by adjusting the Euler angles with a small step angle.

III. HAND POSE ESTIMATION AND MAPPING

A. Hand Coordinate System

From the overview of the system, an RGB camera is adopted to capture human hand images; operators use static hand gestures and hand poses to express interaction intentions to guide the robotic EE to approach the target with expected orientation. To implement both position and orientation teaching, the registration between coordinate systems attached to different parts is necessary. We define the coordinate system of the camera as $X_C Y_C Z_C$, the hand as $X_H Y_H Z_H$, the robotic base

Fig. 3. Overview of the deep neural network for hand gesture recognition

Fig. 4. Hand pose estimation. (a) For index finger recognition: recognized correctly and drawn in green (Left); recognized as small finger and drawn in magenta (Middle); recognized as thumb and drawn in blue (Right). (b) For orientation extraction of the whole hand.

as $X_R Y_R Z_R$, the workpiece as $X_W Y_W Z_W$ which coincides with that of the world, and the robotic EE as $X_E Y_E Z_E$.

B. Hand Pose Estimation

Hand pose estimation is used to regress the 3D coordinates of hand joints (we adopt the 21-joint hand pose model, four joints for each finger plus the wrist joint, see Fig.4 (b)) and then extract the pointing direction of the index finger and the orientation of the whole hand, which will be further utilized to indicate the moving direction and the orientation of the robotic EE. Hand pose estimation is challenging because of the self-similarity, self-occlusion of the hand. Traditionally, hand pose estimation methods based on deep neural networks use 3D pose-labeled dataset, which is not sufficient for accurate and robust estimation. To alleviate the problem, we build our hand pose estimation network upon [24] that aims originally at recovering 3D hand shapes that 3D hand pose can be obtained from.

The deep neural network for hand pose estimation comprises a Stacked Hourglass Network [25] used to extract 2D coordinates of the hand joints, a Residual Network employed to generate a latent feature vector, a Graph CNN [26] utilized to recover hand mesh, a 3D Pose Regressor to regress 3D hand pose from hand mesh, and a Mesh Renderer [27] to render the estimated 3D hand mesh to a depth map from the camera viewpoint.

In [24], the neural network was first trained on a synthetic dataset, which contains a large number of images of high quality. Thereafter, a real dataset, the Stereo Hand Pose (SHP) [28], was employed to fine-tune the network to bridge the gap between synthetic data and real data. Readers are referred to [24] for more details.

Because of self-similarity and self-occlusion, the index finger might be recognized as the small finger or thumb (see Fig. 4 (a)). In order to alleviate this situation, a variable that can measure the bending degree of the finger is introduced, which is defined as

$$b = \frac{d_{12} + d_{23} + d_{34}}{d_{14}}, \quad (1)$$

where d_{ij} represents the distance between the joint i and j of one finger (the joints of a finger from the tip to the MCP are

denoted as 1 to 4, successively). Therefore, the finger which is of the smallest bending degree can be considered as the index finger, which stretches straight.

C. Direction and Orientation Mapping

In our proposal, the pointing direction of the index finger and the orientation of the whole hand, which are obtained through hand post estimation, will be mapped to the moving direction and orientation of the robotic EE.

The 21-joint hand model, in which five fingertips are also viewed as joints, is adopted. As shown in Fig. 5 (a) the pointing direction of the index finger in the camera coordinate system is represented by a vector $\vec{AB}$; Point A represents the MCP (root) of the finger, and Point B the tip. We denote the coordinates of point A as $P_A = (x_A, y_A, z_A)$, B as $P_B = (x_B, y_B, z_B)$; a direction vector parallel with $\vec{AB}$ can be denoted as

$$\vec{s} = \frac{\vec{AB}}{|\vec{AB}|}. \quad (2)$$

Fig. 4 (b) shows that a coordinate system is attached to the hand, which represents the orientation of the whole hand in the camera coordinate system. Specifically, the MCP of the little finger is denoted as A, the tip of the middle finger as B, the MCP of the index finger as C, drawing a vertical through B to line AC crossed at O. Based on the three points A, B, and C, a coordinate frame $X_H Y_H Z_H$ is defined according to the right-hand rule. In order to implement orientation mapping, we define the orientation of the hand as the default one when the three principal axes of coordinate system $X_H Y_H Z_H$ parallel with that of coordinate system $X_C Y_C Z_C$. Similarly, the orientation of the robotic EE is defined as the default orientation when the three principal axes of coordinate system $X_E Y_E Z_E$ parallel with that of coordinate system $X_R Y_R Z_R$. When the operator presents an orientation by the hand, the three Euler angles (i.e. yaw, pitch, and roll) can be calculated. The roll angle α can be obtained by projecting Z_H axis onto plane $X_C O Z_C$, getting vector Z'_H , and measuring the angle between Z_C and Z'_H . Similarly, the pitch angle β can be obtained by projecting Z_H axis onto plane $Y_C O Z_C$, getting vector Z''_H , and measuring the angle between Z_C and Z''_H ; the yaw angle γ can be obtained by projecting Y_H axis onto plane $X_C O Y_C$, getting vector Y'_H , and measuring the angle between Y_C and Y'_H . The aforementioned calculation can formally be expressed as

$$\begin{cases} \text{roll } (\alpha) = \langle Z'_H, Z_C \rangle \\ \text{pitch } (\beta) = \langle Z''_H, Z_C \rangle \\ \text{yaw } (\gamma) = \langle Y'_H, Y_C \rangle \end{cases} \quad (3)$$

where

$$\begin{cases} Z'_H = \text{proj}_{X_C O Z_C}(Z_H) \\ Z''_H = \text{proj}_{Y_C O Z_C}(Z_H) \\ Y'_H = \text{proj}_{X_C O Y_C}(Y_H) \end{cases} \quad (4)$$

After that, the difference between the current hand orientation and the default one can be obtained. Correspondingly, the orientation of the robotic EE can be adjusted according to the calculated orientation difference of the hand, and thus the orientation mapping from the human hand to the robotic EE can

Fig. 5. Direction & orientation mapping. (a) Pointing direction of the index finger. (b) Orientation of the whole hand.

be implemented. Orientation mapping can formally be described as

$$P_c^R = P_d^R + (P_c^H - P_d^H), \quad (5)$$

where superscript R and H represent the robot and hand, respectively; subscript c and d indicate current and default orientations.

IV. HAND GESTURE RECOGNITION

A. Custom Data Set

In order to improve the naturalness of interaction, we designed seventeen classes of static hand gestures according to their usage in humans' daily life. Therefore, a custom hand gesture dataset is necessary for the hand gesture recognition method based on deep neural networks (described below). We built a training dataset and a test dataset. The training dataset contains more than 18,000 hand images, while the test dataset includes more than 4,000 images. The images of the test dataset were presented by a subject, whose hand images did not occur in the training dataset, and the image background is also different from that of the training dataset. Consequently, the performance of the hand gesture recognition network can be measured exactly and robustly using the test dataset.

B. Hand Gesture Recognition

Hand pose estimation, which is able to extract 3D coordinates of hand joints, has the effect of eliminating the background of hand images. In particular, the stacked hourglass network in the hand pose estimation network is capable of extracting features in a multi-scale and fusing them in a multi-stage manner. Therefore, we employ transfer learning methods to make good use of the capacity of the stacked hourglass network to improve the accuracy and robustness of the hand gesture recognition module.

Network Structure: Fig. 2 demonstrates the deep neural network for hand gesture recognition, which comprises a stacked hourglass network and a residual network with attention mechanisms (described below). The stacked hourglass network is of the same structure and weights as its counterpart in the hand pose estimation network. A hand image is first fed into the stacked hourglass network to generate feature maps, which will then be processed by the residual attention network. The end of the hand gesture recognition network is a classifier that consists of two fully connected layers.

Label Smoothing: The last layer of an image classifier is typically a fully-connected layer, which, for each input image x , will output the probability of each label $k \in \{1 \dots K\}$:

$$q(k|x) = \frac{\exp(z_k)}{\sum_{i=1}^K \exp(z_i)} \quad (6)$$

where K is the number of labels. z_i are the logits or unnormalized log probabilities. The loss between predictions and labels can be computed as cross-entropy

$$\text{loss}(p, q) = - \sum_{i=1}^K p_i \log q_i \quad (7)$$

where

$$p_i = \begin{cases} 1, & \text{if } (y = i) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

y is the true label of x . To prevent the model from over-fitting, we adopt label smoothing [30] [31] to reduce the differences between the largest logit and all others so that the confidences of their predictions are more aligned with the accuracies of their predictions. The label smoothing method introduces a small constant to change the probability distribution from (8) to

$$p_i = \begin{cases} 1 - \varepsilon, & \text{if } (y = i) \\ \frac{\varepsilon}{K-1}, & \text{otherwise} \end{cases} \quad (9)$$

where ε is the smoothing factor.

Data Augmentation: Data augmentation techniques are helpful in improving the generalization of the deep neural network. Our data augmentation is two-fold: (1) We adopt conventional data augmentation approaches, such as randomly changing the brightness, contrast, saturation of an image, randomly rotating an image by an angle. (2) We employ another method called mixup [29]. We select two examples (x_i, y_i) and (x_j, y_j) from the training dataset at random, and then construct a virtual training example $(\hat{x}, \hat{y})$ by a weighted linear interpolation of these two examples:

$$\hat{x} = \lambda x_i + (1 - \lambda) x_j, \quad (10)$$

$$\hat{y} = \lambda y_i + (1 - \lambda) y_j, \quad (11)$$

where $\lambda \in [0, 1]$ is a number randomly drawn from the *Beta* (α, α) distribution. In training, we only use the generated virtual examples. After the application of label smoothing and mixup data augmentation, the loss function for hand gesture classification will be

$$\text{loss} = \lambda \text{loss}(p, q^{(1)}) + (1 - \lambda) \text{loss}(p, q^{(2)}), \quad (12)$$

where $q^{(1)}$ and $q^{(2)}$ denote the probability distribution of the two batches of selected examples belonging to each label, respectively.

Attention Mechanisms: Attention mechanisms play an important role in human perception, and many studies [32][33][34] have embedded attention mechanisms in convolutional neural networks to improve performance. We integrate the convolutional block attention module (CBAM) [33] into our residual network. CBAM decomposes the attention mechanism into channel attention and spatial attention. The channel attention is computed as

$$M_c(F) = \sigma(F_a + F_b), \quad (13)$$

$$F_a = \text{MLP}(\text{AvgPool}(F)), \quad (14)$$

$$F_b = \text{MLP}(\text{MaxPool}(F)), \quad (15)$$

where σ indicates the sigmoid function; *AvgPool* and *MaxPool* denote average pooling and max pooling operations, respectively; *MLP* indicates a shared multi-layer perceptron with one hidden layer. The spatial attention is computed as

$$M_s(F) = \sigma(f^{7 \times 7}([\text{AvgPool}(F); \text{MaxPool}(F)])) \quad (16)$$

where σ , *AvgPool*, and *MaxPool* are the same as those in (18); $f^{7 \times 7}$ indicates a convolutional operation with a filter size of 7×7 . Note that feature maps will be concatenated after two types of pooling operations.

Histogram Equalizing: Illumination has negative influences on the performance of the hand gesture classifier. Image histogram equalizing, a traditional image processing method, is employed to relieve this influence. We exercise image histogram equalizing on all images of the training dataset.

We implement our hand gesture classification method within the PyTorch framework. The networks are trained using SGD optimizer with a momentum of 0.9 and weight decay of 0.0001. The weights of the stacked hourglass sub-network are initialized with the weights from [24], and then the whole recognition network is trained with our custom training dataset. The learning rate is set to 0.001 and divided by 10 after every 10 epochs. We train the network for 30 epochs and use the learning rate warmup strategy [35] in the first two epochs to make the training process more stable. We test our network on the test dataset; the average classification accuracy is above 98%.

V. REALIZATION OF ROBOT TEACHING

This section mainly introduces how to implement robot teaching using both hand gestures and hand poses. First of all, in our proposal, the teaching trajectory is separate from the trajectory for the robot to perform actual tasks, which is different from the hand tracking-based method [19]. We teach the robot a trajectory by teaching the position and orientation of the key points of the trajectory and linking them with straight-line segments or arcs. Generally, the control of the robotic EE can be divided into position control and orientation control.

Position control can be carried out by giving a moving direction and a distance. The direction can be indicated by either the six basic spatial directions or the pointing direction of the index finger. For instance, if the current position of the robotic EE is (x_0, y_0, z_0) ; the directional vector of the given direction is $s = (\cos \alpha, \cos \beta, \cos \gamma)$, and the provided step distance is d ; then the coordinates of the target will be (x_t, y_t, z_t) , where

$$\begin{cases} x_t = x_0 + d \cdot \cos \alpha \\ y_t = y_0 + d \cdot \cos \beta \\ z_t = z_0 + d \cdot \cos \gamma \end{cases} \quad (17)$$

When the direction is given by one of the six basic directions, only one of the elements of the directional vector is nonzero. The step distances can be given by a list of various distances, such as 50 cm, 20 cm, ..., 2 mm, 1 mm. We can select a specific

step distance by hand gesture according to the requirement of speed and accuracy.

To make the direction control more intuitive for the operator, the robot should move in a direction that coincides with the direction presented by the operator from their egocentric view. We attach a coordinate system $\{H\}$ to the right hand of the operator (see Fig. 1), and let the X axis point from left to right, Y axis point from back to front, and Z axis point from foot to head. We denote the unit vectors giving the principal directions of coordinate system $\{H\}$ as X_H , Y_H , and Z_H and rewrite them, in terms of the camera coordinate system $\{C\}$, as ${}^C X_H$, ${}^C Y_H$, and ${}^C Z_H$. Therefore, the rotation matrix between coordinate system $\{H\}$ and $\{C\}$ can be described as

$${}^C_H R = \begin{bmatrix} {}^C X_H & {}^C Y_H & {}^C Z_H \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (18)$$

Orientation control can be achieved either by orientation mapping or by the fine-tuning of the three Euler angles. When using orientation mapping, the target Euler angles can be calculated as

$$\begin{cases} \alpha_t^T = \alpha_d^T + \alpha_c^H - \alpha_d^H \\ \beta_t^T = \beta_d^T + \beta_c^H - \beta_d^H \\ \gamma_t^T = \gamma_d^T + \gamma_c^H - \gamma_d^H \end{cases} \quad (19)$$

where α , β , and γ represent roll, pitch, and yaw angles, respectively; the subscript t , d , and c indicate target, default, and current angles, respectively; the superscript T , H denotes the TCP and hand coordinate system, respectively. When using orientation fine-tuning, the target Euler angles of the TCP will be calculated as

$$\begin{cases} \alpha_t = \alpha_o + \alpha_s \\ \beta_t = \beta_o + \beta_s \\ \gamma_t = \gamma_o + \gamma_s \end{cases} \quad (20)$$

where the subscript t , o , and s indicate the target angle, old angle, and step-angle, respectively. In addition, the step distance and step angle can also be adjusted by hand gestures so as to realize accuracy as well as speed control.

Specifically, to teach the robot a regular shape through determining and linking the key points of the shape, the operator should: (1) determine where the key points are; (2) guide the robotic EE to approach the key points along with an expected orientation, one by one; (3) determine the line type by the rule that two points decide a straight line segment, while three points decide an arc; (4) save both the position and orientation of all key points, and the corresponding line types; (5) generate control instructions for the robot.

VI. EXPERIMENTS AND ANALYSES

A. Experiments

To evaluate our proposed human-robot interaction system, a series of elaborately devised experiments are carried out. An RGB camera is employed to capture hand images. A UR3 robot with six Degree-of-Freedom (DOF) is used for the experiments. A gel pen is attached to the robot EE; the distance between the tip of the open and the target is measured, which will then be used to determine whether the location accuracy is acceptable.

Fig. 6. Regular shape to be reconstructed. (a) Placed without a slope. (b) Placed with a slope.

To assess the effectiveness and efficiency of both the position and orientation control of our proposed method, we carried out regular trajectory/shape reconstruction experiments. A shape that consists of two straight line segments and two arcs is introduced as the reference; each straight-line segment is 140 mm in length, and each arc is 70 mm in radius. There are six key points of the shape, four of which connect an arc to a straight-line segment, and two others are the intermediate points of the arcs. To perform the task accurately and rapidly, operators are supposed to use almost all of the predefined hand gestures to control both the position and orientation of the robotic EE in both rapid teaching and accurate teaching manner, and thus the feasibility and usability of the hand gestures will also be confirmed. To effectively validate both the position and orientation control of the robot, we increase the difficulty of the experiment by placing the workpiece onto a flat surface with a slope of about 15 degrees (see Fig. 6 (b)).

Three volunteers, who are students at the age between 22 and 28 and haven't been exposed to the knowledge about robot teaching, are invited to carry out the experiments. Before doing the experiment, they are taught which hand gestures will be used, and how to use them appropriately. Although there are more than ten kinds of hand gestures that will be used, it takes not much time for the operator to remember them since they are designed according to the usage in the human's daily life.

B. Analyses

Fig. 7 (a) displays the teaching trajectory and the reconstructed shape of the experiment in 3D. The blue line segments are the teaching trajectory formed by using basic direction control. The lengths of the line segments are in accordance with the step distance of the moving instructions. The red diamonds indicate the key points of the trajectory/shape, through which the robot will be able to perform actual tasks. The green shape, which consists of two straight line segments and two arcs, represents the reconstructed one. Fig. 7 (b) shows the reconstructed shape in 2D and the details near the six key points. The inner solid line indicates the reference shape to be reconstructed; the outer dashed line represents the shape reconstructed. The gaps between the reference shape and the reconstructed shape near the six key points (from left to right, top to bottom) are 0.12 mm, 0.25 mm, 0.12 mm, 0.16 mm, 0.12 mm, and 0.52 mm, respectively, and the mean value of these six gaps is 0.22 mm. We use the values of these gaps to represent reconstruction errors.

We first compared our method with other natural teaching methods such as hand motion-based method [19] in terms of the reconstruction error. It can rationally be inferred from [19] that the reconstruction error of the trajectory/shape formed by hand motion-based teaching methods might depend scarcely on the

Fig. 7. (a) The moving trajectory of the robot and reconstructed shape. (b) Reconstructed 2D shape and details near six key points. The inner solid line represents the shape to be reconstructed; the outer dashed line represents the shape reconstructed; two parallel blue dashdot lines connect the end points of the two arcs, respectively, and another dashdot line connects the centers of the two arcs.

TABLE II
ERRORS OF TRAJECTORY RECONSTRUCTION

	T	Min. (mm)	Mean (mm)	Max. (mm)
Method [19]	I	1.27	4.51	12.74
Method [36]	I	1.28	3.64	9.9
Our Method	R	0.12	0.22	0.52

T: Trajectory type, I: Irregular, and R: Regular.

trajectory/shape itself. Therefore, it is reasonable to believe that the error of reconstructing a regular trajectory/shape approximates that of an irregular one when using hand motion-based methods. Secondly, we compared our method with the teaching pendant in terms of the time spent in obtaining the position and orientation of the key points of the shape. We compared the operational time of the three volunteers in reconstructing a regular shape placed with and without a slope. It is unbiased to consider that the accuracy of our method is the same as that of the teaching pendant because when the error is less than 1 mm, it is difficult for the operator to tell the difference using the naked eye.

The errors of shape reconstruction are shown in Table II. The data in the first and second rows are from [19], and we calculate the average error of the two tests for the sake of comparison. The mean reconstruction errors of an irregular trajectory using method [19] and [36] are 4.51 and 3.64 mm, respectively, which is much larger than that of our method i.e. 0.56 mm. To the extreme, the minimum errors of method [19] and [36] i.e. 1.27 mm and 1.28 mm are still larger than the maximum error of our method i.e. 0.52 mm. The two arcs of the shape to be reconstructed are of the same radius of 70 mm; the radii of the reconstructed arcs are measured 70.12 mm and 70.15 mm, respectively. Therefore, it can be concluded that our method is more accurate than [19] and [36].

For the sake of simplicity, we only measure the time spent in obtaining the position and orientation of the key points. Although the teaching pendant is a type of device that lacks

TABLE III
OPERATIONAL TIME (S) OF REGULAR SHAPE RECONSTRUCTION

	S	V1	V2	V3
Our Method	W/	584	589	591
	W/O	545	550	557
Teaching pendant	W/	462	474	471
	W/O	427	431	445

S: Slope, V1~V3: Three volunteers, W/: With a slope, and W/O: Without a slope.

naturalness, it can send commands to the robot very quickly through pressing buttons. On the other hand, our method can use bigger step distances and step angles when the distance between the robotic EE and target is relatively large, which can, to some degree, reduce the number of the required instructions and the operational time. The operational times of the three volunteers reconstructing the regular shape with similar accuracy are shown in Table III. It can be seen that for all volunteers the operational times of our method are larger than that of the teaching pendant no matter the shape is placed with or without a slope. However, to program a robot completely, saving the position and orientation of the key points and determining the line type (straight-line segment or arc) between key points is necessary; in this respect, our method is more natural and efficient than the teaching pendant. Therefore, our method is, on the whole, comparable to the teaching pendant in terms of the operational time. It can also be observed that the operational times for the shape placed with a slope are larger than that for the shape placed without a slope when using the same method, due to the difficulty caused by the slope.

VII. CONCLUSION

This paper presented a human-robot interface based on the integration of hand pose estimation and hand gesture recognition to achieve a good balance between naturalness, accuracy, and rapidness. A set of static hand gestures are

designed according to their usage in humans' daily life so that the burden of remembering and presenting them can be reduced as much as possible, and transfer learning and attention mechanisms are employed to improve the recognition accuracy. The teaching process is implicitly divided into rapid teaching stage and accurate teaching stage to implement the balance between accuracy and rapidness. Only RGB hand images are needed in the interface, which makes it applicable to more scenarios. Qualitative analyses and quantitative experimental results have verified the effectiveness and efficiency of our proposed method.

The proposed interface can also be extended to guide mobile robots as well as unmanned aerial vehicles (UAVs) owing to the capacities for both position and orientation control.

REFERENCES

- [1] M. Oh, S. Lee, T. Kim, K. Lee, and J. Kim. "Design of a teaching pendant program for a mobile shipbuilding welding robot using a PDA." *Computer-Aided Design* 42, no. 3 (2010): 173-182.
- [2] M. Kaluarachchi, and F. Annaz. "Gui teaching pendant development for a 6 axis articulated robot." In *International Conference on Intelligent Robotics, Automation, and Manufacturing*, pp. 111-118. Springer, Berlin, Heidelberg, 2012.
- [3] M. Gao, L. Ye, and Y. Yan. "Design and implementation of teach pendant for six degrees of freedom industrial robot." In *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, pp. 1929-1933. IEEE, 2015..
- [4] S. Cho, H. Jin, J. Lee, and B. Yao. "Teleoperation of a mobile robot using a force-reflection joystick with sensing mechanism of rotating magnetic field." *IEEE/ASME Transactions On Mechatronics* 15, no. 1 (2009): 17-26.
- [5] H. Kim, Y. Tanaka, A. Kawamura, S. Kawamura, and Y. Nishioka. "Development of an inflatable robotic arm system controlled by a joystick." In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 664-669. IEEE, 2015.
- [6] M. Bingol, and O. Aydogmus. "Performing predefined tasks using the human-robot interaction on speech recognition for an industrial robot." *Engineering Applications of Artificial Intelligence* 95 (2020): 103903.
- [7] L. Chen, W. Su, Y. Feng, M. Wu, J. She, and K. Hirota. "Two-layer fuzzy multiple random forest for speech emotion recognition in human-robot interaction." *Information Sciences* 509 (2020): 150-163.
- [8] E. Lakomkin, M. Zamani, C. Weber, S. Magg, and S. Wermter. "On the robustness of speech emotion recognition for human-robot interaction with deep neural networks." In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 854-860. IEEE, 2018..
- [9] L. Devillers, M. Tahon, M. Sehili, and A. Delaborde. "Inference of human beings' emotional states from speech in human-robot interactions." *International Journal of Social Robotics* 7, no. 4 (2015): 451-463.
- [10] J. Ding, and J. Shi. "Kinect microphone array-based speech and speaker recognition for the exhibition control of humanoid robots." *Computers & Electrical Engineering* 62 (2017): 719-729.
- [11] D. Xu, X. Wu, Y. Chen, and Y. Xu. "Online dynamic gesture recognition for human robot interaction." *Journal of Intelligent & Robotic Systems* 77, no. 3 (2015): 583-596.
- [12] P. Neto, M. Simão, N. Mendes, and M. Safeca. "Gesture-based human-robot interaction for human assistance in manufacturing." *The International Journal of Advanced Manufacturing Technology* 101, no. 1 (2019): 119-135.
- [13] E. Coupeté, F. Moutarde, and S. Manitsaris. "Multi-users online recognition of technical gestures for natural human-robot collaboration in manufacturing." *Autonomous Robots* 43, no. 6 (2019): 1309-1325.
- [14] B. Chen, C. Hua, B. Dai, Y. He, and J. Han. "Online control programming algorithm for human-robot interaction system with a novel real-time human gesture recognition method." *International Journal of Advanced Robotic Systems* 16, no. 4 (2019): 1729881419861764.
- [15] H. Ai, K. Tang, L. Han, Y. Wang, and S. Zhang. "DuG: Dual speaker-based acoustic gesture recognition for humanoid robot control." *Information Sciences* 504 (2019): 84-94.
- [16] X. Li. "Human-robot interaction based on gesture and movement recognition." *Signal Processing: Image Communication* 81 (2020): 115686.
- [17] G. Du, and P. Zhang. "A novel human-manipulators interface using hybrid sensors with Kalman filter and particle filter." *Robotics and Computer-Integrated Manufacturing* 38 (2016): 93-101.
- [18] G. Du, P. Zhang, and X. Liu. "Markerless human-manipulator interface using leap motion with interval Kalman filter and improved particle filter." *IEEE Transactions on Industrial Informatics* 12, no. 2 (2016): 694-704.
- [19] G. Du, M. Chen, C. Liu, B. Zhang, and P. Zhang. "Online robot teaching with natural human-robot interaction." *IEEE Transactions on Industrial Electronics* 65, no. 12 (2018): 9571-9581.
- [20] F. Gomez-Donoso, S. Orts-Escolano, and M. Cazorla. "Accurate and efficient 3D hand pose regression for robot hand teleoperation using a monocular RGB camera." *Expert Systems with Applications* 136 (2019): 327-337.
- [21] G. Du, G. Yao, C. Li, and X. Liu. "Natural human-robot interface using adaptive tracking system with the unscented Kalman filter." *IEEE Transactions on Human-Machine Systems* 50, no. 1 (2019): 42-54.
- [22] S. Li *et al.*, "Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network." In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 416-422. IEEE, 2019.
- [23] F. Gomez-Donoso, S. Orts-Escolano, and M. Cazorla. "Accurate and efficient 3D hand pose regression for robot hand teleoperation using a monocular RGB camera." *Expert Systems with Applications* 136 (2019): 327-337.
- [24] L. Ge *et al.*, "3d hand shape and pose estimation from a single rgb image." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10833-10842. 2019.
- [25] A. Newell, K. Yang, and J. Deng. "Stacked hourglass networks for human pose estimation." In *European conference on computer vision*, pp. 483-499. Springer, Cham, 2016.
- [26] M. Defferrard, X. Bresson, and P. Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering." *arXiv preprint arXiv:1606.09375* (2016).
- [27] H. Kato, Y. Ushiku, and T. Harada. "Neural 3d mesh renderer." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3907-3916. 2018.
- [28] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang. "3d hand pose tracking and estimation using stereo matching." *arXiv preprint arXiv:1610.07214* (2016).
- [29] H. Zhang, M. Cisse, Y. Dauphin, and D. Lopez-Paz. "mixup: Beyond empirical risk minimization." *arXiv preprint arXiv:1710.09412* (2017).
- [30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.
- [31] R. Müller, S. Kornblith, and G. Hinton. "When does label smoothing help?." *arXiv preprint arXiv:1906.02629* (2019).
- [32] F. Wang *et al.*, "Residual attention network for image classification." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156-3164. 2017.
- [33] S. Woo, J. Park, J. Lee, and I. Kweon. "Cbam: Convolutional block attention module." In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3-19. 2018.
- [34] J. Park, S. Woo, J. Lee, and I. Kweon. "Bam: Bottleneck attention module." *arXiv preprint arXiv:1807.06514* (2018).
- [35] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li. "Bag of tricks for image classification with convolutional neural networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 558-567. 2019.
- [36] H. Lee, J. Kim, and T. Kim. "A robot teaching framework for a redundant dual arm manipulator with teleoperation from exoskeleton motion data." In *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 1057-1062. IEEE, 2014.

Digang Sun received the M.S. degree in computer science from the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, in 2018, where he is currently working toward the Ph.D. degree in computer science.

His research interests include machine learning, computer vision, and human-computer interaction.

Ping Zhang received the B.S. degree in mechanical engineering and the M.S. and Ph.D. degrees in robotics from Tianjin University, Tianjin, China, in 1985, 1988, and 1994, respectively.

He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include intelligent networked robotics, intelligent networked manufacturing, human-computer interaction, and real-time embedded systems.

Mingxuan Chen received the B.S. degree in computer science from Hunan University, Changsha, China, in 2015. He is currently working toward the Ph.D. degree in computer science and technology at the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His research interests include human-swarm interaction and swarm robotics.

Jiaxin Chen is currently working toward the M.S. degree in computer science at the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His research interests include machine vision and human-computer interaction.