Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8871605, 14 pages
https://doi.org/10.1155/2020/8871605

Research Article

A Fusion Recognition Method Based on Multifeature Hidden Markov Model for Dynamic Hand Gesture

Guoliang Chen and Kaikai Ge

School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, Hubei, China

Correspondence should be addressed to Guoliang Chen; glchen@whut.edu.cn

Received 29 April 2020; Revised 11 August 2020; Accepted 28 August 2020; Published 9 September 2020

Academic Editor: Daniele Bibbo

In this paper, a fusion method based on multiple features and hidden Markov model (HMM) is proposed for recognizing dynamic hand gestures corresponding to an operator's instructions in robot teleoperation. In the first place, a valid dynamic hand gesture from continuously obtained data according to the velocity of the moving hand needs to be separated. Secondly, a feature set is introduced for dynamic hand gesture expression, which includes four sorts of features: palm posture, bending angle, the opening angle of the fingers, and gesture trajectory. Finally, HMM classifiers based on these features are built, and a weighted calculation model fusing the probabilities of four sorts of features is presented. The proposed method is evaluated by recognizing dynamic hand gestures acquired by leap motion (LM), and it reaches recognition rates of about 90.63% for LM-Gesture3D dataset created by the paper and 93.3% for Letter-gesture dataset, respectively.

1. Introduction

Dynamic hand gesture recognition is a very intriguing problem in recent years that, if efficiently solved, could be the wealthiest means of communication that can be used. Because of this, many scholars from all over the world have done a lot of theoretical and practical research studies [1]. Compared with static gestures, the meaning of dynamic gestures is more abundant, and it is more common and natural to be an interactive way. But, at the same time, the information of dynamic hand gestures, such as shape and location, varies as time, which consequently increases the difficulty in recognition.

At present, there are two main types of sensors that are capable of sensing hand gestures: wearable sensor or vision-based sensor [2, 3]. The former approach could capture the movement of hands and fingers at the expense of convenience and cost and sufficiently extract information of hand, but it places an additional burden on users and could feel unnatural enough to perform hand gestures. Some advantages of a vision-based sensor are it can be less cumbersome and has more natural interaction than the wearable sensor due to no physical contact with users. However, its computational complexity is quite high for hand detecting, tracking, and extracting [4]. For instance, a hand should be separated from the background before the final recognition, which can be significantly affected by external environmental factors like ambient light. On the contrary, due to the complex 3D movements of hands or fingers, it is difficult to properly understand the performed hand pose based on the extracted information from 2D images [5]. Besides, once the palm surface is not parallel to the camera, for example, the recognition work could be harder.

The classification is a crucial step to recognize hand gestures. Five main classifying methods of hand gesture based on 3D vision can be identified: support vector machines (SVMs), artificial neural network (ANN), template matching (TM), HMM, and dynamic time warping (DTW) [4]. The SVM is a popular classifier for hand gesture recognition, in which support vectors are used to determine the hyperplane to realize the maximum separation of the hand gesture classes [6]. In vision-based hand gesture recognition systems, the ANN is used as a classifier to handle only fundamental and limited hand gestures [7]. When the high-level discriminative 3D hand features are available, the TM is an excellent choice for recognizing hand gestures, which works quite well with the contour- or boundary-based hand features [8]. As the hand gesture is a continuous pattern concerning time, the HMM is found to be the most suitable pattern recognition tool for testing on a moderately large dataset [9]. DTW is an indirect continuous hand gesture recognition approach that automatically aligns the sequences with different lengths and returns the proper distance [10].

Martin Sagayam and Jude Hemanth [11] develop a probabilistic model based on the state sequence analysis in the HMM to recognize hand gestures taken from the Cambridge hand dataset. The experimental results show that the proposed method achieves a 0.98% reduction in error rate and a 1.55% improvement in the recognition rate over that of the Viterbi prediction. Some work combines HMM with other methods for gesture recognition. Zhou et al. [12] use HMM to model the different information sequences of dynamic hand gestures and use BP neural network (BPNN) as a classifier to process the resulting hand gestures modeled by HMM, which achieves a satisfactory real-time performance and an accuracy above 84%. Martin Sagayam and Jude Hemanth [13] propose a hybrid 1D HMM model with artificial bee colony (ABC) optimization. The method is carried out with nine different classes of hand gestures that are used for virtual reality applications. The experimental results show that the average value of the recognition rate with ABC optimization increases by 2.72%, and the average value of the error rate is decreased by 0.47%.

With the emergence and development of deep learning technology, some scholars try to apply the technology for hand gesture recognition. Oyedotun and Khashman [14] apply a convolutional neural network (CNN) and stacked denoising autoencoder (SDAE) to recognize 24 American Sign Language (ASL) hand gestures obtained from a public database, which achieves the recognition rates of 91.33 and 92.83%. Bao et al. [15] propose a deep CNN that can classify hand gestures from the whole image without any segmentation or detection stage information. The method can organize seven sorts of hand gestures in a user-independent manner and achieve an accuracy of 97.1% in the dataset with simple backgrounds and 85.3% in the dataset with complex backgrounds.

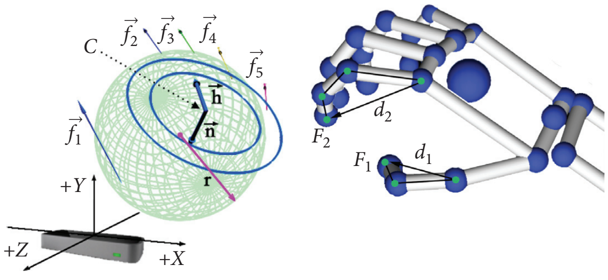
In recent years, 3D sensors, such as binocular cameras, Kinect, and LM, have been applied for hand gesture recognition with excellent performance. LM can detect and track hands and fingers with an accuracy of about 0.01 mm and feedback the gesture information in real time with a sampling rate of 120 fps [16]. Because of its superior performance, many researchers consider that it is a promising 3D sensor and particularly suitable for hand gesture recognition. For instance, Chen et al. [17] extract directional codes of 3D motion trajectory as the feature and exploit a classifier based on SVM to classify letter and number gestures. Ameur et al. [18] extract the positions of fingertips and palm center as features that are then trained with an SVM classifier. Their method reaches an average recognition rate of about 81% with 11 kinds of dynamic gestures. Xu et al. [19] and Zeng et al. [20] also conducted similar studies. Besides t, some researchers are working on dynamic gesture recognition. Lu et al. [21] build two kinds of features and feed them into the hidden conditional neural field classifier to recognize dynamic gestures. Avola et al. [22] propose a long short-term memory (LSTM) and recurrent neural networks (RNNs) combined with an effective set of discriminative features based on both joint angles and fingertip positions to recognize sign language and semaphoric hand gestures, which achieves an accuracy of over 96%. Vamsikrishna et al. [9] propose a low-cost computer-vision-assisted setup based on LM to detect precise movements of palm or finger within the field of view of the sensors. Then, it presents a set of discrete HMM for classifying the gesture sequences performed during rehabilitation.

The paper is aimed at recognizing the hand gestures corresponding to an operator's hand commands in robot teleoperation. For the problem, the paper develops four feature vectors and their extraction models based on 3D information acquired by LM to describe the hand gestures. And then, the article establishes HMMs to calculate the occurrence probabilities of four feature sequences in an unknown hand gesture, respectively. Lastly, the paper uses a weighted algorithm to fuse the occurrence probabilities of four features. The most considerable hazard is taken as can be taken as a recognition result. The rest of the paper is organized as follows. Prophase works of hand gesture recognition are introduced in Section 2. The methods of feature extraction are presented in Section 3, including valid dynamic gesture judgment, feature definition, and feature sequence clustering. HMM training model and hand gesture recognition by fusing the feature probabilities are proposed in Section 4. Section 5 comprises experiments and the result and discussion. Conclusion and possible future extensions are given in Section 6.

2. Prophase Work of Gesture Recognition

2.1. Leap Motion and Data Acquisition. LM, based on time-of-flight technology, mainly consists of three infrared LEDs and two infrared cameras, which can take photos from different directions to obtain gesture information in 3D space [16]. LM has about 150 degrees view field and an effective range of approximately 0.03 to 0.06 meters above itself. LM could feedback data frames that consist of positions and velocities of key points, rotation information, and frame timestamp.

When collecting gestures, LM will establish a right-hand coordinate system, as shown in Figure 1, based on all obtained data such as position, speed, and gesture of human hands. As shown in Figure 1, the five fingertips are denoted by $f_i (i = 1, \ldots, 5)$, and palm center is denoted by C. We mainly focus on the following data: (1) palm normal vector $\overrightarrow{n}$ and palm direction vector $\overrightarrow{h}$, which represent unit vectors perpendicular to the palm plane and point from the palm position toward the fingers, respectively; (2) finger direction vector $\overrightarrow{f}_i$ and the finger extension length points d_i, which represent the unit vector pointing to the point of the finger point F_i and the distance between two points, respectively; (3) instantaneous velocity v_i of five fingertips and instantaneous velocity v_C of the palm center; and (4) coordinate $p_t (x_t, y_t, z_t)$, which represents the coordinate of the palm position in the frame t.

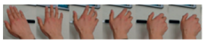
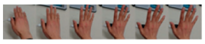
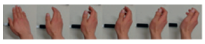
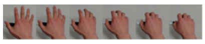
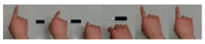
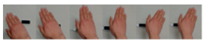
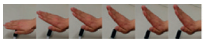
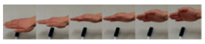


(a) (b)

Figure 1: Data acquisition from leap motion.

2.2. Dynamic Gesture Definition. There are relatively few publicly available hand gesture datasets created by LM-sampled images, especially for dynamic hand gestures in robot teleoperation. We analyze the movement characteristics of the operator's hand command in the robot teleoperation, such as translation and rotation of three degrees of freedom, and create a gesture dataset named LM-Gesture3D, which contains eight different dynamic gestures, as shown in Table 1. All these gestures collected by LM represent some practical operations or command signs and can be performed easily and naturally. Besides, there are similarities among the gestures in some respects, which will be illuminated in more detail later.

Table 1: Definition of LM-Gesture3D dataset.

Num	Gestures	Num	Gestures
G1		G2	
G3		G4	
G5		G6	
G7		G8	

3. Feature Extraction

3.1. Valid Dynamic Gesture Judgment. Despite the fact that LM has many merits, it mainly acts as a gesture data collector similar to a wearable device and camera. Hence, conditions for judging the beginning and the end of a valid dynamic gesture need to be given first. Take LM-Gesture3D as an example; it can be seen that the fingertips and palm center will inevitably produce rapid and continuous displacement when either gesture is performed. Even for a simplest dynamic gesture, click, for example, is no exception. A simple discriminant, based on the above analysis, is established as follows:

$$v = \max\left\{v_C, v_i \mid i = 1, \ldots, 5\right\} > v_\tau, \tag{1}$$

where v_C and v_i are the instantaneous velocity of palm center and fingertips, respectively, and v_τ is the predefined velocity threshold.

When the total number of continuous frames up to 60, v_C and v_i, satisfy discriminant (1), the data frames will be regarded as the original data of a valid dynamic gesture.

As LM is quite sensitive, in both cases when hand makes a slight shaking at rest and the obtained data contain noise, discriminant (1) could be satisfied in a few consecutive frames. So the total number (i.e., 60 frames) is set to eliminate these useless data. In addition, dynamic gesture with a low speed will be judged as invalid by discriminant (1), which means there is a degree of freedom for hand movement.

3.2. Feature Definition. To effectively recognize dynamic gestures, changes in hand posture and position are analyzed separately. The former can be further divided into the bending angle of fingers, opening perspective between fingers, and palm posture. The gesture trajectory can be represented later. Therefore, the paper describes the changes in gestures through the above four features.

The specific extraction process and expression of the four features are as follows.

3.2.1. Palm Attitude Feature. If the palm shape changes little in a dynamic gesture, the change in palm posture can be regarded as the problem of attitude angle calculation of a rigid body. The paper draws lessons from the 3D attitude measurement method, which is pointed out in [23].

As shown in Figure 1, the palm posture in the 3D space at any time could be uniquely determined by palm normal vector $\vec{n}$ and palm direction vector $\vec{h}$. Let $\vec{m} = \vec{h} \times \vec{n}$, then a new coordinate system $[\vec{h}_t, \vec{n}_t, \vec{m}_t]$ can be obtained to represent the palm posture in frame t. We take the initial data frame of the dynamic gesture as the fixed coordinate system and denote it as $[\vec{h}_t, \vec{n}_t, \vec{m}_t]$. So, the change in palm posture between the current frame and the first frame can be represented with three Euler angles:

$$\begin{cases} \psi_t = \arctan\left(-\dfrac{\left(\vec{n}_1 \cdot \vec{h}_t\right)}{\left(\vec{n}_1 \cdot \vec{n}_t\right)}\right), \\ \phi_t = \arcsin\left(\vec{n}_1 \cdot \vec{m}_t\right), \\ \theta_t = \arctan\left(-\dfrac{\left(\vec{h}_1 \cdot \vec{m}_t\right)}{\left(\vec{m}_1 \cdot \vec{m}_t\right)}\right). \end{cases} \tag{2}$$

3.2.2. Bending Angle of Fingers. As we mainly focus on the bending angle of the finger, the thickness of the finger could be regarded as useless, and then, each finger can be simplified to a planar model, as shown in Figure 2. Based on the two models, Hong et al. [24] propose a method to estimate the hand's attitude or instead bending angle of fingers and coordinates of joint points. At all conditions, their method require a merely total length of the finger l_i, visible length of the finger d_i, and several constraint constants. Combining with their research, we define the finger bending angle as

$$\omega_i = \lfloor 100 \times \left(\frac{d_i}{l_i}\right) \rfloor, \quad i = 1, \ldots, 5, \tag{3}$$

where d_i can be obtained directly from LM and l_i equals to d_i when the finger is straight.

In equation (3), l_i is used for normalization in order to make the approach robust to people with hands of different sizes. For l_i, a simple method is proposed to calibrate before data acquisition. The user keeps his/her palm plane parallel to LM and open fingers as straight as possible. When data of total continuous frames satisfy (1) $n_y \longleftarrow 0.94$ and (2) $|\vec{f}_i \cdot \vec{n}| < 0.008, i = 1, \ldots, 5$, up to 30, the obtained visible lengths of five fingers could be recorded as total lengths, where n_y is the component of normal vector $\vec{n}$ along the Y-axis direction in the LM coordinate system.

3.2.3. Opening Angle of Fingers. The other descriptor for the fingers is the opening angle between fingers. As mentioned above, every single finger can be modeled on a plane. Thus, the problem of computing the angle between two fingers can convert to one calculating the angle between two planes. Here, the plane consists of $\vec{h}$, and $\vec{n}$ is taken as the benchmark plane in the computation. Let $\vec{h} \times \vec{n}$ and $\vec{f}_i \times \vec{n}$ be the normal vector of the benchmark plane and finger planes, respectively. So, the opening angle can be calculated as follows:

$$\gamma_i = \arccos\left(\frac{\left(\vec{f}_i \times \vec{n}\right) \cdot (\vec{h} \times \vec{n})}{\left\|\left(\vec{f}_i \times \vec{n}\right)\right\| \times \|(\vec{h} \times \vec{n})\|}\right), \quad i = 1, \ldots, 5. \tag{4}$$

3.2.4. Trajectory Feature. A specific and meaningful trajectory usually accompanies some dynamic gestures, such as circling with a finger (like G5). So, the paper considers the path of the dynamic gesture and extracts a simplified feature for gesture recognition. When LM works, it can detect the palm center's return space coordinates with high accuracy and stability. So the moving trajectory of a hand can be expressed by a series of discrete points. The paper projects the gesture trajectory onto the LM's principal gesture plane, i.e., the XOZ plane. The detailed feature extraction processes are as follows:

(1) Let $(x_1, z_1), \ldots, (x_T, z_T)$ be the discrete points of the 2D gesture trajectory, then the central point $p_o(x_o, z_o)$ of these points can be expressed as follows:

$$p_o(x_o, z_o) = \left(\frac{1}{T}\sum_{t=1}^{T} x_t, \frac{1}{T}\sum_{t=1}^{T} z_t\right). \tag{5}$$

(2) Any point $p_t(x_t, z_t)$ and $p_o(x_o, z_o)$ form a vector of $\overrightarrow{p_o p_t}$ together with the central point p_o as the starting point. Then, the norm of $\overrightarrow{p_o p_t}$ and the direction angles between $\overrightarrow{p_o p_t}$ and the X-axis can be represented as follows:

$$\begin{cases} d_t(p_t, p_o) = \sqrt{(x - x_o)^2 + (z - z_o)^2}, \\ \varphi_t(p_t, p_o) = \dfrac{180}{\pi} \times \tan^{-1}\dfrac{(z_t - z_o)}{(x_t - x_o)}. \end{cases} \tag{6}$$

(3) Norm of the vectors d_t is normalized with the maximum norm $d_{\max}$, thus obtaining δ_t. Besides, direction angles of the vectors φ_t are converted into codes ψ_t according to the angular regions, as shown in Figure 3. δ_t and ψ_t can be computed as follows:

$$\begin{cases} \delta_t = \left\lfloor 20 \times \left(d_t \dfrac{(p_t, p_o)}{d_{\max}}\right)\right\rfloor, \\ \psi_t = \lfloor \varphi_t \dfrac{(p_t, p_o)}{18^\circ} \rfloor + 1. \end{cases} \tag{7}$$

Before coding the direction angle φ_t, we change the coordinate system from the original LM one into the coordinate system, as shown in Figure 3(a), the z' axis of which always points from the central point p_0 to the first point p_1. The obtained trajectory feature δ_t and ψ_t are of scale and rotation invariance based on the operation plane.

Select typical data once for each gesture in the LM-Gesture3D dataset and build their feature diagrams, as shown in Figure 4. Each row in Figure 4 corresponds sequentially to one of the gestures in the LM-Gesture3D. Four descriptions in each row from left to right are palm posture, finger bending angle, finger opening angle, and trajectory, respectively. It is not hard to see that each feature diagram depicts how its corresponding gesture is performed nicely. Gesture with complicated changes usually corresponds to complex feature curves, and vice versa. Different gestures may have similar features. The palm posture feature of G1–G3, for example, is similar to that of G6–G8 finger bending angle, and finger opening angle of G6–G8 is similar to each other. Therefore, it is not easy to distinguish these gestures just with a single feature. Of course, there are some gestures with significantly different features like G1 and G2. So, there is no misrecognition between G1 and G2.

There may be some more distinguishing features that can improve the recognition rate as well as reduce the computation cost for a given gesture. However, considering eight kinds of gestures in LM-Gesture3D that have obvious similarities, we prefer to select a feature set with completeness and redundancy that meets the requirements of

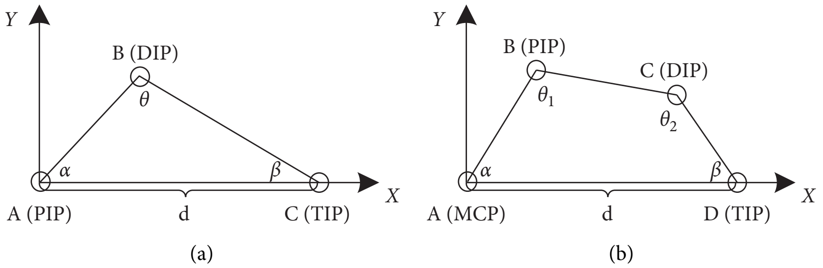


Figure 2: Simplified planar model of fingers: (a) triangular model for the thumb; (b) quadrilateral model for the other four fingers. MCP, PIP, DIP, and TIP represent metacarpophalangeal point, proximal interphalangeal point, distal interphalangeal point, and fingertip, respectively.

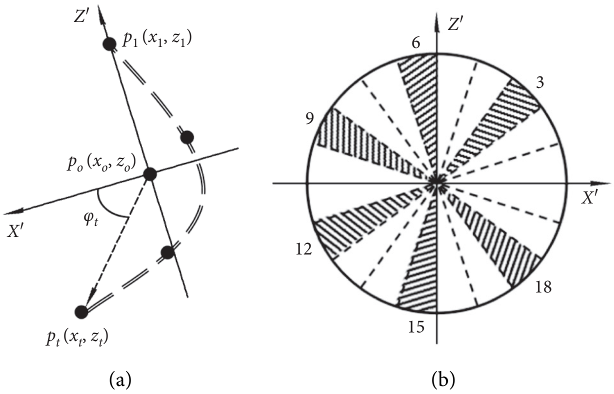


Figure 3: Schematic diagram of normalization process: (a) norm and direction angle of the vectors; (b) angular regions in the XOZ plane.

unified modeling and recognizes the gestures. According to the description of Figure 4, some features in the defined feature set are similar to each other for different gestures. However, there are some distinct features that are also included in the defined feature set. So, on the whole, the collected LM-Gesture3D or other more kinds of dynamic gestures can be adequately represented and distinguished by the defined four types of features.

In all four features, finger bending angle and finger opening angle are not affected by acquisition direction. To verify whether the rest two kinds of features are rotation invariance, we obtain the hand data of the gesture G6 from an experimenter, who is asked to make the gesture G6 twice during the collecting period. Then, we extract the posture feature and trajectory feature from the collected hand data and draw the feature curves, as shown in Figure 5.

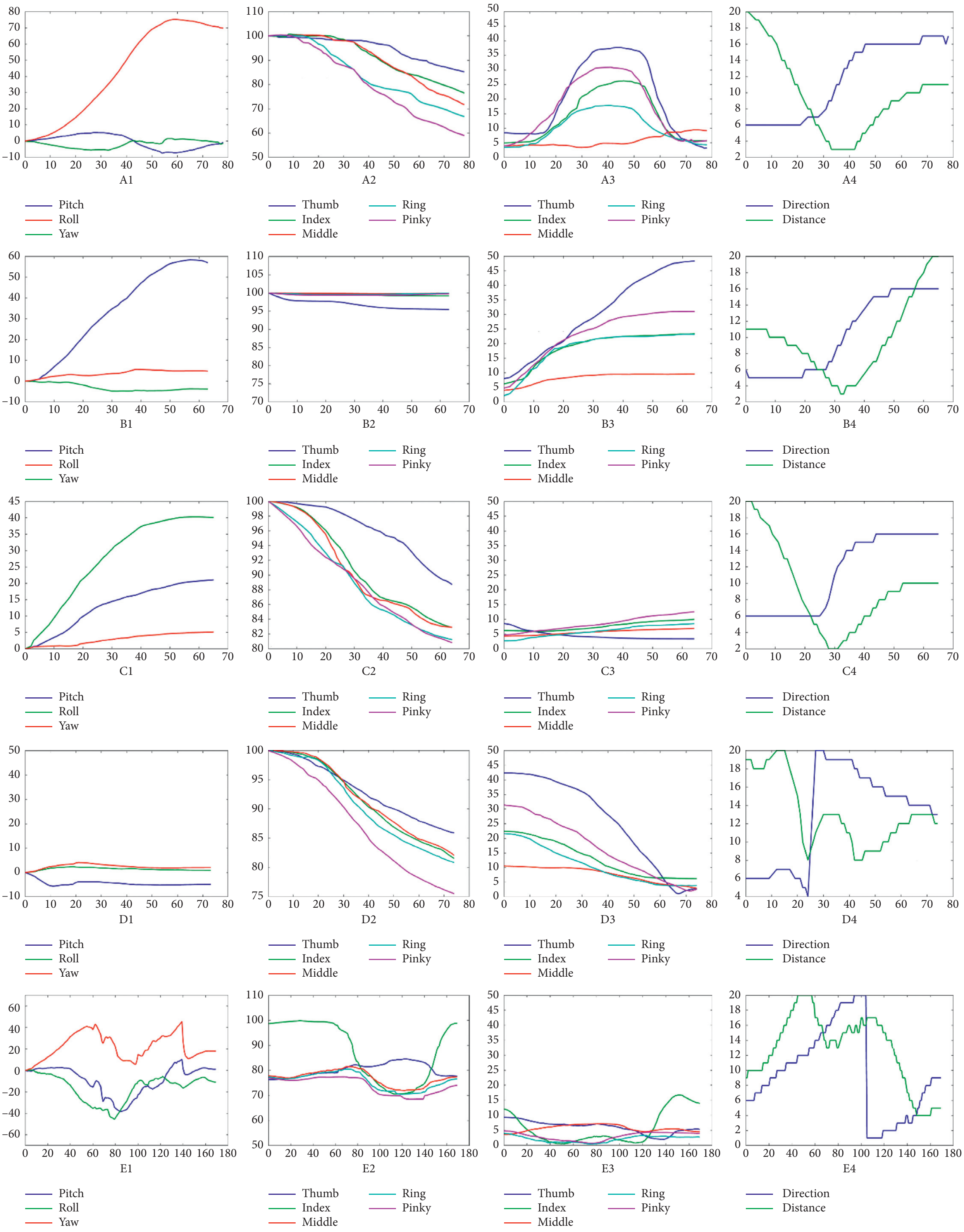
3.3. Feature Sequence Clustering. As shown in Table 2, in a single data frame of a gesture, four features can be represented by $m_i\,(i = 1, 2, 3, 4)$ dimensional vectors, respectively. Accordingly, each feature in T data frames of a dynamic gesture forms $T \times m_i$ dimensional vector sequences. In order to build the model of discrete HMM, K-means algorithm [25] is used to cluster the feature vector in the sequence. After clustering a feature vector into q class, the feature vector sequence can be expressed as $O = \{o_1, \ldots, o_t, \ldots, o_T\}$, where $o_t = 1, \ldots, q$ indicates that the feature vector is closest to the cluster center numbered o_t. In the paper, the cluster number q of four kinds of features is shown in Table 2.

In short, we take the discrete feature sequence composed of cluster tags as inputs of the discrete HMM. Therefore, both the sample data for HMM training stage and the gesture data for HMM recognizing need to go through the steps of feature extraction and clustering.

4. Gesture Modeling and Recognition

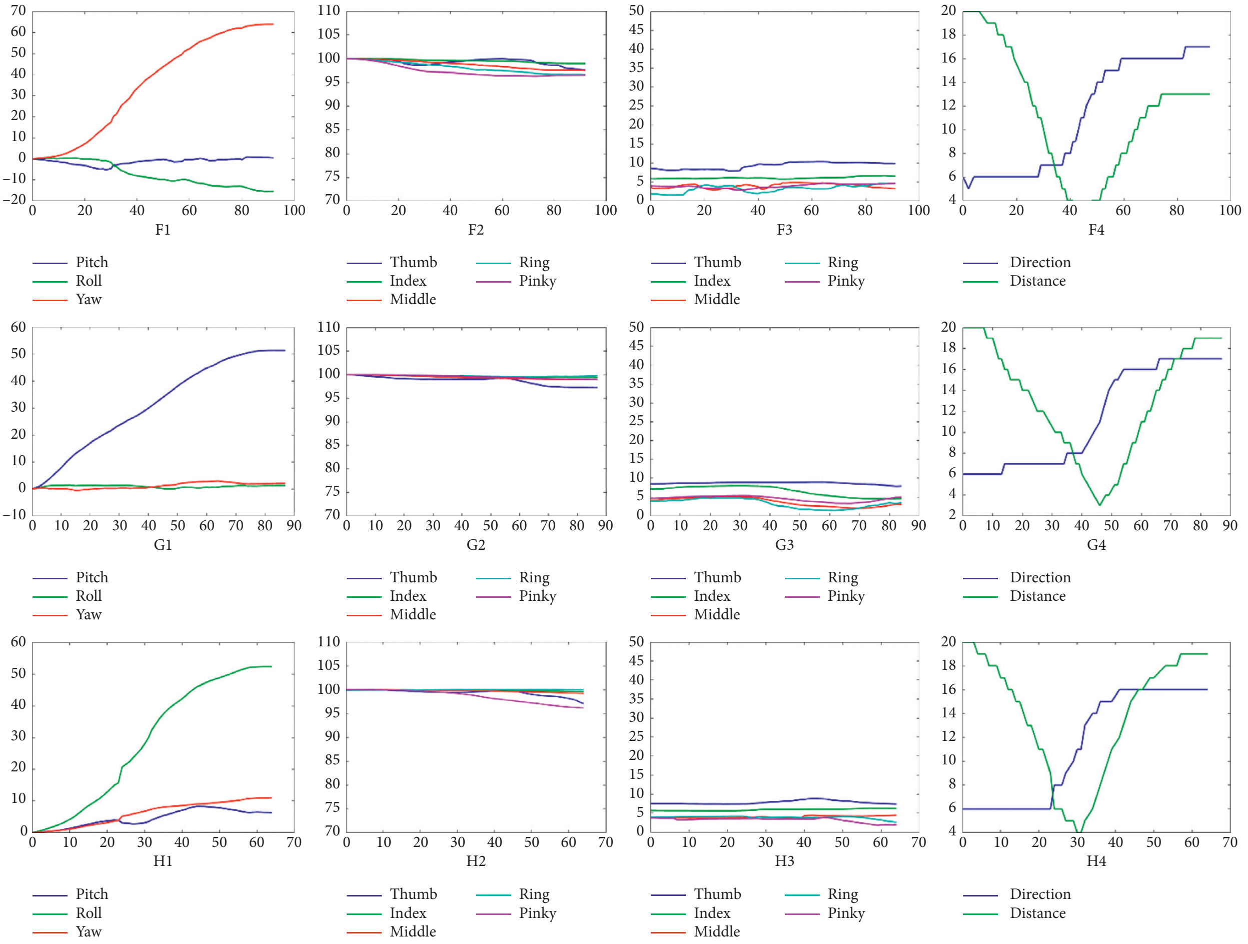
4.1. Recognizing Flow. The recognizing process of gesture is shown in Figure 6, which can be divided into two parts. The first part deals with the accurate gesture segmentation and four features extraction and quantification. The second part includes HMM model training and gesture recognition, both of which are based on the premise of feature sequences extraction.

The formal features of HMM can be expressed with a 5-tuple $(\Omega_X, \Omega_O, \mathbf{A}, \mathbf{B}, \pi)$, where $\Omega_X = \{q_1, \ldots, q_N\}$ is a finite set Markov chain state, and N is the number of states; $\Omega_O = \{V_1, \ldots, V_M\}$ is a finite set of observation symbols, and M is the number of symbols. $\mathbf{A} = (a_{ij})_{N\times N}$ is the matrix of state transition probability, $\mathbf{B} = (b_{ij})_{N\times M}$ is the matrix of observation probability, and $\pi = (\pi_1, \ldots, \pi_N)$ is the initial state probability distribution.



(a)

Figure 4: Continued.



(b)

Figure 4: Features of the gestures in the LMC-Gesture3D training dataset.

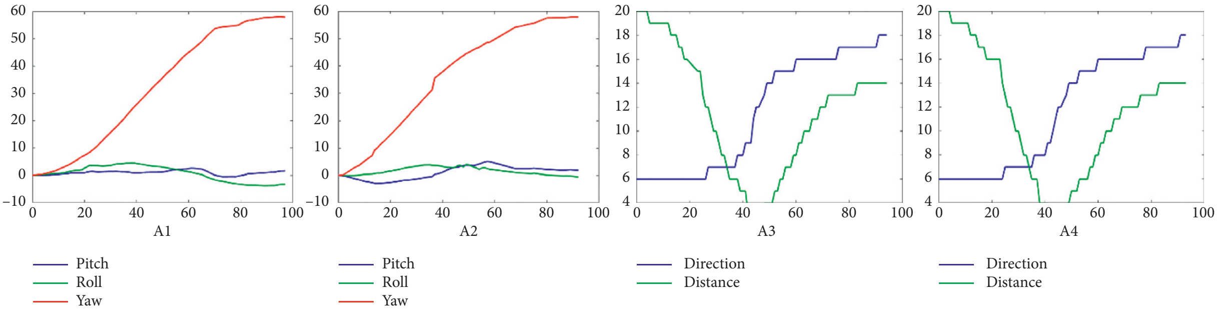


Figure 5: Palm posture and trajectory feature diagrams of gesture G6. a1 and a2 are the feature diagrams of palm posture corresponding to two gestures; a3 and a4 are the feature diagrams of trajectory corresponding to two gestures.

Table 2: Number of cluster centers of four features.

Feature	Feature vector	q
Palm posture	$[\psi_t, \varphi_t, \theta_t]$	16
Finger bending angle	$[\omega_1, \omega_2, \omega_3, \omega_4, \omega_5]$	14
Finger opening angle	$[\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5]$	10
Trajectory	$[\delta_t, \psi_t]$	10

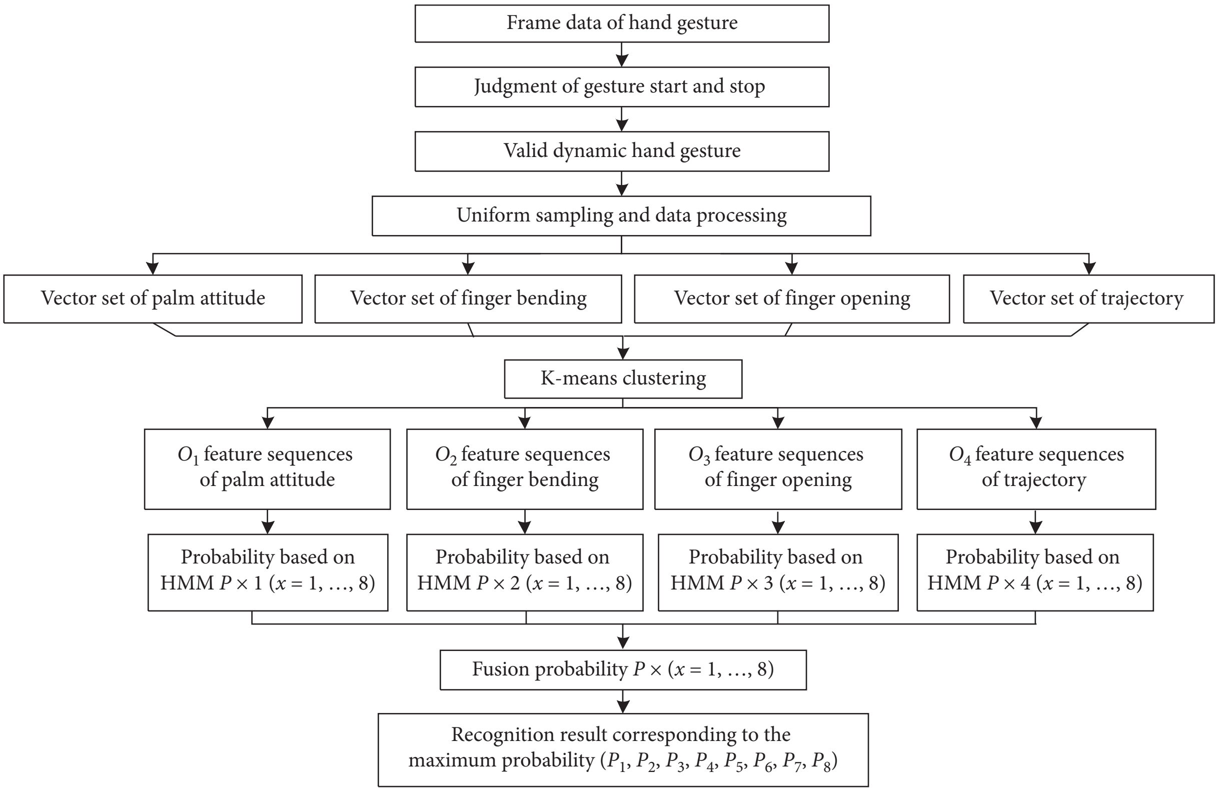

Frame data of hand gesture
Judgment of gesture start and stop
Valid dynamic hand gesture
Uniform sampling and data processing
Vector set of palm attitude
Vector set of finger bending
Vector set of finger opening
Vector set of trajectory
K-means clustering
O_1 feature sequences of palm attitude
O_2 feature sequences of finger bending
O_3 feature sequences of finger opening
O_4 feature sequences of trajectory
Probability based on HMM $P \times 1$ ($x = 1, \ldots, 8$)
Probability based on HMM $P \times 2$ ($x = 1, \ldots, 8$)
Probability based on HMM $P \times 3$ ($x = 1, \ldots, 8$)
Probability based on HMM $P \times 4$ ($x = 1, \ldots, 8$)
Fusion probability $P \times$ ($x = 1, \ldots, 8$)
Recognition result corresponding to the maximum probability ($P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$)

Figure 6: Implementation flow of the proposed method for recognizing dynamic gesture.

4.2. HMM Training. Unlike common one HMM for one kind of gesture modeling pattern, we build one HMM model for each feature, which means that 4 HMM models are adopted to achieve the recognition of each performed unknown gesture. Taking LM-Gesture3D for example, the designed 8 gestures are denoted by $g_u, u = 1, \ldots, 8$; then, for the feature sequence $S_u^v(v = 1, \ldots, 4\ )$ of gesture g_x, the following HMM modeling processes are carried out:

(1) HMM initialization: according to Table 1, in the paper, N is set to be 6. The number of observation symbols M is set as the same value of the number of cluster centers shown in Table 2; the initialization model parameters are described as $\lambda_u^v = (\mathbf{A}, \mathbf{B}, \pi)$.

(2) HMM parameters revaluation: assume that the feature sequence S_u^v consists of K observation sequences $O^{(k)}$, where $k = 1, \ldots, K$, and each observation sequence could be represented as $O^{(k)} = \left\{O_1^{(k)}, \ldots, O_T^{(k)}\right\}$.

For computing $\overline{\pi}_i$, $\overline{a}_{ij}$, and $\overline{b}_{js}$, respectively, the observation sequence $O^{(k)}$ and the original model parameter λ_u^v are substituted into the reestimation equations as follows:

$$\overline{\pi}_i = \sum_{k=1}^{K} \frac{a_1^k(i)\beta_1^k(i)}{P\left(O^{(k)} \middle| \lambda\right)},$$

$$\overline{a}_{ij} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1} a_t^k(i)a_{ij}b_j\left(O_{i+1}^{(k)}\right)b_{t+1}^k(j)/P\left(O^{(k)} \middle| \lambda\right)}{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1} a_t^k(i)\beta_{t+1}^k(i)/P\left(O^{(k)} \middle| \lambda\right)},$$

$$\sum_{k=1}^{K} \sum_{t\,=\,1}^{T_k-1} \quad a_t^k(j)\beta_t^k(j)/P\left(O^{(k)} \middle| \lambda\right)$$

$$\overline{b}_{js} = \frac{o_t^k = v_s}{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1} a_t^k(j)\beta_t^k(j)v/P\left(O^{(k)} \middle| \lambda\right)}, \tag{8}$$

where $1 \le i, j \le N$.

Thus, a new model $\overline{\lambda}_u^v = (\overline{\pi}, \overline{A}, \overline{B})$ is obtained. The above process would be repeated until the parameters in two adjacent iterations meet as follows:

$$\left|\log P\left(O \mid \overline{\lambda}_u^v\right) - \log P\left(O \mid \lambda_u^v\right)\right| < \varepsilon, \tag{9}$$

where $P(O\,|\,\lambda)$ is calculated from the forward-backward algorithm, which indicates the occurrence probability of the observation sequence O under the parameter λ, and ε is the predefined convergence threshold.

The final model parameter $\overline{\lambda}_u^v$ is the optimal parameter of feature sequence S_u^v, that is, the single feature HMM of its corresponding gesture. By repeating the above modeling process for each feature sequence S_u^v of 8 dynamic gestures, we can obtain 32 single-feature HMM models in total.

4.3. Gesture Recognition with HMM Fusion. In the stage of gesture recognition, once original data of an unknown and valid dynamic gesture are obtained, it would be first extracted into 4 observation sequences O_1, O_2, O_3, and O_4. Then, the forward-backward algorithm is used to calculate the occurrence probability $P(O_1\,|\,\overline{\lambda}_u^1)\,(u = 1, \ldots, 8)$ of the observation sequence under 8 single feature HMM $\overline{\lambda}_u^1\,(u = 1, \ldots, 8)$. Similarly, the occurrence probability of the observed sequence O_2, O_3, and O_4 under their corresponding HMM $\overline{\lambda}_u^2, \overline{\lambda}_u^3$, and $\overline{\lambda}_u^4$ can be obtained. For demonstration purposes, we represent the occurrence probabilities $P(O_v\,|\,\overline{\lambda}_u^v)$ as $P_{uv}\,(v = 1, 2, 3, 4)$.

We present an algorithm of weighted probability fusion to compute the probability that an unknown gesture belongs to the gesture u in LM-Gesture3D as follows:

$$P_u^{\mathrm{F}} = \sum_{v=1}^{4} \omega_{uv} P_{uv}, \tag{10}$$

where $\omega_{uv}(0 \le \omega_{uv} \le 1$ and $\sum_{v=1}^{4} \omega_{uv} = 1)$ is the weight of feature v corresponding to the gesture u.

According to equation (10), there are 8 calculation results, in which the maximum is regarded as the recognition result of the unknown gesture.

The paper employs least square method (LSM) to determine ω_{uv} in equation (10). Here is a brief introduction to the LSM weight method. Firstly, we calculate the probabilities of four features for all samples in the training dataset and can obtain $\mathbf{P}^m = \{\mathbf{P}_1^m, \mathbf{P}_2^m, \ldots, \mathbf{P}_8^m\}\,(m = 1, \ldots, L)$, where L is the number of samples and $\mathbf{P}_u^m = (P_{u1}^m, P_{u2}^m, P_{u3}^m, P_{u4}^m)$. Secondly, for the gesture u in LM-Gesture3D, if the sample m belongs to it, we set the probability of the sample m corresponding to the gesture u as follows:

$$\begin{aligned} P_{m,u}^{\mathrm{F}} &= \sum_{v=1}^{4} \omega_{uv}, \\ P_{uv}^{m} &= \boldsymbol{\omega}_u, \\ \left(\mathbf{P}_u^m\right)^{\mathrm{T}} &= p_s. \end{aligned} \tag{11}$$

Else, the probability of the sample m corresponding to the gesture u is set to be as follows:

$$P_{m,u}^{\mathrm{F}} = \sum_{v=1}^{4} \omega_{uv} P_{uv}^{m} = \boldsymbol{\omega}_u \left(\mathbf{P}_u^m\right)^{\mathrm{T}} = \frac{(1 - p_s)}{7}, \tag{12}$$

where $p_s\,(0.5 \le p_s \le 1)$ is a set probability.

Calculating the probabilities of all samples corresponding to the gesture u, we can obtain the following formula:

$$\boldsymbol{\omega}_u \begin{bmatrix} \left(\mathbf{P}_u^1\right)^{\mathrm{T}} \\ \left(\mathbf{P}_u^2\right)^{\mathrm{T}} \\ \vdots \\ \left(\mathbf{P}_u^V\right)^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} P_{u,1}^{\mathrm{F}} \\ P_{u,2}^{\mathrm{F}} \\ \vdots \\ P_{u,V}^{\mathrm{F}} \end{bmatrix}. \tag{13}$$

We use the least square method (LSM) to compute $\omega_u = (\omega_{u1}, \omega_{u2}, \omega_{u3}, \omega_{u4})$ in equation (15). Finally, ω_u is normalized to $\omega_u' = (\omega_{u1}', \omega_{u2}', \omega_{u3}', \omega_{u4}')$. ω_u' is the weight vector of the fusion model.

5. Experiments

To test the performance of the proposed method, several experiments are carried out on a desktop PC with an Intel Core i5-3230M processor and 4 Gb of RAM, and the software environment consists of Visual Studio 2013, Leap Motion SDK 2.3.1 + 3154, and MATLAB 2012a.

5.1. LM-Gesture3D Recognition Experiment. We select four participants with certain experiences in robot teleoperation to join the experiment. Each participant is asked to imitate each gesture in LM-Gesture3D 40 times repeatedly, and LM samples their gestures. So, there are 160 samples of each gesture.

To verify the feasibility of the proposed method, we define the recognition rate as follows:

$$\mathrm{RR} = \frac{N_{\mathrm{Rec}}}{M_{\mathrm{Sam}}} \times 100\%, \tag{14}$$

where N_{Rec} is the number of gestures correctly recognized and M_{Sam} is the total number of gestures recognized.

Firstly, we use K-fold cross-validation to evaluate the recognition performance and stability of the proposed method. In this experiment, K is set to be 10. So, each subset has 128 samples. Figure 7 is the result of K-fold cross-validation, which shows that the recognition rates of different trained models range from 89.8% to 92.9%. The fluctuation ranges of recognition rates of all 10 trained HMM models are about 3%, which shows that the proposed method has a good generalization ability. The average recognition rate of all 10 trained HMM models is about 90.8%, which indicates that the proposed method has a good recognition performance.

Furthermore, we analyze the recognition performance of the proposed method for different types of gestures in LM-Gesture3D. We randomly select 60 samples of each gesture as the testing set and the remaining samples as the training set. Table 3 shows the recognizing results. From the table, we can see that our method has a good representation of the 8 dynamic gestures with the average recognition rate of about 90.6%. The recognition rates for all gestures fluctuate slightly between 88.3% and 91.7%. The recognition rates of G4 and G6–G8 are higher than those of G1–G3 and G5. The reason is that these gestures are relatively simpler and easier for different users to repeat, while the participant's individual habits easily influence G1–G3 and G5. In addition, gestures G1–G3 are easily confused with G6–G8, respectively.

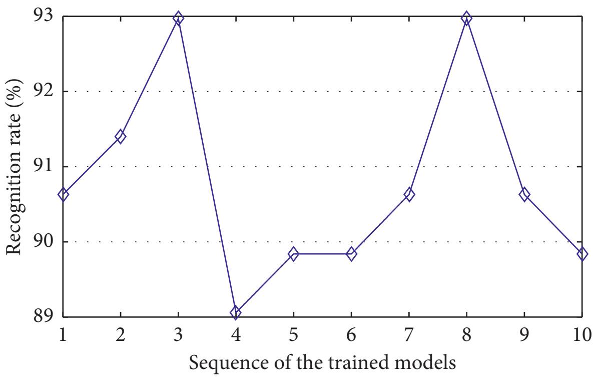


Figure 7: Recognition accuracies of LM-Gesture3D by using k-fold cross-validation.

Table 3: Recognition rate of the proposed method for the gestures in LMC-Gesture3D.

Gesture	G1	G2	G3	G4	G5	G6	G7	G8	RR
G1	54			3		3			90.0
G2		54					6		90.0
G3			53			3		4	88.3
G4	2		3	55					91.7
G5	3		3		55				90.0
G6	5					55			91.7
G7		6					54		90.0
G8			3				2	55	91.7

In general, the recognition results are jointly determined by four kinds of features, and our method based on multiple features and HMM can represent most kinds of complex gestures, which proves that our method is effective.

5.2. Dynamic Gesture Recognition Experiments. This experiment mainly tests our method's recognition rate for two kinds of relatively simple dynamic gestures, which are named letter-gesture dataset and the waving-gesture dataset, respectively. As shown in Figure 8(a), letter-gesture set consists of 6 gestures numbered 1 to 6, which are similar to each other. The waving-gesture dataset contains the rest 6 gestures shown in Figure 8(b). It can be seen that the main feature of two gesture sets is trajectory feature and palm posture feature, respectively.

The gestures in the experiment are sampled from four participants. Each participant is asked to repeat each gesture 50 times. When collecting the letter-gesture dataset, each participant keeps the shape as unchanged as possible and parallel to the horizontal plane of LM. Each gesture's obtained data are further divided into 120 sets of training samples and 80 sets of testing samples.

Chen et al. [17] propose a rapid early recognition system based on SVM to achieve multiclassification among the 36 dynamic gestures (the 3D motion trajectory of the numbers and the alphabet). Chen's method uses LM to capture 3D motion trajectories of the gestures, which is the same as our method. In Chen's method, the orientation angle is utilized as a unique feature of the gesture trajectory projected into the XOZ plane. It is quantized by dividing it by 45° and coded from 1 to 9, which is similar to our method. Chen's method is also used to recognize the gestures in the letter-gesture dataset.

Figure 9 shows the recognition results of our method and Chen's method. Our method and Chen's method get the average recognition rates of 96.0% and 93.5%, respectively. Two approaches have very similar recognition rates. However, the fluctuation of our method's recognition rate with LSM weights is smaller than that of Chen's approach. It shows that our method has better recognition stability than Chen's method.

In addition, the directional code extracted by Chen's method is determined by two neighboring points on the trajectory. In contrast, that of our method is determined by the trajectory points and the central point. At the same time, we also introduce a distance feature. Therefore, the extracted trajectory feature by our method is not affected by the amplitude of the gesture and is of rotation invariance.

Based on the above analysis, we believe that our method performs better than Chen's method.

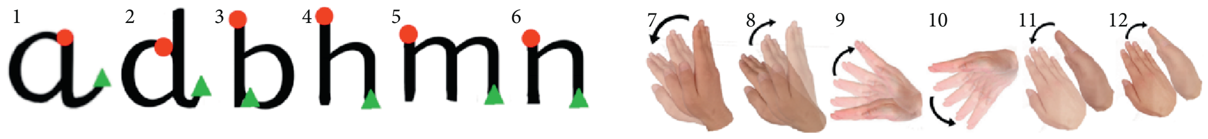
The waving direction of gestures 7–10 in the waving-gesture dataset is from upper right to lower left, from upper left to lower right, from top to bottom, and from bottom to top, respectively. And other gestures 11 and 12 make roughly 90° clockwise and counterclockwise rotations, respectively. It can be seen that this kind of dynamic gesture could be distinguished easily once using palm posture features. We carry an experiment to test the recognition performance of our method aiming at the 6 kinds of gestures. In the experiment, the method of data acquisition and processing is the same as that in the experiment of the letter-gesture dataset.

Pan et al. [26] present a combination method based on rule-based classification and SVM recognizes the gestures, which also use LM to capture real-time frame data of hand motion and define a 14-dimensional feature set including the absolute pose of hand in the 3D coordinate system and the pose changes in the hand between the two frames. Pan's method is also used to recognize the gestures in the waving-gesture dataset.

Figure 10 shows the recognition results of our method and Pan's method. The recognition rates of two methods for gestures 7–12 are all over 90, and the average recognition rates are 90.4% and 90.8%, respectively. The average recognition rate of Pan's method is slightly higher than that of our method.

Compared with our method, Pan's method will lead to more computational costs because it selects high dimension features and adopts a two-step recognizing strategy. Our method has not only a high recognition rate but also has the rotational invariance for selecting the rotation angle based on the initial posture of hand as features. Our method has a good effect on recognizing the wave or rotation gestures, such as those in the waving-gesture dataset.

In addition, all three methods above use LM to sample the gestures. The data of the features defined by three methods can be obtained quickly and accurately by LM. But adopting the camera approach, we have to depend on the



(a)

(b)

FIGURE 8: Gestures in letter-gesture set and waving-gesture set: (a) 6 kinds of gestures in letter-gesture set; (b) 6 kinds of gestures in waving-gesture set.

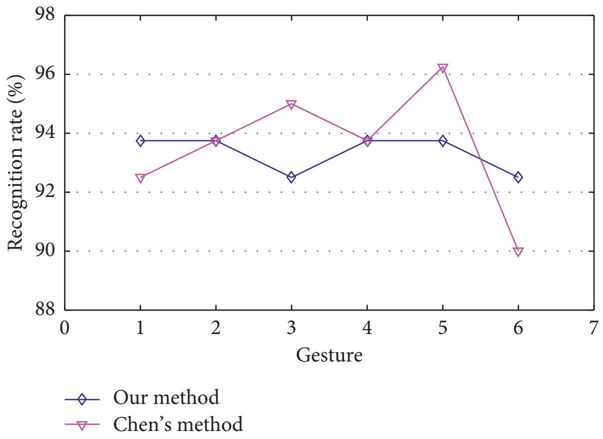


FIGURE 9: Results of using our method and Chen's method to recognize gestures from letter-gesture dataset.

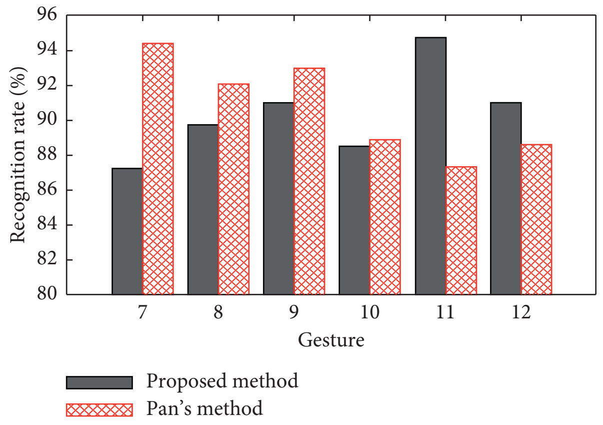


FIGURE 10: Results of using our method and Pan's method to recognize the waving-gesture dataset.

hand area feature to recognize the gestures, which is more complex and challenging. Hence, we can conclude that LM brings excellent benefits to our research.

5.3. Generalization Experiment. A generalization experiment is carried out to verify the adaptability of our method to nonstandard gestures. We select four inexperienced participants for the experiment. In the experiment, each participant is asked to repeat each gesture from LM-Gesture3D 40 times. A total of 1280 different gestures are sampled, which are recognized by the built HMM mode and the same weights in the LM-Gesture3D recognition experiment. The average recognition rate of 90.5% shown in Figure 11 is very similar to that of the LM-Gesture3D recognition experiment. So, the method is adaptable to different nonstandard gestures and has a good generalization ability.

We defined positive prediction value (PPV) and accuracy (ACC) of the gesture Gi ($i = 1, 2, \ldots, 8$) as follows:

$$\mathrm{PPV} = \frac{\mathrm{T}_{\mathrm{G}i}}{\mathrm{T}_{\mathrm{G}i} + \mathrm{F}_{\mathrm{G}i}}, \qquad (15)$$

where $\mathrm{T}_{\mathrm{G}i}$ is the number of gesture Gi correctly recognized and $\mathrm{F}_{\mathrm{G}i}$ is the number of other seven gestures that are incorrectly recognized as Gi.

$$\mathrm{ACC} = \frac{\mathrm{T}_{\mathrm{G}i}}{\mathrm{T}_{\mathrm{G}i} + \sum_{1\le j\le 8, j\ne i} \mathrm{F}_{\mathrm{G}j}}, \qquad (16)$$

where $\mathrm{F}_{\mathrm{G}i}$ is the number of gesture Gi that is incorrectly recognized as gesture Gj.

Table 4 shows the confusion matrix of the generalization experiment using the proposed method. According to Table 4, except for G5 with a PPV of about 0.96, the recognition precisions for the other seven gestures have a small difference ranging from 0.89 to 0.91.

5.4. Comparison Experiment with Other HMM-Based Methods. Here, we compare the recognition performance of the proposed method with other recognition methods based on HMM.

The authors in [11] define three features, including handshape, palm trajectory, and distance from the camera to extract the hand model from image features. It proposes a combinatorial method based on HMM and BPNN. The HMM-BPNN method uses the classical HMM to evaluate and decide the dynamic gesture features and, then, uses the BP neural network to classify the input state sequence.

In this experiment, the experimental samples are from the LM-Gesture3D recognition experiment in Section 5.1, from which 60 samples of each gesture and the remaining samples are randomly selected as the testing and training sets, respectively. The experiment is divided into two parts, including the feature testing and algorithm testing.

The feature testing experiment uses the features defined in the paper [12] to describe the gestures and analyze the HMM-BPNN method's recognition rate. Table 5 shows the recognition results of the experiment. From the table, we can see that the HMM-BPNN method has an average

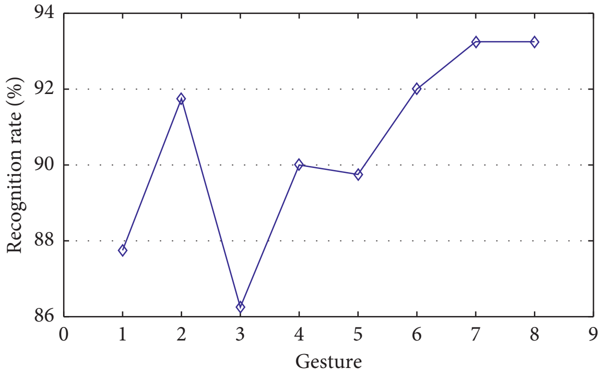


Figure 11: Recognition rate on nontrainer's gestures in LM-Gesture3D.

Table 4: Confusion matrix of the recognition results.

		Gesture samples								PPV
		G1	G2	G3	G4	G5	G6	G7	G8	
Recognition results	G1	140	1	3	5	2	2	1	1	0.90
	G2	2	146	3	2	3	2	3	2	0.90
	G3	1	2	138	3	3	2	1	1	0.91
	G4	6	2	3	144	2	1	1	1	0.90
	G5	1	1	1	1	143	1	1	0	0.96
	G6	7	1	2	2	2	147	2	2	0.90
	G7	1	5	3	1	3	3	150	3	0.89
	G8	2	2	7	2	2	2	1	150	0.89
ACC		0.88	0.91	0.86	0.90	0.89	0.92	0.94	0.94	0.90

recognition rate of only about 50.83% for 8 dynamic gestures. Moreover, for different types of gestures, its recognition rate fluctuates greatly. The main reason for the low recognition rate of the HMM-BPNN method for the gestures in LM-Gesture3D is that the three types of 2D features defined by the method are only suitable for representing simple and highly differentiated gestures but cannot fully represent complex and highly similar gestures, such as G5.

The algorithm testing experiment uses the features defined by our method to describe the gestures and analyze the recognition rate of the HMM-BPNN method again. Table 6 shows the recognizing results of the experiment.

From Table 6, we can see that the HMM-BPNN method has an average recognition rate of about 80.83% for 8 dynamic gestures. The recognition rate of the experiment is 30% higher than that of the feature testing experiment. Moreover, for different types of gestures, its recognition rate fluctuates less. The results show that the paper's features can more effectively represent complex gestures in LM-Gesture3D than that of the HMM-BPNN method.

For the same gesture samples and the same defined features, the recognition rate of our method, shown in Table 3, is more than 90%, which is about 10% higher than that of the HMM-BPNN method. We think there are two main reasons for the relatively low recognition rate of the HMM-BPNN method. Firstly, the input of the BPNN classifier is decided by a maximum assessment of the probabilities of the trained HMM models of four types of features, which does not consider the interference between similar features. Secondly, the BP neural network is prone to fall into local minima, which increases the risk of misrecognition when different sample features have significant similarities.

Table 5: Recognition rate of the HMM-BPNN method of the feature test.

Gesture	G1	G2	G3	G4	G5	G6	G7	G8	RR
G1	35	7	3	5	1	4	2	3	58.3
G2	6	34	4	4	2	3	4	3	56.7
G3	5	5	30	5	2	4	4	5	50.0
G4	4	5	6	31	1	4	4	5	51.7
G5	5	4	3	5	25	5	5	6	41.7
G6	5	4	5	4	4	30	4	4	50.0
G7	4	6	4	4	2	4	29	7	48.3
G8	5	5	3	4	2	5	6	30	50.0

Table 6: Recognition rate of the HMM-BPNN-based method of the algorithm testing.

Gesture	G1	G2	G3	G4	G5	G6	G7	G8	RR
G1	49	2	3	2		3	1	1	81.7
G2	3	48	1	1		2	3	2	80.0
G3	2	1	50	1		2	1	2	83.7
G4	2	2	3	48		1	2	2	80.0
G5	3	2	2	1	48	2		2	80.0
G6	4	2	1	1		50	2	1	83.3
G7	1	4	1	1	1	2	48	2	80.0
G8	2	2	3	2	1		3	47	78.3

6. Conclusion

In the paper, a fusion recognition method based on multiple features and HMM for the dynamic gesture is proposed. We consider both the change in handshape and moving trajectory and build four sorts of hand features with the advantages of being straightforward, simple, and rotation invariance, which bring better operation naturalness and flexibility for the operators. What is more, it offers a further expansion of more kinds of complex dynamic gestures by using these features. For each feature, we have built its corresponding HMM. In the recognition stage, we innovatively present a weighted fusion algorithm to calculate the occurrence probabilities and get the final recognition result. In the above way, the result is not easily affected by a particular feature.

The experimental results show that the proposed method is suitable for relatively simple dynamic gestures like letter gestures and waving gestures. Still, it also has strong robustness for complex dynamic gestures like LM-Gesture3D. The average recognition rate of the proposed method for LM-Gesture3D is up to 90.6%. Besides, the average recognition rate for inexperienced participants is about 90%. These results demonstrate the usability and feasibility of the proposed method.

Like other gesture recognition methods, the proposed method inevitably has certain limitations, and a more in-depth study needs to be carried out. Firstly, as we have adopted four HMMs for each gesture recognition, the algorithm's efficiency remains to be raised. Secondly, we have not yet done more research on the adaptive weight method and their further impact on the recognition rate, which will also be a future research direction.

Data Availability

The research library related to the dissertation will be established in GitHub (https://github.com/glchenwhut), where you can access the folders and find experimental data and lists.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Guoliang Chen conceived the idea, designed the experiments, and wrote the paper. Kaikai Ge helped with the algorithm and to analyze the experimental data.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 61672396.

References

[1] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artificial Intelligence Review*, vol. 43, pp. 1–54, 2015.

[2] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang, "RGB-d-based action recognition datasets: A survey," *Pattern Recognition*, vol. 60, pp. 86–105, 2016.

[3] L. Lamberti and F. Camastra, "Handy: A real-time three color glove-based gesture recognizer with learning vector quantization," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10489–10494, 2012.

[4] H. Cheng, L. Yang, and Z. Liu, "Survey on 3D hand gesture recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1659–1671, 2016.

[5] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 2015.

[6] O. K. Oyedotun, E. O. Olaniyi, A. Helwan, and A. Khashman, "Decision support models for iris nevus diagnosis considering potential malignancy," *International Journal of Scientific & Engineering Research*, vol. 5, no. 12, pp. 419–426, 2014.

[7] N. A. Ibraheem and R. Z. Khan, "Vision based gesture recognition using neural networks approaches: A review," *International Journal of human Computer Interaction*, vol. 3, pp. 1–14, 2012.

[8] W.-Y. Lin and C.-Y. Hsieh, "Kernel-based representation for 2D/3D motion trajectory retrieval and classification," *Pattern Recognition*, vol. 46, no. 3, pp. 662–670, 2013.

[9] K. M. Vamsikrishna, D. P. Dogra, and M. S. Desarkar, "Computer-vision-assisted palm rehabilitation with supervised learning," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 5, pp. 991–1001, 2016.

[10] R. Schramm, C. R. Jung, and E. R. Miranda, "Dynamic time warping for music conducting gestures evaluation," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 243–255, 2015.

[11] K. Martin Sagayam and D. Jude Hemanth, "A probabilistic model for state sequence analysis in hidden markov model for hand gesture recognition," *Computational Intelligence*, vol. 35, no. 1, pp. 59–81, 2019.

[12] L. Zhou, L. S. Zhang, S. Lei, and X. B. Zhang, "Dynamic hand gesture recognition using HMM-BPNN model," in *Proceedings of the IEEE International Conference on Real-time Computing and Robotics (RCAR)*, vol. 6-9, pp. 422–426, Angkor Wat, Cambodia, June 2016.

[13] K. Martin Sagayam and D. Jude Hemanth, "ABC algorithm based optimization of 1-D hidden Markov model for hand gesture recognition applications," *Computers in Industry*, vol. 99, pp. 313–323, 2018.

[14] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Computing and Applications*, vol. 28, no. 12, pp. 3941–3951, 2017.

[15] P. Bao, A. I. Maqueda, C. R. del-Blanco, and N. García, "Tiny hand gesture recognition without localization via a deep convolutional network," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 3, pp. 251–257, 2017.

[16] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[17] Y. Chen, Z. Ding, Y. L. Chen, and X. Wu, "Rapid recognition of dynamic hand gestures using leap motion," in *Proceedings of the IEEE International Conference on Information and Automation*, pp. 1419–1424, Lijiang, China, August 2015.

[18] S. Ameur, A. B. Khalifa, and M. S. Bouhlel, "A comprehensive leap motion database for hand gesture recognition," in *Proceedings of the 7th International Conference on Science of Electronics, Technologies of Information and Telecommunications*, pp. 514–519, Hammamet, Tunisia, December 2016.

[19] Y. Xu, Q. Wang, X. Bai, Y. L. Chen, and X. Wu, “A novel feature extracting method for dynamic gesture recognition based on support vector machine,” in *Proceedings of the IEEE International Conference on Information and Automation*, pp. 437–441, Hailar, China, July 2014.

[20] W. Zeng, C. Wang, and Q. Wang, “Hand gesture recognition using leap motion via deterministic learning,” *Multimed. Tools Application*, vol. 4, pp. 1–22, 2018.

[21] W. Lu, Z. Tong, and J. Chu, “Dynamic hand gesture recognition with leap motion controller,” *IEEE Signal Processing*, vol. 9, pp. 1188–1192, 2012.

[22] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, “Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures,” *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 234–245, 2019.

[23] X. Ji, C. Yu, W. Chen, and D. Dong, “GNSS 3D attitude measurement system based on dual-antenna receiver with common clock,” in *Proceedings of the 2017 Forum on Cooperative Positioning and Service*, pp. 223–227, Harbin, China, May, 2017.

[24] H. Hong, J. Chao, Y. Jin, Z. Zhao, and W. Lin, “Key point model for hand pose estimation based on leap motion,” *Journal of Computer Aided Design & Computer Graphics*, vol. 7, pp. 1211–1216, 2015.

[25] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” *IEEE Transactions on Pattern Analysis*, vol. 7, pp. 881–892, 2007.

[26] J. J. Pan and K. Xu, “Three-dimensional freehand gesture manipulation based on Leap Motion (In Chinese),” *China Science-Paper*, vol. 10, pp. 207–212, 2015.