Received 10 December 2022, accepted 31 January 2023, date of publication 6 February 2023, date of current version 14 February 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3243083

RESEARCH ARTICLE

Keyframe Extraction and Process Recognition Method for Assembly Operation Based on Density Clustering

YONG LIU, QI QIAO, SHENGRUI SHI, XIANG WANG, MINGSHUN YANG, AND XINQIN GAO
Faculty of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an 710048, China
Corresponding author: Yong Liu (liuyong@xaut.edu.cn)

This work was supported in part by the Key Research and Development Program of Shaanxi under Grant 2021SF-421 and Grant 2021SF-422; in part by the Key Scientific Research Program of Shaanxi Provincial Education Department, China, under Grant 20JY047; and in part by the Collaborative Innovation Center of Modern Equipment Green Manufacturing in Shaanxi Province, China.

ABSTRACT A keyframe extraction and process recognition method for assembly operations is proposed based on density clustering to solve the problems of data redundancy and difficulty in obtaining valid data frames from the process of continuous assembly operations. A standard operation gesture set, including dynamic and static actions, was constructed by decomposing the assembly operation. The finger feature variables and comprehensive gesture feature quantized function were defined according to the finger joint structure. Based on searching for local extreme points in the function, the density clustering method was used to extract the keyframes of the assembly operation sequence to eliminate redundant data. Finally, the support vector machine algorithm model and Levinstein distance were determined to complete the keyframe recognition and assembly operation matching. A case study demonstrated that the proposed method could effectively discretize the assembly operation sequence, remove approximately **84%** of redundant data frames, and achieve a comprehensive recognition rate of **92%**.

INDEX TERMS Assembly operation, density clustering, gesture recognition, keyframe extraction, SVM.

I. INTRODUCTION

Human Computer Interaction (HCI) refers to the process of information exchange between humans and computers to accomplish certain tasks, such as voice, touch screen, somatosensory, and gesture interactions. Among these, gesture interaction has always been regarded as an important issue in HCI research. We can recognize gestures and translate them into device-control commands through semantic conversion [1]. With the advancement of gesture data acquisition technology, gesture recognition has become key to accurately conveying interactive instructions. Gesture recognition technology has been applied in many fields such as kinesthetic games, assistant driving, virtual assembly, and augmented reality. It provides users with an abundant, convenient, and immersive experience in HCI.

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

As early as the 1990s, scholars conducted experimental research on gesture recognition. With the development of technology, machine learning and neural networks have become the most common methods for gesture recognition. Kumar [2] proposed a multi-sensor fusion gesture recognition method based on a coupled hidden Markov model. This method overcomes the shortcomings of using the observation state in the Hidden Markov Model (HMM) and provides information interaction in the state space, thus improving the gesture recognition performance. In [3], Jaime implemented a possible gesture-recognition pipeline based on a classic Random Forest classifier for a basic gesture set. Other researchers attempted to apply deep learning to gesture recognition. To fully integrate the advantages of different models, Zhu [4] adopted a hybrid depth model to identify gesture data, which is composed of a convolutional neural network and long-term and short-term memory units. Nunez [5] presented a two-stage training strategy. In the first stage, a Convolutional Neural Network (CNN) is used to extract the relevant features

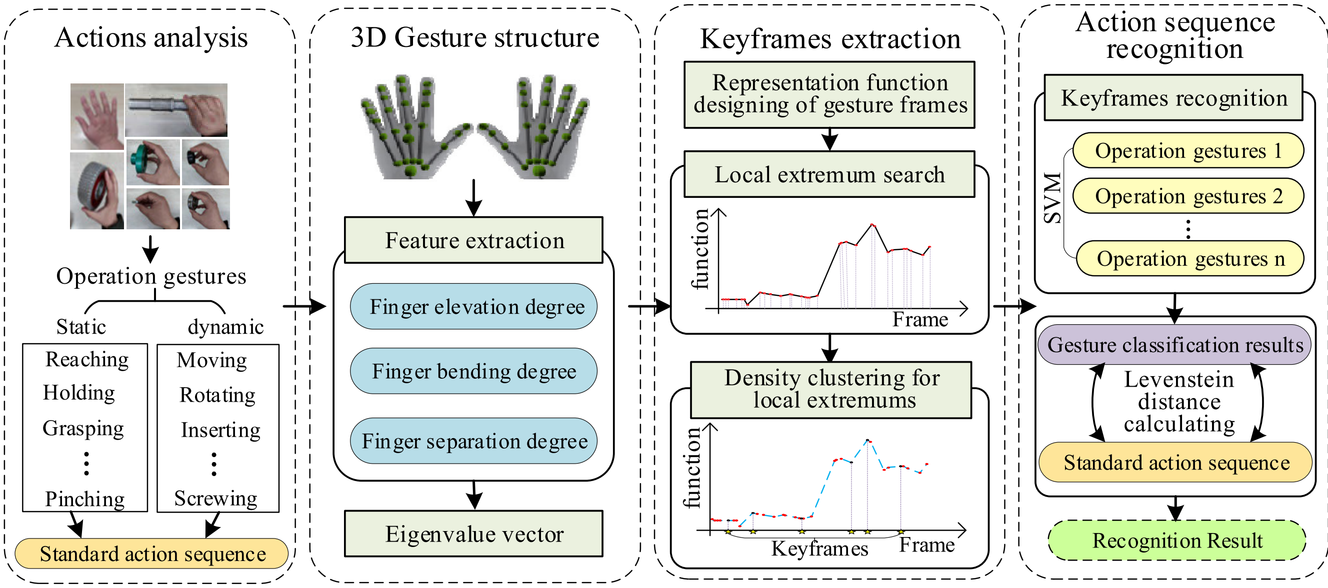


FIGURE 1. Proposed system architecture.

of gestures from 3D skeleton data. In the second stage, Long Short-Term Memory (LSTM) combined with a CNN is used for gesture action recognition. This method yielded good performance results for several benchmark datasets.

With the promotion of image and sensing technology, many types of gesture data can be collected using sophisticated acquisition equipment [6], [7], [8], [9], [10], but the amount of gesture data is large and there is redundancy. Prior work usually emphasized the use of whole data series, which affects the efficiency of learning and recognition, resulting in degraded performance. For complex data, feature extraction can reduce the dimensions and computations of the data. Therefore, feature extraction is widely used in classification tasks, and has achieved good results. In addition, it is also essential to extract the key data and remove redundant data to reduce the time complexity in the long duration and dynamic recognition environment. Currently, there are two main methods for keyframe data extraction: the frame difference algorithm and the clustering algorithm [20], [21], [22], [23], [24], [25]. Feature and keyframe extraction can significantly compress gesture data, which is effective in video data processing and has a good balance between performance and efficiency.

Recognition models typically need to be reconstructed for different objects to achieve recognition accuracy, and a significant amount of time is required for image marking and model training. However, research based on 3D hand gestures can circumvent these problems.

In this paper, a series of 3D digital gestures model of the assembly operation process is obtained by a depth image sensor, and divided into basic gesture elements, which are used to form complex assembly actions; a gesture comprehensive feature quantized function is designed to evaluate the variation degree of gestures frames; in order to compress the number of keyframes, density clustering method is introduced to extract the keyframes. Then, the action and operation process of the assembly are recognized accurately using the Support Vector Machine (SVM) algorithm and by analyzing the Levenstein distance between the identified and target operations. Based on the above methods, in subsequent assembly operation recognition, the recognition of complex actions can be realized only by constantly supplementing or improving the gesture elements and corresponding recognition models. The workflow of the proposed approach is illustrated in Fig. 1.

The remainder of this paper is organized as follows. Section II briefly reviews related work on gesture recognition. Section III analyzes the assembly action and extracts 12 operation gestures. Section IV models 3D gestures and extracts gesture features. In Section V, the feature function is constructed, and keyframes are extracted. In Section VI, the identification of keyframes and assembly actions is discussed, and the accuracy of the methods is verified through experiments. The final section concludes this paper.

II. RELATED WORK

A. GESTURE DATA

According to the different data collection methods, gesture data can be divided into visual-based data [6] and non-visual-based data. Wearable devices [7] and EMG signals [8] are common non-visual-based methods. Wearable devices such as data gloves have the advantages of high recognition accuracy and speed. However, the equipment is complex and heavy, which affects the gesture flexibility. The acquisition of EMG signals is more convenient, but it is significantly affected by noise and is difficult to process. Currently, computer vision is the most widely used gesture-recognition method. EyeToy, Kinect [9] ES8000 and Leap Motion [10] are commonly used devices. Leap Motion uses the binocular recognition principle to capture gestures, and its built-in algorithm can accurately track gestures and collect three-dimensional (3D) data. The visual-based method is more convenient, has a high recognition accuracy, and has gained increasing attention from researchers.

B. GESTURE RECOGNITION

Gesture recognition is widely used in various HCI environments. In this section, we review the research on gesture-recognition methods.

Before the deep learning method, the most important method of the classification was the combination of manual feature extraction and machine learning. Because feature extraction can significantly simplify the data volume, it is also applied to other methods. In [11], the Fourier coefficient amplitude was utilized for extracting features from images, and classification was performed via a Feedforward Neural Network (FNN). In [12], global and local features were combined before using an SVM for finger-spelling recognition. In [13], geometric features, Local Binary Patterns (LBP), distance, and number of fingers were used to extract features from the segmented depth image. Four multiclass SVM kernels were then compared and used to recognize gestures using the extracted feature vector as the input. In [14], Lu used a novel data glove called YoBu to collect data, and an extreme learning machine for gesture recognition. In [15], a view invariant hierarchical parsing method for free-form 3D motion trajectory representation was proposed. Trajectory recognition was achieved using the HMM. In [16], hand gestures were segmented based on thresholding in the YCbCr space. The segmented image was converted to grayscale and resized before being fed into the CNN as input. In [17], parallel CNNs using RGB and depth images as inputs were designed.

The above methods can achieve high accuracy in gesture recognition; however, the cost of computation is excessive. In [18], the authors introduced an effective method to reduce the number of frames of a gesture video by considering the relevant hand poses. This process reduces the processing time. In [19], a new deep-learning neural network model was designed. The network integrates several modules to learn both short-term and long-term features from video inputs and addresses the complexity and performance issues in hand gesture recognition.

Based on the keyframe extraction of the frame difference algorithm, Kim [20] proposed a method based on the compressed domain, which divides the video into multiple shots and determines a certain number of keyframes for each shot using the probability distribution. Sheena [21] calculated the mean and standard deviation of the histogram difference between video frames by segmenting the video shot and obtaining the screening threshold. Finally, the absolute difference in the histogram between the video frames was compared with the threshold for extracting keyframes. In [22], keyframe extraction was addressed as a high-dimensional motion curve simplification problem. Dictionary-based keyframe extraction adopts a dictionary to reconstruct a video, assuming that the video keyframe sequence is the best dictionary [23]. This type of algorithm converts keyframe selection into dictionary learning.

Keyframe extraction mainly uses clustering strategies such as k-means clustering, mean-shift clustering, and density clustering. Jeong [24] removed a small number of redundant frames using a spectral clustering method based on color histogram features, and obtained a concentrated video frame. Then, an accurate content sensing clustering was carried out for each period to obtain the key frames of the video sequence. Tang et al. [25] combined image entropy and density clustering to exploit keyframes from hand-gesture videos for feature extraction, which improves the efficiency of recognition. Mangai [26] proposed a keyframe extraction method using the HSV histogram and k-means clustering for temporal feature-based anomaly detection from surveillance videos.

Different application scenarios have certain requirements for feature selection, which plays a decisive role in the accuracy of the recognition model. A suitable machine learning model can improve the accuracy and efficiency of object recognition. The extraction of keyframes can not only reduce the amount of data input, but also improve the recognition accuracy and computing efficiency of the model.

III. ASSEMBLY OPERATION ACTION ANALYSIS

Taking a decelerator as the research object, each assembly process of its shafts was analyzed to find the therbligs from a series of complicated components and part operations. The therbligs can be summarized and extracted into 12 different operation gestures, as listed in Table 1.

Based on the posture and displacement changes of the gestures, they are divided into static and dynamic work gestures. The static operation gestures include reaching, holding, grasping 1, grasping 2, grasping 3, grasping 4, and pinching. The dynamic operating gestures include moving, rotating, inserting, pressing, and screwing.

IV. GESTURE DATA MODELING

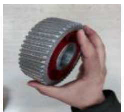
A. HAND STRUCTURE

A depth image sensor (Leap Motion) was used to extract the 3D gesture model in our investigation, which can be expressed by the phalanges and joint poses of the hand. As shown in Fig. 2, the human hand is composed of distal phalanges, middle phalanges, proximal phalanges, and metacarpals, except for the thumb, which lacks middle phalanges. Thus, the hand structure can be defined as a tuple that includes eight parameters, such as fingertip coordinates A_i, distal finger joints B_i, proximal finger joints C_i, finger root joint coordinates D_i, finger length L_i, palm point coordinates O palm normal vector $\vec{Q}$, and palm orientation T (the direction from palm to finger), where $i = 1, 2, 3, 4, 5$ refers to the thumb, index finger, middle finger, ring finger, and little finger.

B. TYPES OF GRAPHICS

According to the basic characteristics of assembly operation, the degrees of elevation P_i^j, separation S_i^j, and curvature H_i^j of finger i ($i = 1, 2, 3, 4, 5$) are defined as the characteristic variables of gesture j. The meaning and calculation method for each characteristic variable are as follows.

TABLE 1. Gesture analysis of parts assembly operation.

Gestures type	Operation Gestures	Object	Describe	Example	Label
	Reaching	None	Hands open		1
	Holding	Shaft	All fingers bend at the same time except the thumb		2
	Grasping 1	Gear	Smaller bend of four fingers and thumb		3
Static	Grasping 2	Bearing Sleeve, Gasket	Bigger bend of four fingers and thumb		4
	Grasping 3	Coupler	Small bend of four fingers and thumb		5
	Grasping 4	End cap, Bearing, Adjusting ring	Big bend of four fingers and thumb		6
	Pinching	shaft key, Bolt	pinch with thumb, index finger, and middle finger together		7
	Moving	All Object	hold object and move without rotation	--	8
	Rotating	All Object	hold object and rotate without moving	--	9
Dynamic	Inserting	Gear, Sleeve, Bearing, Gasket, Coupler	Grab object and move along shaft	--	10
	Pressing	Key, End cap, Adjusting ring	Keep grab gesture and Press object to a position	--	11
	Screwing	Bolt	repeat rotation with grasping posture	--	12

1) P_i^j, the angle between finger i and the palm plane in gesture j, is called the degree of finger elevation.

$$P_i^j = \arccos\left(\frac{(A_i^j - D_i^j)\cdot f^j}{\left\|A_i^j - D_i^j\right\| \times \left\|f^j\right\|}\right) \tag{1}$$

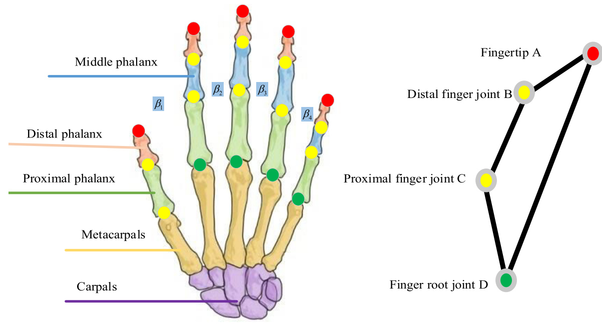


FIGURE 2. Manpower structure diagram, (a) A sketch of human hand, and (b) Finger joint definition.

2) S_i^j is the angle between the two fingers in gesture j, which is called the finger separation degree.

$$S_i^j = \arccos\left(\frac{(A_i^j - D_i^j)\cdot(A_{i+1}^j - D_{i+1}^j)}{\left\|A_i^j - D_i^j\right\| \times \left\|f^j\right\|}\right),$$
$$i+1 = \begin{cases} i+1 & i \le 4 \\ 1 & i = 5 \end{cases} \tag{2}$$

3) H_i^j is the bending degree of finger i in gesture j, which can be described as a quadrilateral formed by the four joint points of the finger, as shown in Fig. 2(b).

$$H_i^j = \frac{\left\|A_i^j - D_i^j\right\|}{L_i^j} \tag{3}$$

Finally, the eigenvalue vector F_j of a single gesture j can be represented by 15 tuples.

$$F_j = (P_1^j, \ldots, P_5^j, S_1^j, \ldots, S_5^j, H_1^j, \ldots, H_5^j) \tag{4}$$

V. KEYFRAMES EXTRACTION BASED ON DENSITY CLUSTERING

In a series of job action data sequences, using keyframes to record and analyze action behavior can significantly improve data processing speed and save storage space. Therefore, this study draws on the image keyframe extraction technology [18] and proposes a keyframe extraction method for gesture eigenvalues based on density clustering. The extraction process includes three steps: (1) calculating the characteristic parameters of the keyframe, (2) finding the local extreme points of the parameters, and (3) performing cluster analysis on the extreme points.

A. QUANTIZED KEYFRAMES REPRESENTATION FUNCTION

Keyframes refer to several action frames with typical representative meanings extracted from a sequence containing multi-frame action data. Keyframes generally appear when an action occurs with prominent or large changes. A quantized keyframe representation function is established, combined with gesture and posture feature variables, to characterize the variation in actions. The function can be defined as

$$K(F_j) = \sum_{i=1}^{5} P_i^j + \sum_{i=1}^{5} S_i^j + \sum_{i=1}^{5} H_i^j \tag{5}$$

where $j = 1, 2, \ldots, n$ is the total number of frames in the gesture data sequence of the assembly activity operation segment. In the gesture operation sequence, the change in gesture is represented by the change in the extreme point on the quantized keyframe representation function curve. The keyframe of the assembly action can be obtained by analyzing the extreme points of the curve.

B. LOCAL EXTREMUM POINT SEARCHING

The extreme points on the curve of the keyframe representation quantization function are composed of multiple local maxima and minima. The local maximum point set can be searched using (6), and the local minimum point set can be obtained using (7).

$$M_u = \{(j, K(F_j)) : K(F_j) > K(F_{j+1}) \text{ and } K(F_j) > K(F_{j-1})\} \tag{6}$$

$$M_d = \{(j, K(F_j)) : K(F_j) < K(F_{j+1}) \text{ and } K(F_j) < K(F_{j-1})\} \tag{7}$$

Therefore, the set of extreme points M_s is expressed as

$$M_s = M_u \cup M_d \tag{8}$$

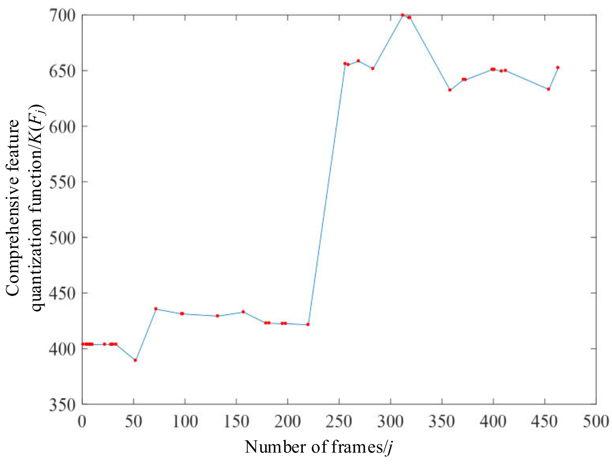


FIGURE 3. The extreme points of quantized keyframes representation function curve.

Representative frames can be extracted from the assembly operation gesture sequence by using the local extremum search method. The preliminary frame set forms a representative gesture sequence on behalf of the assembly process. Fig. 3 shows the extreme point diagram of the quantized keyframe representation function curve of a certain operation gesture sequence. Because of gesture joggling and moving during the operation, many extreme points will be obtained; therefore, it is necessary to eliminate the repeated points or similar points of extreme values to further reduce the amount of data processing. Therefore, this study adopts a clustering method to classify extreme points to obtain more representative keyframes.

C. DENSITY CLUSTERING ALGORITHM

Among the clustering algorithms, the K-means clustering algorithm is simple, easy to implement, and converges quickly. However, it has high requirements for the selection of initial clustering points and may fall into a locally optimal solution [6]. The density-based spatial clustering of applications with noise (DBSCAN) algorithm [27] does not need to determine the cluster center and number in advance; however, the density threshold must be determined first. The clustering effect is not good for samples with uneven density and large spacing. According to the data aspheric distribution characteristics of the research object in this study, clustering by fast search and finding of density peaks (CFDP) was selected [28]. The CFDP algorithm is based on the assumption that the local density of the points around the clustering center is relatively low, and the distance from these points to the clustering center is smaller than that from other clustering centers. For each data point, the CFDP calculates two quantities: the local density of the point, and the distance from this point to a higher local density point. These two quantities depend on the distance between data points.

The clustered point dataset is defined as $M_s = \{m_k\}_{k=1}^{N}$, where N is the number of points in the set. I_s is the index set of points, and $d_{h,l} = \text{dist}(m_h, m_l)$ is the Euclidean distance between point m_h and point m_l, where $k, h, l \in I_s$. There are two main approaches for calculating the local density of data points: the cut-off kernel and the Gaussian Kernel. The cut-off kernel is a density calculation method for discrete values, and the Gaussian kernel is adept at continuous values. It is rare for different data points to have the same local density if the Gaussian kernel is selected to calculate the density. The formula is as follows:

$$\rho_h = \sum_{l \in I_s \backslash \{h\}} e^{-(\frac{d_{h,l}}{d_c})^2}, \quad h \neq l \tag{9}$$

In which, the parameter d_c is the cut-off distance, which needs to be specified in advance according to the application object. The higher ρ_h is, the more data points there will be, and the distance from these data points to m_h is less than d_c.

For point m_h, the density distance is defined as

$$\delta_h = \begin{cases} \min\limits_{l \in I_s} \{d_{h,l}\} & \rho_l > \rho_h; \\ \max\limits_{l \in I_s} \{d_{h,l}\} & \text{otherwise.} \end{cases} \tag{10}$$

For the nonlocal maximum density point, the distance δ_h is calculated in two steps: Find out all points with higher local density than the point m_h, and from these points find the point m_l closest to the point m_h, the distance between m_h and m_l is δ_h; δ_h is the maximum value of the distance between the point with the highest local density and other points.

As shown in Fig. 4, the local density ρ_h and distance δ_h of each extreme point in Fig. 3 were calculated using (9) and (10), and the decision diagram was drawn with ρ_h as the abscissa and δ_h as the ordinate. Only points with high local density and relatively high distance are the cluster centers. According to this rule, we selected six cluster centers, and the data frame of the corresponding point was the keyframe, as shown in Fig. 5. Through density clustering, many redundant local extreme points are removed, and the key frame of the action sequence represented by the classification center point is obtained. This reduces the complexity of subsequent data processing and ensures the reliability of action recognition.

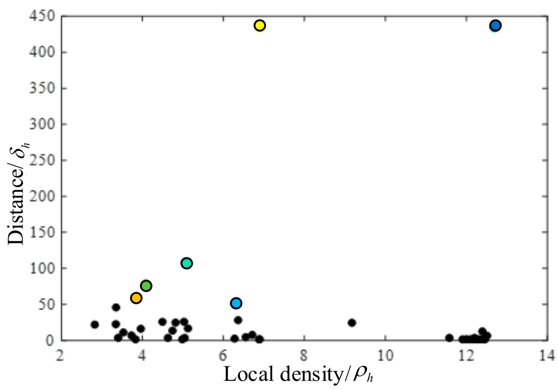


FIGURE 4. Decision diagram of cluster centers.

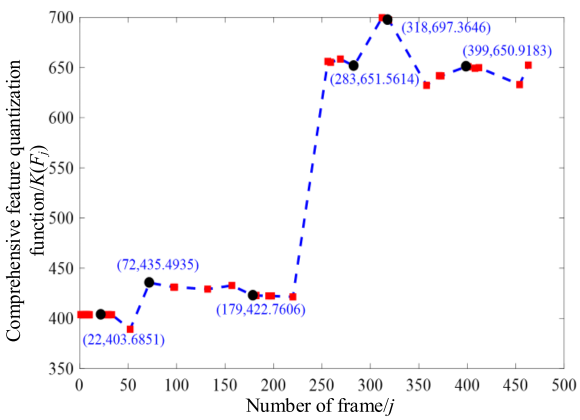


FIGURE 5. The results of keyframes extraction and clustering.

VI. KEYFRAMES EXTRACTION AND RECOGNITION OF OPERATION PROCESS

To evaluate the effectiveness of the keyframe extraction method, taking the assembly operation of the bearing in the shaft assembly of the decelerator as an example, the feasibility and effectiveness of the keyframes extraction algorithm were verified as follows.

A. GESTURE DATA ACQUIRING OF ASSEMBLY SEQUENCE

In the process of bearing assembly, the left hand holds the shafting unmoved, and the right hand completes the entire assembly process of the right-end bearing. The primary process is illustrated in Fig. 6. The assembly action of the right hand was the primary object for recognition in this experiment. Through an analysis of its action elements and gestures, the operation process can be decomposed into five basic therbligs: reaching, grasping, moving, rotating, and pressing. The gesture data sequences of the assembly process were collected using Leap Motion, which required 2.795 s and collected a total of 559 frames data. The feature variables of each frame's gesture data were calculated to constitute the corresponding feature vectors.

B. KEYFRAMES EXTRACTING

According to the keyframe extraction algorithm, the gesture keyframes of the bearing assembly process were extracted as follows:

1) EXTREME POINTS SEARCHING

First, all frames are input into the quantized keyframe representation function, as shown in (5), to construct the feature curve. Subsequently, 81 local extreme points can be identified from the curve using (6) and (7). The results are shown in Fig. 7.

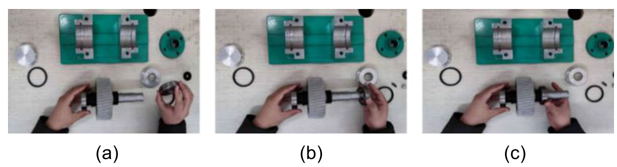


FIGURE 6. Schematic diagram of bearing assembly. (a) Before assembling, (b) In assembling, and (c) After assembling.

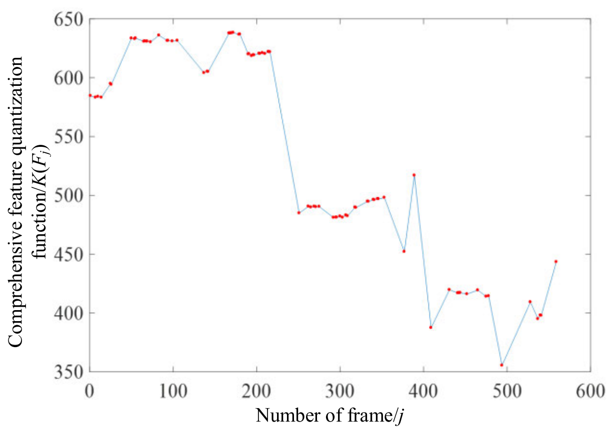


FIGURE 7. The gesture feature curve's extreme points of the bearing assembly operation.

2) CLUSTER ANALYZING

After obtaining the local extreme points, the cut-off distance $d_c = 70.03$ is set to guarantee that the average number of

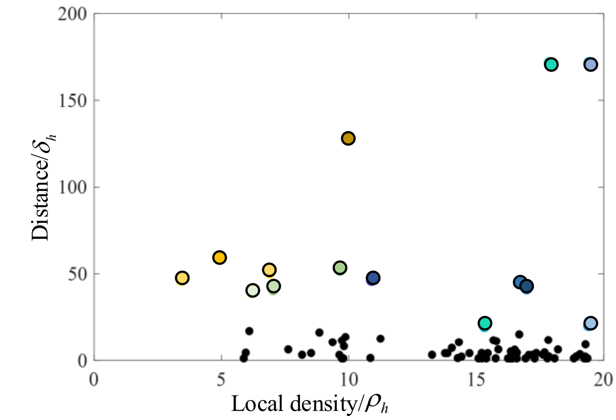


FIGURE 8. **Decision diagram of key frame density clustering for bearing assembly gestures.**

neighbor points for each data point is approximately 2% of the total number of data points. Any neighboring point should be less than d_c from its cluster center point. The Euclidean distance, Gaussian local density value, and density distance between extremum points must be calculated to form the decision diagram shown in Fig. 8.

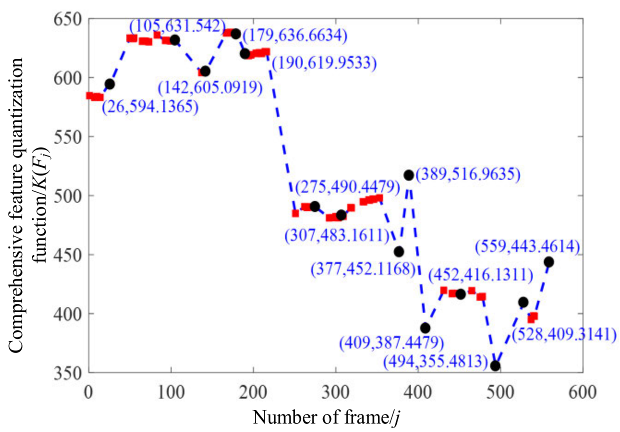


FIGURE 9. **The keyframes that make up the operation process.**

Fourteen keyframe points were selected as the cluster center points from the 81 extreme points. In Fig. 9, the red points are the local extremum points, the black points are the keyframe points, and the serial number and quantized feature value of the keyframe are marked in brackets near the points. Finally, 83.9% of the local extremum points were discarded, which helped reduce the number of invalid gestures and computing resource consumption.

C. KEYFRAME PREDICTION

To identify the gestures of above extracted keyframes, an SVM algorithm model was trained. A total of 7000 static gesture data were collected, 60% of which were used as the training set and 40% as the test set. Through training and testing, the recognition accuracy of the SVM model is 98%.

Keyframe data were input into the trained model to predict the gesture labels. The prediction results are presented in Table 2. Five reaching and nine grasping gestures were predicted from the keyframes, indicating that the representative keyframes could be effectively extracted using the density clustering method.

Meanwhile, it is necessary to consider dynamic actions such as moving and rotating to distinguish the dynamic properties of the keyframes and then achieve a complete match between the keyframe sequence and assembly process sequence.

D. PROCESS SEQUENCE RECOGNITION

Dynamic gestures, such as moving and inserting, involve translation and rotation of the palm while maintaining the grasping posture. Here, we introduce the palm coordinate variate ΔO and palm normal vector variate ΔQ to determine the dynamic features of the gesture translate-on and rotation, respectively, which are defined as follows:

1) The palm center point coordinate ΔO is the distance between the two palm center coordinates (x, y, z).

$$\Delta O = O_h - O_l = \sqrt{(x_h - x_l)^2 + (y_h - y_l)^2 + (z_h - z_l)^2} \tag{11}$$

2) The palm normal vector variate ΔQ is the modulus of the difference between the palm normal vectors (u, v, w) at two positions:

$$\Delta Q = \left|\vec{Q}_h - \vec{Q}_l\right| = \sqrt{(u_h - u_l)^2 + (v_h - v_l)^2 + (w_h - w_l)^2} \tag{12}$$

The ΔO and ΔQ values of two adjacent keyframes were calculated to determine the degree of gesture dynamics, as listed in Table 3.

In actual bearing assembly operations, gesture tremors are inevitable owing to the natural characteristics of the hand. It is necessary to use a threshold value to determine whether the hand is in normal working movement or in natural vibration. According to the work specification and assembly operation requirements, the threshold $\alpha = 50$ for ΔO, and the threshold $\beta = 0.1$ for ΔQ are set as the judgment conditions of the gesture dynamic degree. If the value is greater than the threshold value, it can be determined whether the hand is moving or rotating. Combined with the definition of the work gesture given in Table 1, the final recognition result of the assembly sequence is (reaching, grabing4, moving, rotating, grabing4, inserting, and moving) as shown in Table 3.

Finally, we analyze the matching degree with the standard assembly operation sequence to determine the assembly operation to which the above operation sequence belongs. The Levenshtein distance [29] is introduced for quantitative analysis of the matching degree, and the minimum number of editing operations (replacement, insertion, and deletion) is counted in the conversion from recognized sequence a to standard sequence b. The matching degree

TABLE 2. SVM-based keyframe prediction label.

Frame order	Comprehensive feature quantization value	Prediction label	Frame order	Comprehensive feature quantization value	Prediction label
26	594.14	1 (Reaching)	377	452.12	6 (Grasping 4)
105	631.55	1 (Reaching)	389	516.96	6 (Grasping 4)
142	605.09	1 (Reaching)	409	387.45	6 (Grasping 4)
179	636.66	1 (Reaching)	452	416.13	6 (Grasping 4)
190	619.95	1 (Reaching)	494	355.48	6 (Grasping 4)
275	490.45	6 (Grasping4)	528	409.31	6 (Grasping 4)
307	483.16	6 (Grasping 4)	559	443.46	6 (Grasping 4)

TABLE 3. Dynamic feature judgment of two adjacent keyframes ($\alpha = 50$, $\beta = 0.1$).

Frame order h	Frame order l	ΔO	$\Delta O \geq \alpha$?	ΔQ	$\Delta Q \geq \beta$?	Decision outcomes
26	105	149.0213	Y	0.0994		1 (Reaching)
105	142	85.9868	Y	0.1147	Y	
142	179	63.3285	Y	0.0743		
179	190	20.4757		0.0261		
190	275	34.8112		0.0675		6 (Grasping4)
275	307	83.2392	Y	0.0868		8 (Moving)
307	377	93.7663	Y	0.0751		
377	389	24.2660		0.1629	Y	9 (Rotating)
389	409	23.7093		0.0839		6 (Grasping4)
409	452	4.5141		0.0263		
452	494	75.1117	Y	0.1357	Y	10 (Inserting)
494	528	129.3995	Y	0.2394	Y	
528	559	75.5729	Y	0.0582		8 (Moving)

TABLE 4. The matching degree between the identified operation sequence and the standard assembly sequence.

Assembly object	Label	Standard assembly sequence	Levenstein distance	Matching degree
Shaft	1	Reaching、Grasping、Moving、Rotating	4	0.429
Gear	2	Reaching、Grasping 1、Moving、Rotating、Inserting	3	0.571
Bearing Sleeve	3	Reaching、Grasping 2、Moving、Rotating、Inserting	3	0.571
Coupler	4	Reaching、Grasping 3、Moving、Rotating、Inserting	3	0.571
Adjustment ring	5	Reaching、Grasping 4、Moving、Rotating、Pressing	3	0.571
Bearing	6	Reaching、Grasping 4、Moving、Rotating、Inserting	2	0.714
Key	7	Reaching、Pinching、Moving、Rotating、Pressing	4	0.429
Bolt	8	Reaching、Pinching、Moving、Rotating、Screwing	4	0.429

$R_{a,b}$ is expressed as

$$R_{a,b} = 1 - \frac{\text{lev}_{a,b}}{\text{Max}(\text{len}_a, \text{len}_b)} \tag{13}$$

where len_a and len_b are the lengths of sequences a and b respectively, and $\text{lev}_{a,b}$ is the Levinstein distance of conversion from a to b.

By calculating the matching degree in the bearing assembly experiment, the recognized sequence was found to have the highest matching degree of 0.714 with the standard assembly operation sequence of bearings, as shown in Table 4. The recognition results show that the proposed method is feasible and effective and can accurately identify the target operation.

At the same time, the final matching degree also shows that there are certain differences between the recognized process and the standard process. The main reason for this is the high flexibility of the hand joints. It is difficult to maintain consistency in operation and ease mixing in redundant gestures. Therefore, it is necessary to further verify the recognition accuracy and reproducibility of the method.

E. EXPERIMENTAL VERIFICATION OF RECOGNITION ACCURACY

To verify the recognition accuracy of the above method, eight standard assembly operation sequences, listed in Table 4, were studied experimentally. A total of 160 assembly operation process sequences from the four tests were captured to constitute a test set for the eight sequences. The recognition results for the assembly sequences were obtained via feature extraction, keyframe recognition, and sequence matching, as shown in Fig. 10.

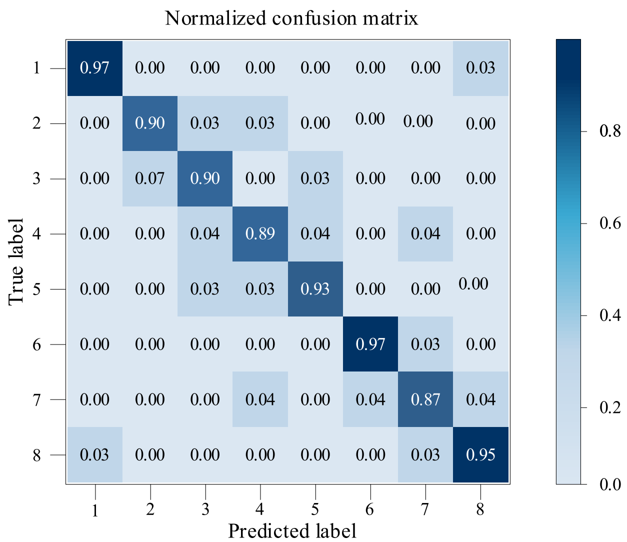


FIGURE 10. Experimental data confusion matrix.

From the recognition results, the recognition rate of the process sequence of coupler and key is low, the reason should be due to the high similarity between the defined grasping gestures. For bearings and shafts, the assembly operation sequences are quite different, and thus, the recognition accuracy is higher. Overall, the proposed process sequence recognition method could effectively identify assembly operations with an average recognition accuracy of 92.25%.

VII. CONCLUSION

This study is based on the assembly operation recognition of decelerator shaft parts. It conducts an analysis and modeling of assembly operations and gestures. A density cluster-based keyframe extraction method and process sequence recognition method are presented to achieve an effective judgment of the assembly operation.

1) The assembly action was decomposed into 12 basic operation gestures from both the static and dynamic aspects, which was conducive to the discretization of the process sequence.

2) Based on the analysis of the hand shape and structure, a 15-tuple hand gesture vector model was established to represent the operation postures of the hand.

3) A comprehensive feature quantization function of the gesture state was designed to obtain local extremum points from the trend curve of the action change. The keyframes of the process sequence were extracted using the density clustering method, and 83.9% of the data frames were redundant and were eliminated.

4) According to the dynamic representation parameters of gestures, dynamic and static gestures are subdivided to form a complete process sequence for recognition and matching with standard operation sequences. Levinstein distance was used to measure the matching degree of the recognized process sequence. The test experiments showed that the average recognition rate reached 92.25%.

The operation keyframes extracting method based on density clustering can discretize the continuous operation process, and at the same time can reduce redundant data frames, which is helpful to improve the recognition efficiency and accuracy. Subsequently, process recognition of different assembly operations can be realized by establishing a broad basic hand gesture database, which lays the foundation for robot operation process planning and teaching.

REFERENCES

[1] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2015.

[2] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "Coupled HMM-based multi-sensor data fusion for sign language recognition," *Pattern Recognit. Lett.*, vol. 86, pp. 1–8, Jan. 2017.

[3] J. Lien, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 142:1–142:19, Jul. 2016.

[4] G. Zhu, L. Zhang, L. Yang, L. Mei, S. A. A. Shah, M. Bennamoun, and P. Shen, "Redundancy and attention in convolutional LSTM for gesture recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1323–1335, Apr. 2020.

[5] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vélez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognit.*, vol. 76, pp. 80–94, Apr. 2018.

[6] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *J. Imag.*, vol. 6, no. 8, p. 73, Jul. 2020.

[7] X. Yang, X. Sun, D. Zhou, Y. Li, and H. Liu, "Towards wearable a-mode ultrasound sensing for real-time finger motion recognition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 6, pp. 1199–1208, Jun. 2018.

[8] I. Ketyko, F. Kovacs, and K. Z. Varga, "Domain adaptation for sEMG-based gesture recognition with recurrent neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 14–19.

[9] C. Zhu, J. Yang, Z. Shao, and C. Liu, "Vision based hand gesture recognition using 3D shape context," *CAA J. Autom. Sinica*, vol. 8, no. 9, pp. 1–14, Sep. 2019.

[10] W. Zeng, C. Wang, and Q. Wang, "Hand gesture recognition using leap motion via deterministic learning," *Multimedia Tools Appl.*, vol. 77, no. 21, pp. 28185–28206, 2018.

[11] A. A. Zare and S. H. Zahiri, "Recognition of a real-time signer-independent static Farsi sign language based on Fourier coefficients amplitude," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 5, pp. 727–741, May 2018.

[12] T. Pariwat and P. Seresangtakul, "Thai finger-spelling sign language recognition using global and local features with SVM," in *Proc. 9th Int. Conf. Knowl. Smart Technol. (KST)*, Feb. 2017, pp. 116–120.
[13] P. Sharma and R. S. Anand, "Depth data and fusion of feature descriptors for static gesture recognition," *IET Image Process.*, vol. 14, no. 5, pp. 909–920, Mar. 2020.
[14] D. Lu, Y. Yu, and H. Liu, "Gesture recognition using data glove: An extreme learning machine method," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2016, pp. 1349–1354.
[15] J. Yang, J. Yuan, and Y. Li, "Parsing 3D motion trajectory for gesture recognition," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 627–640, Jul. 2016.
[16] P. Nakjai and T. Katanyukul, "Hand sign recognition for Thai finger spelling: An application of convolution neural network," *J. Signal Process. Syst.*, vol. 91, no. 2, pp. 131–146, Feb. 2019.
[17] Q. Gao, "Static hand gesture recognition with parallel CNNs for space human-robot interaction," presented at the Intell. Robot. Appl., Wuhan, China, Aug. 2017.
[18] E. J. E. Cardenas and G. C. Chavez, "Multimodal hand gesture recognition combining temporal and pose information based on CNN descriptors and histogram of cumulative magnitudes," *J. Vis. Commun. Image Represent.*, vol. 71, Aug. 2020, Art. no. 102772.
[19] W. Zhang, J. Wang, and F. Lan, "Dynamic hand gesture recognition based on short-term sampling neural networks," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 1, pp. 110–120, Jan. 2021.
[20] K. W. Kim, "A new framework for automatic extraction of key frames using DC image activity," *KSII Trans. Internet Inf. Syst. (TIIS)*, vol. 8, no. 12, pp. 4533–4551, Nov. 2014.
[21] C. V. Sheena and N. K. Narayanan, "Key-frame extraction by analysis of histograms of video frames using statistical methods," *Proc. Comput. Sci.*, vol. 70, pp. 36–40, Dec. 2015.
[22] T. Miura, T. Kaiga, T. Shibata, H. Katsura, K. Tajima, and H. Tamamoto, "A hybrid approach to keyframe extraction from motion capture data using curve simplification and principal component analysis," *IEEJ Trans. Electr. Electron. Eng.*, vol. 9, no. 6, pp. 697–699, Nov. 2014.
[23] X. Li, B. Zhao, and X. Lu, "Key frame extraction in the summary space," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1923–1934, Jun. 2018.
[24] D.-J. Jeong, H. J. Yoo, and N. I. Cho, "A static video summarization method based on the sparse coding of features and representativeness of frames," *EURASIP J. Image Video Process.*, vol. 2017, no. 1, pp. 1–14, Dec. 2016.
[25] H. Tang, H. Liu, W. Xiao, and N. Sebe, "Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion," *Neurocomputing*, vol. 331, no. 2, pp. 424–433, 2019.
[26] P. Mangai, M. K. Geetha, and G. Kumaravelan, "Temporal features-based anomaly detection from surveillance videos using deep learning techniques," in *Proc. 2nd Int. Conf. Artif. Intell. Smart Energy (ICAIS)*, Feb. 2022, pp. 490–497.
[27] Y. Aref, K. Cemal, Y. Asef, and S. Amir, "Automatic fuzzy-DBSCAN algorithm for morphological and overlapping datasets," *J. Syst. Eng. Electron.*, vol. 31, no. 6, pp. 1245–1253, Dec. 2020.
[28] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[29] J. Beernaerts, E. Debever, M. Lenoir, B. D. Baets, and N. Van de Weghe, "A method based on the levenshtein distance metric for the comparison of multiple movement patterns described by matrix sequences of different length," *Expert Syst. Appl.*, vol. 115, pp. 373–385, Jan. 2019.

YONG LIU was born in Xinzhou, Shanxi, China, in 1981. He received the B.S. and M.S. degrees in industrial engineering and the Ph.D. degree in mechanical engineering from the Xi'an University of Technology, in 2003 and 2009, respectively. From 2010 to 2014, he was a Lecturer with the Mechanical Engineering Department, Xi'an University of Technology. Since 2015, he has been an Assistant Professor. He is the author of more than 60 articles, and holds 20 patents. His research interests include production process control and optimization, gesture recognition of assembly operations, and the application of IoT technology.

QI QIAO was born in Shangqiu, Henan, China, in 1997. She received the B.S. degree in industrial engineering from the Xi'an University of Technology, in 2016, where she is currently pursuing the master's degree in mechanical engineering. Her research interest includes action recognition in industrial environment based on machine vision.

SHENGRUI SHI was born in Xi'an, Shaanxi, China, in 1998. He received the B.S. degree in industrial engineering from the Xi'an University of Technology, where he is currently pursuing the master's degree in mechanical engineering. He holds one patent of invention and four software copyrights patents. His research interests include digital twins and the application of IoT.

XIANG WANG was born in Shangluo, Shaanxi, China, in 1999. He received the bachelor's degree in industrial engineering from the Xi'an University of Technology, in 2016, where he is currently pursuing the master's degree in mechanical engineering. He holds one patent. His research interests include digital twins, the application of IoT technology, and fault diagnosis and life prediction.

MINGSHUN YANG was born in Xixia, Henan, China, in 1974. He received the B.S. degree in mechanical engineering from the Northeast Heavy Machinery Institute, in 1995, the M.S. degree in mechanical engineering from Yanshan University, in 1998, and the Ph.D. degree from Xi'an Jiaotong University, in 2003. He is currently a Professor with the School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology. His research interests include advanced manufacturing technology, production planning and control, and product quality management.

XINQIN GAO was born in Lvliang, Shanxi, China, in 1976. He received the Ph.D. degree in mechanical engineering, in 2008. He is currently a Professor and a Supervisor of Ph.D. candidate with the Xi'an University of Technology. His publication has appeared in *Production Planning & Control*, the *International Journal of Computer Integrated Manufacturing*, and *Enterprise Information Systems*. His research interests include the intelligent manufacturing, decision making, and game theory. His research was supported by the National Natural Science Foundation of China.