# 3D Hand Trajectory Recognition with H-ELM

Jingjing Gao [a)], Yinwei Zhan

*School of Computer, Guangdong University of technology, Guangzhou 510006, China*

[a)] Corresponding author: jjgaogdut@foxmail.com

**Abstract.** In this paper, we present a method to extract features of dynamic gestures and use the Hierarchical Extreme Learning Machine (H-ELM) for gesture recognition. We use a Kinect sensor to record the motion of the three joint points of palm, wrist and elbow. The relation among the three trajectories is extracted as a gesture feature. Then the key nodes in the trajectory are extracted by calculating the U-Chord Curvature algorithm for eliminating redundant nodes and simplifying calculation. Then, through the sparse automatic coding and hierarchical training of the H-ELM, the input of automatic coding is approximate to the original input, and the reconstruction error is reduced. The experiment proves that H-ELM is faster than SVM and original ELM, and the recognition accuracy is higher.

**Key words:** gesture recognition; 3D trajectory; U-chord curvature; H-ELM.

## INTRODUCTION

As an important human-computer interaction, gesture has natural, intuitive and non-contact features, and has always been the frontier and hot spot of research at home and abroad. At present, the recognition of dynamic gestures mainly depends on the movement of the gestures and the change of hand shape. The gesture track is to recognize the meaning of gestures in accordance with some rules, and the application of gesture track information has a strong portability and extensibility. For different acquisition devices, different researches on 2D trajectory and 3D trajectory are presented. The feature of 2D trajectories are velocity, direction angle, position and so on. Singha et al [1] distinguishes the manual alphabet of different strokes according to the direction, location and distance information of the locus. Guo et al [2] designs a solution to identify the fingertip, wrist and arm from the depth map. Then the VR scene is manipulated by gesture commands differentiated by the specified feature points.

3D gesture recognition technology is a natural and efficient way to track and identify hand movements in the air. It is a natural and efficient human-computer interaction method. The stereo vision based on multi-angles information fusion can obtain 3D spatial information and improve the accuracy of gesture recognition, however, the algorithm is complex and difficult to meet the real-time requirement [3]. Kinect, Leap Motion and other RGB-D devices can record the coordinates of (x,y,z) directly by estimating the depth of the object. Bashir et al [4] used the geometric features of the trajectory directly from the Centroid Distance Function (CDF). This feature cannot describe the shape information of the trajectory completely and cannot distinguish different trajectories. Wu et al [5] proposes a context descriptor, based on the curvature window by selecting the method of curvature trajectory point of the largest and separated as context point, can fully retain the shape of the 3D trajectory information, at the same time has the RST independence.

DTW is put forward to deal with continuous change information. Ding et al [6] according to the angle changes of the key point of joint when people doing gestures, to derive an appropriate amount of feature vector for different categories of gestures, then using DTW algorithm to match the gestures with the categories. This algorithm is complex in calculation, and it is limited by the size and sequence length of the training set. Especially in the case that the training set is too large. Therefore, it is not suitable for the application in the real world. Beh et al [7] proposes a simple and effective modeling method based on the angle and angle variation of the hand trajectory and uses HMM to model the gesture trajectory. Wu et al [8] uses the Gaussian-Bernouilli Deep Delief Network (DBN) to handle skeletal dynamics, and a 3D Convoluted Neural Network (3DCNN) to fuse depth map and RGB image. Then using HMM to extrapolate the sequence of actions. HMM contains time and space information, and the algorithm is complex and requires a lot of training data.

Extreme Learning Machine (ELM) is a new machine learning method which can overcome the slow learning speed of neural network and the slow convergence speed in large-scale training samples in SVM. It's easy to use, and do not need to adjust the parameters. It is a kind of fast learning algorithm compared with the traditional BP algorithm. It has high generalization performance similar to BP and SVM and can well meet the requirements of real-time dynamic gesture recognition.

This paper uses the Kinect sensor to capture the dynamic trajectory data required by the experiment, without the influence of factors such as illumination and background and improves the stability and robustness of gesture trajectory recognition. A relative trajectory descriptor based on kinematics relation of multi-motion components is introduced. Among them, relative trajectory concept is defined based on direction and distance variation, which is beneficial to obtain the relative motion feature of relative motion trajectory of each child trajectory. Then, a new descriptor is constructed by combining the differential invariant of the root path and the distance of each relative trajectory. Then we train ELM to realize the dynamic trajectory recognition of trajectory 1 to 9.

## GESTURE TRAJECTORY PREPROCESSING

Gesture trajectory comes from the hand joint trajectory collected by Kinect, including position coordinates and depth coordinates of the three skeletal points of palm, wrist and elbow. The location of the marked skeletal point is shown in Fig.1. Kinect's frame rate is 30fps. Since it's difficult to define the starting position of dynamic gestures, we discard 10% of track points at the beginning and end of the track.

Because the trajectory lengths of different gestures are diverse, the same gesture may have different lengths of hand gestures because of different operator speeds. So, we need to resample the trajectory so that all hand gesture trajectories have a uniform length L. Calculate the average of the trajectory points of all trajectories. The normalized trajectory length is twice the average length. For trajectories below the average length, we use interpolation to increase its length; for trajectories that exceed the average length, we remove redundant points in the trajectory.

In order to maintain the relative position and scale of the trajectory point, we follow the method of [9] to normalize the trajectory and redesign the coordinates of the trajectory point so that they are evenly arranged according to the length of the trajectory and the position of the first point. Assume that the coordinate of the 3D track point before resampling can be expressed as $S_0 = \{x_p, y_p, z_p\}_{p=1}^{L_0}$. The normalized expression is

$$x_p^{'} = \frac{x_p - \min(\{x_p\})}{\max(\{x_p\}) - \min(\{x_p\})} \tag{1}$$

$$\mathrm{y}_p^{'} = \frac{y_p - \min(\{y_p\})}{\max(\{y_p\}) - \min(\{y_p\})} \tag{2}$$

$$\mathrm{z}_p^{'} = \frac{z_p - \min(\{z_p\})}{\max(\{z_p\}) - \min(\{z_p\})} \tag{3}$$

The normalized trajectory can be represented by a column vector

$$V = [x_1 ... x_L, y_1 ... y_L, z_1 ... z_L]^{\mathrm{T}} \tag{4}$$

Then the range of track coordinate values transforms to [0, 1].

In order to avoid unsmooth noise points in the track affecting the recognition result, we use the method in [10] to smooth the trajectory. For each point of each axis is calculated the mean value among its previous four neighbors and its four forward neighbors.

# TRAJECTORY FEATURE

Because dynamic gestures contain space and time information, so the 3D trajectory representations should be more efficient. The joint feature used in this paper is a 9-dimensional feature vector composed of the palm position feature and the spherical features formed by the wrist and elbow.

Because the dynamic gesture trajectory is a 3D discrete curve, calculating the trajectory feature using Euclidean distance is computationally expensive. And due to the high dimension of the trajectory feature, excessively redundant calculations will lead to a reduction in the real-time performance of the system. Therefore, this paper proposes a method based on the U-Chord Curvature [11] to calculate the point with larger curvature in the normalized trajectory, and then extracts the trajectory feature as a basis for classification.

## Root Trajectory Local Features

Gesture track acquired in real scene, even if they are samples of the same gesture, cannot guarantee that they have the same starting point. In order to reduce the error, all the image frame data of the trajectory of the palm is child trajectory from the coordinates of the starting frame to obtain the trajectory. The new coordinate expression for the track point sequence is $\Delta V$ . (x, y, z) are the coordinates of the track point, t represents the number of frames, N is the number of points in the track.

$$\triangle V\text{=}(x_t^{'}, y_t^{'}, z_t^{'}) = (x_t - x_1, y_t - y_1, z_t - z_1 \mid t \in [1, N]) \tag{5}$$

## Child Trajectory Spherical Coordinate Features

The movement of the hand involves the movements of the palms, wrists, elbows and shoulders, which satisfy the constraints of kinematics and have a certain time-space relationship [12]. Therefore, dynamic gesture recognition should include the feature of the trajectory between different parts, which helps to improve the accuracy of gesture recognition. Shao et al [13] proposed the concept of relative trajectory to satisfy the above situation.

Firstly, a representative trajectory is selected from a plurality of trajectories as a root trajectory, and the remaining trajectories are defined as child trajectory. In order to combine the spatial and temporal relationships between each child trajectory and the root locus simultaneously, a spherical coordinate system is introduced to describe the relative motion feature of each child trajectory relative to the root locus. From the space-time relationship between the root locus and the child trajectory, we can infer the pose of the dynamic locus. In this paper, we use the trajectory of the palm as the root trajectory S(t), where N is the number of frames, the root trajectory S(t) is

$$S(t)\text{=}(x_t^{'}, y_t^{'}, z_t^{'} \mid t \in [1, N]) \tag{6}$$

S($t_1$) is the trajectory of the wrist and S($t_2$) is the trajectory of the elbow. The child trajectory $\Delta S_1$ and $\Delta S_2$ are respectively expressed as the relative positions of the wrist and elbow relative to the palm of the hand. The expression of the child trajectory is

$$\Delta S_1\text{=}S_1\text{-}S\text{=}\{\Delta \mathrm{x}_t^1,\Delta \mathrm{y}_t^1,\Delta \mathrm{z}_t^1 | t \in [1,\mathrm{N}]\} \tag{7}$$

$$\Delta S_2\text{=}S_2\text{-}S\text{=}\{\Delta \mathrm{x}_t^2,\Delta \mathrm{y}_t^2,\Delta \mathrm{z}_t^2 | t \in [1,\mathrm{N}]\} \tag{8}$$

It combines the spatial and temporal relationships of the three body parts and effectively describes the sports connection of hand, wrist and elbow joints.

We represent $\Delta S_1$ and $\Delta S_2$ with the spherical coordinate system and represent the trajectory by two angles and a distance. Assume that $\theta$ is the y-axis inclination, $\phi$ is the azimuth angle from the z-axis in the xz plane, r is the radius of the spherical coordinate system, and N is the number of frames of the trajectory. The child trajectory spherical coordinates are expressed as

$$\Delta S_1 \triangleq \{\theta_t^1,\ \phi_t^1,\ \mathrm{r}_t^1 | t \in [1,\mathrm{N}]\} \tag{9}$$

$$\Delta S_2 \triangleq \{\theta_t^2,\ \phi_t^2,\ \mathrm{r}_t^2 | t \in [1,\mathrm{N}]\} \tag{10}$$

Therefore, the feature of the joint trajectory and the relative child trajectory spherical coordinate system of the wrist and elbow are defined as

$$f = (x_t^{'}, y_t^{'}, z_t^{'}, \theta_t^1, \phi_t^1, r_t^1, \theta_t^2, \phi_t^2, r_t^2) \tag{11}$$

## U-Chord Curvature Based Trajectory Key Points Extraction

Because the dynamic gestures are collected in space, the amplitude of the gesture waving is large, the trajectory is long, and the gesture trajectory does not vary greatly. So, the track contains more redundant points. So, we use an algorithm that computes the curvature size to extract the key points in the trajectory [11]. Relative to other algorithms, U-Chord Curvature has invariance to translation and rotation.

Firstly, we calculate the U-chord curvature for all trajectory points on the 3D gesture. For the ith sample point $p_i$ on the curve, if the distance between $\mathrm{p}_{i+u}$ and $\mathrm{p}_{i\text{-}u}$ is equal and the value is u, the U-chord curvature can be expressed as

$$c_i = s_i \sqrt{1 - (\tfrac{D_i}{2U})^2} \tag{12}$$

Among them, it is the symbol of the U-chord curvature defined in the literature [11]. This paper only uses the value of the curvature value, so it is set to 1. $D_t = \|p_{i-u} p_{i+u}\|$ is the Euclidean distance between $p_{i\text{-}u}$ and $p_{i+u}$. After the hand gesture trajectory in this article is resampled, the distance between adjacent track points is approximately equal to M, then U=uM in formula (12). Therefore, given the parameter u, the U-chord curvature of $p_i$ on the hand gesture trajectory can be found simply by calculating the value $D_i$. Take the first 80% of the trajectory point of curvature as the feature point of the trajectory.

For the root trajectory, the trajectory points with relatively large curvatures are calculated, and the remaining points are discarded. According to the number of frames t, the corresponding points in the child trajectories are found, which simplifies the calculation and shortens the calculation time.

## HIERARCHICAL EXTREME LEARNING MACHINE

In the past few years, Extreme Learning Machine (ELM) [14] has become a unique feature. That is, the training speed is very fast, and it has universal approximating and classification capabilities. It has become an application area for pattern recognition and artificial intelligence. An increasingly important study. The character of Extreme Learning Machine is randomly generating the connection weight between the input layer and the hidden layer and the threshold of the hidden layer neuron. It only needs to set the number of neurons in the hidden layer, and there is no need to adjust the parameters in the training process. The only optimal solution can be obtained. However, ELM is a single hidden layer feed-forward neural network, the training sample is always the initial training sample set, which limits the network's robustness and generalization ability. Therefore, this paper adopts an extension of the extreme learning machine, namely the Hierarchical Extreme Learning Machine (H-ELM) [15] as a classifier applied to gesture recognition. The algorithm uses ELM sparse automatic coding, so that the output after the automatic encoding approximates the original input and minimizing the reconstruction error. H-ELM has better generalization performance than the original ELM.

## Extreme Learning Machine

The original over-limit learning machine lacks good generalization performance and robustness in gesture recognition applications. Compared with the original over-limit learning machine, H-ELM has better generalization performance, faster recognition rate and higher recognition accuracy, and improves the overall learning performance.

The H-ELM algorithm is mainly divided into two phases: unsupervised hierarchical feature extraction and supervised classification. As shown in Fig.2, the previous stage extracts the multi-layer sparse matrix of the input data, and the final stage is based on the original ELM based regression for the final decision. ELM is a universal learning method that is effectively applied to a single-hidden layer feedback neural networks (SLFNs). The hidden layer parameters of ELM are randomly generated and do not require any sub sequent adjustments.

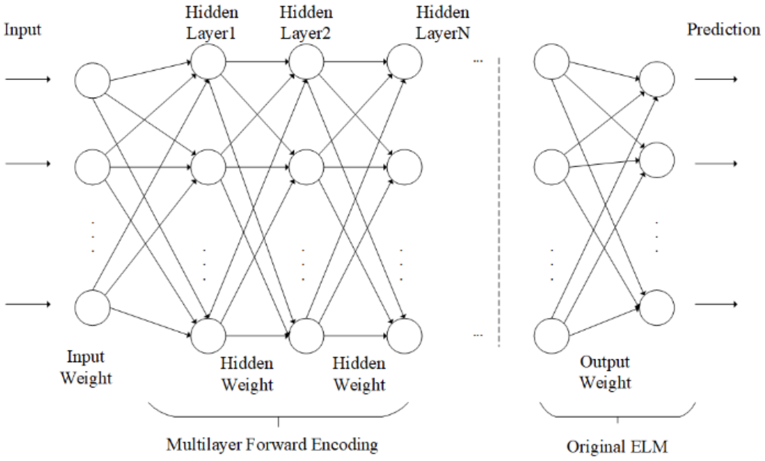

**FIGURE 2.** Structure of H-ELM learning algorithm.

Application of H-ELM algorithm as classifier in gesture recognition are as follows. Given a training set T $T=\{(x_i,t_i)\,|\,x_i\in R^d,t_i\in R^m,i=1,2,...,N\}$, hidden node output function $f_K(x)=\sum_{i=1}^{K}G_i(x,a_i,b_i)\beta_i$.

(1) Set the number of hidden layers, and the number of hidden nodes K, activation function and other relevant parameters. The training data and the corresponding label matrix are imported as output matrix X.

(2) Randomly generate input weights a and offsets b and base them orthogonally.

(3) Calculate the hidden layer output matrix H.

(4) Calculate the output weight $\beta=H^{+}T$ . Where $H^{+}$ is the Moore-Penrose (MP) generalized inverse matrix .

(5) Calculate the output matrix of the first layer $Y=\beta\cdot H$ .

(6) The output of layer 1 is then passed through the activation function as the input to layer 2. Until the output of the nth layer is calculated.

## Sparse Self-Encoding

In the first phase of the H-ELM algorithm, unsupervised multi-layer feature coding uses ELM sparse automatic coding. Automatic encoding is used as a feature extractor in a multi-layer learning framework to make the encoding output approximate the original input. It is known that the autoencoder aims to learn a function $h_\theta(x)\simeq x$ . Then to minimize the error of reconstructing. In mathematics, the input data x in the automatic coding can get a higher-level representation y through a deterministic mapping:

$$y=h_\theta(x)=G(A\cdot x+b) \tag{13}$$

where θ = {A, b}, $G(\cdot)$ is the activation function, A are he hidden weights and b is the bias. The final hidden layer indicates that y is then mapped back to the original input space to get the reconstructed input z:

$$z = h_{\theta'}(y) = G(A' \cdot y + b'), \theta' = \{A', b'\} \tag{14}$$

Therefore, the sparse autoencoder is a problem of reconstructing the input. The reconstructed input x can be regarded as an ELM learning problem. The weighted matrix A' is obtained by solving the regularized minimum mean square error.

Unlike sparse autoencoder based on BP neural networks that requires iterative adjustment of weights, ELM theory has shown that any input data can be estimated by random mapping of input weights. If the ELM theory is satisfied when training the encoders, then the initialization of the encoder is completed, and no adjustment of any weight is needed, which can reduce the training time.

## EXPERIMENT

This document was prepared using the AIP Proceedings template for Microsoft Word. It provides a simple example of a paper and offers guidelines for preparing your article. Here we introduce the paragraph styles for Level 1, Level 2, and Level 3 headings.

Because there is no published 3D gesture trajectory database on the Internet, the data set we used in the experiment was collected by Kinect v1.0. The data set includes a total of 9 dynamic gestures from 1 to 9 and records the trajectory coordinates of hand, wrist, and elbow of right hand. The database comes from 6 individuals. Each gesture is collected 50 times. A total of 450 data samples are stored in the xml file. Fig.3 shows the processed gesture trajectory of number 7, in which the blue line represents the hand, the red line represents the wrist, and the yellow line represents the elbow.

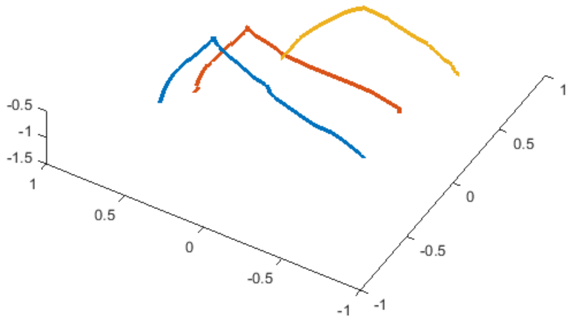

**FIGURE 3.** Gesture Trajectory of Number 7.

The experimental platform is MATLAB 2014b running on the Windows10 operating system. We use the aforementioned data acquisition and feature extraction methods to extract the right-hand position and spherical coordinate features, and using SVM classifier, the original ELM and H-ELM for comparison. The hidden nodes of ELM algorithm are set to 75, the number of hidden layers is set to 3 and the activation function is RBF. Table 1 shows the recognition effect of dynamic gestures of number 1 to 9:

**TABLE 1.** The average recognition rate of numbers 1~9.

| Gesture | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 99.8% | 99.2% | 98.6% | 97.5% | 96.4% | 98.3% | 99.1% | 97.5% | 98.2% |

Table 2 shows the comparison of SVM, ELM and H-ELM. It can be seen that the speed of ELM and H-ELM is obviously better than SVM.

**TABLE 2.** Comparison of different classifier recognition effects.

| Classifier | Mean Accuracy (%) | Mean Test Time (s) |
|---|---|---|
| SVM | 96.8 | 0.9014 |
| ELM | 97.5 | 0.0062 |
| H-ELM | 98.3 | 0.0055 |

## CONCLUSION

In this paper, we use the concept of relative trajectories to establish the space-time descriptors for multiple 3D gesture trajectories in space and extract the trajectory key points based on the U-Chord Curvature to identify dynamic gestures from 1 to 9. The layered limit learning machine is applied to dynamic gesture recognition, which reduces the computational complexity and improves the classification accuracy. Hierarchical Learning Machine not only inherited the feature of rapid learning of the original Extreme Learning Machine, but also reduced the reconstruction error by unsupervised and multi-level feature extraction and retained the accuracy of feature classification. Experiments show that this method can meet the real-time and accuracy requirements of dynamic gesture recognition, and its performance is better than SVM algorithm.

## ACKNOWLEDGMENTS

This work was supported by Project of Science and Technology Program of Guangdong (grant no. 2017B010110015) and Project of Natural Science Foundation of China (grant no. 61502159).

## REFERENCES

1. J. Singha, S. Misra, and R. H. Laskar. "Effect of variation in gesticulation pattern in dynamic hand gesture recognition system." Neurocomputing No. 208(2016), pp. 269-280.
2. S. Guo, M. Zhang, Z. Pan, and M. Sun. "Gesture Recognition Based on Pixel Classification and Contour Extraction." International Conference on Virtual Reality and Visualization IEEE, (2015), pp.93-100.
3. H. Aghajan, and W. Chen. "Layered and Collaborative Gesture Analysis in Multi-Camera Networks." IEEE International Conference on Acoustics, Speech and Signal Processing IEEE, (2007), pp. IV-1377-IV-1380.
4. X. Wu, X. Mao, L. Chen, Y. Xue, and A. Rovetta. "Point Context: An Effective Shape Descriptor for RST-Invariant Trajectory Recognition." Journal of Mathematical Imaging & Vision 56.3(2015), pp. 1-14.
5. F. I. Bashir, A. A. Khokhar, and S. Dan. "View-invariant motion trajectory-based activity classification and recognition." Multimedia Systems 12.1(2006), pp. 45-54.
6. I. J. Ding, and C. W. Chang. "Feature design scheme for Kinect-based DTW human gesture recognition." Multimedia Tools & Applications 75.16(2016), pp. 9669-9684.
7. J. Beh, D. Han, and H. Ko. "Rule-based trajectory segmentation for modeling hand motion trajectory." Pattern Recognition 47.4(2014), pp. 1586-1601.
8. D. Wu, L. Pigou, P. J. Kindermans, N. Le, L. Shao, and J. Dambre, et al."Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition." IEEE Transactions on Pattern Analysis & Machine Intelligence 38.8(2016), pp. 1583-1597.
9. W. Y. Lin, and C. Y. Hsieh. "Kernel-based representation for 2D/3D motion trajectory retrieval and classification." Pattern Recognition 46.3(2013), pp. 662-670.
10. D. R. Faria, and J. Dias. "3D hand trajectory segmentation by curvatures and hand orientation for classification through a probabilistic approach." IEEE International Conference on Intelligent Robots and Systems IEEE, 2009:1284-1289.
11. J. J. Guo, and B. J. Zhong. "U-Chord Curvature: A Computational Method of Discrete Curvature." Pattern Recognition & Artificial Intelligence 27.8(2014), pp. 683-691.
12. Jr. Gabriel, J. Brostow, and J. K. Hodgins. "Automatic Joint Parameter Estimation from Magnetic Motion Capture Data." Georgia Institute of Technology (2000).
13. Z. Shao, and Y. F. Li. "A new descriptor for multiple 3D motion trajectories recognition." (2013), pp. 4749-4754.
14. G. B. Huang, Q. Y. Zhu, and C. K. Siew. "Extreme learning machine: Theory and applications." Neurocomputing 70.1(2006), pp. 489-501.
15. J. Tang, C. Deng, and G. B. Huang. "Extreme Learning Machine for Multilayer Perceptron." IEEE Transactions on Neural Networks & Learning Systems 27.4(2016), pp. 809.