Image Processing, Electronics and Computers
L. Trajkovic et al. (Eds.)

doi:10.3233/ATDE240460

Research on Hand Motion Recognition Algorithm Based on STS-GCN

Ruimin ZHANG[1], Xiaodi WEI, Jianjun HAI and Shuqiang DU
College of Computer and Artificial Intelligence, Lanzhou Institute of Technology, Lanzhou, Gansu 730050, China

Abstract. In order to solve the problem that the spatial structure and temporal dynamic structure of skeletal data are not clearly and fully utilized when using hand bone data for action recognition, a spatio-temporal synchronous graph convolution network with combined attention mechanism is designed. Using the method of 2D estimation and triangulation, the feature is projected into a single 3D volume, the 3D heat map is output, and the 3D joint coordinates are obtained by soft-argmax operation on the heat map. The spatial dimension and the temporal dimension of the bone data are separated, the spatial dimension is encoded according to the order of the related nodes, and the same related nodes are encoded in the temporal dimension, and the spatial embedding matrix and the temporal embedding matrix are obtained. The matrix is synchronously added to the spatio-temporal network sequence. The experiment is tested on the SHREC2017 data set, and compared with some representative gesture recognition methods, the results show that the algorithm has achieved good results in hand action recognition.

Keywords. 3D Skeletal, Space-time synchronization, Graph convolution, Hand movements

1. Overview

Recognizing human hand gestures is an important topic in the field of computer vision. Enabling robots to have hands as dexterous as human hands is a challenging problem in artificial intelligence and robotics research [1]. User manipulation of objects through hand gestures is a very important mode of interaction, especially the hand's posture plays a central role in understanding and implementing hand-object interactions and gesture-based action recognition.

Hand gestures can be categorized into two types based on hand and finger movements: (1) Coarse gestures (such as swiping, waving up, down, left, and right, defined more by the movement of the hand). (2) Fine gestures (such as grabbing, pinching, zooming, rotating, defined by finger movements). Hand skeletal data can process precise information of hand shapes, providing skeletal joint data of the hand and fingers in the form of a complete 3D skeleton corresponding to 22 joints, which will aid in the recognition and analysis of gestures and hand actions. In response to the issue of partial hand joint occlusion caused by the limited field of view of RGB cameras, some researchers have proposed a method based on a multi-camera system [2]. This approach alleviates the

[1] Corresponding Author: Ruimin ZHANG, ruimin_zhang126@126.com.

high degree of occlusion during gesture interaction but is costly and complex to operate. Chen and others [3,4] designed a motion feature-enhanced recursive neural network, sequentially encoding each finger and the entire hand's skeletal joints. Lee et al. [5,6] developed a real-time learning deep network to recognize dynamic gestures, demonstrating good recognition performance. However, manually crafted features are insufficient in describing high-level semantic information, and their generalization capabilities are limited. Liu et al. [7] treated hand gesture 3D skeletal joints as pseudo-images, using CNN to extract features of each frame, bypassing skeletal connections, and utilizing multi-scale features in image segmentation for gesture recognition. This achieved good recognition accuracy on a custom gesture database, but the range of recognizable gestures is limited.

Although dynamic gesture recognition algorithms have achieved commendable recognition effects, hand movements are fast and occur within a small range, necessitating further exploration of the spatiotemporal information and dependencies in the action execution process [8]. Additionally, hand gestures are flexible and varied, rich in meaning, and often ambiguous, making it challenging for researchers to design a recognition method applicable to all gestures. The "hand gesture recognition algorithm based on STS-GCN (Space-time synchronization graph convolution network)" utilizes 2D + triangulation to obtain 3D skeletal joint data of the hand. Then, through STS-GCN, it captures the 3D skeletal sequence graph of the hand, thereby tracking and recognizing hand actions.

2. 3D Hand Pose Estimation

2.1 3D Hand Pose Estimation Framework

Hand pose estimation is the process of modeling a person's hand as a collection (for example, using the main joints of fingers and palms) and locating their positions in hand images (2D hand pose estimation) or simulating the positions of hand parts in 3D space [9] (3D hand pose estimation). Learning from 3D hand poses is more effective than merely using image/video features, thus making pose-based action recognition more efficient. The 3D hand pose estimation framework is shown in Figure 1.

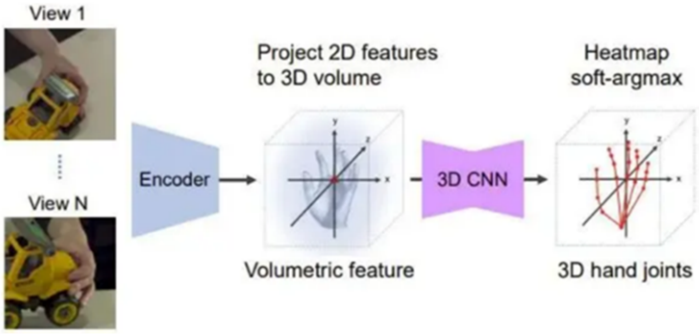

Figure 1. 3D Hand Pose Estimation Framework.

Initially, sampling is performed from videos to obtain input images at a rate of eight RGB images per frame. The Encoder is an autoencoder that extracts 2D keypoint features for each view. Then, the features are projected into a single 3D volume using a 2D

estimation + triangulation method. Finally, the volume features are refined through a 3D-CNN (three-dimensional convolutional network) and output as 3D heat maps. On the heat maps, the soft-argmax operation is employed to obtain the coordinates of the three-dimensional joint points.

2.2 2D Estimation + Triangulation

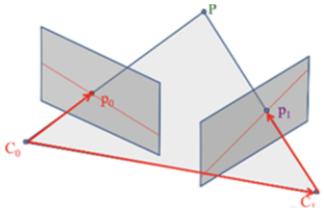

Figure 2. 2D Estimation.

As shown in Figure 2, P is a three-dimensional point in space, C_0 and C_1 are two different observation points, and C_0-C_1-P is an epipolar plane formed by the line, intersecting the left and right planes, placing $C_0 - C_1 - P - p_0 - p_1$ on the same plane. p_0 and p_1 are two-dimensional points on the image plane, which are converted into three-dimensional directional vectors and represented in the $X_0Y_0Z_0$ coordinate system of C_0 as in Equation (1):

$$\overrightarrow{C_0p_0}\,|_{X_0Y_0Z_0} = x_0 = \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix} = K^{-1}q_0 \tag{1}$$

In Equation (1), x_0 is the coordinate of p_0 on the normalized plane, and q_0 is its homogeneous pixel coordinate.

x_1 is the coordinate of p_1 on the C_1 normalized plane, and q_1 is its homogeneous pixel coordinate, with the normalized vector of p_1 in the C_1 coordinate system represented as in Equation (2).

$$\overrightarrow{C_1p_1}\,|_{X_1Y_1Z_1} = x_1 = \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = K^{-1}q_1 \tag{2}$$

The normalized vector of p_1 in the C_0 coordinate system is described as in Equation (3).

$$\overrightarrow{C_1p_1}|_{X_0Y_0Z_0} = R \bullet \overrightarrow{C_1p_1}|_{X_1Y_1Z_1} = Rx_1 \tag{3}$$

In the $X_0Y_0Z_0$ coordinate system, substituting the relationship between normalized coordinates and homogeneous pixel coordinates yields Equation (4):

$$\begin{aligned} &(K^{-1}p_0)^T t^\wedge R(K^{-1}p_1) = 0 \\ &\Rightarrow q_0^T K^{-T} t^\wedge R K^{-1} q_1 = 0 \end{aligned} \tag{4}$$

The essential matrix is defined as in Equation (5):

$$E = t^\wedge R \tag{5}$$

The epipolar constraint simplifies to Equation (6), which succinctly gives the spatial relationship of two matching points, and the 2D estimation problem is to find R, t based on E.

$$x_0^T E x_1 = 0 \tag{6}$$

For any given E, there are two possible t and R corresponding to it, as shown in Equations (7) and (8).

$$t_1 = U \begin{pmatrix} 0 \\ 0 \\ \sigma \end{pmatrix} \qquad R_1 = U R_Z(-\frac{\pi}{2}) V^T \tag{7}$$

$$t_1 = U \begin{pmatrix} 0 \\ 0 \\ -\sigma \end{pmatrix} \qquad R_1 = U R_Z(\frac{\pi}{2}) V^T \tag{8}$$

After obtaining the two-dimensional coordinates of key points in multiple views, three-dimensional information is recovered through triangulation. Triangulation is an algorithm that calculates the three-dimensional spatial coordinates of feature points based on the pixel coordinates of matched feature points in consecutive frames and the camera motion between these frames, R and t, as illustrated in reference [10]. Intuitively, triangulation addresses the problem of estimating the actual pose of an object from two images taken by cameras at known relative positions: obtaining the three-dimensional structure of corresponding points on two-dimensional images through triangulation.

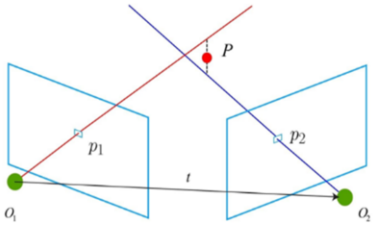

Figure 3. Triangulation.

As shown in Figure 3, the relationship between O_1 and O_2 is known, and P_1 and P_2 are also known. To determine the three-dimensional spatial coordinates of point P, it is necessary to solve the depth of two points, s_1, s_2. According to the definition of epipolar geometry, let x_1, x_2 be the normalized coordinates of the two feature points, then they satisfy Equation (9):

$$s_2 x_2 = s_1 R x_1 + t \tag{9}$$

Left multiplying both sides of Equation (9) by the antisymmetric matrix $x_2^\wedge$ of x_2 yields Equation (10):

$$s_2 x_2^\wedge x_2 = s_1 x_2^\wedge R x_1 + x_2^\wedge t \Rightarrow s_1 x_2^\wedge R x_1 + x_2^\wedge t = 0 \tag{10}$$

Once s_1 is obtained, s_2 can be determined from Equation (9), leading to Equation (11):

$$\begin{cases} s_1 K^{-1} p_1 = P_{O_1} \\ s_2 K^{-1} p_2 = P_{O_2} \end{cases} \tag{11}$$

Where P_{O_1} denotes the three-dimensional coordinates of point P in camera coordinate system O_1, and P_{O_2} denotes the three-dimensional coordinates of point P in camera coordinate system O_2.

3. STS-GCN Hand Gesture Recognition Algorithm

3.1 Graph Convolution Operation

For the 3D skeleton of the hand, it is inherently a natural graph structure. All the joint nodes of the hand skeleton can be considered as vertices of the graph, and the connections

between skeletal joint nodes can be represented by lines between points. The implicit relationships within the joint nodes are obtained using the Graph Convolutional Network (GCN) method [11-12]. Represent skeletal joint nodes as three-dimensional vectors, corresponding to the x、y、z coordinates, with each joint node represented as $v_1 = (x_1, y_1, z_1)$, the set of joint nodes as $V = \{v_i\}_{i=1}^{n}$, and the connections between joint nodes as $W = \{w_i\}_{i=1}^{n}$, where $w = v_i - v_2$. Define the hand skeletal data graph as an undirected graph $G = (V, W)$, where W is the set of edges, and the vertex set V can be represented by the matrix $X = (v_1, v_2 ... v_n), v_i \in \mathbb{R}^c$, $M \in \mathbb{R}^{n \times n}$ represents the adjacency matrix of X, and $N \in \mathbb{R}^{n \times n}$ represents the degree matrix, satisfying $N_{ij} = \sum_i M_{ij}$. Therefore, the graph convolution operation is defined as Equation (12):

$$g_\theta \otimes X = (I + N^{\frac{1}{2}} M N^{\frac{1}{2}}) X^T A \tag{12}$$

In the equation, $\otimes$ represents the graph convolution operation, g is the filter, which is a key parameter of A, $A \in \mathbb{R}^{c \times c}$. Substituting $\hat{M} = M + I$ and $\hat{N} = \sum_i \hat{M}_{ij}$ into Equation (12) yields Equation (13):

$$g_\theta \otimes X = \hat{N}^{\frac{1}{2}} \hat{M} \hat{N} - \frac{1}{2} X^T A \tag{13}$$

Hammond D K et al. proved that the graph convolution operation of g_θ can be approximated by a R-th order Chebyshev polynomial as shown in Equation (14) [13]:

$$F_{out} = \sum_{r=0}^{R} \theta_r^{'} T_r(\hat{L}) F_{in} \tag{14}$$

Where $\theta_r^{'}$ represents Chebyshev coefficients, $L = N^{\frac{1}{2}} M N$, $T_0 = 1$, $T_1 = \hat{L}$. The recursive formula of the Chebyshev polynomial can be defined as Equation (15).

$$T_r(\hat{L}) = 2\hat{L} T_{r-1}(\hat{L}) - T_{r-2}(\hat{L}) \tag{15}$$

3.2 Space-Time Synchronization Graph Convolutional Network

The Space-Time Synchronization Graph Convolutional Network (STS-GCN) with combined attention designed in this paper is illustrated in Figure 4. The network captures

the 3D skeletal sequence graph of the hand for tracking and recognizing hand actions. The space-time synchronization graph convolution part consists of spatial and temporal branches. The spatial branch is responsible for extracting multi-scale spatial features, mainly comprising one GCN and two ResNet structures. The temporal branch focuses on extracting finger position and motion features, primarily consisting of one GCN, one 2D-ResNet, and one ResNet structure, where the 2D-ResNet more effectively extracts temporal features.

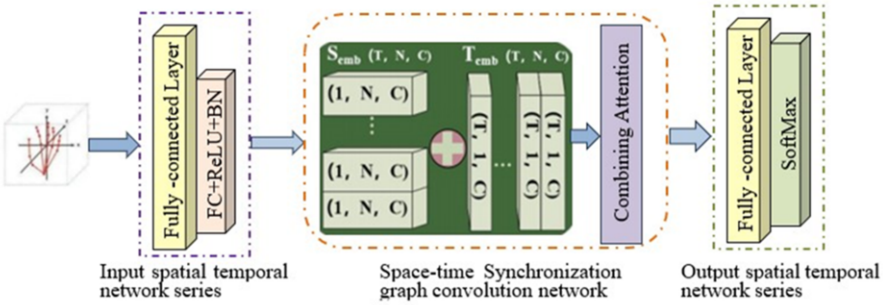

Figure 4. Space-Time Synchronization Graph Convolutional Network Framework.

Unlike typical graph convolutional networks, which treat each frame's skeleton as a separate graph, this paper separates the spatial and temporal dimensions of skeletal data. It encodes the spatial dimension according to the sequence of joints and encodes the same joints in the temporal dimension according to their chronological order. This encoding generates spatial and temporal embedding matrices, which are added to the space-time network sequence, enhancing the network's ability to model the spatiotemporal correlations in the data. This enables the network to analyze and recognize gestures more precisely.

For the input skeletal sequence $X \in R^{T\times N\times C}$, where T is the number of frames in the sample sequence, N is the number of nodes in a frame, and C is the number of feature dimensions. The spatial position vector of the data sequence is set to $S_{pos} \in [1,2,...N]$, and the temporal position vector is $T_{pos} \in [1,2,...T]$. The spatial and temporal position vectors are encoded using sine and cosine functions, as shown in Equations (16) and (17):

$$f(pos,i) = pos / 1000^{\frac{2i}{C}}, i = 1,2,...C \tag{16}$$

$$PE(pos,i) = \begin{cases} \sin(f(pos,i)) & \text{if i\%2=0} \\ \cos(f(pos,i)) & \text{if i\%2=1} \end{cases} \tag{17}$$

Where pos represents the position of the element, i is the dimension of the position encoding vectors, and % is the modulo operation. According to Equations (16)

and (17), the time information matrix $T \in R^{T\times C}$ and the spatial information matrix $S \in R^{N\times C}$ of the skeletal sequence can be obtained, as shown in Equations (18) and (19):

$$T = \begin{cases} \sin(f(T_{pos}, i)) & \text{if} \quad \text{i\%2=0} \\ \cos(f(T_{pos}, i)) & \text{if} \quad \text{i\%2=1} \end{cases} \tag{18}$$

$$S = \begin{cases} \sin(f(S_{pos}, i)) & \text{if} \quad \text{i\%2=0} \\ \cos(f(S_{pos}, i)) & \text{if} \quad \text{i\%2=1} \end{cases} \tag{19}$$

Since the tensor matrices T and S have different sizes, tensor expansion is required to match their dimensions with X before adding these two matrices.

3.3 Adjacency Matrix

The relationships between nodes are constrained by spatial distance and temporal sequence. Typically, these constraints are integrated into an adjacency matrix. Define M as the adjacency matrix of a graph composed of three consecutive time steps. This matrix is used to adjust the weight coefficient matrix S of the nodes, allowing S to converge to a reasonable value. For the sample sequence $X_C^K \in R^{3N\times C}$, the adjacency matrix $M \in R^{3N\times 3N}$ is defined as in Equation (20):

$$M_{t_p \to t_q} = \begin{cases} 1 & \text{if} \quad \mathrm{v}_i^k \to v_j^k, i, j = 1, 2, \ldots 3N \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Where $v_i^k \to v_j^k$ represents the connection from node v_i^k to node v_j^k, $p, q = 1, 2, 3$ and $0 \le |p - q| \le 1$. t is the time step in the space-time graph, and $0 < t \le 3$. The structure of the adjacency matrix is shown in Figure 5, where $M_{t_p \to t_q}$ represents the connection from time step t_p to time step t_q. The diagonals represent the adjacency matrices of the spatial graph at three time points, while the other four represent connections across time steps.

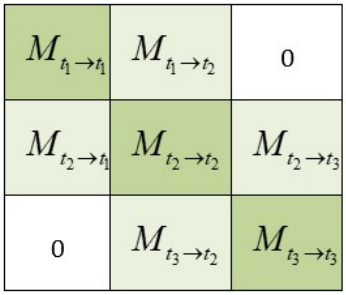

Figure 5. Adjacency Matrix M.

3.4 Combined Attention Mechanism

In the attention module, the combined attention mechanism simultaneously utilizes spatial, temporal, and channel aspects. Combined attention is a linear combination of individual self-attentions and is defined as in Equations (21) and (22):

$$H_m = Attention(QW_m^Q, KW_m^k, PW_m^P) \quad , \quad m = 1,2,...M \tag{21}$$

$$MultiH(Q,K,P) = Concat(H_1, H_2...H_M)W^o \tag{22}$$

Where H_m represents the output of the m attentions, Q, K, P is the query-key-value matrix, and W_m^Q, W_m^K, W_m^P is the weight matrix corresponding to the three linear fully connected layers of the m -th attention. Since the sequence in the space-time graph can be defined as Equation (23):

$$X_c^k = \{v_j^k = v_{t,i} \mid t = k, k+1, k+2, i = 1,2...N, k = 1,2,...T-2, j = 1,2,...3N\} \in R^{3N \times C} \tag{23}$$

Where v_j^k represents the j -th node in the k -th space-time graph, and $v_{t,i}$ represents the i -th skeletal node in the t -th frame. The query-key-value matrix for the m -th attention of this sequence is represented as in Equations (24), (25), and (26):

$$Q^m = QW_m^Q = X_c^k W_Q W_m^Q \in R^{3N \times d} \tag{24}$$

$$K^m = KW_m^K = X_c^k W_k W_m^k \in R^{3N \times d} \tag{25}$$

$$P^m = PW_m^K = X_c^k W_P W_m^p \in R^{3N \times d} \tag{26}$$

Where $Q^m = [q_1^m, q_2^m, ...q_{3N}^m]^T$, $K^m = [k_1^m, k_2^m, ...k_{3N}^m]^T$, $P^m = [p_1^m, p_2^m, ...p_{3N}^m]^T$ and $q_j^m, k_j^m, p_j^m \ \ (j = 1,2,...3N, m = 1,2,...M)$ are the query-key-value vectors corresponding to the m -th node feature of the i -th attention, with vector dimensions d . W_m^Q, W_m^K, W_m^P are the weight matrices of the fully connected layer for the m -th attention. The overall output of the sample sequence X can then be represented as in Equation (27):

$$Y^m = Attention(Q^m, K^m, P^m) = soft\max(\frac{Q^m (K^m)^T}{\sqrt{d}})P^m \in R^{3N \times d} \quad (27)$$

The output of the combined attention is given by Equation (28).

$$MultiH(Q\ , K,\ \ P) = Concat(Y^1, Y^2, ...Y^M) \in R^{M \times 3N \times d} \quad (28)$$

4. Experimental Results and Analysis

4.1 Experimental Environment Setup

The hardware requirements for the experiment include a CPU: AMD Ryzen 5 3600 @4.2 GHz, GPU: NVIDIA GeForce RTX 3060, memory: 32 GB, and an ASUS TUF-GeForce RTX3080TI-O12G-GAMING graphics card with 12 GB of video memory. Software development tools used include: Python3 downloaded and installed from the official website; Python editor PyCharm installed; Anaconda installed; pandas library installed; OpenCV library installed; CUDA downloaded and installed; cuDNN downloaded and installed; and PyTorch installed.

4.2 Dataset

To validate the effectiveness of the algorithm proposed in this paper, experiments were conducted on the SHREC2017 public dataset, comparing it with some representative gesture recognition methods. The dataset includes gestures classified into two categories: Coarse gestures (such as sliding and other hand movements) and Fine gestures (hand shapes). In gesture recognition algorithms, it is necessary to consider the subtle differences between these gestures. Each gesture was performed by 28 participants in 1 to 10 instances using two methods, producing 2800 sequences with lengths ranging from 20 to 50 frames. All participants were right-handed, and sequences were tagged according to their gestures, number of fingers used, performer, and experiment. In this experiment, 14 gesture labels were used to tag sequences for evaluating the algorithm. The structure directory of the dataset is shown in Figure 6.

Figure 6. Dataset Structure Directory.

4.3 Evaluation Metrics

For hand gesture recognition, the mean Average Precision (mAP) and Cumulative Matching Characteristic (CMC) are used on different datasets to assess the overall average state of recognition and the accuracy of the recognition results. mAP calculates the average precision (AP) value for all samples to be recognized, where AP is the result of precision and recall. The formulas for calculating precision and recall are shown in Equations (29) and (30):

$$precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{29}$$

$$recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{30}$$

The mAP value is the average of precision and recall calculated for all query samples. The CMC/Rank(K) metric reflects whether there is a correct match in the top K samples of the gallery results in retrieval. If there is a match, the value is 1; otherwise, it is 0, as defined in Equation (31):

$$CMC / Rank(K) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}1, The\ first\ K\ retrieval\ results\ have\ correct\ samples\\ 0, The\ first\ K\ retrieval\ results\ do\ not\ have\ correct\end{cases} \tag{31}$$

Where N is the total number of samples, and K is the evaluation metric parameter.

4.4 Experimental Results and Analysis

By adjusting relevant parameters and selecting a better-performing group, a comparison is made with ST-GCN, SIT-GCN, HiVideoDarwin, CNN for Skeleton, Two-stream RNN, etc. The results are shown in Table 1. It can be observed that STS-GCN improves recognition accuracy by approximately 1.8% compared to SIT-GCN and about 6.5% compared to ST-GCN. In terms of time consumption, STS-GCN also uses nearly 4 seconds less than the least time-consuming HiVideoDarwin. Overall, STS-GCN demonstrates better gesture recognition accuracy and speed.

Table 1. Accuracy Comparison of STS-GCN with Other Networks

Network model	Precision (%)	Time consumption/s
ST-GCN[14]	86.67	873.1
SIT-GCN [15]	91.38	895.5
HiVideoDarwin[16]	74.92	859.4
CNN for Skeleton[17]	91.23	910.2
Two-stream RNN[18]	91.79	879.7
3s_net_TTM[19]	92.11	863.8
STS-GCN	93.15	855.3

5. Conclusion

This paper presents a Space-Time Synchronization Graph Convolutional Network (STS-GCN) with a combined attention mechanism. The network input is 3D skeletal joint data, where the spatial and temporal dimensions of the skeletal data are separated. The joint nodes are encoded using sine and cosine functions for spatial and temporal position vectors, resulting in spatial and temporal embedding matrices. These matrices are synchronously added to the space-time network sequence, enhancing the network's capability to model spatiotemporal correlations in data, thereby enabling more accurate analysis and recognition of gestures. Compared to other algorithms, while the STS-GCN algorithm achieves commendable recognition results, there is still significant room for improvement in recognition accuracy and time consumption. Future work will continue to research optimization strategies for this algorithm.

Acknowledgement

Funded by the 2023 Gansu Province Higher Education Innovative Ability Improvement Project; Project Number: 2023A-163.

Funded by the 2023 Gansu Province Innovation and Entrepreneurship Training Plan Project; Project Number: DC202302-35.

References

[1] Withana A, Kaluarachchi T, Singhabahu C, et al. WaveSense: Low power voxeltracking technique for resource limited devices. Proceedings of the Augmented Humans International Conference. 2020: 1-7.

[2] QIAN W, WANG G Z, LI G P. Improved robust algorithm for real-tiem traffic light detection with YOLOv5. Computer Science and Exploration, 2022, 16(1):231-241.

[3] Chen L, Lin S Y, Xie Y, et al. DGGAN: Depth-image guided generative adversarial networks for disentangling rgb and depth images in 3d hand pose estimation. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020: 411-419.
[4] Zhou Y, Habermann M, Xu W, et al. Monocular real-time hand shape and motion capture using multi-modal data. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 5346-5355.
[5] Lee M, Bae J. Deep Learning Based Real-Time Recognition of Dynamic Finger Gestures Using a Data Glove. IEEE Access, 2020, 8(8):219923-219933.
[6] SHI L, ZHANG Y, CHENG J, et al. Decoupled spatial-temporal attention network for skeleton based action recongnition. Computer Vision-ACCV 2020, 2020: 38-53.
[7] Liu J, Liu Y, Wang Y, et al. Decoupled Representation Learning for Skeleton-Based Gesture Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, 5750-5759.
[8] XIONG X, WU H, MIN W, et al. Traffic police gesture recognition based on gesture skeleton extractor and multichannel dilated graph convolution network. Electronics, 2021, 10(5):551.
[9] LIAO Y, XIONG P, MIN W,et al. Dynamic sign language recognition based on video sequence with BLST M-3D residual networks. IEEE Access,, 2019: 38044-38054.
[10] Hammond D K, Vandergheynst P, and Gribonval R. Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 2011, 30(2): 129-150.
[11] GUO Y L, WANG H Y, HU Q Y, et al. Deep learning for 3D point clouds A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(12):4338-4364.
[12] Zhang Yifan, Cao Congqi, Cheng Jian, et al. EgoGesture: a new dataset and benchmark for egocentric hand gesture recognition. IEEE Transactions on Multimedia, 2018, 20(5): 1038-1050.
[13] Köpüklü, O, Gunduz A., Kose N., et al. Real-time hand gesture detection and classification using convolutional neural networks// 2019 14th IEEE International Conference on Automatic Face and Gesture Recognition. Lille, France: IEEE Press, 2019: 1-8.
[14] Chen L, Lin S Y, Xie Y, et al. Mvhm: A large-scale multi-view hand mesh benchmark for accurate 3d hand pose estimation. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021: 836-845.
[15] Gomez-Donoso F, Orts-Escolano S, Cazorla M. Large-scale multiview 3d hand pose dataset. Image and Vision Computing, 2019, 81(1): 25-33.
[16] Wang H, Wei W, Liang W. Hierarchical motion evolution for action recognition // Proc of the 3rd IAPR Asian Conference on Pattern Recognition,2016:574-578.
[17] DuY, Fu Y, Wang L. Skeleton based action recognition with convolutional neural network // Proc of the 3rd IAPR Asian Conference on Pattern Recognition,2016:579-583.
[18] WangH, WangL. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks // Proc of 2017 IEEE Conference on Computer Vision and Pattern Recognition,2017:3633-3642.
[19] Li C, Zhang X, Liao L, etal. Skeleton-ased gesture recognition using several fully connected layers with path signature features and temporal transformer module.arXiv:1811, 07081, 2018.