Received 26 April 2016; | Revised 6 June 2016; | Accepted 6 June 2016

DOI: xxx/xxxx

ARTICLE TYPE

Online Human Action Recognition with Spatial and Temporal Skeleton Features Using a Distributed Camera Network

Guoliang Liu*[1] | Qinghui Zhang[1] | Yichao Cao[1] | Guohui Tian[1] | Ze Ji[2]

[1]School of Control Science and Engineering, Shandong University, Jinan, China

[2]School of Engineering, Cardiff University, Cardiff, UK

Correspondence

*Guoliang Liu, School of Control Science and Engineering, Shandong University, Jinan, China. Email: liuguoliang@sdu.edu.cn

Funding Information

This research was supported by the National Key Research and Development Program of China, Grant/Award Number: 2018YFB1306500; National Natural Science Foundation of China, Grant/Award Number: 91748115; National Natural Science Foundation of China, Grant/Award Number: 61603213; Young Scholars Program of Shandong University, Grant/Award Number: 2018WLJH71; Fundamental Research Funds of Shandong University; Taishan Scholars Program of Shandong Province.

Abstract

Online action recognition is an important task for human centered intelligent services. However, it remains a highly challenging problem due to the high varieties and uncertainties of spatial and temporal scales of human actions. In this paper, the following core ideas are proposed to deal with the online action recognition problem. First, we combine spatial and temporal skeleton features to represent human actions, which include not only geometrical features, but also multi-scale motion features, such that both spatial and temporal information of the actions are covered. We use an efficient 1D Convolutional Neural Network (CNN) to fuse spatial and temporal features and train them for action recognition. Second, we propose a group sampling method to combine the previous action frames and current action frames, which are based on the hypothesis that the neighbouring frames are largely redundant, and the sampling mechanism ensures that the long-term contextual information is also considered. Third, the skeletons from multi-view cameras are fused in a distributed manner, which can improve the human pose accuracy in the case of occlusions. Finally, we propose a Restful style based client-server service architecture to deploy the proposed online action recognition module on the remote server as a public service, such that camera networks for online action recognition can benefit from this architecture due to the limited onboard computational resources. We evaluated our model on the datasets of JHMDB and UT-Kinect, which achieved highly promising accuracy levels of 80.1% and 96.9%, respectively. Our online experiments show that our memory group sampling mechanism is far superior to the traditional sliding window.

KEYWORDS:

online action recognition, kinetic skeleton feature, 1D CNN, camera network

1 | INTRODUCTION

Online action recognition plays an important role in many applications, such as elderly care, medical rehabilitation, security surveillance, and human-robot interaction and collaboration. There are many sensors can be used to capture human actions, e.g., RGB camera, RGBD camera, IMU (Inertial Measurement Unit), and 3D laser scanner. RGB cameras are the most common type of sensor for analyzing human actions [1 2 3], because of its many advantages, such as the characteristics of images that provide rich and naturally interpretable information for humans and, on the other hand, its acceptable prices, compared to other costly sensors,

such as IMU and laser scanners. However, 2D images are projected from the 3D world, such that human action recognition can be affected by the view point. In contrast, recently RGBD cameras, e.g. Microsoft Kinect, Intel realsense, and Asus Xtion, are becoming popular. They can directly output depth information of the environment for 3D human skeleton pose detection[4 5] that is invariant to the view direction. In addition, RGBD cameras' advantages of low cost, real time 3D reconstruction capability and easy-to-use features drive its popularity in the field of human pose estimation and action recognition.

In recent years, deep learning techniques[6 7 8 9 10] are prevalent for target recognition tasks. The skeleton of a human is one of the most commonly used models for representing human poses that can be captured using RGBD sensors. A human skeleton include a number of spatially connected bone joints. Each joint has a coordinate in the RGB image or depth image, such that the skeleton can be represented by a compact vector of values with a fixed length, which can save computational cost to analyze human poses and actions. Currently, most of the human action recognition methods using skeleton data are offline processing, which use recorded video clips of fixed lengths as input[11]. Some attempt to predict the start frame and end frame of an action in the sequential images[12 13]. On the other hand, online action recognition is still an onging open problem, which is difficult to be solved, since only previous frames of the current time are available and the start point of an action is unknown[14 15]. Furthermore, the real time requirement of online action recognition is the other challenging issue, due to the expensive computational cost of the recent deep learning algorithms.

In this paper, we focus on online action recognition using 3D skeleton sequences derived from RGBD sensors. The novelties of our ideas are as follows:

(1) We introduce a group of kinetic skeleton features that can capture both of the spatial and temporal features of the human action, which includes joint collection distance features, multi-scale motion features and geometrical features. We employ a fast, small and efficient neural network to combine these kinetic skeleton features, which can return a competitive performance for action recognition.

(2) We propose a group sampling mechanism to handle the uncertainty of the temporal scale of the actions, such that long-term contextual information can be considered for action recognition.

(3) A distributed multi-view information fusion method is used for human pose fusion, such that each camera can have more accurate skeleton data by fusing information from neighbour camera nodes.

(4) A Restful style client-server architecture for distributed action recognition of camera networks is proposed, such that all camera nodes as clients can send their local kinetic skeleton features to the server which loads the proposed online action recognition module in advance, and return the recognition results to the clients.

The rest parts of the paper are structured as follows. We first discuss related works in the field of action recognition using skeleton data in Section 2, and then introduce our kinetic skeleton features, group sampling, 1D CNN neural network, distributed information fusion and client-server architecture for online action recognition in Section 3. The demonstrated experiments on the public datasets and our laboratory datasets can be found in Section 4. Finally, the paper is concluded in Section 5.

2 | RELATED WORKS

Most of the current action recognition methods are offline, where the skeleton sequence data for processing are segmented for each action in advance, such that we can easily label these data and train the learning methods. In addition, it is convenient to use segmented data for numerical accuracy analysis. Furthermore, the computation cost of the algorithm is not considered a key problem due to the offline processing requirement. H. Wang and L. wang[16] proposed a new two-stream Recurrent Neural Network (RNN) architecture to model temporal dynamics and spatial structures of skeletons for action recognition. In order to improve the generalization ability of the model, they further developed a data expansion technology based on 3D transformation, including rotation and scaling transformation. Finally, the recognition accuracy of the algorithm is verified on two public datasets. Liu et al.[17] proposed a Long short-term memory (LSTM) model-based action recognition method, which simultaneously extracts the action-related information in the temporal and spacial domains. Wang et al.[18] proposed the Joint Trajectory Maps (JTM), which represents the spatial configuration and dynamics of joint trajectories as three texture images through color encoding. Then, they use CNN for action recognition. The original skeleton data can be affected by the view direction, so more advanced geometrical features can be extracted from original joints. For instance, Thien Huynh-The et al.[19] proposed a novel skeleton-to-image encoding technique to exploit pose features for a more robust action representation. Zhang et al.[20] designed eight geometric features to represent raw skeleton data, including joint-to-joint distances, joint-line distances, joint-to-plane distances, etc. These hand-crafted features are used as the input of the LSTM network for action recognition. They finally proved that

properly defining hand-crafted features for a basic model can be superior. Yasin et al.[21] presented a novel method, which relies on keyframes extracted from action sequences. The extracted keyframes provide information that is free from redundancy, but carries the most relevant details about the action that exists in the motion.

Online action recognition refers to the capability of predicting the category of ongoing action based on the observed data up to the present. This means that we need to predict the category of action before they are completely executed. There are fewer works about online action recognition than offline algorithms. Most of current online algorithms deal with RGB videos. Jiang et al.[22] proposed a novel Dual 3D convolutional Network (D3DNet) with two complementary lightweight branches to learn spatio-temporal models for video-based human action recognition. The method proposed by Geest et al.[23] takes an RGB video stream as input, and outputs the class of the action in real time. You et al.[24] proposed Action4DNet to generate 4D volumes of the environment, track each person in the volume and infer the actions of each subject, for situations with multiple people. One of the challenging problems of online action recognition is the unknown starting point of each action of interest. A sliding window with fixed scale proposed by Zanfir and Mihai[25] is used to extract the frames for recognition. The sliding window is simple and easy for use, but its limitation is the fixed scale of the sliding window, since the temporal length of the action is unknown and different due to the variety of the action. In addition, the sliding window method can lose long-term context information, such that it has very low accuracy for such situations. To solve these problems, we propose a group sampling mechanism that balances the data that are far away from the present frame, and those that are relatively close.

Data fusion based on sensor network is an effective solution to viewpoint variation and occlusion problems in human action recognition. Aggarwal et al.[26] point out that multiple cameras can not only extend range of perception, but also solve problems caused by occlusions by other targets. However, due to the limitations of network bandwidths, it is not feasible to transmit the mass of video data between network nodes. Therefore, it is important to figure out what sensor information should be handled locally at the sensor node and what information should be shared and merged with other nodes. Compared to centralized networks, distributed networks require less traffic and energy consumption, which aroused extensive interest in the study. B. Song et al.[27] propose using the Kalman consensus filter (KCF) to track multiple targets, and fuse the similarity scores of neighboring cameras for action recognition using 2D images. As shown in[28,29], the KCF cannot handle naive node and information redundancy problems. Therefore, A.T. Kamal et al.[28,30] introduce an information weighted consensus filter (IWCF)-based distributed human tracking method, which solves the naivety and redundancy problem by giving less weight to the prior information, when new information contribution is fused, since the redundancy information is present only in the prior information[29]. Furthermore, Liu et al.[31] improve the original IWCF by using Metropolis weighting during consensus, which shows improved convergence rate. The nonlinear versions and square-root extensions of IWCF can be founded in[32,33,30].

3 | PROPOSED METHOD

In this section, we introduce our ideas to extract advanced features from original skeleton sequences for handling the problems caused by the view changes, and discuss the idea of group sampling to extract the frames from previous frames for handling the problem of unknown starting point of corresponding actions. Finally, a CNN is used for action recognition. The overall flowchart of the proposed method can be seen in Fig. 1 .

3.1 | Advanced kinetic skeleton feature representation

To fully describe the actions of a human, we use not only the advanced spatial geometrical information of the human joints, but also the temporal motion features.

The joint collection distances (JCD) features, which are location viewpoint invariant features, are first proposed in[34]. If each human skeleton has N joints with corresponding Cartesian coordinates $g_i^k = (x, y, z)$ for the k_{th} frame and the i_{th} joint, the joint collection distances can be calculated as follows:

$$F^k = \begin{bmatrix} \left\| g_2^k g_1^k \right\| & & \\ \vdots & \ddots & \\ \vdots & \cdots & \ddots \\ \left\| g_N^k g_1^k \right\| & \cdots & \cdots & \left\| g_N^k g_{N-1}^k \right\| \end{bmatrix} \tag{1}$$

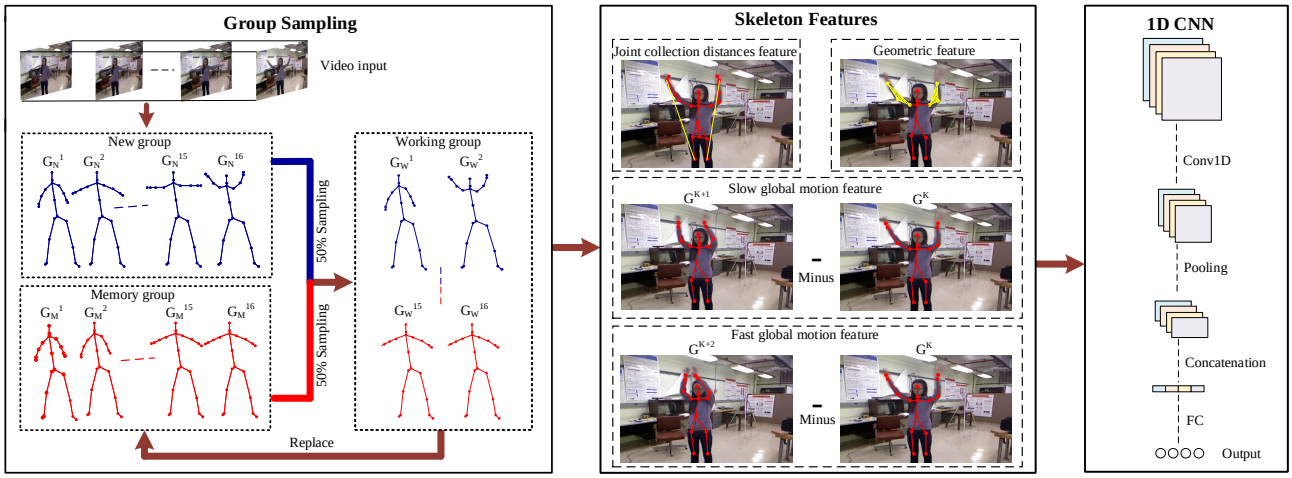


FIGURE 1 The overall flowchart of the proposed online human action recognition method using group sampling with advanced geometry and motion skeleton features, where 1) a new group means the new skeleton sequences from the camera, 2) memory group is used for storing history skeleton sequence, and 3) a working group is a skeleton sequence sampled from the new group and memory group. G_N^i denotes the i_{th} frame of the new group, G_M^j denotes the j_{th} frame of the memory group. The skeleton features used in the paper include joint collection distances, geometric feature and global motion feature, where the red points are detected joints, the yellow lines are the examples of extracted features, minus means the Euclidean distance between two skeletons. A 1D CNN is used as the classifier for action recognition, which concatenates all features and output the recognition result.

where $\left\| g_i^k g_j^k \right\|$ represents the Euclidean distance between joint g_i^k and joint g_j^k. Since F^k is a symmetry matrix, we only use the lower triangular matrix as JCD features.

In addition to the JCD features, we also explore the feature information from joint orientations, joint-line distances, line-line angles, joint-plane distances, line-plane angles, plane-plane angles from original skeleton joints according the work presented in[20]. To reduce the information redundancy, we select these lines and planes according to the following rules[20]:

- Lines: $x_{g_1g_2}$ is a line connected by joint g_1 and g_2, which satisfy one of the following constraints: (1) g_1 and g_2 are directly adjacent in the human structure. (2) One of g_1 and g_2 is the end joint (like head joint, left or right hand joint, left or right foot joint), and the other is the joint separated by a joint in the human structure. (3) g_1 and g_2 are both end joints.
- Planes: $P_{g_1g_2g_3}$ is a plane determined by a triangle formed by g_1, g_2 and g_3. Only five planes that correspond to body, two arms and two legs are considered.

According to these selected lines and planes, six types of geometric features are chosen as shown in Table 1 and Fig. 2 , where gg_o is the direction from joint g_1 to g_2, gx_d is the distance from joint g to line $x_{g_1g_2}$, xx_a is the angle between line $x_{g_1g_2}$ and $x_{g_3g_4}$, gP_d is the distance from joint g to plane $P_{g_1g_2g_3}$, xP_a is the angle between line $x_{g_1g_2}$ and plane $P_{g_3g_4g_5}$ normal vector, PP_a is the angle between plane $P_{g_1g_2g_3}$ normal vector and plane $P_{g_4g_5g_6}$ normal vector. The calculation methods of above features are shown in Table 1 , where *unit* is the unit vector, $S_{\triangle gg_1g_2}$ is the area of triangle $\triangle gg_1g_2$, *arccos* is the inverse trigonometrical function, $\left\| \overrightarrow{g_1g_2} \right\|$ is the Euclidean distance between joint g_1 to g_2, $\odot$ represents dot product, and $\otimes$ denotes cross product of two vectors. Here, we do not use repetitive features caused by symmetry of the human body.

The geometrical features only depict the spatial relations of the human skeleton joints, whereas the temporal information is missing. However, temporal information is important for human action recognition to represent a skeleton sequence. Therefore, we further employ the global motion features by differentiating the spatial positions G^k of human skeleton joints between the k_{th} frame and the $k_{th} + s$ frame, where s is the temporal scale, k refers to the frame, G^k represents the set of joint points of

TABLE 1 Geometric feature calculation methods and feature description

Feature	Symbol	Calculation Methods	Description
Joint Orientation	gg_o	$gg\_o(g_1, g_2)= unit(\overrightarrow{g_1 g_2})$	Direction from joint g_1 to g_2
Joint Line Distance	gx_d	$gx\_d(g, x_{g_1 g_2}) = 2S_{\triangle g g_1 g_2} / \left\|\overrightarrow{g_1 g_2}\right\|$	Distance from joint g to line $x_{g_1 g_2}$
Line Line Angle	xx_a	$xx\_a(x_{g_1 g_2}, x_{g_3 g_4}) = \arccos(gg\_o(g_1, g_2)^T \bigodot gg\_o(g_3, g_4)$	Angle between line $x_{g_1 g_2}$ and $x_{g_3 g_4}$
Joint Plane Distance	gP_d	$gP\_d(g, P_{g_1 g_2 g_3}) = (g - g_1) \bigodot gg\_o(g_1, g_2) \bigotimes gg\_o(g_3, g_4)$	Distance from joint g to plane $P_{g_1 g_2 g_3}$
Line Plane Angle	xP_a	$xP\_a(x_{g_1 g_2}, P_{g_3 g_4 g_5}) = \arccos(gg\_o(g_1, g_2)) \bigodot gg\_o(g_3, g_4) \bigotimes gg\_o(g_3, g_5)$	Angle between line $x_{g_1 g_2}$ and plane $P_{g_3 g_4 g_5}$ normal vector
Plane Plane Angle	PP_a	$PP\_a(P_{g_1 g_2 g_3}, P_{g_4 g_5 g_6}) = \arccos(gg\_o(g_1, g_2)) \bigotimes gg\_o(g_1, g_3) \bigodot gg\_o(g_3, g_4) \bigotimes gg\_o(g_3, g_5)$	Angle between plane $P_{g_1 g_2 g_3}$ normal vector and plane $P_{g_4 g_5 g_6}$ normal vector

the human in the k_{th} frame. Here we use two scales for capturing the fast motion Y^k_{fast} and slow motion Y^k_{slow} respectively, i.e., $s = 1, 2$. The motion features are calculated as

$$Y^k_{slow} = G^{k+1} - G^k, k \in \{1, 2, \cdots, K-1\} \tag{2}$$

$$Y^k_{fast} = G^{k+2} - G^k, k \in \{1, 2, \cdots, K-2\} \tag{3}$$

3.2 | Group sampling mechanism

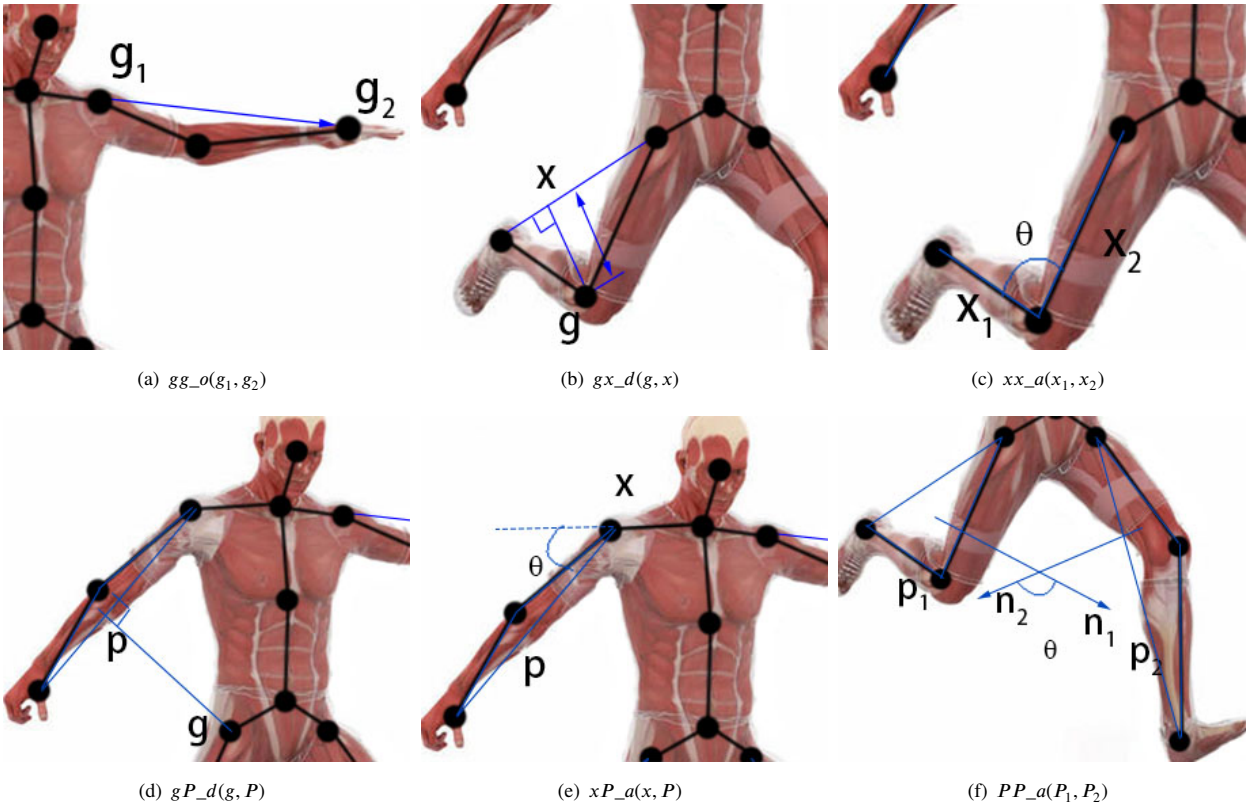
For online action recognition, the unknown start and end time of an action is a challenging problem compared to offline action recognition, which uses segmented action sequences. The sliding window method is the popular method for traditional online action recognition[25], which has a fixed window size and can lose long-term context information. We here propose a group sampling mechanism to balance the information that include a window of varying time duration from the current frame. A fixed number of frames are chosen as the input of the action classifier. The sampling function is defined as:

$$T = \lfloor \frac{j}{N} \rfloor - 1, \mathrm{j} \geq \mathrm{N} \tag{4}$$

$$W^T = \begin{cases} Q^0 & \text{if T} = 0 \\ \left\{0.5^T Q^0\right\} \bigcup_{t=1}^{T} \left\{0.5^{T-t+1} Q^t\right\} & \text{if T} > 0 \end{cases} \tag{5}$$

where j is the number of frames currently received, N is the number of sampling frames required for the action classifier input, T is the number of corresponding sampling step, when the j_{th} frame is received, and it starts with 0, W^T is a working group including skeleton frames obtained in the T_{th} sample for the current classification, Q^t is a new queue group that stores N consecutive frames of data before the sampling step t in the data stream, and 0.5 means 50% sampling of the data. We assign 16 to N. In addition, we define the $\bigcup_{t=1}^{T}$ operator as

$$\{A\} \bigcup_{x=1}^{X} \{B(x)\} = \{A\} \bigcup \{B(1)\} \bigcup \{B(2)\} \bigcup \cdots \cdots \bigcup \{B(X)\} \tag{6}$$



(a) $gg\_o(g_1, g_2)$ (b) $gx\_d(g, x)$ (c) $xx\_a(x_1, x_2)$

(d) $gP\_d(g, P)$ (e) $xP\_a(x, P)$ (f) $PP\_a(P_1, P_2)$

FIGURE 2 Six advanced spatial geometric features used in this paper.

where x is a variable, X is a constant, A is a fixed set, and $B(x)$ is a set relative to the variable x.

At the beginning of the skeleton sequence ($T = 0$), we use all N frames of data received in the current data stream:

$$W^0 = Q^0 \tag{7}$$

For the third sampling ($T = 2$), the sampling equation is shown as:

$$W^2 = \{0.5^2 Q^0\} \bigcup \{0.5^2 Q^1\} \bigcup \{0.5^1 Q^2\} \tag{8}$$

where W^2 consists of three parts, including 25% of Q^0, 25% of Q^1 and 50% of Q^2. It shows that the latest frames have higher probabilities to be chosen than older frames, whereas long-term contextual information is also considered. The specific steps of the memory group sampling algorithm are shown in Algorithm 1.

To store these sampled frames, we use a memory group M, which will be replaced by the working group W after each sampling step, such that W^T can also be espressed by M as

$$W^T = \begin{cases} Q^0 & \text{if T} = 0 \\ \{0.5Q^t\} \bigcup \{0.5M\} & \text{if T} > 0 \end{cases} \tag{9}$$

The update of memory group M ensures that the image frame at a closer time has a greater sampling density, which means it has greater weight to be chosen as the input candidate of the classifier. The sampled frames are fed into the classifier at each time step for real-time prediction. The recognition result at the current and previous moments are averaged to obtain the final prediction result.

We summarize the memory group sampling method in Algorithm 1. The new queue group Q, memory group M and work group W are initially empty. The new queue group receives and buffers joint data frames until it reaches N frames. The work group W is constructed by sampling 50% from Q and 50% from M. We then feed the working group to the classifier to obtain the action recognition result p. The average of the current result p and previous result p_a is the output result.

Algorithm 1 Online Action Recognition Based on Group Sampling Mechanism

Require: Live stream L of human skeleton, trained classifier C, the size of sampling group N.

Initialize the empty new queue group Q to save the sampled N frames
Initialize the memory group M and work group W
Initialize output average recognition probability p_a
while New frame available from L **do**
    Add frame f_i to queue Q
    if $i\%N$ **then**
        W = sample 50% Q and sample 50% M
        Feed W to the classifier C to get recognition probability p
        p_a = average p_a and p
        $M = W$
        empty queue Q
        Output action recognition probability p_a
    end if
end while

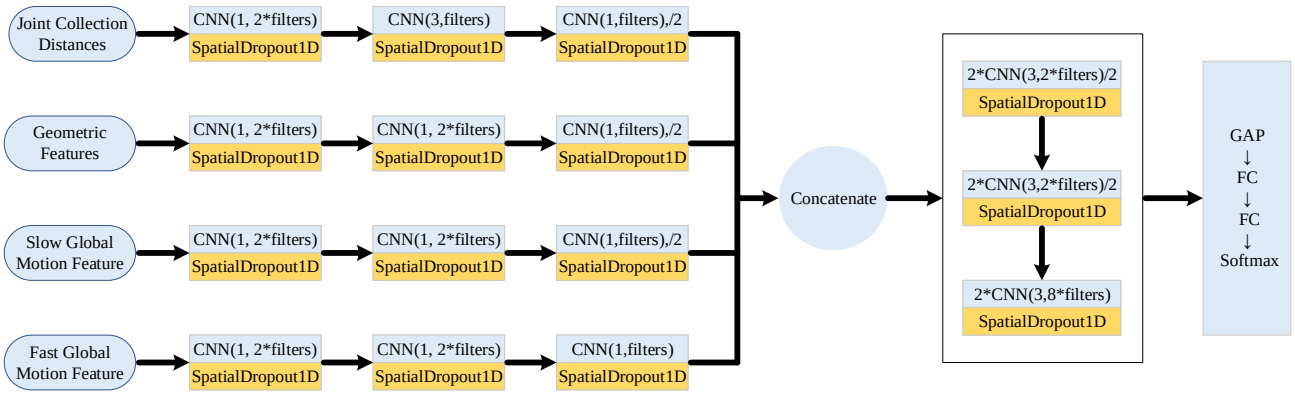


FIGURE 3 The network architecture of 1D CNN for action recognition, where "2*CNN(3, 2×filters), /2" indicates two 1D CNN layers (kernel size = 3, channels = 2×filters) and a Maxpooling layer (strides = 2), SpatialDropout1D indicates one 1D space dropout layer for suppressing overfitting, GAP represents Global Average Pooling, and FC stands for Fully Connected Layers.

3.3 | 1D CNN for online action recognition

Considering the fast speed and competitive recognition accuracy, we here use a 1D CNN to train action classifier for online recognition. The 1D convolutional neural network can be used to learn temporal sequential data. Unlike the 2D convolutional neural network, the convolutional kernel of 1D convolutional neural network convolves along one dimension, which is simpler and faster. The network architecture is shown in Fig.3 .

In fact, the indices of human joints are not locally related. For example, the human head, left shoulder, and right shoulder are physiologically connected, but their indices, which are defined by the datasets, are not continuous. In addition, the local correlation of joints is different when the human performs different actions. Therefore, the part before "concatenate" of the neural network is designed to automatically learn the correlation between joints, so as to better improve the accuracy of recognition. Next, we splice the vectors obtained from the joint correlation learning to achieve multi-angle features fusion. Then we use a 1D convolutional layer for temporal information modelling. Finally, the feature map is expanded into a one-dimensional vector, two Fully Connected layers (FC) are used for classification, and the Softmax function is used in the output layer to get the probability of actions.

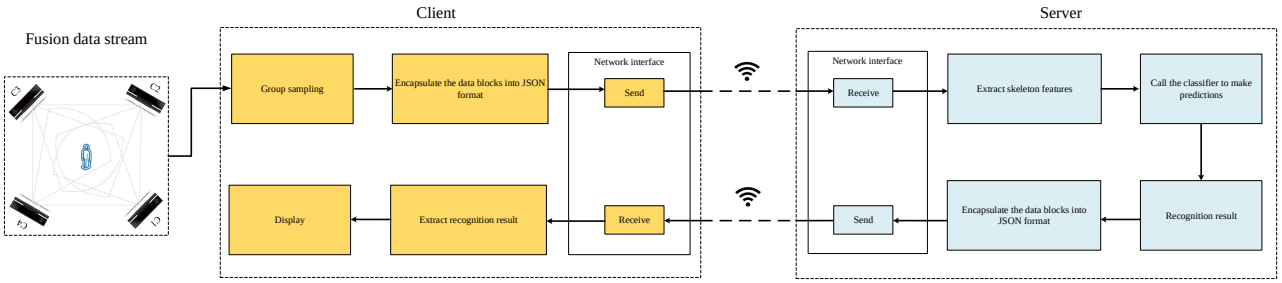


FIGURE 4 The client-server architecture using the Restful style for online action recognition using a camera network.

3.4 | Skeleton fusion using a distributed RGBD camera network

We here consider the RGBD camera network with N nodes, which construct an undirected graph $G = (C, E)$, where $C = \{1, 2, 3, \cdots, N\}$ denotes a vertex set and $E \subset \{\{i, j\}|i, j \subset C\}$ means the edge set. The neighboring nodes of the i node can be defined as $\mathcal{N}_i = \{j \in C|i, j \in E\}$. Each sensor node can have a measurement of the target human joint, which will be used to update the state of the estimator in the information weighted consensus filter (IWCF). The target human skeleton joints can have a dynamic model as

$$x(t+1) = Fx(t) + w \tag{10}$$

where $x(t) = (p_x(t), p_y(t), p_z(t), v_x(t), v_y(t), v_z(t))^T$ is the state vector including the joint 3D position and velocity, F is the state transition matrix and $w \sim N(0, Q)$ is the process Gaussian noise with zero mean and covariance Q. The measurement model for each joint is

$$z(t) = Hx(t) + v \tag{11}$$

where z is the measurement of the joint, H is the observation matrix having a Gaussian noise $v \sim N(0, R)$ with zero mean and covariance R. The dynamic model can be used for prediction and measurement model can be used for state updating using consensus as shown in Algorithm 2, where $u_{i,k} = H_{i,k}^T R_{i,k}^{-1} z_{i,k}$ and $U_{i,k} = H_{i,k}^T R_{i,k}^{-1} H_{i,k}$ are information contributions of the current measurement z.

The original IWCF algorithm in[30], uses a deterministic value $\epsilon = 0.65/\Delta_{max}$, which is not optimal for convergence as we show in[31]. As an alternative way, the Metropolis weights have been proposed to be used with consensus algorithms, and have been shown to be faster in terms of convergence rate than the maximum-degree weights[35]. The Metropolis weights at time step k can be defined as

$$\epsilon_{i,j,k} = \begin{cases} \frac{1}{1+max\{d_{i,k}, d_{j,k}\}} & if\ j \in \mathcal{N}_i \\ 1 - \sum_{j \in \mathcal{N}_i} \epsilon_{i,j,k} & if\ i = j \\ 0 & otherwise \end{cases} \tag{12}$$

where $d_{i,k}$ and $d_{j,k}$ are the degrees of the node i and node j respectively.

3.5 | The client-server architecture for distributed action recognition

The action recognition algorithms based on CNN require GPU computing power, which is not suitable for sensor networks with limited computational resources. Therefore, we here propose a client-server architecture for online action recognition using a distributed camera network as shown in Fig. 4 . The design of the client-server architecture follows the Restful style, due to its high extension capabilities for massive parallel service asks. The API's messaging protocol between the server and the client is based on HTTP, and the data stream format is based on JSON.

On the client side, we capture real time image sequences from each camera, and use OpenPose[36] for human joints detection. Then, we estimate the 3D coordinates of the joints and fuse the joints from cameras of different perspective. Furthermore, we encapsulate the data packets of skeleton sequences into the JSON format, send them to the server by the network, and wait for the server response. When the server receives the skeleton sequences, it will calculate the advanced kinetic features of the skeleton sequences and use the classifier deployed on it for action recognition.

Algorithm 2 IWCF based skeleton fusion

- Initialization:

  Total consensus iteration steps L, process noise Q and measurement noise R.

- For $k = 1, \cdots, \infty$:

  1. Prediction for the next time step:

$$\hat{x}_{i,k} = F_k x_{i,k-1} \tag{13}$$

$$\hat{Y}_{i,k} = (F_k Y_{i,k-1}^{-1} F_k^T + Q_k)^{-1} \tag{14}$$

$$\hat{y}_{i,k} = \hat{Y}_{i,k} \hat{x}_{i,k} \tag{15}$$

  2. Perform consensus:

$$v_{i,k}^0 = \frac{1}{N} \hat{y}_{i,k} + u_{i,k} \tag{16}$$

$$V_{i,k}^0 = \frac{1}{N} \hat{Y}_{i,k} + U_{i,k} \tag{17}$$

  for $l = 1$ to L **do**

  (a) Send $v_{i,k}^{l-1}$ and $V_{i,k}^{l-1}$ to all neighbors $j \in \mathcal{N}_i$

  (b) Receive $v_{j,k}^{l-1}$ and $V_{j,k}^{l-1}$ from all neighbors $j \in \mathcal{N}_i$

  (c) Update consensus terms

$$v_{i,k}^l = \epsilon_{i,j,k} \sum_{j \in \mathcal{N}_i} v_{j,k}^{l-1} \tag{18}$$

$$V_{i,k}^l = \epsilon_{i,j,k} \sum_{j \in \mathcal{N}_i} V_{j,k}^{l-1} \tag{19}$$

  end for

  3. Compute the posterior at k time step:

$$y_{i,k} = N v_{i,k}^L \tag{20}$$

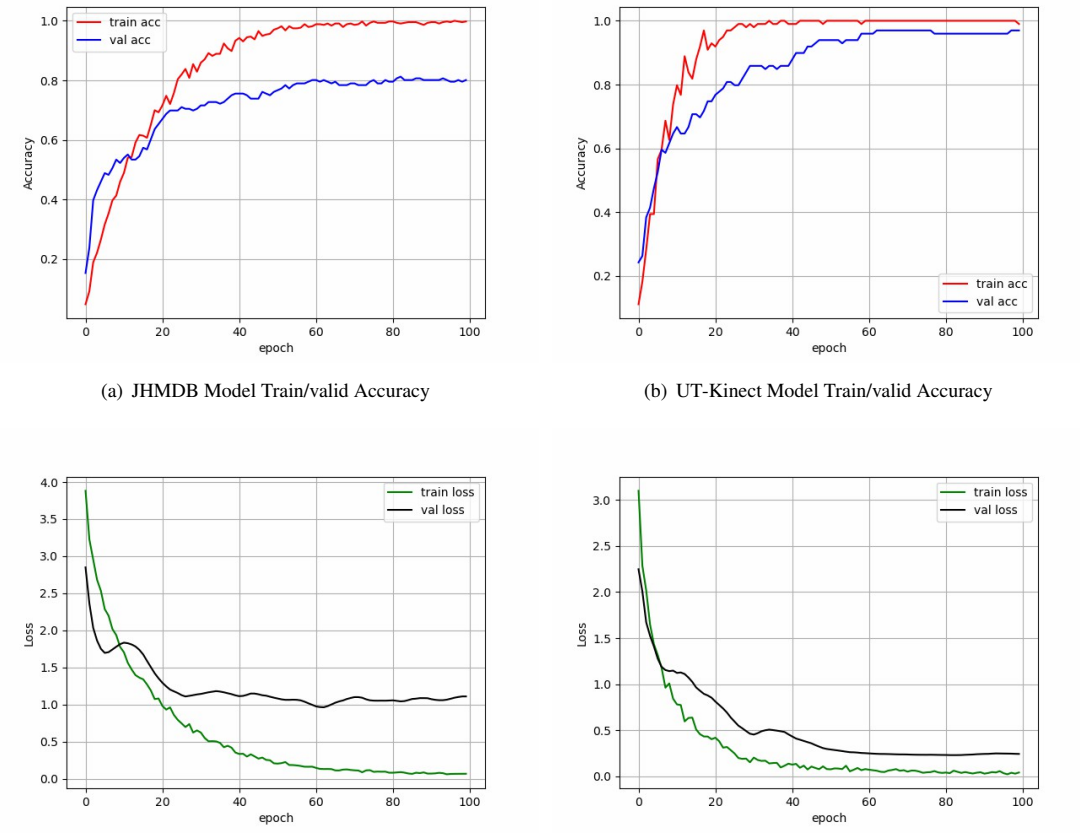
$$Y_{i,k} = N V_{i,k}^L \tag{21}$$

$$x_{i,k} = Y_{i,k}^{-1} y_{i,k} \tag{22}$$

On the server side, it uses the flask network framework to deploy the trained classifier model as a Restful style application program, and create an interface to provide human action classification service for the clients. The service program receives client data and corresponding call requests. Received data will be pre-processed next, and the classifier model will be loaded into the server memory for action prediction. The status of whether the prediction is successful or not will be recorded, encapsulating the status and results into the JSON format that will be sent to the client.

4 | EXPERIMENTS

To demonstrate the performance of the proposed method, we use two public skeleton-based datasets: the JHMDB dataset[37] and UT-Kinect dataset[38], and our laboratory dataset collected using a distributed RGBD camera network. The JHMDB dataset contains video clips and skeleton sequences of single person actions, and we only use the skeleton sequences. The UT-Kinect



(a) JHMDB Model Train/valid Accuracy

(b) UT-Kinect Model Train/valid Accuracy

(c) JHMDB Model Train/valid Loss

(d) UT-Kinect Model Train/valid Loss

FIGURE 5 The training performance of the proposed model. Subgraphs a and b show the training and validation accuracies of the model on JHMDB and UT-Kinect datasets, respectively. Subgraphs c and d show the training and validation losses of the model on JHMDB and UT-Kinect datasets, respectively.

dataset contains 200 sequences of 10 action classes with the 3D skeleton data from depth cameras. Every action is recorded twice for each subject.

To train the neural network, we use a computer with a Nvidia TITAN X GPU. The recognition frame rate can reach 20fps, which can meet the requirements of real-time applications. The Adam[39] optimizer is used for learning. We set the initial learning rate to 0.001. When the loss function value of the validation set does not decrease after more than 5 epochs of training, the learning rate is reduced at a rate of 0.5 until it is reduced to 0.00001. A total of 100 epochs were trained. Fig. 5 shows the trend of training and validation accuracies, training and validation losses.

4.1 | JHMDB dataset

Our algorithm achieves a higher classification accuracy than ChainedNet, EHPI, PoTion and DDNet on the JHMDB dataset as shown in Table 2 . Compared to our method, ChainedNet[40] uses a 3D CNN classifier and directly inputs the original joint sequences. However, it only achieves 56.8% with its recognition rate. EHPI[41] encodes original joint coordinates as the color information over a fixed period of time, such that the motion of joints can be seen from the color changes, and the color image is fed into a CNN for classification. Similarly, the recognition accuracy is 65.5% with EHPI. PoTion[42] extracts the joint heatmaps for each frame and colorize them using a color that depends on the relative time in the video clip. For each joint, they aggregate them across all frames, which constitutes coded images that are further stacked together as an action representation for classification. PoTion achieves a 67.9% recognition rate. DDNet[34] performs relatively better by achieving a recognition rate of 77.2%. DDNet employs 1D CNN network for action classification using the JCD feature and global motion feature. It is slightly less accurate

TABLE 2 Offline action recognition rate on JHMDB dataset. The table shows that our method has an accuracy of 2.9% higher than the state-of-the-art method

Methods	Accuracy(%)
ChainedNet (ICCV17)[40]	56.8
EHPI (ITSC19)[41]	65.5
PoTion (CVPR18)[42]	67.9
Spatial-Temporal Attention (2018)[43]	73.5
iDT+FV+T-CNN(2020)[44]	74.4
DDNet(2019)[34]	77.2
Ours	**80.1**

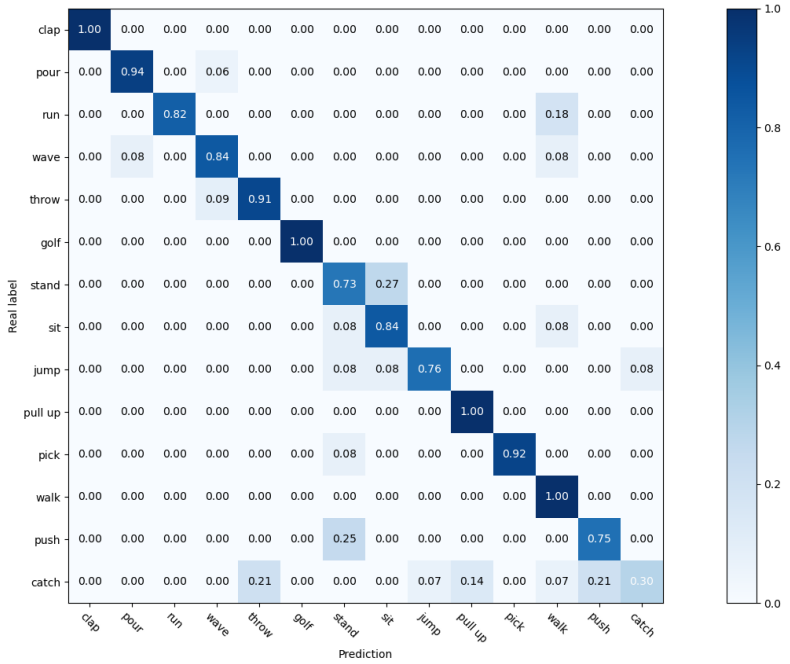


FIGURE 6 The confusion matrix of offline action recognition on JHMDB dataset with an average recognition rate of 80.1%

than ours since our method explores more advanced geometrical features from the original skeleton data as shown in Fig. 2 . This method also loses a considerable amount of spatial and temporal information. The methods we proposed use joint collection distances and multi-scale motion information to guide the data-driven feature learning. It introduces rich spatial geometric features to model the spatial information between joint in more detail. 1D CNN is used to better model the temporal information of the action sequences. Thus, we achieve higher classification accuracy. The confusion matrix of the offline action recognition rate on the JHMDB dataset is shown in Fig. 6 . The confusion matrix shows the action recognition rate of each class and also shows their confusion rate with other classes at corresponding rows. The x-axis of confusion matrix is the prediction label, the y-axis is the real label, and the diagonal value indicates the correct recognition rate of the corresponding category.

4.2 | UT-Kinect dataset

For the UT-Kinect dataset, we use half of the samples for training and the other half for testing. Our methods achieve a competitive performance compared to the state of the art methods as shown in Table 3 . The Skeleton Joint Features-based method proposed by[45] computes the frame difference and pairwise distance of skeleton joints positions to characterize the spatial information of the joints in 3D space, and has achieved a 87.9% recognition accuracy. The Elastic Functional Coding based method

TABLE 3 Offline action recognition rate on UT-Kinect dataset. The table shows that our method has an accuracy of 0.8% higher than the state-of-the-art method

Methods	Accuracy(%)
SkeletonJointFeatures(2013)[45]	87.9
ElasticFunctionalCoding (2015)[46]	94.9
GeoFeat(2017)[20]	95.9
GFT (2019)[47]	96.0
CNN With Attention Mechanism (2020)[48]	96.1
Ours	**96.9**

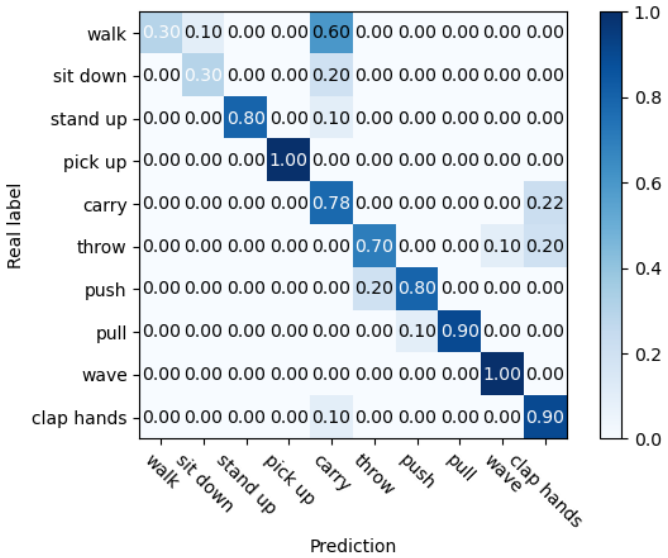


FIGURE 7 The confusion matrix of online action recognition using the sliding window method on UT-Kinect dataset with an average recognition rate of 74.7%

proposed by[46] achieves 94.9% in recognition accuracy. This method employs the TSRVF space that provides an elastic metric between two trajectories on a manifold to learn the latent variable space of human actions, and proposes mfPCA for compact and robust representation of features. GeoFeat[20] uses advanced geometric features to model human action sequences, and uses a three-layer LSTM network to classify actions. It achieves a 95.9% recognition accuracy. GFT[47] leverages skeletal temporal graph structures to represent body joints, and the graph transform GFT is utilized to extract representations of human motion data. GFT achieves 96% in recognition accuracy. Compared to these methods, we use advanced geometric features and multi-scale motion features to capture spatial and temporal information of human action, and employ a 1D CNN for action classification. We achieve a promising recognition accuracy of 96.9%.

4.3 | Online action recognition

To demonstrate the effectiveness of the group sampling mechanism for online action recognition using skeleton sequences, we compare the proposed method to the sliding window method. We first define online action recognition rate as the ratio of the number of positive samples to the total number of samples. The size of the input samples for the online action classifiers is 16, so the sliding window has a fixed size as shown in[25]. The results are shown in Table. 4 , which indicates that group sampling achieves a higher mean accuracy. The confusion matrix of online action recognition using the sliding window method on UT-Kinect dataset is shown in Fig. 7 , while the confusion matrix of online action recognition using the proposed group sampling

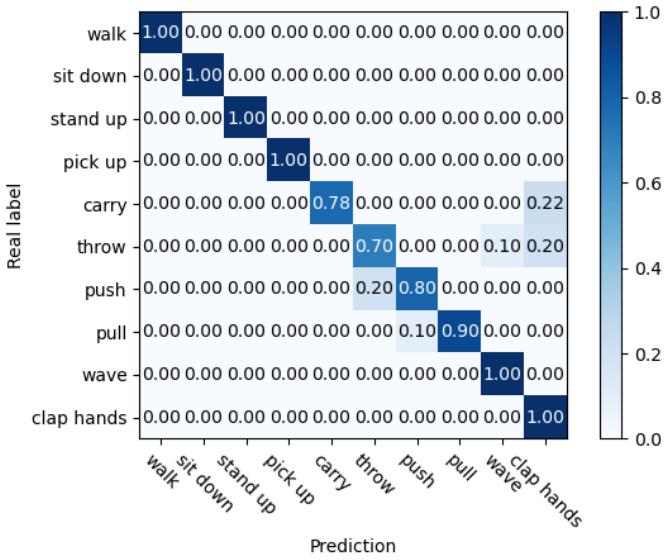


FIGURE 8 The confusion matrix of online action recognition using the proposed group sampling method on UT-Kinect dataset with an average recognition rate of 92%

TABLE 4 Online action recognition rate on UT-Kinect dataset. The table shows that the group sampling method achieves higher mean accuracy than sliding window method.

	Walk	Sit down	Stand up	Pick up	Carry	Throw	Push	Pull	Wave	Clap hands	Average
Group sampling	100%	100%	100%	100%	78%	70%	80%	90%	100%	100%	92%
Sliding window method	30%	30%	80%	100%	78%	70%	80%	90%	100%	90%	74.7%

method on UT-Kinect dataset is shown in Fig. 8 . The confusion matrix shows the action recognition rate of each class and also shows their confusion rate with other classes at corresponding rows. The x-axis of the confusion matrix is the prediction label, the y-axis is the real label, and the diagonal value indicates the correct recognition rate of the corresponding category. For short action sequences, the sliding window method has similar performance with ours, such as pick up, carry, throw, push, pull, wave. However, the sliding window can not handle long action sequences well, such as walk, sit down, stand up and clap hands due to its fixed window size. For those actions including walk, sit down and stand up, which consist of different behavior patterns in a long period of time, the sliding window method can only cover frames within a certain window size, which can destroy the structure of the behavior pattern, e.g., more than half of the walk sequences are misclassified as the carry actions in the confusion matrix. On the contrary, the group sampling mechanism can recognize the behavior patterns correctly for the whole sequences, such that it achieves a higher accuracy.

4.4 | Distributed action recognition

Data fusion based on camera networks is an effective solution to viewpoint variation and occlusion problems in human action recognition. In general, a distributed camera network is more robust than a centralized camera network on sensor node failures, and easier for size expansion of the network for covering large areas. Therefore, we build a distributed three-dimensional vision sensor network in an indoor environment, which employs four Kinect sensors (C1, C2, C3 and C4) to cover the whole field of view as shown in Fig. 9 . We use an NVIDIA Jetson TX2 board for local data processing of each Kinect. These four cameras are calibrated in a coarse-to-fine manner, and connected with each other using a local area network. The NTP protocol is used to synchronize the time of the four sensor computing nodes. We then use the OpenPose algorithm [36] to estimate the coordinates of two-dimensional human joints. The 3D coordinates of the joints can be derived from the depth images of the Kinect cameras.

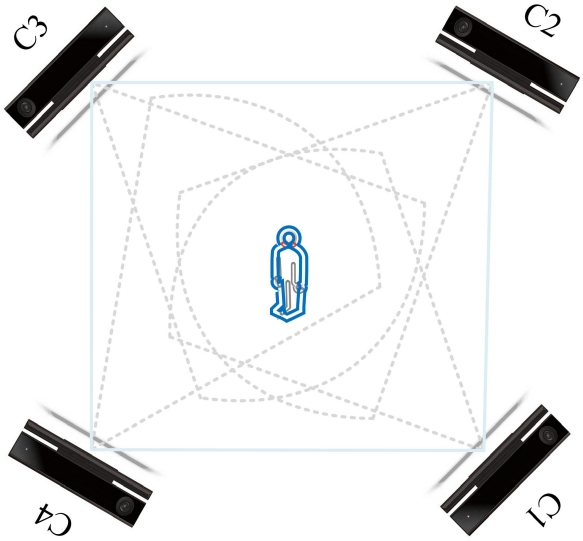


FIGURE 9 The spatial position of Kinects around the actor. We here use four Kinect V2 cameras with four Jetson TX2 computation nodes to construct the distributed sensor network, which covers a rectangular area for multiple view fusion. The topology structure of our distributed camera network is a circle loop compared with the centralized network.

(a) Right-front RGBD camera.

(b) Left-front RGBD camera.

(c) Left-back RGBD camera.

(d) Right-back RGBD camera.

FIGURE 10 Distributed action recognition using a camera network of four RGBD cameras (Kinects). The actors stand in the middle of the rectangle area, and perform a number of actions with different facing directions. The gray images are corresponding depth images.

To fuse the 3D positions of joints from different views, we use a distributed information consensus filter proposed by[49], such that each sensor can fuse the information from all other views after a limited number of consensus iteration steps.

We collected nineteen actions by nine actors, namely bowling, reaching out, hand clap, waving, knocking, drawing a cross (x), drawing circles, left kicking, right kicking, front kicking, sitting, standing, bending, bending arm, squat, tennis serving, shooting, stepping and drawing triangle. The actors stand in the middle position as shown in Fig. 10 , and each action is repeated three times by facing three different directions. As we can see in Table. 5 , our algorithm achieves competitive recognition rates for most of the daily actions with an average processing rate of $20 fps$. The confusion matrix of the distributed online action

TABLE 5 Distributed online action recognition rate on laboratory dataset using Our Method. The table shows competitive recognition rate for most of the daily actions

Bowling	Reach out	Hand clap	Waving	Knocking
86.7%	100%	100%	86.7%	66.7%
Draw x	Draw circles	Left kicking	Right kicking	Front kicking
73.3%	53.3%	100%	100%	100%
Sit	Stand	Bend	Bend arm	Squat
100%	100%	93.3%	86.7%	100%
Tennis serve	Shoot	Step	Drawing triangle	Average
86.7%	80%	93.3%	93.3%	89.5%

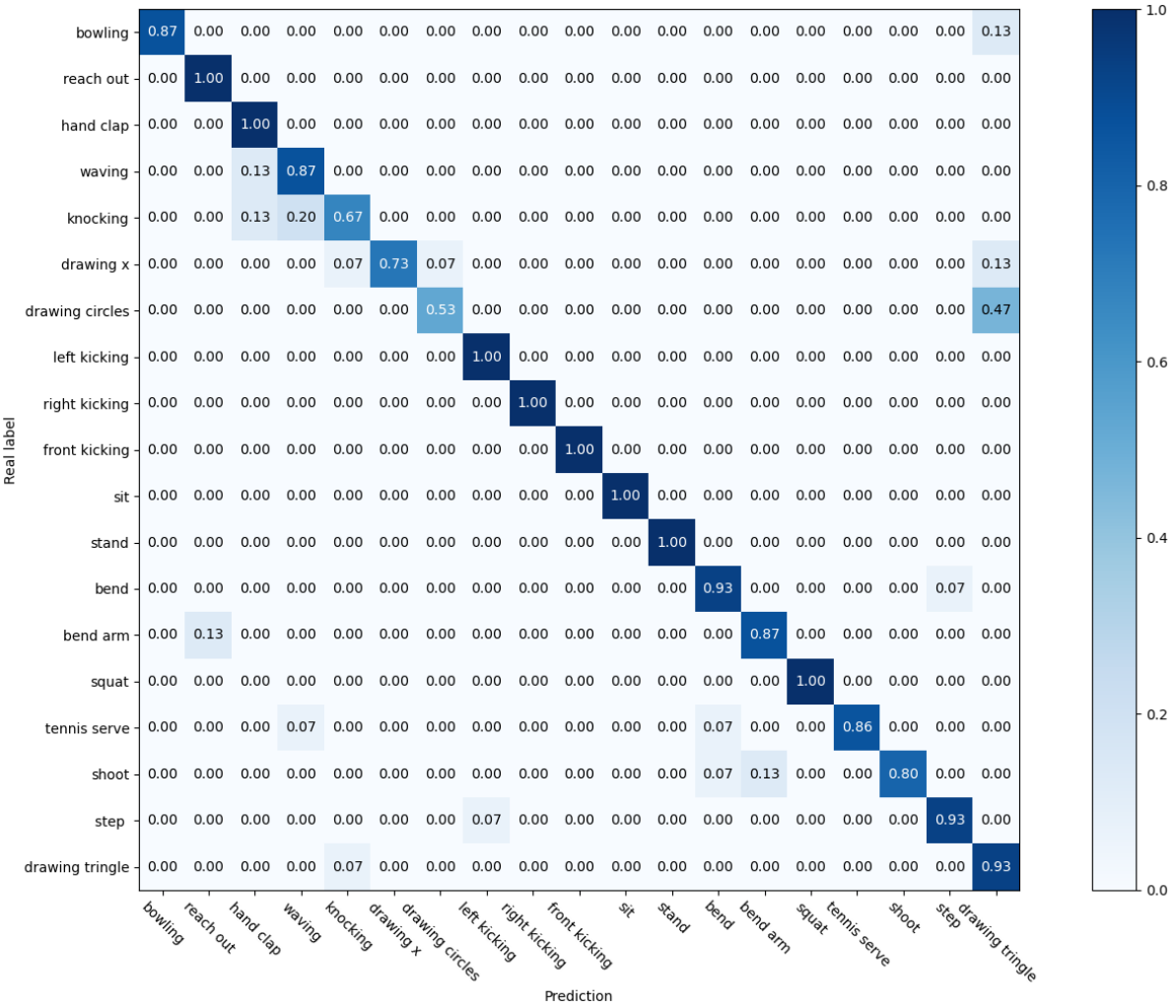


FIGURE 11 The confusion matrix of distributed online action recognition using the proposed group sampling method on our laboratory dataset

recognition using the proposed group sampling method on our laboratory dataset is shown in Fig. 11 . The confusion matrix shows that each type of action achieves a high recognition rate.

5 | CONCLUSION

In order to solve the problem of online human action recognition, we propose a group sampling based 1D CNN action classifier using both spatial and temporal kinetic skeleton features. The group sampling is superior to the traditional sliding window method, since it can capture long term contextual information, while the nearby frames have higher sampling densities. In addition, we combine the JCD feature, advanced geometrical feature and global motion feature to represent human action information, such that the spatial and temporal information are considered. Furthermore, a simple and effective 1D CNN is used for online classification of concatenated multiple skeleton features. Considering the high computation demands of CNN and limited onboard computational resources of a distributed camera network, we propose a client-server architecture to deploy the proposed action recognition module on the remote server, such that all camera nodes can request for action recognition services simultaneously. Finally, we demonstrate our method on the JHMDB and UT-Kinect datasets and our laboratory datasets. The results show that the proposed method outperform other methods with competitive performance.

In this work, we only focus on cases with one single person in the scene. However, in the real world, it would be a highly demanding feature to be able to process multiple people that co-exist and interact in the same scene. In the future, we will focus on action recognition with multiple people.

ACKNOWLEDGEMENTS

This research was supported by the National Key R&D Program of China (2018YFB1306500), National Natural Science Foundation of China (91748115, 61603213), Young Scholars Program of Shandong University (2018WLJH71), the Fundamental Research Funds of Shandong University, and the Taishan Scholars Program of Shandong Province.

References

1. Donahue J, Hendricks LA, Guadarrama S, et al. Long-term recurrent convolutional networks for visual recognition and description. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR):2625-2634; 2015.
2. Zhang Y, Liu X, Chang M, Ge W, Chen T. Spatio-temporal phrases for activity recognition. In: European Conference on Computer Vision:707–721; 2012.
3. Cheng G, Huang Y, Wan Y, Buckles BP. Exploring Temporal Structure of Trajectory Components for Action Recognition. *International Journal of Intelligent Systems.* 2015;30(2):99–119.
4. Zhu W, Lan C, Xing J, et al. Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence:3697–3703; 2016.
5. Yong D , Wang W, Wang L. Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition:1110-1118; 2015.
6. Mustaqeem , Kwon S. Att-Net: Enhanced emotion recognition system using lightweight self-attention module. *Applied Soft Computing.* 2021;102:107101.
7. Mustaqeem , Kwon S. A CNN-Assisted Enhanced Audio Signal Processing for Speech Emotion Recognition. *Sensors.* 2020;20(1).
8. Anvarjon T, Mustaqeem , Kwon S. Deep-Net: A Lightweight CNN-Based Speech Emotion Recognition System Using Deep Frequency Features. *Sensors.* 2020;20(18).
9. Mustaqeem , Kwon S. MLT-DNet: Speech emotion recognition using 1D dilated CNN based on multi-learning trick approach. *Expert Systems with Applications.* 2021;167:114177.
10. Xiang Q, Wang X, Song Y, Lei L, Li R, Lai J. One-dimensional convolutional neural networks for high-resolution range profile recognition via adaptively feature recalibrating and automatically channel pruning. *International Journal of Intelligent Systems.* 2021;36(1):332-361.

11. Kong Y, Tao Z, Fu Y. Deep sequential context networks for action prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition:1473–1481; 2017.

12. Gao J, Yang Z, Chen K, Sun C, Nevatia R. Turn tap: Temporal unit regression network for temporal action proposals. In: Proceedings of the IEEE international conference on computer vision:3628–3636; 2017.

13. Lea C, Flynn MD, Vidal R, Reiter A, Hager GD. Temporal convolutional networks for action segmentation and detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition:156–165; 2017.

14. Li Y, Lan C, Xing J, Zeng W, Yuan C, Liu J. Online human action detection using joint classification-regression recurrent neural networks. In: European Conference on Computer Vision:203–220; 2016.

15. Baek S, Kim K, Kim T. Real-time online action detection forests using spatio-temporal contexts. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV):158–167; 2017.

16. Wang H, Wang L. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition:499–508; 2017.

17. Liu J, Shahroudy A, Xu D, Wang G. Spatio-temporal lstm with trust gates for 3d human action recognition. In: European conference on computer vision:816–833; 2016.

18. Wang P, Li Z, Hou Y, Li W. Action recognition based on joint trajectory maps using convolutional neural networks. In: Proceedings of the 24th ACM international conference on Multimedia:102–106; 2016.

19. Huynh-The T, Hua C, Kim D. Encoding Pose Features to Images With Data Augmentation for 3-D Action Recognition. *IEEE Transactions on Industrial Informatics.* 2020;16(5):3100-3111.

20. Zhang S, Liu X, Xiao J. On geometric features for skeleton-based action recognition using multilayer lstm networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV):148–157; 2017.

21. Yasin H, Hussain M, Weber A. Keys for Action: An Efficient Keyframe-Based Approach for 3D Action Recognition Using a Deep Neural Network. *Sensors.* 2020;20(8):2226.

22. Jiang S, Qi Y, Zhang H, Bai Z, Lu X, Wang P. D3D: Dual 3D Convolutional Network for Real-time Action Recognition. *IEEE Transactions on Industrial Informatics.* 2020;:1-1.

23. De Geest R, Gavves E, Ghodrati A, Li Z, Snoek C, Tuytelaars T. Online action detection. In: European Conference on Computer Vision:269–284; 2016.

24. You Q, Jiang H. Action4D: Online Action Recognition in the Crowd and Clutter. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR):11849-11858; 2019.

25. Zanfir M, Leordeanu M, Sminchisescu C. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In: Proceedings of the IEEE international conference on computer vision:2752–2759; 2013.

26. Aggarwal JK, Cai Q. Human motion analysis: A review. *Computer vision and image understanding.* 1999;73(3):428–440.

27. Song B, Kamal A, Soto C, Ding C, Farrell J, Roychowdhury A. Tracking and activity recognition through consensus in distributed camera networks.. *IEEE Transactions on Image Processing.* 2010;19(10):2564-2579.

28. Kamal AT, Farrell JA, Roy-Chowdhury AK. Information Weighted Consensus Filters and Their Application in Distributed Camera Networks. *IEEE Trans. Autom. Control.* 2013;58(12):3112-3125.

29. Liu G, Tian G. Square-Root Sigma-Point Information Consensus Filters for Distributed Nonlinear Estimation. *Sensors.* 2017;17(800).

30. Kamal AT, Bappy JH, Farrell JA, Roy-Chowdhury AK. Distributed Multi-Target Tracking and Data Association in Vision Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2016;38(7):1397-1410.

31. Liu G, Zhao Y. Information weighted consensus filtering with improved convergence rate. In: Chinese Control Conference:8356-8359; 2016.

32. Katragadda S, Sanmiguel JC, Cavallaro A. Consensus protocols for distributed tracking in wireless camera networks. In: International Conference on Information Fusion:1-8; 2014.

33. Chen Y, Zhao Q, An Z, Lv P, Zhao L. Distributed Multi-Target Tracking Based on the K-MTSCF Algorithm in Camera Networks. *IEEE Sens. J.*. 2016;16(13):5481-5490.

34. Yang F, Wu Y, Sakti S, Nakamura S. Make Skeleton-Based Action Recognition Model Smaller, Faster and Better. In: Proceedings of the ACM Multimedia Asia:1–6; 2019.

35. Xiao L, Boyd S, Lall S. A scheme for robust distributed sensor fusion based on average consensus. In: International Symposium on Information Processing in Sensor Networks:63–70; 2005.

36. Cao Z, Hidalgo G, Simon T, Wei SE, Sheikh Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2021;43(1):172-186.

37. Jhuang H, Gall J, Zuffi S, Schmid C, Black MJ. Towards Understanding Action Recognition. In: 2013 IEEE International Conference on Computer Vision:3192-3199; 2013.

38. Xia L, Chen C, Aggarwal JK. View invariant human action recognition using histograms of 3d joints. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops:20–27; 2012.

39. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.* 2014;.

40. Zolfaghari M, Oliveira GL, Sedaghat N, Brox T. Chained Multi-stream Networks Exploiting Pose, Motion, and Appearance for Action Classification and Detection. In: 2017 IEEE International Conference on Computer Vision (ICCV):2923-2932; 2017.

41. Ludl D, Gulde T, Curio C. Simple yet efficient real-time pose-based action recognition. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC):581-588; 2019.

42. Choutas V, Weinzaepfel P, Revaud J, Schmid C. PoTion: Pose MoTion Representation for Action Recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition:7024-7033; 2018.

43. Du W, Wang Y, Qiao Y. Recurrent Spatial-Temporal Attention Network for Action Recognition in Videos. *IEEE Transactions on Image Processing.* 2018;27(3):1347-1360.

44. Sheng B, Li J, Xiao F, Li Q, Yang W, Han J. Discriminative Multi-View Subspace Feature Learning for Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology.* 2020;30(12):4591-4600.

45. Zhu Y, Chen W, Guo G. Fusing Spatiotemporal Features and Joints for 3D Action Recognition. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops:486-491; 2013.

46. Anirudh R, Turaga P, Su J, Srivastava A. Elastic functional coding of human actions: From vector-fields to latent variables. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR):3147-3155; 2015.

47. Kao J, Ortega A, Tian D, Mansour H, Vetro A. Graph Based Skeleton Modeling for Human Activity Analysis. In: 2019 IEEE International Conference on Image Processing (ICIP):2025-2029; 2019.

48. Zhu K, Wang R, Zhao Q, Cheng J, Tao D. A Cuboid CNN Model With an Attention Mechanism for Skeleton-Based Action Recognition. *IEEE Transactions on Multimedia.* 2020;22(11):2977-2989.

49. Liu G, Tian G, Li J, Zhu X, Wang Z. Human Action Recognition Using a Distributed RGB-Depth Camera Network. *IEEE Sens. J.*. 2018;18(18):7570-7576.

How to cite this article: G. Liu, Q. Zhang, Y. Cao, G. Tian, and Z. Ji (2021), Online Human Action Recognition with Spatial and Temporal Skeleton Features Using a Distributed Camera Network, . *Int J Intell Syst.* , *2021*.