Global Co-occurrence Feature Learning and Active Coordinate System Conversion for Skeleton-based Action Recognition

Sheng Li Tingting Jiang Tiejun Huang Yonghong Tian
NELVT, Department of Computer Science, Peking University, China
{li_sheng, ttjiang, tjhuang, yhtian}@pku.edu.cn

Abstract

Skeleton-based action recognition has attracted more and more attention in recent years. Besides, the rapid development of deep learning has greatly improved the performance. However, the current exploration of action co-occurrence is still not comprehensive enough. Most existing works only mine co-occurrence features from the temporal or spatial domain seperately, and it's common to combine them in the end. Different from previous works, our approach is able to learn temporal and spatial co-occurrence features integratedly and globally, which is called spatio-temporal-unit feature enhancement (STUFE). In order to better align the skeleton data, we introduce a novel method for skeleton data preprocessing called active coordinate system conversion (ACSC). A coordinate system can be learned automatically to transform skeleton samples for alignment. By the way, the proposed methods are compatible with current two types of mainstream models, the CNN-based and GCN-based models. Finally, on the two benchmarks of NTU-RGB+D and SBU Kinect Interaction, we validated our methods based on two mainstream models. The results show that our methods achieve the state-of-the-art.

1. Introduction

In the past few years, human action recognition has become an active area of research, due to its wide applications, ranging from surveillance to human-computer interaction and virtual reality. Human pose, also known as skeleton, can be used as a kind of data modality for action recognition. Unlike RGB video, human skeleton sequences can provide very effective information only with a limited amount of data. [9] first verified the validity of skeletal sequence on discriminant actions from a biological perspective. Now there are many devices can directly provide solutions for real-time skeleton sequence output. Intel RealSense [11] and Microsoft Kinect [36] are the most

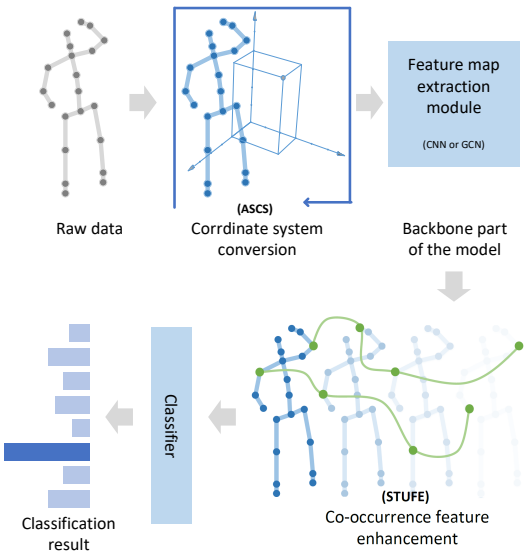

Figure 1: This is an overview of our approach at the data flow level. Take the original skeleton data as input, then pass ACSC and STUFE, and finally complete the classification.

commonly used. The popularity of these devices has greatly enhanced the utility of skeleton-based action recognition.

Preprocessing of skeleton datas is common in skeleton-based action recognition. Most of these operations are designed for data augmentation or reducing the noise of the data. In order to get a better expression of the skeleton data, MTLN [10] converts the skeleton data from a Cartesian coordinate system to a cylindrical coordinate system. And the experimental results validates the effectiveness of the operation. However, such coordinate transformations are fixed for all the skeleton samples without enough flexibility, which leaves room for improvement.

On the other hand, one of the most critical problems has always been how to extract useful features from the skeleton sequences. Many early works tried the approach of hand-crafted features [29, 31, 25], e.g. some probability based approaches [31] constructed features from dynamic motion

and static skeletons. Since 2015, manual feature methods have been gradually surpassed by deep neural network methods. Recurrent Neural Networks (RNNs) was first introduced to skeleton sequence modeling. These RNN-based methods, including H-RNN [4] and LSTM [18, 24, 33, 34, 13, 19] focus on modeling the information in the time domain and are lack of the ability of modeling the spatial information. Subsequently, the introduction of the Convolutional Neural Networks (CNN) solved the above problem of RNN. The feature extraction capabilities of CNN have been verified on multiple computer vision tasks. Applying CNN to skeleton-based action recognition requires formatting the skeleton sequence into an image form. [3] are the first to convert the joints and timing of skeleton sequences to the height and width of the images, followed by an image classifier. Further improvements were made in HCN [15]. It enhanced the spatial co-occurrence features learning abilities by modifying the convolution operation. But it does not consider temporal domain modeling. A few years later, graph-based convolutions (GCN) have emerged in this area. Most recent related work is aimed to enhance feature extraction in temporal or spatial(joint) domain based on ST-GCN [30].

In this paper, we consider learning the spatio-temporal features integratedly and globally. 1) 'integratedly' means that the modeling of temporal information and spatial information is integrated. Differently, some previous work such as STGR [14] designed a temporal module and a spatial module separately which are responsible for temporal modeling and spatial modeling respectively. These methods do not satisfy 'integratedly'. 2) 'Globally' means that the model has the ability to learn global information directly, both in terms of time and spatial dimensions. To the best of our knowledge, most previous works did not have both of these properties with in one model.

For many actions, spatio-temporal co-occurrence is especially important. Take the action of 'touching head' for example, people usually pay attention to the position of the hand at the beginning of the action, and then are concerned with the position of the head later. So the early 'hand' and the later 'head' are the key to distinguish this action category. Considering that the previous methods did not integratedly consider temporal and spatial domain, we have designed a method spatio-temporal-unit feature enhancement (STUFE) to learn co-occurrence features across temporal and spatial domains, and this method has good versatility and can be directly used in the current mainstream CNN and GCN architectures. To the best of our knowledge, this is the first time to learn co-occurrence features across spatial-temporal domain instead of doing it separately. Besides co-occurrence mining, we also consider the problem of skeleton data preprocessing. According to the previous work [10], the skeleton sequence in the cylindrical coordinate system is more helpful for action recognition than it in the Cartesian coordinate system. Inspired by this finding, we propose a method of active coordinate system transformation (ACSC) that converts each skeleton sequence into a more suitable space. The entire conversion process is differentiable, and its internal parameters can be automatically updated under the guidance of the gradient. The workflow of our method is shown in Fig. 1.

Our contributions can be summarized as follows: 1) For the first time, we propose a spatio-temporal co-occurrence feature learning method for skeleton-based action recongnition. 2) We design an active coordinate system transformation that can better align the skeleton data for action recognition.

2. Related Work

Action recognition is one of the key tasks in computer vision. There have been many approaches so far. According to the difference of the modal of the input data, they can be classified as RGB-based, depth-map based, skeleton-based, and others [27]. For example, a RGB-based method[21] uses video data as input, then a pose estimate method is introduced for feature extraction. A typical depth-map based method[26] uses depth map as input data, then using CNNs for feature learning and classification. At the same time, there is a method[2] that fuses multimodal data. Here, we focus on the relevant work of skeleton-based action recognition methods.

So, in this section, we first briefly cover the research related to skeleton-based recognition. Then, we summarize the relevant methods from the perspective of co-occurrence exploration.

2.1. Skeleton-based Action Recognition

The acquisition of skeleton data is becoming more and more convenient, thanks to the popularity of low-cost depth cameras [11, 36] and the rapid development of related technologies such as human posture estimation [1] for obtaining skeleton data from a single picture. This has led to the skeleton-based action recognition, attracting more attention in the academia.

Unlike the methods based on handcrafted features, the methods of learning feature representation directly from raw data, relying on the powerful feature learning capabilities of deep neural networks, have become the mainstream in recent years. As mentioned before, the RNN method [37, 35, 33] was once the mainstream. Then, a spatio-temporal graph [7] is introduced, which is used to express the relationships among body parts into the RNN. Although rough, this is the prototype of the follow-up GCN idea. Subsequently, skeleton sequences are manually transformed into images to feed into CNN. CNN-based methods

[3, 10, 17] gradually became dominant with its super feature extraction capability.

Nevertheless, the recent rise of the GCN-based approaches such as [30, 14] has shown greater potential with its superior spatial relationship modeling capabilities. The GCN method is derived from graph convolution. The earliest graph convolution is based on Graph Fourier Transform [23, 6, 5]. ST-GCN [30] first used the spatial domain graph convolution method to complete the skeleton-based action recognition. [14] has added dynamic routing based on ST-GCN, which can build more flexibly relationships among the joints of the human body. Similarly, in [28] a motif notation is introduced to model each joint in a skeleton.

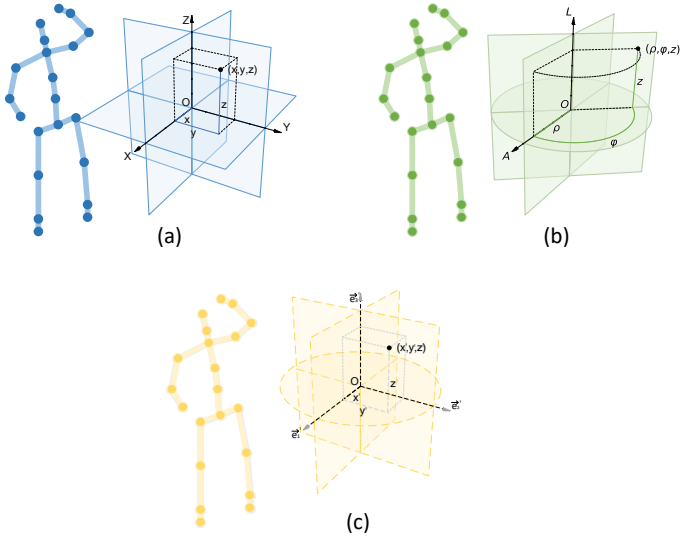

Figure 2: The idea of the ACSC method, (a) is the raw data, (b) is the data transformed in the cylindrical coordinate system, and (c) is the data we learned in the new coordinate system.

Although CNN-based and GCN-based methods are comparable in performance, GCN has better interpretability than CNN. The advantage of the CNN method is that since CNN has been successfully used in many fields, there are more experiences on CNN's network structure design and training. The proposed method in this work is fully compatible with both models.

2.2. Co-occurrence Exploration

Human actions are usually associated with a specific set of joint points at certain time steps. The human brain uses this when distinguishing an action. For example, if someone identify the action of 'touching back' (backache), he or she needs to focus on the hand and then turn the attention to the back. Therefore, how to learn co-occurrence has always been crucial for skeleton-based action recognition. Most of the work has explored this aspect, even though some did not mention the concept of co-occurrence. DeepLSTM [37] introduced the fully connected layer to complete the co-occurrence learning in the spatial domain. Global spatial co-occurrence can also be learned by transforming the feature map in the channel and joint dimensions [15]. Previous work only learn co-occurrence from spatial domain or the temporal domain separately. For example, the spatial graph router and the temporal graph router are respectively introduced to extract co-occurrence features independently, and finally the two are combined [14] . Different from these, our approach directly performs co-occurrence learning across both temporal and spatial domains.

3. Method

In this section, we introduce the on active coordinate system conversion (ASCS) and spatio-temporal-unit feature enhancement (STUFE). They have good compatibility and are compatible with CNN-based and GCN-based models.

3.1. Active Coordinate System Conversion

The skeleton data of a person at time step t can be denoted as $X_t = \{X_t^1, X_t^2, ..., X_t^j, ..., X_t^J\}$ where X_t^j refers to the coordinate value of the joint j at time t and J is the number of joints. For a complete skeleton sequence, it can be formulated as $X = \{X_1, X_2, ..., X_t, ..., X_T\} \in \mathbb{R}^{D\times T\times J}$, where T is the number of frames in the sequence and D is the depth dimension which is the dimensionality of the coordinate space. For 3D skeleton data, $D = 3$.

The original skeleton data is based on the Cartesian coordinate system. $[x, y, z]$ denotes the original coordinate value. The process of coordinate system conversion is represented as operator $\mathscr{A}$. Then, the cylindrical coordinate transformation process proposed by [10] can be expressed as:

$$\mathscr{A}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sqrt{x^2+y^2} \\ arcsin(y/\sqrt{x^2+y^2}) \\ z \end{pmatrix}. \quad (1)$$

Although good performance is achieved with the conversion, we think that there might be a coordinate system more suitable than the cylindrical coordinate system. We continue to advance on this direction and propose ACSC which is to generalize $\mathscr{A}$.

In detail, some formulation is needed. In 3D space, the base of the original coordinate system is defined as $E = [\vec{e}_1, \vec{e}_2, \vec{e}_3]$. The transformed base can be formulated as $E' = [\vec{e_1}', \vec{e_2}', \vec{e_3}']$. Obviously, in the inferring process, the coordinate values are involved in the calculation, and the base does not participate in any calculation. Here, $\vec{v} = [x, y, z]$ and $\vec{v}' = [x', y', z']$ represent initial coordinate values and transformed coordinate values, respectively. Each skeleton joint corresponds to a vector in space, and the

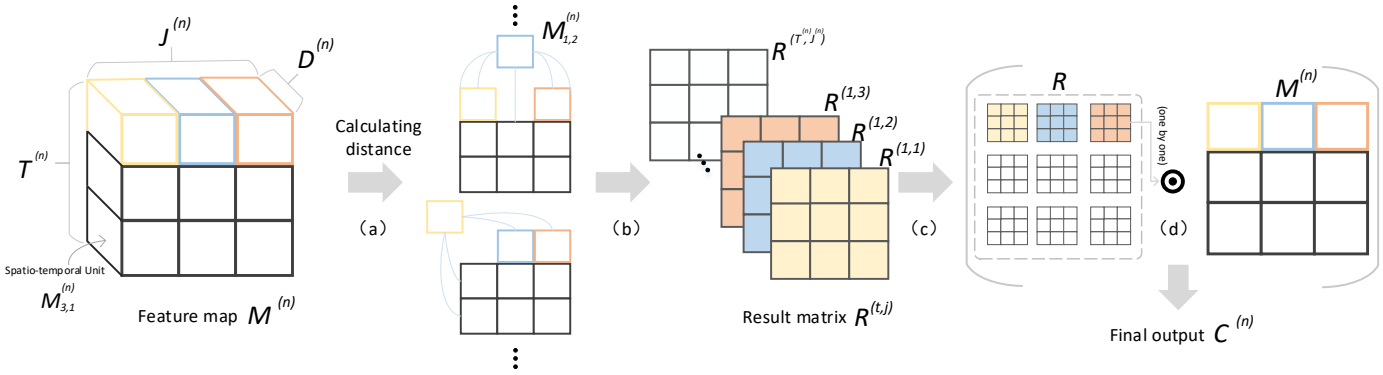

Figure 3: Overview of the key steps of the STUFE algorithm. Feature map $M_{t,j}^{(n)}$ is input. First step a) is to calculate the distance between every two units. A matrix R consisting of all the results can be obtained after the step c) . Finally, after $\odot$ summation in step (d), the $C^{(n)}$ used for attention mechanism can be obtained.

vector can be expressed as a base vector multiplied by a coordinate value. In any case, there must be

$$E\vec{v} \equiv E'\vec{v}', \tag{2}$$

$$(\vec{e}_1, \vec{e}_2, \vec{e}_3) \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = (\vec{e_1}', \vec{e_2}', \vec{e_3}') \cdot \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}. \tag{3}$$

The transformation of the coordinate system is essentially the conversion of the base. Here, $\mathscr{A}$ is still used to represent the base transformation. The process of base conversion of the coordinate system is,

$$E = E'\mathscr{A}. \tag{4}$$

We can get $E\vec{v} = E'\mathscr{A}\vec{v}$. According to Eqn. (4), $E\vec{v} = E'\mathscr{A}\vec{v} = E'\vec{v}'$. In this way, the relationship of $\vec{v}$ and $\vec{v}'$ that we are most concerned about can be found.

$$\vec{v}' = \mathscr{A}\vec{v}. \tag{5}$$

Finding a better coordinate system is actually to find a better base conversion operator $\mathscr{A}$. The change that $\mathscr{A}$ can make is shown in Fig. 2. Unlike the fixed form of cylindrical coordinate system conversion, we aim to transform $\mathscr{A}$ into a learnable form.

Ideally $\mathscr{A}$ should have following properties: **1)** The new coordinate system generated by $\mathscr{A}$ should be orthogonal. **2)** The transformation itself should be nonsingular, otherwise the dimensionality of the newly generated coordinate system will be smaller than that of the original space. And this will lose a lot of spatial information in the original skeleton sequence, just as 3D data is projected onto a 2D plane.

In fact, it is difficult to find constraints on the orthogonality to satisfy property **1)** rigorously. Thus, we choose an approximation strategy such that $\mathscr{A}$ can be approximated by a multi-layer perceptron (MLP) and the transformation of the coordinate system can be learned. In order to satisfy property **2)**, based on MLP, the initial values of the parameters must be selected carefully. Because if the initial parameters of the MLP happen to cause singularity of the skeleton data, subsequent training will be difficult to take a turn for the better. In order to ensure that the initial parameters are meaningful and satisfy the property **2)**, the training is not started immediately, but the MLP is used as a self-encoder for first training. In other words, $\mathscr{A}$ at this time just serves as 'identity matrix', which maps the base of the Cartesian coordinate system to itself. Then, MLP is used to learn the transform process.

Since the coordinate transformation involves only the primitive input data, it has good versatility. It is compatible with two mainstream models. The experimental details are shown in Section 4.

3.2. Spatio-temporal-unit Feature Enhancement

Co-occurrence is very important for the recognition of actions. Our method of spatio-temporal-unit feature enhancement (STUFE) is compatible with both CNN and GCN. This is because the feature maps of CNN and GCN are very similar. Thus, we use CNN as an example to illustrate the method.

The input skeleton data $X \in \mathbb{R}^{D\times T\times J}$ can be regarded as a standard 3D tensor and the subsequent feature maps can be as well where D, T, J have same meaning as Section 3.1. For the CNN model, the feature map of the n^{th} layer can be noted as $M^{(n)}$. $M^{(n)} \in \mathbb{R}^{D_n\times T_n\times J_n}$, where D_n, T_n, J_n is the depth,height (time dimension) and width (joint dimension) of the feature map of n^{th} layer. We splits the feature map $M^{(n)}$ into basic spatio-temporal units $M_{t,j}^{(n)}$ and each unit $M_{t,j}^{(n)} \in \mathbb{R}^{D_n\times 1\times 1}$. A feature map $M^{(n)}$ have $T\times J$ units totally. These units with high-level semantic features are the key to our exploration of co-occurrence across spatio-temporal domain.

The key steps of STUFE are shown in Fig. 3.

Start with Step (a) in Fig. 3. Here, we introduce a distance function $Distance(unit_a, unit_b)$, so that we can calculate the distance between the unit $M_{t,j}^{(n)}$ and the remaining units one by one. This distance function will be described in detail later. Then just as (b) shown in Fig. 3, the calculation result can be written in the form of a matrix with the size of $T \times J$, representing the distance among $M_{t,j}^{(n)}$ and all units. Here we record this result matrix as $R^{(t,j)}$. Similarly, each unit $M_{t,j}^{(n)}$ has the corresponding matrix $R^{(t,j)}$ through the above calculation. In total there are $T \times J$ matrices like $R^{(t,j)}$. Let R denotes the stack of the matrices. For feature map $M^{(n)}$, R can be reshaped such that $R \in \mathbb{R}^{(T_n \times J_n) \times (T_n \times J_n)}$, shown as (c) in Fig. 3. Note that R contains the distance of any two of spatio-temporal feature units in a specific semantic space. It also plays an important role in the visualization in Section 4.3.4. However, the information currently contained in R cannot be directly translated into the discriminative ability of the model. Step (d) is to solve this problem. Next we elaborate Step (a) and Step (d).

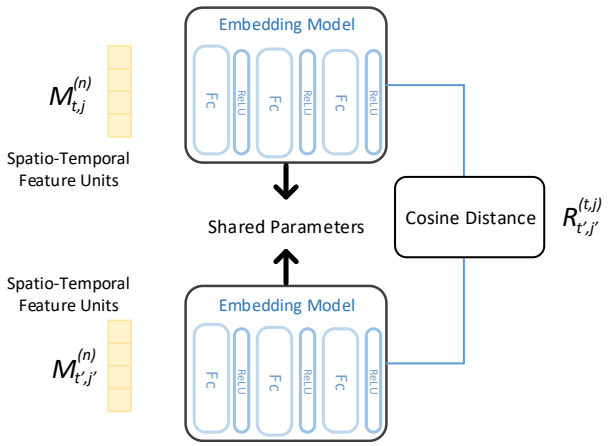

Figure 4: Illustration of the distance function. First perform embedding on each unit, then calculate the cosine distance.

For Step (a) which is about the distance function, we want to allow units to map to a new space in which the cosine distance between two units with co-occurrence can be larger. Instead of directly calculating the cosine distance between the feature units, we perform embedding firstly as shown in Fig. 4. This embedding process can not be supervised directly, so this process needs to be restricted. Here, a similar approach to ACSC in Section 3.1 is adopted. The embedding model is separately trained by the way of autoencoder. This ensures that the embedding model has meaningful initial parameters when the training begins. Subsequent training also limits the learning rate of the parameters of this part. The magnitude of the parameter change is limited, making this part of the training more like finetuning.

For Step (d), although R contains the spatio-temporal correlation information we want, its size is too large to be used directly. To solve this problem, we used the attention mechanism. As long as a group of units with co-occurrence is enhanced by weighted summation, the effect of increasing attention is achieved. In order to clarify this process, now we define an operation $\odot$,

$$A \odot B = \sum_{i=1}^{M} \sum_{j=1}^{N} A_{ij} \cdot B_{ij}, \tag{6}$$

where $A \in \mathbb{R}^{M \times N}$ and $B \in \mathbb{R}^{D \times M \times N}$. Here, A_{ij} is a scalar and B_{ij} is a vector of size D, So $A_{ij} \cdot B_{ij}$ is still a vector of size d. $R^{(t,j)} \odot M^{(n)}$ gives a tensor of the same shape as a unit $M_{t,j}^{(n)}$. Just as (d) shown in Fig. 3. Here we introduce $C^{(n)}$,

$$C_{t,j}^{(n)} = R^{t,j} \odot M^{(n)}. \tag{7}$$

When all calculations are completed, the shape of $C^{(n)}$ will have the same shape as the original feature map $M^{(n)}$. The whole process can be expressed in the form of an algorithm, as shown in Algorithm 1. The final $C^{(n)}$ is an enhanced feature map $M^{(n)}$. To make the whole process smoother, we did not directly replace the original feature map $M^{(n)}$ with $C^{(n)}$, but instead adopted a way to add $M^{(n)}$ to $C^{(n)}$ with suitable weight as following formula:

$$M_{final}^{(n)} = \varphi C^{(n)} + M^{(n)}, \tag{8}$$

where φ is not a hyperparameter, but is gradually increased from 0 as a normal parameter.

$M_{final}^{(n)}$ is the final enhanced feature map that can be directly involved in subsequent CNN operations. So far spatio-temporal-unit-based co-occurrence feature learning has been completed.

4. Experiments

In this section, we evaluated the proposed ASCS and STUFE on two major benchmarks NTU-RGB+D and SBU Kinect Interaction. We first introduced the evaluation metrics and experimental details, and finally analyzed the results. All experiments are conducted on GTX2080Ti.

4.1. Datasets

4.1.1 NTU-RGB+D

The NTU-RGB+D[22] is currently the most widely used and largest skeleton-based action recognition dataset. This dataset contains 56000 action samples in 60 action classes. These samples were taken by 40 volunteers in a laboratory environment. The sensor uses Microsoft's Kinect II[36],

Algorithm 1 algorithm for co-occurrence exploration

Input: $M^{(n)}$ feature map of the n^{th} layer
Parameter: Function $Distance()$
Output: $C^{(n)}$with the same shape as the $M^{(n)}$

for all $t \in [1, T_n], j \in [1, J_n]$ **do**
  Initialize $R^{(t,j)}$.
  for all $t' \in [1, T_n], j' \in [1, J_n]$ **do**
    $R^{(t,j)}_{t',j'} \leftarrow Distance(M^{(n)}_{t,j}, M^{(n)}_{t',j'})$
  end for
  Initialize $S \leftarrow 0$
  for all $t' \in [1, T_n], j' \in [1, J_n]$ **do**
    $S \leftarrow R^{(t,j)}_{t',j'} \cdot M^{(n)}_{t',j'} + S$
  end for
  $C^{(n)}_{t,j} \leftarrow S$
end for
return $C^{(n)}$

Method	CS	CV
Lie Group [25] *CVPR 2014*	50.1	52.8
H-RNN [4] *CVPR 2015*	59.1	64.0
Deep RNN [22] *CVPR 2016*	59.3	64.1
Deep LSTM [22] *CVPR 2016*	60.7	67.3
PA LSTM [22] *CVPR 2016*	62.9	70.3
ST-LSTM [18] *ECCV 2016*	62.9	70.3
STA-LSTM [24] *AAAI 2017*	73.4	81.2
Visualization CNN [20] *PR 2017*	76.0	82.6
VA-LSTM [33] *ICCV 2017*	79.4	87.6
Temporal Conv [12] *CVPRW 2017*	74.3	83.1
Clips + CNN + MTLN [10] *CVPR 2017*	79.6	84.8
Skepxels [17] *arXiv 2017*	81.3	89.2
HCN [15] *IJCAI 2018*	86.5	91.1
RHCN [Described in Sec. 4.2]	84.2	90.7
3D-POSE-S2 [21] *CVPR 2018*	82.4	86.7
ST-GCN [30] *AAAI 2018*	81.5	88.3
SR-TSL [30] *ECCV 2018*	84.8	92.4
motif-GCNs [28] *AAAI 2019*	84.2	90.2
STGR-GCN [14] *AAAI 2019*	**86.9**	92.3
RHCN + ACSC + STUFE	**86.9**	**92.5**

Table 1: Recognition performance on NTU-RGB+D dataset. We report the accuracies on cross-subject (CS) and cross-view (CV) benchmarks.

which can directly output the three-dimensional coordinates of each joint of the human body. Each sample consists of the skeleton containing 25 joints and each sample contains up to two subjects. All action classes are roughly divided as daily action, medical action, and interactive action. The authors of the NTU-RGB+D recommend two evaluation protocols. **1)Cross-subject** (CS): According to the different subjects, 40,320 samples are divided into training sets, and 16560 are used for testing. **2)Cross-view** (CV):Under this protocol, the training data comes from cameras at view 2 and 3, while the data from cameras with view 1 is used for testing. The two parts consist of 37,920 and 18,960 samples, respectively. In order to facilitate comparison with previous work, We use top1 accuracy as a benchmark.

4.1.2 SBU Kinect Interaction

SBU Kinect Interaction [32] is a dataset that focuses on two person interaction. All data is collected by Kinect. The entire dataset contains 282 skeleton sequences divided into 8 action classes. The whole skeleton sequence consists of 6822 frames. Since the acquisition device (Kinect I) is relatively behind the NTU-RGB+D (Kinect II), each skeleton contains only 15 joints. The evaluation protocol uses the subject-independent 5-fold cross validation recommended by the author [32].

4.2. Implementation Details

For the basic model of CNN, we chose HCN[15] for testing. Since the HCN model is not open source, it can only be reproduced according to the paper. However, in doing so, the performance will be slightly different from the original text. Here we call it RHCN. By the way, there is a performance gap between origin HCN (98.6) and RHCN(97.4) that we reproduced. This is because the author of HCN adjusted the network structure for the SBU dataset. Here, for a fair comparison, our RHCN is identical on both two datasets. For the multi-person samples, the following processing method is adopted. First, we default that each sample contains two persons. All samples in the SBU Kinect Interaction [32] contain exactly two persons, but some samples in the NTU-RGB+D [22] contain only one. For the single-person sample, we extend it by filling in zeros, and then use the element-wise maximum operation for feature fusion. Because according to experience[15], using such operations for fusion can reduce the impact of padded zeros. The model is optimized by the Adam optimization algorithm, and the learning rate is set to 0.001. The learning rate of the ASCS module and the STUFE module is set to 0.0001. The weight decay is 10^{-4} and the batch size is 64. Setting the learning rate in this way is to limit the fluctuation of parameters in ASCS and STUFE, which is especially important in the early stage of training.

For the GCN-based approach, the open source ST-GCN is directly used as the basis. Here, the main part of the model is optimized by the SGD algorithm with a learning rate of 0.01. Similarly, the proposed module uses a smaller learning rate of 0.001.Weight decay and batch size are the same as RHCN. All implementation work is based on the deep learning framework Pytorch 1.0.

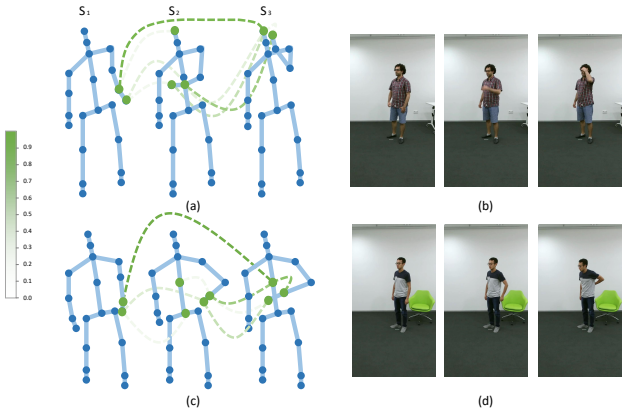

Figure 5: For the visualization of $R^{(t,j)}$, (a)(b)(c)(d) consists of three subgraphs, which represent three 'time points' S_1, S_2 and S_3. These samples are from the NTU-RGB+D dataset and (a)(b), (c)(d) are 'touching head' and 'touching back'.

Methods	Accuracy (%)			
	X=ST-GCN		X=RHCN	
	SBU	NTU	SBU	NTU
X	94.3	81.5	97.4	84.2
X + CCS	94.4	81.5	97.5	84.3
X + ACSC	94.8	81.7	97.7	84.9
X + STUFE	95.2	82.3	98.3	86.1
X + ACSC + STUFE	95.6	82.5	98.7	86.9

Table 2: Ablation study on the SBU Kinect Interaction dataset and NTU-RGB+D dataset CS benchmark. CCS refers to the cylindrical coordinate system.

4.3. Analysis of Results

4.3.1 Comparison on NTU-RGB+D

On NTU-RGB+D [22] dataset, our performance exceeds all previous methods under CV protocol, including the latest approach [28, 14] just introduced in 2019 as shown in Table 1. Our method exceeded it by 0.2% in top-1 accuracy. There are similar results in the CS protocol, except that we tied with STGR-GCN [14]. Even so, the number of parameters size is still smaller than STGR-GCN. Verifying the effectiveness of STUFE and ACSC is more important to us. So we verified them on CS protocol of NTU-RGB+D dataset based on both RHCN and ST-GCN as shown in Table 2. The method we propose can produce 2.7% improvement based on RHCN. The results fully demonstrate the effectiveness of our approach.

4.3.2 Comparison on SBU Kinect Interaction

The ablation study based on the SBU Kinect Interaction dataset is also shown in Table 2. Both ACSC and STUFE have been fully validated. In particular, for STUFE, there is a gain of 0.9% in the case where the base performance is already high. In Table 3, we compare the performance of proposed method to other methods'. Although we emphasize that our main contribution is not absolute performance, we still outperforms the state-of-the-art approaches.

Method	Accuracy (%)
Raw Position [32] *CVPRW 2012*	49.7
Joint feature [8] *ICMEW 2014*	86.9
CHARM [16] *ICCV2015*	86.9
H-RNN [4] *CVPR 2015*	80.4
ST-LSTM [18] *ECCV 2016*	88.6
Co-occurrence-LSTM [37] *AAAI 2016*	90.4
STA-LSTM [24] *AAAI 2017*	91.5
ST-LSTM + Trust Gate [18] *ECCV 2016*	93.3
VA-LSTM [33] *ICCV 2017*	97.6
RHCN + ACSC + STUFE	**98.7**

Table 3: Action recognition performance for skeleton-based models on the SBU Kinect Interaction dataset. We report Top-1 accuracy.

Method	training time (s)	increments(%)
RHCN	7023	baseline
RHCN+ACSC	7213	+2.71%
RHCN+STUFE	7721	+9.94%
RHCN+ACSC+STUFE	7962	+13.37%

Table 4: The increase in model's training time when ACSC and STUFE are added.

4.3.3 Learning cost

In addition to performance metrics, we also consider the learning time cost of the proposed method. As shown in Table 4. We compared the increase in training time after the ACSC and STUFE was introduced to RHCN respectively. From the results, ACSC and STUFE did not bring too much computational cost. We think this is reasonable compared to the performance improvement it brings.

4.3.4 Visualization

We have visualized R of GCN model, as shown in Fig.5. R contains the distance information between all spatio-temporal units. Considering that the size of R is very large the size of which is $T_n^2 \times J_n^2$, it is difficult to directly visualize it, so we simplified it. The length of the original 64 frames are merged as three segments, denoted as S_1, S_2, and S_3, respectively. At the same time, the hand joints are simplified, and the 25 joints are eventually merged into 21. Finally, the distance values are normalized and the distances with small values are filtered out. The larger cosine distance

value (the higher the correlation), the darker the color of the connecting line between units. It can be found that for 'touching back', the 'hand' at S_1 is highly correlated with the 'back' at S_3 . This is in line with our previous speculation about co-occurrence.

5. Conclusion

In this paper, we propose two novel methods for skeleton-based action recognition. Active coordinate system conversion (ACSC) can actively convert the coordinate system of the skeleton sample to a new coordinate system that is more conducive to model discriminationd, and STUFE can mine the co-occurrence features across temporal and spatial domain. Moreover, the proposed methods are compatible with the current mainstream GCN-based model and CNN-based model. In order to verify the effectiveness of proposed methods, we tested on two benchmarks and the accuracy eventually surpassed the previous state-of-the-art.

Acknowledgment

This work was partially supported the Natural Science Foundation of China under contracts 61572042, 61527804, 61425025, 61825101. This work was partially supported by Qualcomm. We also acknowledge the high-performance computing platform of Peking University for providing computational resources.

References

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 7291–7299, 2017. 2

[2] S. Das, A. Chaudhary, F. Bremond, and M. Thonnat. Where to focus on for human action recognition? In *IEEE Winter Conference on Applications of Computer Vision*, pages 71–80. IEEE, 2019. 2

[3] Y. Du, Y. Fu, and L. Wang. Skeleton based action recognition with convolutional neural network. In *Asian Conference on Pattern Recognition*, pages 579–583. IEEE, 2015. 2, 3

[4] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015. 2, 6, 7

[5] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Conference on Neural Information Processing Systems*, pages 2224–2232, 2015. 3

[6] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015. 3

[7] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 5308–5317, 2016. 2

[8] Y. Ji, G. Ye, and H. Cheng. Interactive body part contrast mining for human interaction recognition. In *IEEE International Conference on Multimedia and Expo Workshop*, pages 1–6. IEEE, 2014. 7

[9] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973. 1

[10] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. A new representation of skeleton sequences for 3D action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 4570–4579. IEEE, 2017. 1, 2, 3, 6

[11] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik. Intel RealSense stereoscopic depth cameras. *arXiv preprint arXiv:1705.05548*, 2017. 1, 2

[12] T. S. Kim and A. Reiter. Interpretable 3D human action analysis with temporal convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop Workshop*, pages 1623–1631. IEEE, 2017. 6

[13] I. Lee, D. Kim, S. Kang, and S. Lee. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *IEEE International Conference on Computer Vision*, pages 1012–1020, 2017. 2

[14] B. Li, X. Li, Z. Zhang, and F. Wu. Spatio-temporal graph routing for skeleton-based action recognition. *AAAI Conference on Artificial Intelligence*, 2019. 2, 3, 6, 7

[15] C. Li, Q. Zhong, D. Xie, and S. Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *arXiv preprint arXiv:1804.06055*, 2018. 2, 3, 6

[16] W. Li, L. Wen, M. Choo Chuah, and S. Lyu. Category-blind human action recognition: A practical recognition system. In *IEEE International Conference on Computer Vision*, pages 4444–4452, 2015. 7

[17] J. Liu, N. Akhtar, and A. Mian. Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition. *arXiv preprint arXiv:1711.05941*, 2017. 3, 6

[18] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal LSTM with trust gates for 3D human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016. 2, 6, 7

[19] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016. 2

[20] M. Liu, H. Liu, and C. Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017. 6

[21] M. Liu and J. Yuan. Recognizing human actions as the evolution of pose estimation maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 6

[22] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 1010–1019, 2016. 5, 6, 7

[23] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013. 3

[24] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI Conference on Artificial Intelligence*, volume 1, pages 4263–4270, 2017. 2, 6, 7

[25] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a Lie group. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2014. 1, 6

[26] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. O. Ogunbona. Action recognition from depth maps using deep convolutional neural networks. *IEEE Transactions on Human-Machine Systems*, 46(4):498–509, 2015. 2

[27] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera. Rgb-d-based human motion recognition with deep learning: A survey. *Computer Vision and Image Understanding*, 171:118–139, 2018. 2

[28] Y. Wen, L. Gao, H. Fu, F. Zhang, and S. Xia. Graph CNNs with motif and variable temporal block for skeleton-based action recognition. *AAAI Conference on Artificial Intelligence*, 2019. 3, 6, 7

[29] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 20–27. IEEE, 2012. 1

[30] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018. 2, 3, 6

[31] X. Yang and Y. Tian. Effective 3D action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 25(1):2–11, 2014. 1

[32] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2012. 6, 7

[33] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. *arXiv preprint arXiv:1703.08274*, 2017. 2, 6, 7

[34] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *Winter Conference on Applications of Computer Vision*, pages 148–157. IEEE, 2017. 2

[35] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer LSTM networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 148–157. IEEE, 2017. 2

[36] Z. Zhang. Microsoft Kinect sensor and its effect. *IEEE Multimedia*, 19(2):4–10, 2012. 1, 2, 5

[37] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *AAAI Conference on Artificial Intelligence*, volume 2, page 6, 2016. 2, 3, 7