Computer Aided Geometric Design 86 (2021) 101964

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Multi-derivative physical and geometric convolutional embedding networks for skeleton-based action recognition

Guoli Yan [a,*], Michelle Hua [b], Zichun Zhong [a]

[a] Wayne State University, Detroit, MI, United States
[b] Cranbrook Schools, Bloomfield Hills, MI, United States

ARTICLE INFO

Article history:
Available online 9 March 2021

Keywords:
Skeleton-based action recognition
Multi-derivative
Physical and geometric embedding
Multi-task learning

ABSTRACT

Action involves rich geometric and physical properties hidden in the spatial structure and temporal dynamics. However, there is a lack of synergy in investigating these properties and their joint embedding in the existing literature. In this paper, we propose a multi-derivative physical and geometric embedding network (PGEN) for action recognition from skeleton data. We model the skeleton joint and edge information using multi-derivative physical and geometric features. Then, a physical and geometric embedding network is proposed to learn co-occurrence features from joints and edges, respectively, and construct a unified convolutional embedding space, where the physical and geometric properties can be integrated effectively. Furthermore, we adopt a multi-task learning framework to explore the inter-dependencies between the physical and geometric properties of the action, which significantly improves the discrimination of the learned features. The experiments on the NTU RGB+D, NTU RGB+D 120, and SBU datasets demonstrate the effectiveness of our proposed representation and modeling method.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Human action recognition is an important research area in computer vision, computer graphics, virtual reality, etc., which has a wide range of applications, such as human-computer interaction, intelligent video surveillance, sports analysis and robotics. In this area, recognizing human actions from RGB videos has been studied for a long time, and recently, a lot of progress has been made Ji et al. (2012); Simonyan and Zisserman (2014a); Carreira and Zisserman (2017). However, it still suffers from the cluttered background, appearance variation, viewpoint change, occlusion, etc. On the other hand, with the advancement of low-cost depth cameras (e.g., Microsoft Kinect Zhang (2012)) and pose estimation algorithms Shotton et al. (2011); Yub Jung et al. (2015), the skeleton-based action recognition has attracted increasing research attention in recent years Du et al. (2015); Song et al. (2017); Yan et al. (2018). Compared with RGB videos, the skeletal representation is more robust to variations of camera locations, human appearances and backgrounds.

For skeleton-based action recognition, recent works mainly focus on graph convolutional networks (GCNs) for spatio-temporal modeling of skeleton sequences, due to its retainment of more spatial information with the graph structures Yan

---

The code (and data) in this article has been certified as Reproducible by Code Ocean: https://codeocean.com/. More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.

E-mail addresses: guoliyan@wayne.edu (G. Yan), mhua23@cranbrook.edu (M. Hua), zichunzhong@wayne.edu (Z. Zhong).

https://doi.org/10.1016/j.cagd.2021.101964
0167-8396/© 2021 Elsevier B.V. All rights reserved.

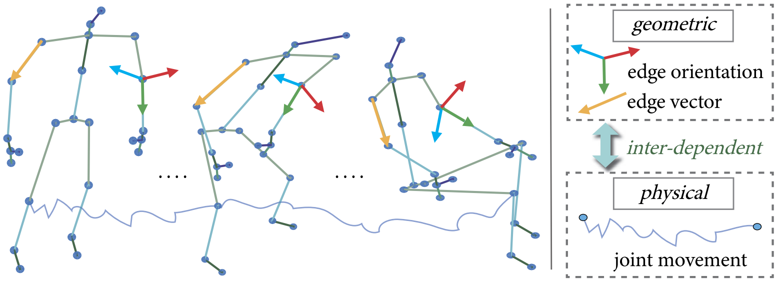

Fig. 1. The physical and geometric skeleton features. The multi-derivate geometric features are obtained from the edge vectors and orientations, while the multi-derivative physical features are derived from joint movement trajectories.

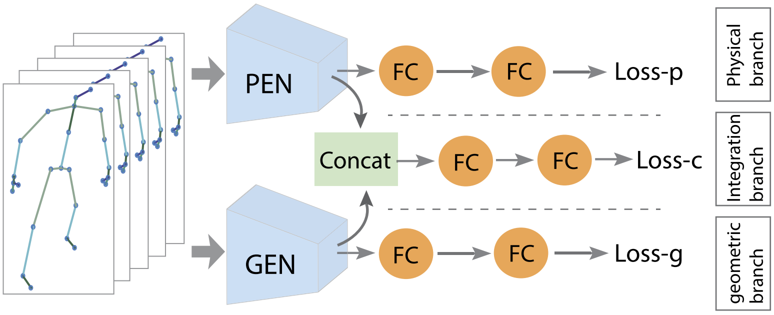

Fig. 2. Architecture of the entire network. The PEN (physical embedding network) and GEN (geometric embedding network) learn co-occurrence features from joints and edges, respectively, and map them to a unified convolutional embedding space for integration. FC indicates the fully connected layer. Loss-p, Loss-g and Loss-c are the losses of the three branches. A multi-task framework is adopted to explore the inter-dependencies between physical and geometric properties.

et al. (2018); Si et al. (2019); Shi et al. (2019a,b); Li et al. (2019). However, with an adjacency matrix used at each convolutional layer and a two-phase convolution for space and time, the GCN-based methods are typically quite time-consuming. By organizing the joints as columns and frames as rows in the same pseudo image, CNN-based methods perform convolutions on the input or internal feature maps directly and allow the spatial and temporal information to be integrated simultaneously Ke et al. (2017); Li et al. (2017, 2018), which is a much faster solution. But the representation used in existing CNN-based methods is still suboptimal due to the lack of connectivities among joints Ke et al. (2017); Li et al. (2018). It is still a challenging problem to represent and model the spatial and temporal information hidden in the action effectively and efficiently.

From our observation, action involves rich geometric and physical properties hidden in the spatial configuration and temporal dynamics. In order to understand the spatio-temporal evolutions of human poses, it is critical to represent the physical and geometric properties and integrate them effectively. To this end, we propose a multi-derivative physical and geometric convolutional embedding network for spatio-temporal analysis of human actions. Specifically, we represent skeleton joints with multi-derivative physical features (joint position, velocity and acceleration) and edges with multi-derivative geometric features (edge vector, orientation and their derivatives), as partially illustrated in Fig. 1. Each feature is organized as pseudo images by arranging frames as rows and joints or edges as columns. A novel physical and geometric embedding network (PGEN) with a multi-task framework (see Fig. 2) is proposed to explore the co-occurrence features as well as inter-dependencies between the physical and geometric properties. Our main contributions can be summarized as follows:

- We present a physical embedding network and a geometric embedding network to learn co-occurrence features from joints and edges, respectively, and map them into a unified embedding space to explore the inter-dependencies between the physical and the geometric properties;
- We propose novel multi-derivative features of skeleton joints and edges including joint acceleration, edge orientation and its derivative, to enhance the joint and edge representations used in existing works;
- The proposed CNN-based method outperforms the state-of-the-art approaches on three benchmarks. Even with more involving features, its time performance is about 7 times faster than the GCN-based method Shi et al. (2019b).

2. Related work

2.1. Representation of skeleton sequences

In the skeleton-based human centered research domains, skeleton sequences are often represented with physical features Ke et al. (2017); Li et al. (2017, 2018); Yan et al. (2018); Du et al. (2015) or geometric features Jain et al. (2016); Fragkiadaki et al. (2015); Martinez et al. (2017); Zhang et al. (2017b), and organized as chain of joints Fragkiadaki et al. (2015); Martinez et al. (2017); Du et al. (2015); Zhang et al. (2017b), pseudo images Ke et al. (2017); Li et al. (2017, 2018) or graphs Jain et al. (2016); Yan et al. (2018), in order to model the spatio-temporal evolution of human action states. For example, Fragkiadaki et al. (2015) and Jain et al. (2016) adopted the exponential map Grassia (1998) expressed joint orientation representation for motion prediction. To capture the spatio-temporal interactions between body parts, Jain et al. (2016) used spatio-temporal graphs (st-graphs) to structure the joint features. Zhang et al. (2017b) presented eight types of geometric features to model the spatial relations among joints, lines and planes for action recognition. Recently, Li et al. (2018) employed both the 3D coordinates of joints and the temporal difference of joint positions as joint features, and organized them as pseudo images. Different from the above methods, we represent skeleton sequences using both physical features of joints and geometric features of edges, and utilize their multi-order derivatives to constitute a novel and more comprehensive feature modeling.

2.2. Skeleton-based action recognition

In this section, we mainly review the recent deep learning methods for skeleton-based action recognition. Generally, there are three major types, including (1) using RNNs to capture the spatio-temporal evolutions from joint chains Du et al. (2015); Liu et al. (2016); Song et al. (2017); Zhang et al. (2017a); Lee et al. (2017); (2) employing CNNs to aggregate the spatial and temporal information hierarchically from the pseudo images Ke et al. (2017); Li et al. (2017, 2018); and (3) using GCNs to learn from the spatio-temporal graphs Yan et al. (2018); Si et al. (2019); Shi et al. (2019a,b); Li et al. (2019).

RNN-based methods represent each skeleton as a chain of joints and employ RNNs to capture the spatio-temporal evolutions. For instance, Song et al. (2017) used a multi-layer LSTM network for action recognition, with a spatial attention module to manipulate the input sequence data and a temporal attention module to assign different weights for each frame. CNN-based methods represent the skeleton sequences as pseudo images to allow CNNs be applied directly. For example, Ke et al. (2017) generated four gray images with respect to four reference joints, for each channel of the joint coordinates, and then, used the pre-trained VGG19 Simonyan and Zisserman (2014b) model to extract CNN features for action recognition. Li et al. (2018) generated 3-channel pseudo images instead. They proposed a Hierarchical Co-occurrence Network (HCN) to learn global features from all joints. Different from these methods, we construct pseudo images for both features of joints and edges, and explore their co-occurrences and inter-dependencies using the convolutional embedding networks and the multi-task framework. Recently, Yan et al. (2018) used a spatial-temporal graph representation of skeleton sequences and employed a GCN for action recognition. However, the information transmission between two correlated joints is not effective if they are not directly connected. To alleviate this, Shi et al. (2019a) used a directed graph to represent the skeleton and learned adaptive graph structures during training, which allows actional correlated joints to be connected directly. Different from these methods, the interactions of joints and edges in our method are not constrained by the connectivities due to our co-occurrence feature learning modules used in the network.

3. Multi-derivative physical and geometric convolutional embedding networks

The overall architecture of the proposed physical and geometric embedding method is shown in Fig. 2. We first calculate multi-derivative physical features from skeleton joints and multi-derivative geometric features from skeleton edges. Then, the physical embedding network (PEN) and geometric embedding network (GEN) are applied to learn from the physical and geometric features, respectively, and form a unified convolutional embedding space for integration. We use a multi-task learning framework to explore the inter-dependencies of the physical and geometric features. The entire network is trained in an end-to-end manner.

3.1. Multi-derivative physical features of joints

The movement of joint positions over time contains rich information for inferring different actions. For example, the temporal displacement of each joint between two consecutive frames, i.e., the velocity information, is commonly used for learning temporal evolutions in skeleton sequences Li et al. (2018); Si et al. (2018); Shi et al. (2019a). However, the acceleration information, which can be represented as the temporal difference of joint velocities, has not been explored in previous deep learning methods. Intuitively, different action classes embrace different distributions of joint accelerations along temporal dimension, which can serve as an auxiliary clue to distinguish between different classes. Therefore, we explicitly model the acceleration of joint movement and combine it with the position and velocity to represent the joint property multi-derivatively.

Specifically, the raw skeleton at frame t is represented as $\boldsymbol{P}^t \in \mathbb{R}^{N\times 3}$, where N is the number of joints, $\boldsymbol{P}^t_i \in \mathbb{R}^3$ denotes the 3D coordinate of i-th joint at time t. Then the joint accelerations at time t can be calculated as: $\boldsymbol{A}^t = \boldsymbol{V}^{t+1} - \boldsymbol{V}^t$, where $\boldsymbol{V}^t = \boldsymbol{P}^{t+1} - \boldsymbol{P}^t$. $\boldsymbol{V}^t$ and $\boldsymbol{A}^t$ can be considered as the first-order and second-order derivative of the joint coordinates, respectively. Finally, the multi-order derivatives are organized as $\boldsymbol{P}, \boldsymbol{V}, \boldsymbol{A} \in \mathbb{R}^{T\times N\times 3}$ tensors, where T is the number of frames in a sequence. Temporal interpolation is applied to the velocity and acceleration information in order to have consistent sequence length. The $\boldsymbol{P}$, $\boldsymbol{V}$ and $\boldsymbol{A}$ tensors can be viewed as pseudo images, whose rows, columns, and channels correspond to frames, joints, and 3D features, respectively.

3.2. Multi-derivative geometric features of edges

To better capture the spatio-temporal information in the skeleton data, we also represent the skeleton sequence from a geometric perspective. The geometric features are obtained from the skeleton edges. As shown in Fig. 3(b), a skeleton model can be transformed into a hierarchical tree structure, where the parent-child relationships are determined between connected joints once the tree root is settled. The connections from parent to child joints are referred as edges, denoted as $\varepsilon_i = \langle p_i, c_i\rangle$, where p_i and c_i are the corresponding indices of parent joint and child joint of edge i.

Edge orientations. First, we present a novel edge orientation feature. The skeletal human body is an articulated system of rigid segments (edges). At each time step, the edges present different orientation relationships with respect to the camera space as they move. In order to quantitatively describe the orientations, we define a local coordinate system for each edge. The edge orientation is then defined as yaw, pitch, and roll angles of the local coordinate system to align with the camera view, as shown in Fig. 3(a). Specifically, each local coordinate system is an orthonormal basis of $\mathbb{R}^3$ space, which is constituted by three unit basis vectors, i.e., $\boldsymbol{u}_{t,i}$, $\boldsymbol{v}_{t,i}$ and $\boldsymbol{n}_{t,i}$, where $t \in \{1, ..., T\}$ is the frame index and $i \in \{1, ..., N-1\}$ is the edge index. The direction of $\boldsymbol{v}_{t,i}$ is consistent with the edge direction pointing from parent joint p_i to child joint c_i, i.e., $\boldsymbol{v}_{t,i} = (\boldsymbol{P}^t_{c_i} - \boldsymbol{P}^t_{p_i}) \big/ \left\| \boldsymbol{P}^t_{c_i} - \boldsymbol{P}^t_{p_i} \right\|$ where $\boldsymbol{P}^t_{c_i}$ denotes the 3D coordinate of c_i-th joint at time t. Different from $\boldsymbol{v}_{t,i}$, the direction of $\boldsymbol{n}_{t,i}$ is considered in two cases. For the limb edges, the $\boldsymbol{n}_{t,i}$ vector is orthogonal to both edge $\varepsilon_i = \langle p_i, c_i\rangle$ and edge $\varepsilon_{i-1} = \langle p_{i-1}, c_{i-1}\rangle$, therefore, $\boldsymbol{n}_{t,i}$ can be obtained by $\boldsymbol{n}_{t,i} = \frac{(\boldsymbol{P}^t_{c_i}-\boldsymbol{P}^t_{p_i})\times(\boldsymbol{P}^t_{p_{i-1}}-\boldsymbol{P}^t_{c_{i-1}})}{\left\|(\boldsymbol{P}^t_{c_i}-\boldsymbol{P}^t_{p_i})\times(\boldsymbol{P}^t_{p_{i-1}}-\boldsymbol{P}^t_{c_{i-1}})\right\|}$, where $p_i = c_{i-1}$, and "×" denotes cross product operator. For other edges, i.e., edges of torso and head, the direction of $\boldsymbol{n}_{t,i}$ is consistent with the facing direction of the human body. We can get forward facing directions by taking the cross product of the y-axis and the vector across two shoulders, or the vector across two hips. It is distance-dependent as for taking which facing direction as $\boldsymbol{n}_{t,i}$'s direction. Finally, the $\boldsymbol{u}_{t,i}$ vector can be obtained by $\boldsymbol{u}_{t,i} = \boldsymbol{v}_{t,i} \times \boldsymbol{n}_{t,i}$. And the $\boldsymbol{n}_{t,i}$ vector can be corrected as $\boldsymbol{n}_{t,i} = \boldsymbol{u}_{t,i} \times \boldsymbol{v}_{t,i}$. In this way, the set of vectors $\{\boldsymbol{u}_{t,i}, \boldsymbol{v}_{t,i}, \boldsymbol{n}_{t,i}\}$ forms an orthonormal basis in 3D space. Then, the rotation matrix aligning the local basis with the global camera view can be expressed as:

$$\boldsymbol{R}_{t,i} = \begin{bmatrix} \boldsymbol{u}_{t,i} \\ \boldsymbol{v}_{t,i} \\ \boldsymbol{n}_{t,i} \end{bmatrix} = \begin{bmatrix} u^x, u^y, u^z \\ v^x, v^y, v^z \\ n^x, n^y, n^z \end{bmatrix}. \tag{1}$$

Finally, the Euler angle representation of the rotation can be expressed as follows if taking the rotation in xyz order:

$$\begin{aligned} \alpha &= atan2(n^y, n^z), \\ \beta &= atan2\left(-n^x, \sqrt{(n^y)^2 + (n^z)^2}\right), \\ \gamma &= atan2(v^x, u^x), \end{aligned} \tag{2}$$

where α, β and γ are the rotation angles around the x, y and z-axis, respectively. Then the edge orientation can be represented as a 3D vector $\boldsymbol{o}_{t,i} = (\alpha, \beta, \gamma)$.

The derivatives of edge orientation. The derivative of edge orientation can be simplified as the difference of orientations between two consecutive frames, i.e., $\boldsymbol{o}'_{t,i} = \boldsymbol{o}_{t+1,i} - \boldsymbol{o}_{t,i}$, where $\boldsymbol{o}'_{t,i}$ is the Euler angle representation. It implies the geometric rotation direction and magnitude at time t.

Edge vectors and their derivatives. We also include edge vectors Shi et al. (2019b,a) and their temporal derivatives to enrich the edge representation. We denote edge vectors as $\boldsymbol{e}_{t,i} = \boldsymbol{P}^t_{c_i} - \boldsymbol{P}^t_{p_i}$, where p_i and c_i are the indices of parent joint and child joint of edge i (see Fig. 3(b)), and edge derivatives as $\boldsymbol{e}'_{t,i} = \boldsymbol{e}_{t+1,i} - \boldsymbol{e}_{t,i}$.

Similar to the physical features of joints, the geometric features of edges in a sequence, i.e., the orientations $\boldsymbol{O} = \left(\boldsymbol{o}_{t,i}\right)$, orientation differences $\boldsymbol{O}' = \left(\boldsymbol{o}'_{t,i}\right)$, edge vectors $\boldsymbol{E} = \left(\boldsymbol{e}_{t,i}\right)$ and edge vector differences $\boldsymbol{E}' = \left(\boldsymbol{e}'_{t,i}\right)$, are organized as 3D tensors of shape $T \times (N-1) \times 3$, where T is the number of frames and $N-1$ is the number of edges. The $\boldsymbol{O}$, $\boldsymbol{O}'$, $\boldsymbol{E}$ and $\boldsymbol{E}'$ tensors can be viewed as pseudo images, whose rows, columns, and channels correspond to frames, edges, and 3D features, respectively.

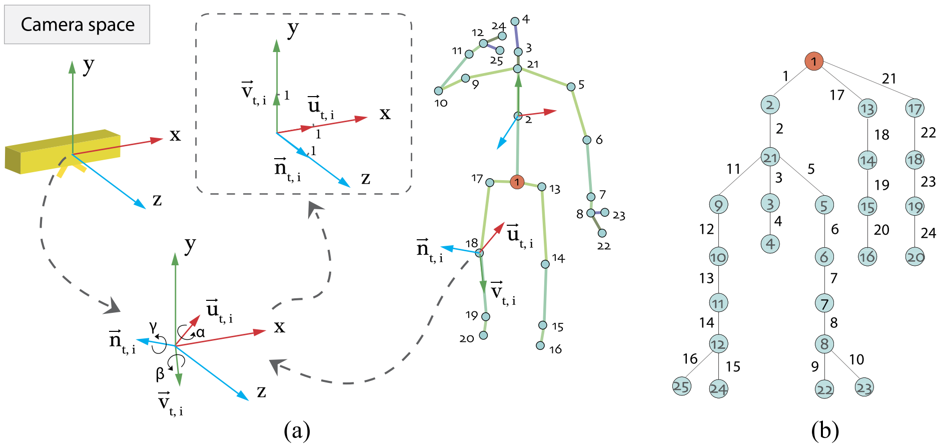

Fig. 3. (a) Illustration of the edge orientation. $\boldsymbol{u}_{t,i}$, $\boldsymbol{v}_{t,i}$ and $\boldsymbol{n}_{t,i}$ are mutually orthogonal unit vectors that constitute a local basis of edge i at time t. The orientation is defined as the Euler angle represented rotation for aligning the local basis of an edge with the camera space. (b) A tree representation of the skeleton that displays the joint hierarchy relationships. The "base of the spine" is chosen as the root node (red color). The numbers attached to the joints and edges indicate the indices of joints and edges. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

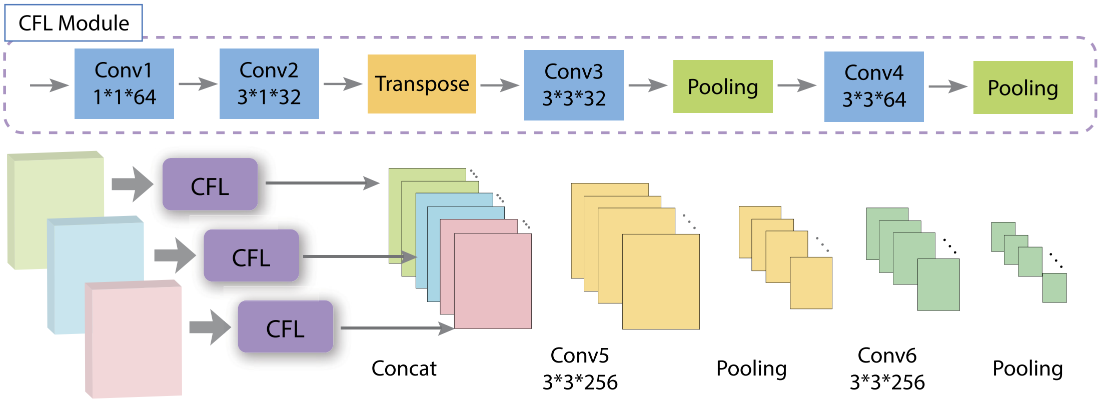

Fig. 4. The architecture of the physical embedding network (PEN). The position, velocity, and acceleration data are separately sent to three co-occurrence feature learning (CFL) modules. The CFL outputs are concatenated along channels before feeding to the next layer. The architecture of the geometric embedding network (GEN) is similar to PEN, except that the GEN is trained with four geometric features of edges and the CFL module is used to learn the co-occurrence feature of edges rather than joints.

3.3. Physical and geometric embedding network

Co-occurrence feature learning module. Our PGEN network incorporates the co-occurrence feature learning (CFL) module. As shown in Fig. 4, it contains 4 convolutional layers. Temporal dynamics is encoded with convolutions with kernel size of 3 along the time dimension, except for the first layer with kernel size of 1. The output tensor of the second convolution layer is transposed to make the joint dimension as channel, so that joints' information can be aggregated globally through element-wise summation across channels during the convolutional operation.

Physical embedding network. The physical embedding network (PEN) is learned with the multi-derivative physical features of skeleton joints. As shown in Fig. 4, the position, velocity, and acceleration data are separately sent to three co-occurrence feature learning (CFL) modules, and fused at a higher layer by concatenating their feature maps along channel dimension. After that, two convolutional layers together with two max-pooling layers are employed to learn high level motion patterns from the skeleton joints.

Geometric embedding network. The structure of the geometric embedding network (GEN) is similar to the PEN shown in Fig. 4. The only difference is that the GEN is trained with four geometric features of edges, as described in Section 3.2, and the CFL module here is used to learn the co-occurrence feature of edges rather than joints.

Multi-task learning framework. As shown in Fig. 2, two linear layers are appended after the PEN, with neuron sizes of 256 and C, respectively, where C is the number of classes. The second linear layer generates the class scores $\boldsymbol{s}^p = (s_1, s_2, \ldots, s_C)$ for the input sequence. Then the probability of being the i-th class can be calculated using the softmax function,

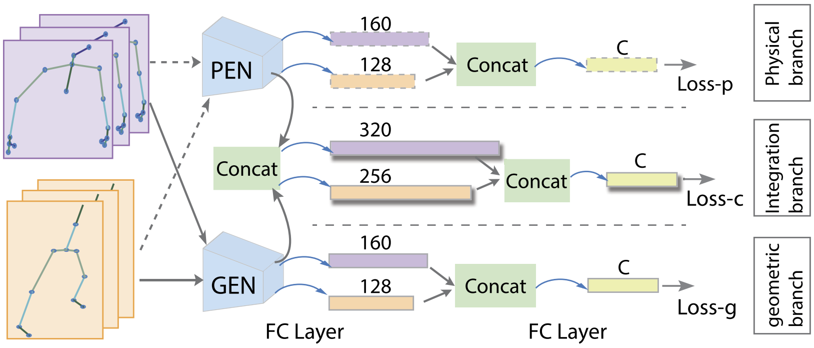

Fig. 5. Architecture of the part-based model. The upper-body inputs and features are denoted by purple, while the lower-body inputs and features by orange. The rectangles in dotted boxes (physical branch), solid-line boxes (geometric branch) and shadowed boxes (integrate branch) denote the output features of fully connected (FC) layers. The numbers on top are the neuron sizes of FC layers. C is the number of classes.

$$\hat{y}_i^p = \frac{e^{s_i}}{\sum_{k=1}^{C} e^{s_k}}, i = 1, ..., C. \tag{3}$$

Also, two linear layers are appended after the GEN, with neuron sizes of 256 and C, respectively. Similarly, we can get the class scores $\boldsymbol{s}^g$ for the C classes, and the probability of being the i-th class $\hat{y}_i^g$ by softmax. To combine both the physical information of the joints and the geometric information of the edges, we concatenate the outputs of PEN and GEN along channel dimension. The concatenated features are then fed into two linear layers with neuron sizes of 512 and C, respectively, to produce the class scores $\boldsymbol{s}^c$ for the C classes. The probability of being the i-th class can be also calculated, which is denoted as $\hat{y}_i^c$.

Each of the classifications performed on the physical features of the joints, the geometric features of the edges and the concatenated features is treated as a separate task. We use the cross entropy loss to learn with supervision for each task. The final loss function is defined as the sum of the three losses, i.e.,

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_p + \mathcal{L}_g + \mathcal{L}_c \\ &= -\sum_{i=1}^{C} y_i \log \hat{y}_i^p - \sum_{i=1}^{C} y_i \log \hat{y}_i^g - \sum_{i=1}^{C} y_i \log \hat{y}_i^c\end{aligned} \tag{4}$$

where $\boldsymbol{y} = (y_1, ..., y_C)$ is the one-hot encoding vector of the ground-truth label.

Through multi-task learning, the inter-dependencies of the physical and the geometric information are further explored during back-propagation, which brings performance boosting for each single task. In the test phase, the class scores of the concatenated features ($\boldsymbol{s}^c$) are used for predicting the action class.

3.4. Part-based model

The co-occurrence feature learning (CFL) module aims to learn co-occurrence features from all joints (or edges) of a skeleton sequence, which is beneficial for modeling actions with long-range joint interactions, such as putting on a shoe. However, for actions involving interactions within only a small local region of joints, it may dilute the effective information due to the lack of focus. Therefore, in addition to modeling on the whole human body, we also model on human parts to capture co-occurrence features within a part. Specifically, we consider two body parts, i.e., the upper body and the lower body. As shown in Fig. 5, for each part, the physical and geometric embeddings are combined, as the whole-body-based model, and the features of the upper and lower parts are combined after the first linear layer. Finally, the class scores of the whole-body-based and part-based models are fused to boost the performance.

4. Experiments

4.1. Datasets and settings

NTU RGB+D dataset Shahroudy et al. (2016) is a large-scale dataset for 3D action recognition, with over 56,000 action sequences and 4 million frames, collected from 40 different subjects using three cameras. It covers 60 action classes, including 49 classes of actions performed by single person, and 11 classes of two-person interactions. The actions are captured from 80 distinct camera views. The 3D coordinates of 25 body joints are provided for each skeleton. There are two standard evaluation protocols in this dataset, i.e., Cross-Subject (CS) and Cross-View (CV). In CS setting, samples from 20 subjects are used for training, and the remaining 20 subjects are for testing. In CV setting, samples of front and side views constitute the training set, and the remaining samples of 45° views are for testing.

Table 1
Accuracy (%) comparison on the NTU RGB+D for Cross-Subject (CS) and Cross-View (CV) settings.

Methods	CS	CV
HBRNN-L Du et al. (2015)	59.1	64.0
Part-aware LSTM Shahroudy et al. (2016)	62.9	70.3
ST-LSTM+TrustGate Liu et al. (2016)	69.2	77.7
STA-LSTM Song et al. (2017)	73.4	81.2
GCA-LSTM Liu et al. (2017b)	74.4	82.8
VA-LSTM Zhang et al. (2017a)	79.4	87.6
ST-GCN Yan et al. (2018)	81.5	88.3
SR-TSL Si et al. (2018)	84.8	92.4
HCN Li et al. (2018)	86.5	91.1
AS-GCN Li et al. (2019)	86.8	94.2
2s-AGCN Shi et al. (2019b)	88.5	95.1
AGC-LSTM Si et al. (2019)	89.2	95.0
DGNN Shi et al. (2019a)	89.9	**96.1**
PL-GCN Huang et al. (2020)	89.2	95.0
NAS-GCN Peng et al. (2020a)	89.4	95.7
PGEN (w/o partition)	89.3	94.9
PGEN (w/ partition)	89.7	95.0
PGEN-fusion	**90.3**	95.6

NTU RGB+D 120 dataset Liu et al. (2019) is the extended version of NTU RGB+D dataset. It contains over 114,000 action sequences and 8 million frames collected from 106 subjects with 120 action classes, including 94 classes of single-person actions and 26 classes of two-person interactions. It has 32 collection setups with various camera heights and distances, resulting 155 views from three cameras. Each skeleton has 25 joints. Compared to the actions in NTU RGB+D, the added actions in NTU RGB+D 120 dataset involve more object-related and fine-grained motions that are hard to distinguish. There are two evaluation settings in this dataset. One is Cross-Subject evaluation, i.e., samples from 53 subjects are used for training and the remaining 53 subjects are for testing. The other is Cross-Setup evaluation, i.e., samples with even setup IDs as the training set while samples with odd setup IDs as the testing set.

SBU Kinect Interaction dataset Yun et al. (2012) contains 282 skeleton sequences and 6822 frames. Each frame contains two human subjects' skeletons performing an interaction and each skeleton consists of 15 joints. There are 8 classes of two-person interactions. A 5-fold cross validation is used for evaluation.

4.2. Implementation details

For the three datasets, we translate the original 3D coordinates to the new coordinate system with the origin at the center of the "middle of the spine" joints of the two bodies in each frame. For the samples with only one person, the second body is padded with zeros.

For data augmentation, at each iteration we randomly crop a sub-sequence from a training sample with a ratio of $\mathcal{R}$, which is in range $[0.6, 1]$. For the NTU RGB+D and NTU RGB+D 120, the cropped sequences are normalized to a fixed length, $T = 64$, by interpolation. For the SBU dataset, $T = 16$. In testing, we center-crop a sub-sequence with a ratio of 0.95. To alleviate overfitting, dropout with a probability of 0.5 is employed for conv4, conv5, conv6 and the first FC layer of each branch. In addition, since the SBU dataset is rather small, it uses a simplified version of CFL module, i.e., the conv4 layer is removed and the output channels of the first three layers are reduced to 32, 16, and 16, respectively.

The learning rate is initialized to 0.001 and decayed by multiplying 0.99 after each epoch. We use the Adam Kingma and Ba (2014) optimizer for optimization. The batch size is set to 64, 64, 16, and the network is trained with 400, 400, 200 epochs for the NTU RGB+D, NTU RGB+D 120 and SBU dataset, respectively. The source code of the framework will be made available later in Github.

4.3. Comparison to other state-of-the-arts

We compare the performance of our method with recent state-of-the-art methods on skeleton-based action recognition using the NTU RGB+D, NTU RGB+D 120 and SBU datasets. The results are shown in Table 1, Table 2 and Table 3. In the tables, PGEN (w/o partition) means the whole-body-based physical and geometric embedding method, PGEN (w/ partition) is the part-based method that models the upper and lower body parts separately, and PGEN-fusion is the fused network of the above two.

On the NTU RGB+D dataset (Table 1), compared with SR-TSL Si et al. (2018), which is the current best LSTM-based method for skeleton-based action recognition, the results of our method are 5.5% and 3.2% better in the cross-subject setting and cross-view setting, respectively. This is because SR-TSL models spatial and temporal information separately, while our method can integrate the spatial and temporal properties simultaneously through convolution. HCN Li et al. (2018) is the current best CNN-based method. Our method outperforms HCN by 3.8% and 4.5% for cross-subject and cross-view,

Table 2
Accuracy (%) comparison on the NTU RGB+D 120 for Cross-Subject (X-Sub) and Cross-Setup (X-Set) settings.

Methods	X-Sub	X-Set
Dynamic Skeleton Hu et al. (2015)	50.8	54.7
ST-LSTM+Trust Gate Liu et al. (2016)	55.7	57.9
GCA-LSTM Liu et al. (2017b)	58.3	59.2
Skeleton Visualization Liu et al. (2017c)	60.3	63.2
Two-Stream Attention LSTM Liu et al. (2017a)	61.2	63.3
Multi-Task+RotClips Ke et al. (2018)	62.2	61.8
Pose Evolution Map Liu and Yuan (2018)	64.6	66.9
ST-GCN Yan et al. (2018)	70.7	73.2
SR-TSL Si et al. (2018)	74.1	79.9
2s-AGCN Shi et al. (2019b)	82.5	84.2
AS-GCN Li et al. (2019)	77.7	78.9
3s RA-GCN Song et al. (2020)	81.1	82.7
Mix Dimension Peng et al. (2020b)	80.5	83.2
PGEN (w/o partition)	83.0	84.1
PGEN (w/ partition)	81.7	83.5
PGEN-fusion	**83.9**	**85.2**

Table 3
Accuracy performance comparison on the SBU Interaction dataset. The average results of the 5-fold cross validation are shown in the table.

Methods	Acc. (%)	Year
Co-occurrence RNN Zhu et al. (2016)	90.4	2016
ST-LSTM+Trust Gate Liu et al. (2016)	93.3	2016
STA-LSTM Song et al. (2017)	91.5	2017
GCA-LSTM Liu et al. (2017b)	94.1	2017
VA-LSTM Zhang et al. (2017a)	97.2	2017
Multi-Task CNN + RotClips Ke et al. (2018)	94.2	2018
HCN Li et al. (2018)	98.6	2018
VA-fusion Zhang et al. (2019)	98.3	2019
PGEN (w/o partition)	**98.9**	-
PGEN (w/ partition)	96.5	-
PGEN-fusion	**98.9**	-

respectively. The accuracy improvement indicates the importance of exploiting the connectivities of joints. DGNN Shi et al. (2019a) is the current state-of-the-art on the NTU RGB+D dataset, which is a GCN-based method. Compared with DGNN, our result is better for cross-subject evaluation, which reflects that our proposed multi-derivative physical and geometric features are discriminative to characterize actions intrinsically, while less affected by the performers. For the cross-view setting, DGNN performs better due to the adaptive graph structures it applied at each convolutional layer. However, this requires additional computational cost with high time complexity, which may make it hard to extend on larger datasets.

Since NTU RGB+D 120 is a newly published dataset, many recent state-of-the-art methods have not been evaluated on this dataset. We include all the results of RNN-based and CNN-based methods reported in Liu et al. (2019) as well as the results of recent GCN-based methods that reported in Song et al. (2020) for comparison in Table 2. Our method outperform the reported RNN-based and CNN-based methods by a large margin. Compared with GCN-based methods, our method consistently outperform ST-GCN Yan et al. (2018) by 13.2% and 12.0%, AS-GCN Li et al. (2019) by 6.2% and 6.3%, and 2s-AGCN Shi et al. (2019b) by 1.4% and 1.0% for the cross-subject and cross-setup evaluation, respectively.

On the SBU Interaction dataset (Table 3), our method achieves 98.9% accuracy, which is 1.7% better than the current best LSTM-based method for this dataset (VA-LSTM Zhang et al. (2017a)). VA-fusion Zhang et al. (2019) implements a fusion of VA-LSTM and VA-CNN networks. Our method outperforms VA-fusion by 0.6%. HCN Li et al. (2018) is the current state-of-the-art method on the SBU dataset. Our method still outperforms HCN by 0.3%.

To verify the efficiency advantage of our CNN-based method over the prevalent GCN-based methods, we also conduct the time performance comparison between our method and two GCN-based methods ST-GCN Yan et al. (2018) and 2s-AGCN Shi et al. (2019b). According to our experiments on CPU (i7-6700), the test times of a single sample using ST-GCN, 2s-AGCN, and our whole model are 0.232s, 0.667s, and 0.085s, respectively. Hence our method is 7∼8 times faster than 2s-AGCN and 2∼3 times faster than ST-GCN at the inference phase.

4.4. Model analysis

In Table 4, we compare our model with several variants on the NTU RGB+D dataset for cross-subject evaluation. Specifically, we design the following baselines: “PEN + MLP” denotes the PEN module followed by two fully connected layers. “GEN + MLP” denotes the GEN module followed by two fully connected layers. Compared with “PEN + MLP”, the baseline

Table 4
Comparison with the baselines on the NTU RGB+D dataset with CS setting.

	Methods	Acc. (%)
W/o partition	PEN + MLP	87.1
	GEN + MLP	87.3
	PEN w/o second-order feature	86.2
	GEN w/o second-order feature	86.8
	PEN + GEN w/o multi-task	87.4
	PG-network w/o splitting	88.0
	PEN + GEN (PGEN)	89.3
W/ partition	PEN + GEN (PGEN)	89.7
	PGEN-fusion	**90.3**

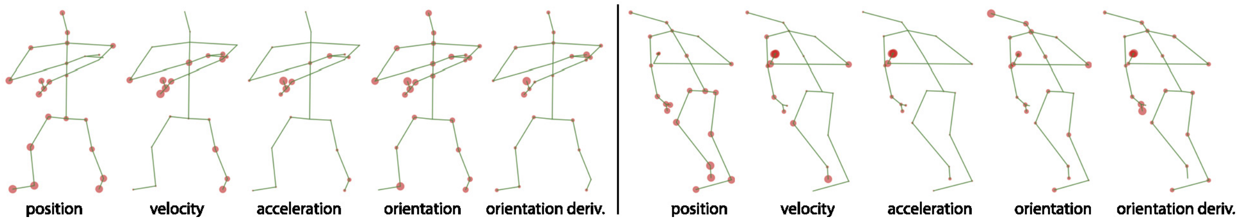

Fig. 6. Visualization of individual joint/edge response of a certain frame learned by our PGEN at conv2. The left panel shows the “writing” case while the “touch chest” on the right. The feature inputs to which the joint/edge responses correspond are listed at the bottom.

“PEN w/o second-order feature” is trained without the acceleration feature. Compared with “GEN + MLP”, the baseline “GEN w/o second-order feature” is trained without the derivative of orientation feature. Compared with our proposed PGEN, “PEN + GEN w/o multi-task” is trained without multi-task learning, i.e., the loss function $\mathcal{L} = \mathcal{L}_c$. Compared with PGEN, “PG-network w/o splitting” denotes the network trained with all the multi-derivative features, without splitting into physical branch and geometric branch.

Compared with “PEN + MLP”, the accuracy of “PEN w/o second-order feature” is dropped by 0.9%. Compared with “GEN + MLP”, the accuracy of “GEN w/o second-order feature” is dropped by 0.5%. These results show the effectiveness of using multi-derivative physical and geometric features. The accuracy of “PEN + GEN w/o multi-task” is dropped by 1.9% compared with the full model, which demonstrates the effectiveness of using a multi-task learning framework to explore the inter-dependencies of the physical and geometric information during back-propagation. Compared with “PG-network w/o splitting”, our PGEN improves the accuracy by 1.3%, which verifies the effectiveness of the learned embedding space, where the physical and geometric properties are better integrated.

4.5. *Visualization*

In this section, we use visualization techniques to intuitively understand what PGEN learns from the physical and geometric feature inputs. Fig. 6 shows the individual joint/edge response of a certain frame learned by our PGEN at conv2. The area of joint indicates the magnitude of joint response. The edge response is indicated by the area of edge’s end joint. The physical features well capture the joint movement, while the geometric features (showing only two due to space) capture the posture/shape well. We can observe that the new acceleration and edge orientation/derivative feature complement the traditional joint and edge features by emphasizing attention on high dynamic joints and edges, which improves the performance. Our PGEN can correctly recognize both cases, while “GEN+MLP” baseline misclassifies the “writing” case as “reading” and “PEN+MLP” baseline misclassifies the “touch chest” case as “touch back”. That is because “writing” and “reading”, which have similar postures, are hard to distinguish with geometric features only, and “touch chest” and “touch back”, which have similar movement of arms, are hard to distinguish with physical features only.

Second, we visualize the heatmaps of class activation Zhou et al. (2016) at the last convolutional layer (conv6) of the PEN and GEN modules. The heatmaps are resized to the input size of the network, with height 64, i.e., the number of frames. As shown in Fig. 7, the first column shows the heatmaps from PEN, while the second column shows the heatmaps from GEN. The subsequent columns show the skeletons from a series of frames in the corresponding input sample. The first row shows the visualization results of a “drink water” sample, while the second row is a “brush hair” sample. From the first row, we find that the PEN heatmap highlight the region that corresponding to a period of raising the right arm (see red bounding boxes on the heatmap and the skeleton sequence), while the GEN heatmap highlight the region that corresponding to a period of drinking water (see black bounding boxes). From the second row, we find that both PEN and GEN heatmaps almost highlight the same region that corresponding to a period of brushing hair with right arm. Based on these, we conclude that PEN tend to focus on time periods with significant joint movements, while GEN tend to focus on time periods with discriminative body postures/shapes—whether accompanied with tiny motion (e.g. the period of drinking water) or big

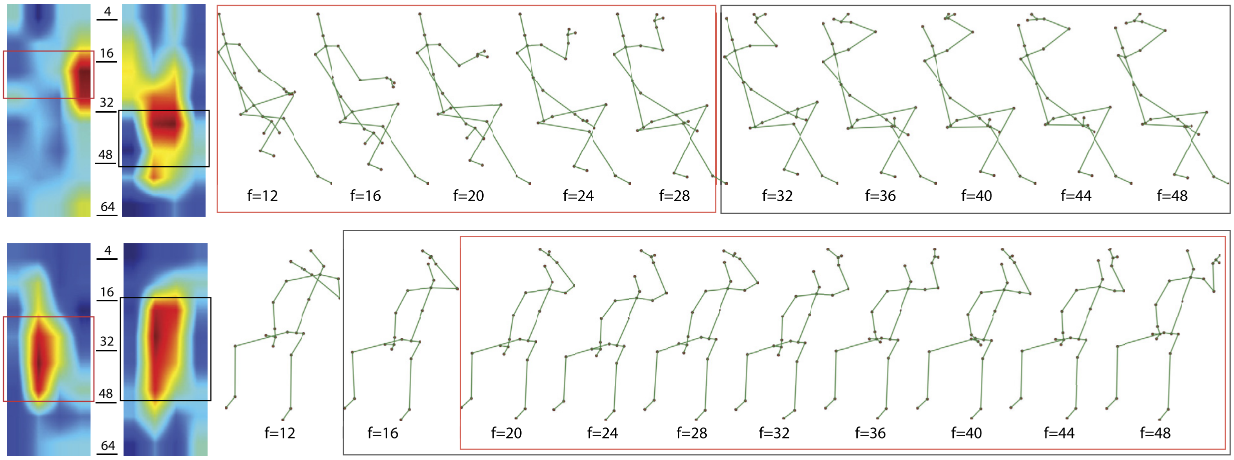

Fig. 7. Visualization of PGEN. The first column and second column show the conv6 heatmaps from PEN and GEN, respectively. The subsequent columns show the skeletons from the corresponding input sample, with frame indices listed below. The first row shows a "drink water" sample, while the second row is a "brush hair" sample.

motion (e.g., the period of brushing hair). Through exploiting feature inter-dependencies in the unified embedding, PGEN takes the merits of both PEN and GEN by complementing each other.

5. Conclusion

We have presented a coordinated, multi-derivative physical and geometric embedding network (PGEN) for skeleton-based action recognition. The skeleton joints and edges are represented with multi-derivative physical and geometric features, respectively. A physical and geometric convolutional embedding network is applied to learn co-occurrence features from joints and edges and construct a unified embedding space of physical and geometric features, where they can be integrated effectively. Our multi-task learning framework is useful for further exploring the inter-dependencies of physical and geometric properties to obtain a more discriminative feature representation for action recognition. The experiments on the NTU RGB+D, NTU RGB+D 120, and SBU datasets have verified the effectiveness and superiority of our proposed representation and learning framework. The proposed framework has the potential to be extended to other motion analysis problems involving different physical and geometric properties. We will employ our framework for non-articular, deformable motion analysis in the future, e.g., the heart motion abnormality recognition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Carreira, J., Zisserman, A., 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308.

Du, Y., Wang, W., Wang, L., 2015. Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1110–1118.

Fragkiadaki, K., Levine, S., Felsen, P., Malik, J., 2015. Recurrent network models for human dynamics. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4346–4354.

Grassia, F.S., 1998. Practical parameterization of rotations using the exponential map. J. Graph. Tools 3, 29–48.

Hu, J.F., Zheng, W.S., Lai, J., Zhang, J., 2015. Jointly learning heterogeneous features for RGB-D activity recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5344–5352.

Huang, L., Huang, Y., Ouyang, W., Wang, L., 2020. Part-level graph convolutional network for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 11045–11052.

Jain, A., Zamir, A.R., Savarese, S., Saxena, A., 2016. Structural-RNN: deep learning on spatio-temporal graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5308–5317.

Ji, S., Xu, W., Yang, M., Yu, K., 2012. 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. 35, 221–231.

Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F., 2017. A new representation of skeleton sequences for 3D action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3288–3297.

Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F., 2018. Learning clip representations for skeleton-based 3D action recognition. IEEE Trans. Image Process. 27, 2842–2855.

Kingma, D.P., Ba, J., 2014. Adam: a method for stochastic optimization. arXiv preprint. arXiv:1412.6980.

Lee, I., Kim, D., Kang, S., Lee, S., 2017. Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1012–1020.

Li, C., Zhong, Q., Xie, D., Pu, S., 2017. Skeleton-based action recognition with convolutional neural networks. In: Proceedings of the IEEE International Conference on Multimedia & Expo Workshops, pp. 597–600.

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

Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q., 2019. Actional-structural graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3595–3603.

Liu, J., Shahroudy, A., Perez, M.L., Wang, G., Duan, L.Y., Chichung, A.K., 2019. NTU RGB+D 120: a large-scale benchmark for 3D human activity understanding. IEEE Trans. Pattern Anal. Mach. Intell. 42, 2684–2701.

Liu, J., Shahroudy, A., Xu, D., Wang, G., 2016. Spatio-temporal LSTM with trust gates for 3D human action recognition. In: European Conference on Computer Vision. Springer, pp. 816–833.

Liu, J., Wang, G., Duan, L.Y., Abdiyeva, K., Kot, A.C., 2017a. Skeleton-based human action recognition with global context-aware attention LSTM networks. IEEE Trans. Image Process. 27, 1586–1599.

Liu, J., Wang, G., Hu, P., Duan, L.Y., Kot, A.C., 2017b. Global context-aware attention LSTM networks for 3D action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1647–1656.

Liu, M., Liu, H., Chen, C., 2017c. Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognit. 68, 346–362.

Liu, M., Yuan, J., 2018. Recognizing human actions as the evolution of pose estimation maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1159–1168.

Martinez, J., Black, M.J., Romero, J., 2017. On human motion prediction using recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2891–2900.

Peng, W., Hong, X., Chen, H., Zhao, G., 2020a. Learning graph convolutional network for skeleton-based human action recognition by neural searching. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence.

Peng, W., Shi, J., Xia, Z., Zhao, G., 2020b. Mix dimension in Poincaré geometry for 3D skeleton-based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1432–1440.

Shahroudy, A., Liu, J., Ng, T.T., Wang, G., 2016. NTU RGB+D: a large scale dataset for 3D human activity analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1010–1019.

Shi, L., Zhang, Y., Cheng, J., Lu, H., 2019a. Skeleton-based action recognition with directed graph neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7912–7921.

Shi, L., Zhang, Y., Cheng, J., Lu, H., 2019b. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12026–12035.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A., 2011. Real-time human pose recognition in parts from single depth images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1297–1304.

Si, C., Chen, W., Wang, W., Wang, L., Tan, T., 2019. An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1227–1236.

Si, C., Jing, Y., Wang, W., Wang, L., Tan, T., 2018. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In: Proceedings of the European Conference on Computer Vision, pp. 103–118.

Simonyan, K., Zisserman, A., 2014a. Two-stream convolutional networks for action recognition in videos. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 568–576.

Simonyan, K., Zisserman, A., 2014b. Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556.

Song, S., Lan, C., Xing, J., Zeng, W., Liu, J., 2017. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 4263–4270.

Song, Y.F., Zhang, Z., Shan, C., Wang, L., 2020. Richly activated graph convolutional network for robust skeleton-based action recognition. IEEE Trans. Circuits Syst. Video Technol. https://doi.org/10.1109/TCSVT.2020.3015051.

Yan, S., Xiong, Y., Lin, D., 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, pp. 7444–7452.

Yub Jung, H., Lee, S., Seok Heo, Y., Dong Yun, I., 2015. Random tree walk toward instantaneous 3D human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2467–2474.

Yun, K., Honorio, J., Chattopadhyay, D., Berg, T.L., Samaras, D., 2012. Two-person interaction detection using body-pose features and multiple instance learning. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 28–35.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N., 2017a. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2117–2126.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N., 2019. View adaptive neural networks for high performance skeleton-based human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. 41, 1963–1978.

Zhang, S., Liu, X., Xiao, J., 2017b. On geometric features for skeleton-based action recognition using multilayer LSTM networks. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 148–157.

Zhang, Z., 2012. Microsoft kinect sensor and its effect. IEEE Multimed. 19, 4–10.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016. Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., Xie, X., 2016. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 3697–3703.