View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition from Skeleton Data

Pengfei Zhang [1]*, Cuiling Lan [2]†, Junliang Xing [3], Wenjun Zeng [2], Jianru Xue [1], Nanning Zheng [1]
[1] Xi'an Jiaotong University, Shannxi, China [2] Microsoft Research Asia, Beijing, China
[3] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
zpengfei@stu.xjtu.edu.cn, {culan,wezeng}@microsoft.com, jlxing@nlpr.ia.ac.cn, {jrxue,nnzheng}@mail.xjtu.edu.cn

Abstract

Skeleton-based human action recognition has recently attracted increasing attention due to the popularity of 3D skeleton data. One main challenge lies in the large view variations in captured human actions. We propose a novel view adaptation scheme to automatically regulate observation viewpoints during the occurrence of an action. Rather than re-positioning the skeletons based on a human defined prior criterion, we design a view adaptive recurrent neural network (RNN) with LSTM architecture, which enables the network itself to adapt to the most suitable observation viewpoints from end to end. Extensive experiment analyses show that the proposed view adaptive RNN model strives to (1) transform the skeletons of various views to much more consistent viewpoints and (2) maintain the continuity of the action rather than transforming every frame to the same position with the same body orientation. Our model achieves significant improvement over the state-of-the-art approaches on three benchmark datasets.

1. Introduction

Recognizing human actions has remained one of the most important and challenging problems in computer vision. Demands on human action recognition techniques are growing very fast and have expanded in many domains, such as visual surveillance, human-computer interaction, video indexing/retrieval, video summary, and video understanding [27, 42].

Considering the differences in inputs, human action recognition can be categorized into color video-based and 3D skeleton-based approaches. While color video based human action recognition has been extensively studied over the past few decades, 3D skeleton based human representation for action recognition has recently attracted a lot of research attention because of its high level representation and robustness to variations of viewpoints, appearances, and surrounding distractions [2, 10, 28, 47]. Biological observations from the early seminal work of Johansson suggest that humans can recognize actions from just the motion of a few joints of the human body, even without appearance information [19]. Besides, the prevalence of cost-effective depth cameras such as Microsoft Kinect [48], Intel RealSense [1], dual camera devices, and the advance of a powerful technique of human pose estimation from depth [34] make 3D skeleton data easily obtainable. Like the many previous works listed in the survey paper [10], we focus on skeleton-based action recognition.

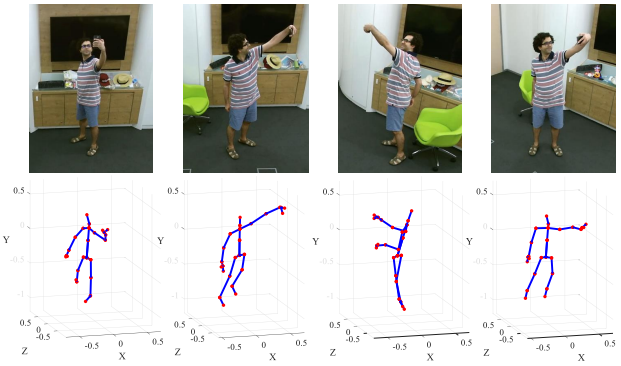


Figure 1: Skeleton representations of the same posture captured from different viewpoints (different camera position, angle, and the subject orientation) are very different.

One of the main challenges in skeleton-based human action recognition is the complex viewpoint variations when capturing human action data. First, in a practical scenario, the capturing viewpoints of the camera differ among different sequences, *e.g.*, the facing angle, position of the camera, resulting in large differences among skeleton representations. Second, the actor could conduct an action towards different orientations. Moreover, he/she may dynamically change his/her orientations as time goes on. As illustrated

*This work was done when P. Zhang was an intern at Microsoft Research Asia.
†Corresponding author.

2380-7504/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCV.2017.233

in Fig. 1, the skeleton representations of the same posture are rather different when captured from different viewpoints. In practice, the variation of the observation viewpoints makes action recognition a very challenging problem [2, 16]. Attempts have been made in previous works to overcome the view variations for robust action recognition [16, 30, 3, 32, 20, 7, 33, 41, 23, 15, 22, 43, 44, 25, 49, 29, 8]. Most of these works, however, are designed for color video-based human recognition. The investigation of view invariance for skeleton-based human recognition, however, still remains under explored.

There are only a few attempts in previous works to consider the effect from view variations. A general treatment employs a pre-processing step to transform the 3D joint coordinates from the camera coordinate system to a person-centric coordinate system by placing the body center at the origin, followed by rotating the skeleton such that the body plane is parallel to the (x, y)-plane, to make the skeleton data invariant to absolute location, and the body orientation [45, 39, 5, 51, 18, 31, 24, 35]. Such a pre-processing gains partial view-invariant. However, it also has many drawbacks. On one hand, it loses partial motion information, *e.g.*, the moving trajectory and speed of the body center, and the changing dynamics of the body orientation. For example, the action of walking becomes walking in the same place and the action of dancing with body rotating becomes dancing with body facing a fixed orientation. On the other hand, the processing (*i.e.*, translation, rotation) is not explicitly designed with the target of optimizing action recognition in mind but is based on human defined criteria, which reduces the space for exploiting optimal viewpoints. How to design a system which provides superior viewpoint for action recognition is still an under-explored problem, and warrants more investigation.

In this work, we address the view variation problem for high performance skeleton-based action recognition. Instead of processing the 3D skeletons based on human defined criteria for solving view variations, we propose a view adaptation scheme which automatically regulates the observation viewpoint at each frame to obtain the skeleton representation under the new view. Note that the regulation of the viewpoint of the camera is equivalent to the transformation of the skeleton to a new coordinate system. To this end, as shown in Fig. 2, we design a view adaptive RNN with LSTM architecture to learn and determine the appropriate viewpoints based on the input skeleton. The skeleton newly represented in the determined observation viewpoint is used for easier action recognition by a main LSTM network. With the objective of maximizing recognition performance, the entire network is end-to-end trained to encourage the view adaptation subnetwork to learn and determine suitable viewpoints.

To summarize, we make the following contributions.

- We propose a self-regulated view adaption scheme which re-positions the observation viewpoints dynamically to facilitate better recognition of the action from skeleton data.
- We integrate the proposed view adaption scheme into an end-to-end LSTM network which automatically determines the "best" observation viewpoints during recognition.
- We have made many observations and analyses of the results from the view adaptation model. We find that the proposed model automatically regulates the skeletons to more consistent observation viewpoints while maintaining the continuity of an action.

Based on the above contributions, we present an end-to-end, high performance action recognition system. Extensive experiment analyses and evaluations demonstrate its strong ability to overcome the view variation problem, and its state-of-the-art performance on three benchmark datasets.

2. Related Work

2.1. View Invariant Action Recognition

Human actions may be observed from arbitrary camera viewpoints in realistic scenes. This factor is a barrier for the development of efficient action recognition techniques. Researchers have paid much attention to this issue and designed view-invariant approaches for action recognition from color videos [16, 30, 3, 32, 20, 7, 33, 41, 23, 15, 22, 43, 44, 25, 49, 29, 8]. One category of approaches requires multiple view videos for training [15, 8, 41, 44, 25]. For example, the 3D histogram of Oriented Gradients based Bag of Words model [41] is learned from all viewpoints of data to provide robustness to view changes. Another category of approaches designs view-invariant feature representations [20, 30, 3] like self-similarity descriptors [20] or descriptions based on trajectory curvature [30, 3]. There is also a category of approaches that employ knowledge transfer-based models [7, 23, 22, 49, 50, 29]. They find a view independent latent space in which features from different views are directly comparable. Considering the different domains of the color videos and skeleton sequences, the approaches designed for color videos cannot be directly extended to skeleton-based action recognition.

As a comparison, the study of viewpoint influences on skeleton-based action recognition is under-explored. The commonly used strategies are monotonous where a pre-processing of skeleton is performed [45, 39, 5, 51, 18, 31, 24, 35]. Unfortunately, they result in the loss of partial relative motion information. Sequence-based pre-processing, which performs the same transformation on all frames with the parameters determined from the first frame so that the motion is invariant to the initial body position and initial orientation, can preserve motion information. However, since

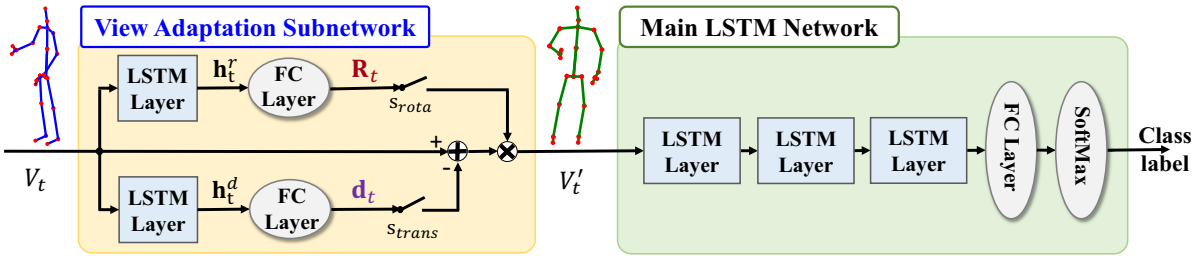


Figure 2: Architecture of our end-to-end view adaptive RNN, which consists of a View Adaptation Subnetwork, and a Main LSTM Network. The View Adaptation Subnetwork determines the suitable observation viewpoint at each time slot. With the skeleton representations under the new observation viewpoints, the main LSTM network determines the action class.

the human body is not rigid, the definition of the body plane by the joints of "hip", "shoulder", "neck" is not always suitable for the purpose of orientation alignment [40]. After the alignment of such a defined body plane, a person who is bending over will have his/her legs obliquely upward. Wang et al. [40] use only the up-right pose frames in a sequence to determine the body plane by averaging the rotation transformation. However, a sequence may not contain an up-right pose.

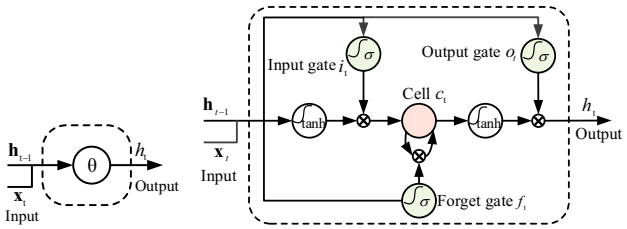
In contrast to the above works, we leverage a content-dependent view adaptation model to automatically learn and determine the suitable viewpoints for each frame.

2.2. RNN for Skeleton-based Action Recognition

Earlier works used hand-crafted features for action recognition from the skeleton [10, 45]. Many recent works leverage the Recurrent Neuron Networks to recognize human actions from raw skeleton input, with feature learning and temporal dynamic modeling achieved by the neuron networks. Du et al. [5] proposes an end-to-end hierarchical RNN for action recognition which takes each body part as input to each RNN subnetwork and fuses the output of subnetworks hierarchically. Zhu et al. [51] propose the automatic exploration of the co-occurrence of discriminative skeleton joints in an LSTM network using group sparse regularization. In the part aware LSTM model [31], the memory unit of the LSTM model is separated to part-based sub-cells to push the network towards learning long-term context representations for each individual part. To learn both the spatial and temporal relationships among joints, the spatial-temporal LSTM network extends the deep LSTM architecture to two concurrent domains, *i.e.*, the temporal domain and the spatial domain [24]. To further exploit joint discriminations, the spatial-temporal attention model [35] further introduces the attention mechanism into the network to enable it to selectively focus on discriminative joints of the skeleton within one frame, and pay different levels of attention to the outputs from multiple frames.

Most of the above works take the center and orientation aligned skeletons as input to the RNNs, by using the human defined alignment criteria. In contrast, our model automatically determines the observation viewpoints and thus the skeleton representations for efficient action recognition.

3. RNN and LSTM Overview



(a) (b)

Figure 3: Structures of the neurons. (a) RNN; (b) LSTM.

To make the paper self-contained, in this section we briefly review the Recurrent Neural Network (RNN), and the RNN with Long Short-Term Memory (LSTM) [12], based on which our framework is built.

RNN is a powerful model for sequential data modeling and feature extraction, which allows the previous information to persist [9, 26]. Fig. 3 (a) shows an RNN neuron, where the output response $\mathbf{h}_t$ at time step t is determined by the input $\mathbf{x}_t$ and the hidden outputs from RNN themselves at the last time step $\mathbf{h}_{t-1}$. However, such a standard RNN faces the vanishing gradient effect in practice [12, 11, 9], which is not very capable of handling long-term dependencies. The advanced RNN architecture of LSTM [12] mitigates this problem. Fig. 3 (b) shows an LSTM neuron. The key to LSTM is the cell state $\mathbf{c_t}$, which is kind of like a conveyor belt [26]. The removal of the previous information or addition of the current information to the cell state are regulated with linear interactions by the forget gate $\mathbf{f_t}$ and the input gate $\mathbf{i_t}$.

4. View Adaptation Model using LSTM

We propose an end-to-end LSTM network with a view adaptation module for skeleton-based human action recog-

nition. Fig. 2 shows the overall architecture of the proposed network, which consists of a View Adaptation Subnetwork and a Main LSTM Network. In the following subsections, we first formulate the problem of observation viewpoint regulation. Then we describe our proposed view adaptation network in detail, which is capable of adaptively determining the most suitable observation viewpoints frame by frame.

4.1. Problem Formulation

The raw 3D skeletons are recorded corresponding to the camera coordinate system (global coordinate system), with the origin located at the position of the camera sensor. To be insensitive to the initial position of an action and to facilitate our study, for each sequence, we translate the global coordinate system to the body center of the first frame as our new global coordinate system $\mathcal{O}$. Note that the input skeleton V_t to our system as in Fig. 2 is the skeleton representation under this global coordinate system.

One can choose to observe an action from suitable views. Thanks to the availability of the 3D skeletons captured from a fixed view, it is possible to set up a movable virtual camera and observe the action from new observation viewpoints as illustrated in Fig. 4. With the skeleton at frame t re-observed from the movable virtual camera viewpoint (observation viewpoint), the skeleton can be transformed to a representation under the movable virtual camera coordinate system, which is also referred to as the observation coordinate system $\mathcal{O}'_t$.

Given a skeleton sequence $\mathcal{S}$ with T frames, under the global coordinate system $\mathcal{O}$, the j^{th} skeleton joint on the t^{th} frame is denoted as $\mathbf{v}_{t,j} = [x_{t,j}, y_{t,j}, z_{t,j}]^{\mathrm{T}}$, where $t \in (1, \cdots, T)$, $j \in (1, \cdots, J)$, J denotes the total number of skeleton joints in a frame. We denote the set of joints in the t^{th} frame as $V_t = \{\mathbf{v}_{t,1}, \cdots, \mathbf{v}_{t,J}\}$.

For the t^{th} frame, assume the movable virtual camera is placed at a suitable viewpoint, with the corresponding observation coordinate system obtained from a translation by $\mathbf{d}_t \in \mathbb{R}^3$, and a rotation of α_t, β_t, γ_t radians anticlockwise around the X-axis, Y-axis, and Z-axis, respectively, of the global coordinate system. Therefore, the representation of the j^{th} skeleton joint $\mathbf{v}'_{t,j} = [x'_{t,j}, y'_{t,j}, z'_{t,j}]^{\mathrm{T}}$ of the t^{th} frame under this observation coordinate system $\mathcal{O}'_t$ is

$$\mathbf{v}'_{t,j} = [x'_{t,j}, y'_{t,j}, z'_{t,j}]^{\mathrm{T}} = \mathbf{R}_t \times (\mathbf{v}_{t,j} - \mathbf{d}_t). \quad (1)$$

$\mathbf{R}_t$ can be represented as

$$\mathbf{R}_t = \mathbf{R}^x_{t,\alpha} \times \mathbf{R}^y_{t,\beta} \times \mathbf{R}^z_{t,\gamma}, \quad (2)$$

where $\mathbf{R}^y_{t,\gamma}$ denotes the coordinate transform for rotating the original coordinate system around the Y-axis by β_t radians anticlockwise, which is defined as

$$\mathbf{R}^y_{t,\beta} = \begin{bmatrix} \cos(\beta_t) & \sin(\beta_t) & 0 \\ -\sin(\beta_t) & \cos(\beta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Similarly, $\mathbf{R}^x_{t,\alpha}$ and $\mathbf{R}^z_{t,\gamma}$ denote the coordinate transforms for rotating the original coordinate system around the X-axis by α_t radians, and around the Z-axis by γ_t radians anticlockwise, respectively.

Note that all the skeleton joints in the t^{th} frame share the same transform parameters, *i.e.*, $\alpha_t, \beta_t, \gamma_t, \mathbf{d}_t$, considering that the changing of viewpoints is a rigid motion. Given these transform parameters, the skeleton representation $V'_t = \{\mathbf{v}'_{t,1}, \cdots, \mathbf{v}'_{t,J}\}$ under the new observation coordinate can be obtained from (1). Besides, the viewpoints can vary for different frames. The key problem becomes how to determine the viewpoints of the movable virtual camera.

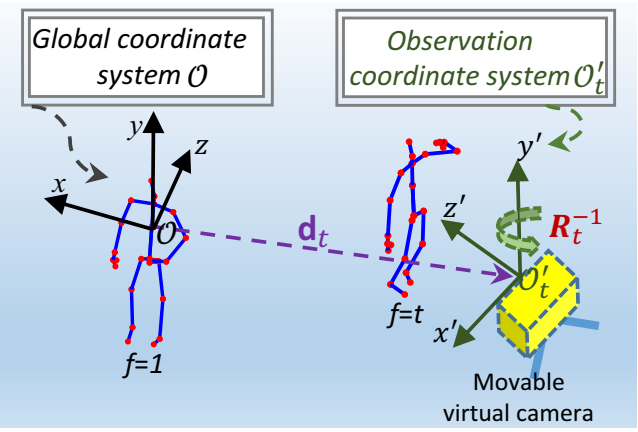


Figure 4: Illustration of the regulation of the observation viewpoint (movable virtual camera). A skeleton sequence is a record of the skeletons from the first frame $f = 1$ to the last frame $f = T$ under the global coordinate system $\mathcal{O}$. The action can be re-observed by a movable virtual camera under the observation coordinate systems. For the t^{th} frame, the observation coordinate system is at a new position $\mathbf{d}_t$ with a rotation of α_t, β_t, γ_t radians anticlockwise around the X-axis, Y-axis, and Z-axis, respectively, corresponding to the global coordinate system. The skeleton can then be represented under this observation coordinate system $\mathcal{O}'_t$.

4.2. View Adaptive Recurrent Neural Network

We use a View Adaptation Subnetwork to automatically determine the observation viewpoints, *i.e.*, $\alpha, \beta, \gamma, \mathbf{d}_t$ (as discussed in section 4.1), and use a Main LSTM Network to learn the temporal dynamics and perform the feature abstractions from the view-regulated skeleton data for the action recognition, from end to end, as shown in Fig. 2.

View Adaptation Subnetwork. A regulation of observation viewpoint corresponds to the re-positioning of the movable virtual camera, which can be described by the translation and rotation of this virtual camera (observation coordination system). At a time slot corresponding to the t^{th} frame, with the skeleton V_t as input, two branches of LSTM subnetworks are utilized to learn the rotation parameters $\alpha_t, \beta_t, \gamma_t$ to obtain the rotation matrix $\mathbf{R}_t$, and the translation vector $\mathbf{d}_t$, corresponding to the global coordinate system.

The branch of rotation subnetwork for learning rotation parameters consists of an LSTM layer, and a full connection (FC) layer. The rotation parameters are obtained as

$$[\alpha_t, \beta_t, \gamma_t]^{\mathrm{T}} = \mathbf{W}_r \mathbf{h}_t^r + \mathbf{b}_r, \tag{4}$$

where $\mathbf{h}_t^r \in \mathbb{R}^{N\times 1}$ is the hidden output vector of the LSTM layer with N denoting the number of LSTM neurons, $\mathbf{W}_r \in \mathbb{R}^{3\times N}$ and $\mathbf{b}_r \in \mathbb{R}^{3\times 1}$ denote the weight matrix and offset vector of the FC layer, respectively. With the rotation parameters, the rotation matrix $\mathbf{R}_t$ is obtained by (2).

The branch of translation subnetwork for learning translation parameters consists of an LSTM layer, and a FC layer. The translation vector $\mathbf{d}_t$ is calculated as

$$\mathbf{d}_t = \mathbf{W}_d \mathbf{h}_t^d + \mathbf{b}_d, \tag{5}$$

where $\mathbf{h}_t^d \in \mathbb{R}^{N\times 1}$ is the hidden output vector of its LSTM layer, $\mathbf{W}_d \in \mathbb{R}^{3\times N}$ and $\mathbf{b}_d \in \mathbb{R}^{3\times 1}$ denotes the weight matrix and offset vector of the FC layer. Under the observation viewpoint of the t^{th} frame, the representation of the skeleton V_t' is then obtained through (1).

Note that to obtain an efficient view adaptation subnetwork, we have experimented with many alternative designations and found the current design very efficient. First, we use separated LSTM layers for the rotation and translation model learning rather than using shared LSTM layers because the rotation and translation are different operations which are difficult to learn from the shared LSTM neurons. Second, we use the same skeleton input for both the rotation branch subnetwork and the translation branch subnetwork rather than taking the output of one branch (*e.g.*, translation / rotation) as the input of another (*e.g.*, rotation / translation). This is because the learning of the model under the consistent global coordinate system is easier.

Main LSTM Network. The LSTM network is capable of modeling long-term temporal dynamics and automatically learning feature representations. Similar to the designs in [51, 35], we build a main LSTM network by stacking three LSTM layers, followed by one FC layer with a SoftMax classifier. The number of neurons of the FC layer is equal to the number of action classes.

End-to-End Training. The entire network is end-to-end trainable. We use cross-entropy loss as the training loss [35]. The gradients of loss flow back not only within each subnetwork, but also from the Main LSTM Network to the View Adaptation Subnetwork. Let us denote the loss back-propagated to the output of the View Adaptation Subnetwork by $\boldsymbol{\epsilon}_{v'_{t,j}}$, where $j \in (1, \cdots, J)$ and J is the number of skeleton joints. Then, the loss back-propagated to the output of the branch for determining the translation vector of $\mathbf{d}_t$ is

$$\boldsymbol{\epsilon}_{\mathbf{d}_t} = \sum_{j=1}^{j=J} \frac{\partial \mathbf{v}'_{t,j}}{\partial \mathbf{d}_t} \odot \boldsymbol{\epsilon}_{v'_{t,j}}, \tag{6}$$

where $\odot$ denotes element-wise product. Similarly, the loss back-propagated to the output of the branch for determining the rotation parameters can be obtained. For example, the loss back-propagated to the output of β_t is

$$\boldsymbol{\epsilon}_{\beta_t} = \sum_{j=1}^{j=J} \frac{\partial \mathbf{v}'_{t,j}}{\partial \mathbf{R}_t} \frac{\partial \mathbf{R}_t}{\partial \beta_t} \odot \boldsymbol{\epsilon}_{v'_{t,j}}. \tag{7}$$

With the end-to-end training feasible, the view adaptation model is guided to select the suitable observation viewpoints for enhancing recognition accuracy.

Our scheme has the following characteristics. Firstly, it automatically chooses the suitable observation viewpoints based on the contents, rather than using human predefined criteria. Secondly, the view adaptation model is optimized for the purpose of high accuracy recognition.

5. Experiment Results

We evaluate the effectiveness of our proposed view adaptation scheme on three benchmark datasets. In-depth analyses are made on the NTU dataset. To better understand the model, visualizations of the skeleton representations under the observation viewpoints are given.

5.1. Datasets and Settings

NTU RGB+D Dataset (NTU) [31]. This Kinect captured dataset is currently the largest dataset with RGB+D videos and skeleton data for human action recognition, with 56880 video samples. It contains 60 different action classes including daily actions, mutual, and health-related actions. Samples are captured from 17 setups of cameras, where in different setups, the height and distances of the cameras to the subjects are different. For each setup, the three cameras were located at the same height but from different horizontal angles: -45^o (camera 2), 0^o (camera 1), $+45^o$ (camera 3). Each subject performed each action twice, once facing towards the left camera and once towards the right camera. Each subject has 25 joints. The standard evaluations include Cross-Subject (CS) evaluation, where the 40 subjects are split into training and testing groups, and Cross-View (CV) evaluation, where the samples of cameras 2 and 3 are used for training while those of camera 1 for testing.

SBU Kinect Interaction Dataset (SBU) [46]. This Kinect captured dataset is an interaction dataset with two subjects, containing 282 sequences of 8 classes with subject independent 5-fold cross validation. Each subject has 15 joints.

SYSU 3D Human-Object Interaction Set (SYSU) [13]. This Kinect captured dataset contains 12 actions performed by 40 subjects. It has 480 sequences. Each subject has 20 joints. We evaluate performance on two standard protocols [13]. For setting-1, half of samples are used for training and the rest for testing for each activity. For setting-2, half of subjects are used for training and the rest for testing. 30-fold cross validation is utilized. Downsampling the sequences in temporal is performed on this dataset in considering that the maximum length of the sequences is high.

Implementation Details. We build our frameworks based on the platform of Keras [4] toolbox with theano [38]. Dropout [36] with a probability of 0.5 is used to alleviate overfitting. Gradient clipping similar to [37] is used by enforcing a hard constraint on the norm of the gradient (to not exceed 1) to avoid the exploding gradient problem. Adam [21] is adapted to train all the networks, and the initial learning rate is set as 0.005.

In our network design, we use 100 LSTM neurons in each LSTM layer for the NTU and the SYSU datasets. To avoid overfitting, we use 50 LSTM neurons in each LSTM layer for the SBU dataset, which has much smaller numbers of training samples than that of the NTU and the SYSU datasets. We set the batch sizes for the NTU, SYSU, and SBU dataset to 256, 64, and 8, respectively. For the View Adaptation Subnetwork, we initialize the full connection layer parameters to zeros for efficient training.

5.2. Comparisons to Other State-of-the-Art

We show the performance comparisons of our proposed view adaptation scheme (*VA-LSTM*) with other state-of-the-art approaches in Table 1, Table 2, and Table 3 for the NTU, SBU and SYSU datasets, respectively. We can see that our scheme significantly outperforms the state-of-the-art approaches by about 6%, 4%, 1% in accuracy for the NTU, SBU, SYSU dataset respectively.

5.3. Efficiency of the View Adaptation Model

To validate the effectiveness of the proposed view adaptation model, we make two sets of comparisons as summarized in Table 4. One set of comparisons evaluates the efficiency among the different pre-processing based methods and our proposed scheme. Another set of results evaluates the efficiency of the view adaptation models.

VA-LSTM is our proposed final view adaptation scheme which automatically regulates the observation viewpoints in the network. This is the scheme where both the translation and rotation branches are connected, *i.e.*, the switch s_{rota} and s_{trans} are on as in Fig. 2. *VA-trans-LSTM* is our scheme which only allows the translation of the viewpoint, *i.e.*, the switch s_{rota} is off while s_{trans} is on. In comparison, *S-trans+LSTM* is our baseline scheme without enabling the view adaptation model, *i.e.*, the switch s_{rota} and s_{trans} are both off, where $V'_t = V_t$. Note that the input V_t is the same as that of our view adaptation schemes, where the global coordinate system is moved to the body center of the first frame for the entire sequence to be insensitive to the initial position (see section 4.1). We refer to this pre-processing as sequence level translation, *i.e.*, *S-trans*. *VA-rota-LSTM* is our scheme which only allows the rotation of the viewpoints, *i.e.*, the switch s_{rota} is on while s_{trans} is off.

Table 1: Comparisons on the NTU dataset with Cross-Subject and Cross-View settings in accuracy (%).

Methods	CS	CV
Skeleton Quads [6]	38.6	41.4
Lie Group [39]	50.1	52.8
Dynamic Skeletons [13]	60.2	65.2
HBRNN-L [5]	59.1	64.0
Part-aware LSTM [31]	62.9	70.3
ST-LSTM (Tree Traversal) + Trust Gate [24]	69.2	77.7
STA-LSTM [35]	73.4	81.2
VA-LSTM	**79.4**	**87.6**

Table 2: Comparisons on the SBU dataset in accuracy (%).

Methods	Acc. (%)
Raw skeleton [46]	49.7
Joint feature [46]	80.3
Raw skeleton [17]	79.4
Joint feature [17]	86.9
HBRNN-L [5]	80.4
Co-occurrence RNN [51]	90.4
STA-LSTM [35]	91.5
ST-LSTM (Tree Traversal) + Trust Gate [24]	93.3
VA-LSTM	**97.2**

Table 3: Comparisons on the SYSU dataset in accuracy (%).

Methods	setting-1	setting-2
LAFF [14]	–	54.2
Dynamic Skeletons [13]	75.5	76.9
VA-LSTM	**76.9**	**77.5**

From Table 4, we observe that the proposed final view adaptation scheme outperforms the baseline scheme *S-trans+LSTM* by 3.4% and 5.3% in accuracy for CS and CV settings, respectively, thanks to the introduction of the proposed view adaptation module.

One may wonder how the performance is when using the pre-processed skeletons, basing on the widely used human defined processing criteria, before inputing to the Main LSTM Network. Such pre-processings can be considered as the human defined rules for determining the viewpoints. We name the pre-processing based schemes in the manner of *C+LSTM*, where *C* indicates the pre-processing strategy, *e.g.*, *F-trans+LSTM*. The 3^{rd} to 7^{th} rows show the perfor-

Table 4: Comparisons of pre-processing methods and our view adaptation model on the NTU dataset in accuracy (%).

	Methods	CS	CV
wo/ pre-proc.	Raw + LSTM	66.3	73.4
Pre-proc.	S-trans + LSTM	76.0	82.3
	F-trans + LSTM	75.1	80.5
	S-trans&S-rota + LSTM	76.4	85.4
	S-trans&F-rota + LSTM	75.0	85.1
	F-trans&F-rota + LSTM	74.1	83.9
View adap.	VA-trans-LSTM	77.7	84.9
	VA-rota-LSTM	79.4	87.1
	VA-LSTM	**79.4**	**87.6**

mance of schemes using different pre-processing strategies. *F-trans* means performing frame level translation to have the body center at the coordinate system origin for each frame. *S-rota* means the sequence level rotation with the rotation parameters calculated from the first frame, which is to fix the X-axis to be parallel to the vector from "left shoulder" to "right shoulder", Y-axis to be parallel to the vector from "spline base" to "spine", and Z-axis as the new $X \times Y$. Similarly, *F-rota* means the frame level rotation. *F-trans&F-rota* means both *F-trans* and *F-rota* are performed, which is similar to the pre-processing in [31, 24, 35]. The scheme *Raw+LSTM* in the 2^{nd} row denotes a scheme which uses the original skeleton without any pre-processing as the input to the Main LSTM Network. Note that for 3D skeletons, the distance of a subject to the camera does not influence the scale of the skeletons. Therefore, the scaling operation is not considered in our framework.

From the comparisons in Table 4, we have the following observations and conclusions. (1) Our final scheme significantly outperforms the commonly used pre-processing strategies. In comparison with *F-trans&F-rota+LSTM* [31, 24, 35], our scheme achieves improvement by 5.3% and 3.7% in accuracy for CS and CV settings, respectively. In comparison with *S-trans&S-rota+LSTM*, our scheme achieves improvement by 3.0% and 2.2% in accuracy. (2) When only the rotation (or the translation) is allowed for adjusting the viewpoints, our scheme still consistently outperforms the schemes with human defined rotation (or translation) pre-processing. (3) Frame level pre-processing is inferior to the sequence level pre-processing, because the former loses more information, *e.g.*, the motion across frames. (4) Being insensitive to the initial position of an action, *S-trans+LSTM* significantly outperforms the scheme with raw skeletons as input *Raw+LSTM*.

5.4. Visualization of the Learned Views

At each frame, the view adaptation subnetwork determines the observation viewpoint (by re-localizing the virtual movable camera) and then transforms the input skeleton V_t to the representation V_t' in the new viewpoint for optimizing recognition performance. We visualize the representations V_t and V_t' for better understanding of our model.

Fig. 1 shows the skeletons from different sequences captured from different viewpoints of the same posture. Interestingly, the transformed skeletons (green) of various viewpoints have much more consistent viewpoints, i.e., frontal viewpoint here. Another example is shown in Fig. 6 with the skeleton frames of the same action performed by differ-

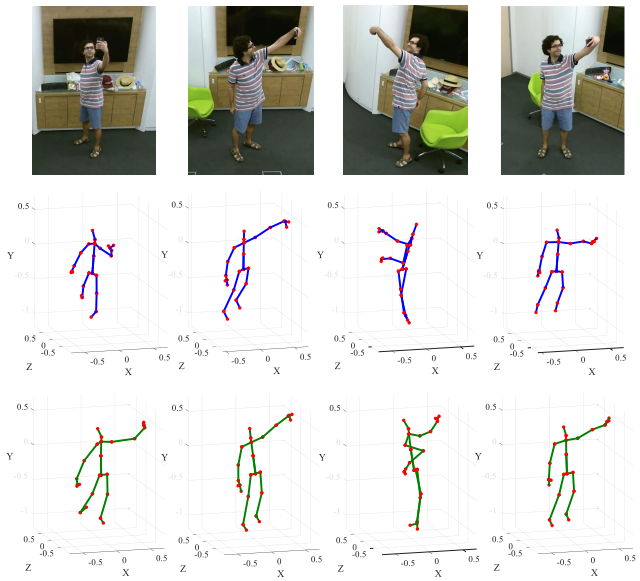


Figure 5: Frames of the same posture captured from different viewpoints for the same subject. 2^{nd} row: original skeletons. 3^{rd} row: skeleton representations from the observation viewpoints of our model. Note the third skeleton is very noisy due to occlusion during Kinect shooting.

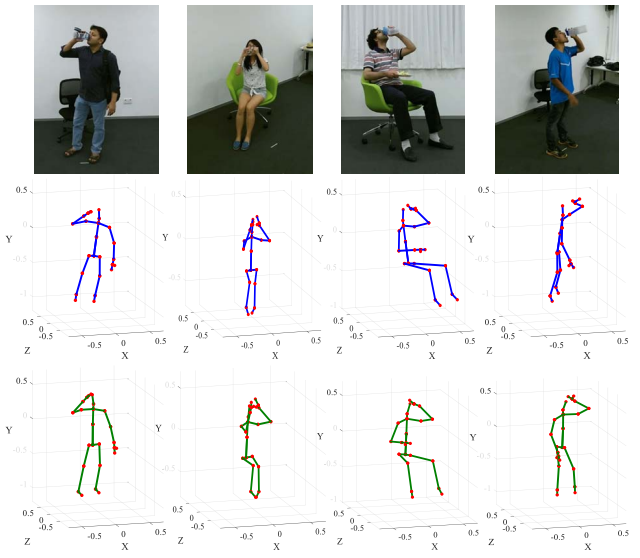
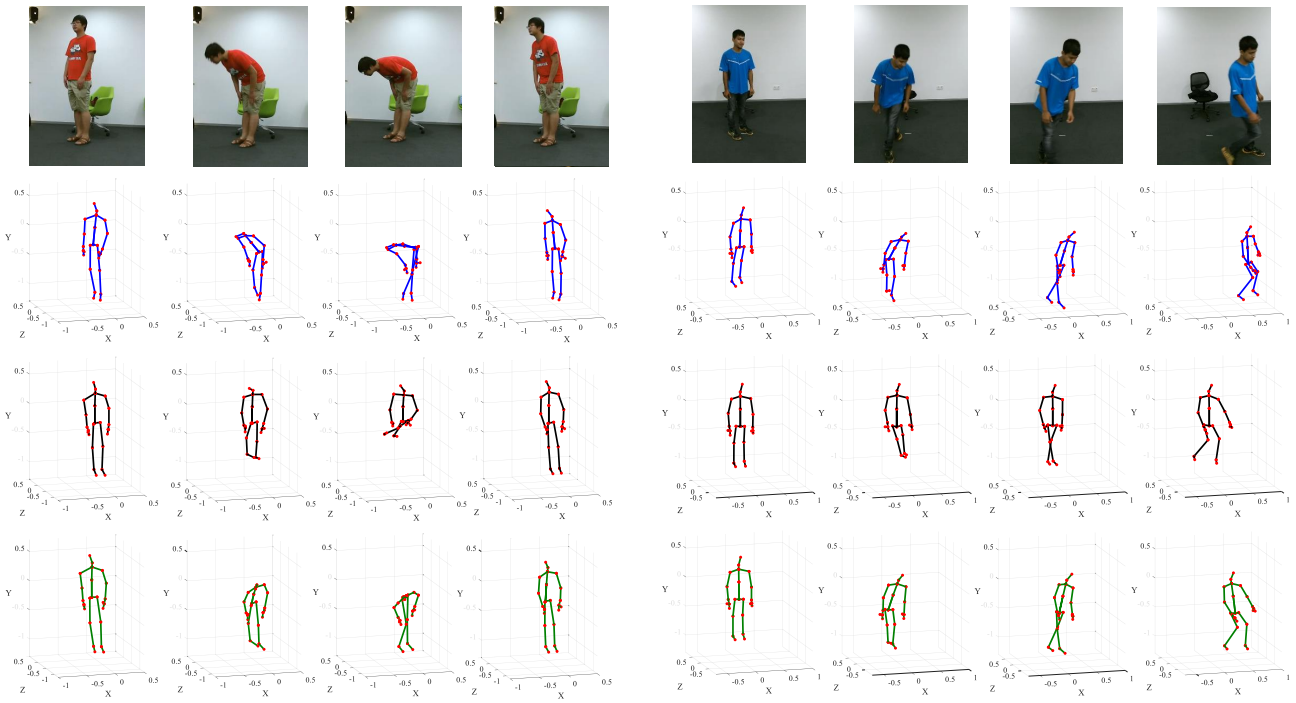


Figure 6: Frames of the same action "drinking" captured from different viewpoints for different subjects. 2^{nd} row: original skeletons. 3^{rd} row: skeleton representations from the observation viewpoints of our model.



(a) (b)

Figure 7: Frames from sequences of actions: (a) "bow"; (b) "staggering". 2^{nd} row: original skeleton. 3^{rd} row: skeleton after the pre-processing with *F-trans*&*F-rota*. 4^{th} row: skeleton representation from the observation viewpoints of our model.

ent subjects. We can see that they are transformed to similar viewpoints. A similar phenomenon is observed in different actions and sequences.

To visualize the skeleton representations in the sequence along time, we show some frames of an action under the original and new observation viewpoints in Fig. 7. We can see that after our view adaptation model is applied, the subjects even for different actions are oriented toward a more consistent view. Different from frame level pre-processing (as in the 3^{rd} row), the transformed skeletons among frames are continuous and looks much natural. In Fig. 7 (a) of action "bow", the orientation of the body after the processing of our model is parallel to X-axis while the legs after frame level pre-processing becomes obliquely upward. In Fig. 7 (b) of action "staggering", the position changes of the subject after the processing of our model remain while such motion is lost for the pre-processing results.

From observations, we find that the learned view adaptation model tends to (1) regulate the observation viewpoints to present the subjects as if observed in a consistent viewpoint cross sequences and actions; (2) maintain the continuity of an action without losing much of the relative motions.

Optimized with the target of maximizing the recognition performance, the proposed view adaptation model is much effective in choosing the suitable viewpoints. The consistency of viewpoints for various actions/subjects overcomes the challenge caused by the diversity of viewpoints in video capturing, enabling the network to focus on the learning of action-specific features. Besides, unlike some pre-processing strategy, the valuable motion information is preserved.

6. Conclusion

We present an end-to-end view adaptation model for human action recognition from skeleton data. Instead of following the human predefined criterion to re-position the skeletons for action recognition, our network is capable of regulating the observation viewpoints to the suitable ones by itself, with the optimization target of maximizing recognition performance. It overcomes the limitations of the human defined pre-processing approaches by exploiting the optimal viewpoints through the content dependent recurrent neuron network model. Experiment results demonstrate that the proposed model can significantly improve the recognition performance on three benchmark datasets and achieve state-of-the-art results.

Acknowledgements

Junliang Xing is partly supported by the Natural Science Foundation of China (Grant No. 61672519), Jianru Xue is partly supported by National Key Research and Development Plan 2016YFB1001004.

References

[1] Intel RealSense. https://software.intel.com/en-us/realsense.

[2] J. K. Aggarwal and L. Xia. Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, 48:70–80, 2014.

[3] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. View-invariant motion trajectory-based activity classification and recognition. *Multimedia Systems*, 12(1):45–54, 2006.

[4] F. Chollet. Keras. https://github.com/fchollet/keras, 2015.

[5] 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.

[6] G. Evangelidis, G. Singh, and R. Horaud. Skeletal quads: Human action recognition using joint quadruples. In *IEEE International Conference on Pattern Recognition*, pages 4513–4518, 2014.

[7] A. Farhadi and M. K. Tabrizi. Learning to recognize activities from the wrong view point. In *European Conference on Computer Vision*, pages 154–166, 2008.

[8] J.-g. Feng and X. Jun. View-invariant human action recognition via robust locally adaptive multi-view learning. *Frontiers of Information Technology & Electronic Engineering*, 16(11):917–920, 2015.

[9] A. Graves. Supervised sequence labelling with recurrent neural networks. *Volume 385 of Studies in Computational Intelligence*, 2012.

[10] F. Han, B. Reily, W. Hoff, and H. Zhang. Space-time representation of people based on 3D skeletal data: A review. *arXiv preprint arXiv:1601.01006*, 2016.

[11] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A field guide to dynamical recurrent neural networks, 2001.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for RGB-D activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5344–5352, 2015.

[14] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, and J. Lai. Real-time RGB-D activity prediction by soft regression. In *European Conference on Computer Vision*, pages 280–296, 2016.

[15] A. Iosifidis, A. Tefas, and I. Pitas. View-invariant action recognition based on artificial neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 23(3):412–424, 2012.

[16] X. Ji and H. Liu. Advances in view-invariant human motion analysis: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):13–24, 2010.

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

[18] M. Jiang, J. Kong, G. Bebis, and H. Huo. Informative joints based human action recognition using skeleton contexts. *Signal Processing: Image Communication*, 33:29–40, 2015.

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

[20] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *European Conference on Computer Vision*, pages 293–306, 2008.

[21] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] R. Li and T. Zickler. Discriminative virtual views for cross-view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2855–2862, 2012.

[23] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *IEEE International Conference on Computer Vision*, pages 3209–3216, 2011.

[24] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal LSTM with trust gates for 3D human action recognition. *arXiv preprint arXiv:1607.07043*, 2016.

[25] B. Mahasseni and S. Todorovic. Latent multitask learning for view-invariant action recognition. In *IEEE International Conference on Computer Vision*, pages 3128–3135, 2013.

[26] C. Olah. LSTM. http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

[27] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[28] L. L. Presti and M. La Cascia. 3d skeleton-based human action classification: a survey. *Pattern Recognition*, 53:130–147, 2016.

[29] H. Rahmani and A. Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2458–2466, 2015.

[30] C. Rao and M. Shah. View-invariance in action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 316–322, 2001.

[31] 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*, pages 1010–1019, 2016.

[32] Y. Shen and H. Foroosh. View-invariant action recognition using fundamental ratios. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2008.

[33] Y. Shen and H. Foroosh. View-invariant action recognition from point triplets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1898–1905, 2009.

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

[35] 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*, pages 4263–4270, 2017.

[36] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[37] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[38] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

[39] 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.

[40] J. Wang, Z. Liu, and Y. Wu. Learning actionlet ensemble for 3d human action recognition. In *Human Action Recognition with Depth Cameras*, pages 11–40. 2014.

[41] D. Weinland, M. Özuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. In *European Conference on Computer Vision*, pages 635–648, 2010.

[42] D. Weinland, R. Ronfard, and E. Boyerc. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[43] X. Wu and Y. Jia. View-invariant action recognition using latent kernelized structural svm. In *European Conference on Computer Vision*, pages 411–424, 2012.

[44] X. Wu, H. Wang, C. Liu, and Y. Jia. Cross-view action recognition over heterogeneous feature spaces. In *IEEE International Conference on Computer Vision*, pages 609–616, 2013.

[45] 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 Workshops*, pages 20–27, 2012.

[46] 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 Workshops*, pages 28–35, 2012.

[47] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang. Rgb-d-based action recognition datasets: A survey. *Pattern Recognition*, 60:86–105, 2016.

[48] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, 2012.

[49] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi. Cross-view action recognition via a continuous virtual path. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2690–2697, 2013.

[50] J. Zheng and Z. Jiang. Learning view-invariant sparse representations for cross-view action recognition. In *IEEE International Conference on Computer Vision*, pages 3176–3183, 2013.

[51] 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*, pages 3697–3703, 2016.