3D Action Recognition from Novel Viewpoints

Hossein Rahmani, and Ajmal Mian
School of Computer Science and Software Engineering, The University of Western Australia
hossein@csse.uwa.edu.au, ajmal.mian@uwa.edu.au

Abstract

We propose a human pose representation model that transfers human poses acquired from different unknown views to a view-invariant high-level space. The model is a deep convolutional neural network and requires a large corpus of multiview training data which is very expensive to acquire. Therefore, we propose a method to generate this data by fitting synthetic 3D human models to real motion capture data and rendering the human poses from numerous viewpoints. While learning the CNN model, we do not use action labels but only the pose labels after clustering all training poses into k clusters. The proposed model is able to generalize to real depth images of unseen poses without the need for re-training or fine-tuning. Real depth videos are passed through the model frame-wise to extract view-invariant features. For spatio-temporal representation, we propose group sparse Fourier Temporal Pyramid which robustly encodes the action specific most discriminative output features of the proposed human pose model. Experiments on two multiview and three single-view benchmark datasets show that the proposed method dramatically outperforms existing state-of-the-art in action recognition.

1. Introduction

Video based human action recognition is challenging because significant intra-action variations exist due to changes in viewpoint, illumination, visual appearance (such as color and texture of clothing), scale (due to different human body sizes or distances from the camera), background and speed of performing an action. Some challenges have been simplified by the use of real-time depth cameras (*e.g.* Kinect) that capture the texture and illumination invariant human body shape and simplify human segmentation. However, variations due to viewpoint remains a major challenge and is explicitly addressed in this paper.

Many methods [23, 31, 34, 36–38, 44, 47, 48, 55–57, 63, 67, 68, 74] have been proposed which achieve impressive action recognition results when videos are acquired from a common viewpoint. However, their performance degrades sharply under viewpoint changes [38, 39, 60]. This is because the same human pose appears quite different when observed from different viewpoints. To cope with this problem, view-invariant approaches [17, 39–42, 54, 59, 60] have been recently proposed for action recognition in videos acquired from novel views. Most of these methods operate on RGB videos [17, 20, 41, 42, 60] or skeleton data [54, 59]. Joints extraction methods are inaccurate and sometimes fail when subject is not in the upright or frontal view position [39, 64]. Moreover, view-invariant information can be more reliably extracted from depth videos [39]. For instance, [39, 40] have achieved higher accuracy by extracting view-invariant local spatio-temporal features from depth videos. However, their performance is limited by the discriminative power of the local features [60].

To overcome these drawbacks, we propose a depth video based cross-view action recognition method that consists of two main steps: (1) learning a general view-invariant human pose model from synthetic depth images, and (2) modeling the temporal action variations. The former is a deep CNN which represents different human body shapes and poses observed from numerous viewpoints in a view-invariant high-level space. However, learning such a model requires a large corpus of training data containing a large number of human body poses observed from many viewpoints. Such data is not publicly available and is very expensive to acquire and label. Our solution is to generate the training data synthetically but in the most realistically possible way. To achieve this, we fit realistic synthetic 3D human models to real mocap data [2] and then render each pose from a large number of viewpoints as shown in Fig. 1.

We learn a *single* model for all poses and views without using action labels and show that our model generalizes to real depth images of unseen human poses acquired from novel views without re-training or fine-tuning. Our learned model operates on a frame by frame basis transferring human pose in each frame to a high-level view-invariant representation. Our motivation for using a frame based CNN model comes from the findings [21] that a single frame model performs equally well as the multiframe CNN model. Since actions are performed over a period of time, modeling the temporal structure of videos is performed in the next stage. Many methods [15, 41, 51, 60] model the temporal

variations of videos using optical flow. However, optical flow is not reliable in the presence of noise and lack of texture [34] which is especially the case for depth videos [39]. Moreover, in spatio-temporal matching, temporal misalignments can also become a source of errors. We propose a representation which is robust to depth noise and temporal misalignments. Our representation is a group sparse Fourier Temporal Pyramid that extracts features from the view-invariant high-level representation layer of the proposed CNN model. We capitalize on the fact that the output of different neurons in the CNN representation layer contributes differently to each human pose and hence each action. Thus, we learn action specific sparse neurons-sets for accurate classification. New action classes can be efficiently added to our framework as it requires retraining the action classifier only while using the same learned CNN model.

Experiments on two benchmark multiview human action datasets *i.e*. Northwestern-UCLA Multiview Action3D [60] and UWA3D Multiview Activity II [40], and comparison with state-of-the-art show that our method achieves 12% and 13% higher accuracies respectively than the nearest competitor (Section 6). To show that our method performs equally good in the single/known view case, we provide comparative results on three single-view benchmark human action datasets, MSR Gesture3D [57], MSR Action Pairs3D [34] and MSR Daily Activity3D [58] in the supplementary material.

2. Related Work

Action recognition methods can be divided into three categories based on the type of video data *i.e*. RGB, skeleton or depth. This section discusses related work in each category as well as deep learning based methods.

RGB Videos: Some methods use view-invariant spatio-temporal features [5,35,43,62] and others infer the 3D scene structure through geometric transformations to achieve view invariance in RGB videos [13, 52, 69]. Recently, knowledge transfer based methods [11, 12, 17, 27–29, 41, 60, 61, 66, 72, 73] have become popular that find a set of transformations in feature space such that the features extracted from different views are comparable. For example, Wang *et al*. [60] proposed a cross-view video action representation by discovering the compositional structure in spatio-temporal patterns and geometrical relations among different views. They trained a spatio-temporal AND-OR graph structure by learning a separate linear transformation for each body part between different views. Thus, for action recognition from a novel view, all learned transformations are used for exhaustive matching and the results are combined with an AND-OR Graph.

Skeleton Videos: Skeleton-based methods [10, 45, 54, 59, 65] generally use the human joint positions, extracted by the OpenNI tracking framework [50], as interest points. For example, Wang *et al*. [59] proposed the histogram of occupancy pattern of a fixed region around each joint in each frame. They also proposed a data mining technique to discover the most discriminative joints for each action class. Vemulapalli *et al*. [54] proposed a body part-based skeleton representation to model their relative geometry and modeled human actions as curves in the Lie group. For robustness to viewpoint variations, they rotate the skeletons such that the ground plane projection of the vector from left hip to right hip is parallel to the global x-axis. It is important to note that the human joints extraction methods (such as [50]) are not accurate and sometimes fail when the human is not in the upright or frontal view position [39, 40, 64].

Depth Videos: Action recognition from depth videos has recently become more popular due to the availability of real-time cost-effective sensors. For instance, Oreifej and Liu [34] proposed a histogram of oriented 4D normals (HON4D) for action recognition. Yang and Tian [67] extended HON4D by concatenating the 4D normals in the local neighbourhood of each pixel as its descriptor. However, these descriptors must be extracted from interest points, *e.g*. joint positions, when the subjects significantly change their locations. To overcome this problem, Xia and Aggarwal [64] proposed a method to filter the depth sensor noise and extract more reliable spatio-temporal interest points. However, their approach is sensitive to the speed of performing actions [39]. Although, these methods achieve impressive accuracies for action recognition from a fixed view (mostly frontal), their performance drops sharply when recognition is performed on videos acquired from novel views [39]. More recently, Rahmani *et al*. [39, 40] proposed Histogram of Oriented Principal Components (HOPC) to first detect and then describe spatio-temporal interest points which are repeatable and robust to viewpoint variations. This method directly processes the 3D pointclouds and calculates the HOPC descriptor at every point.

Deep Learning Methods: Due to the impressive results of deep learning on image classification [25] and object detection [14], several attempts have been recently made to train deep networks for action recognition [9, 15, 19, 21, 41, 46, 51]. Ji *et al*. [19] proposed a deep 3D convolutional neural network (CNN) where convolutions are performed in 3D feature maps from spatial and temporal dimensions. However, Karapathy *et al*. [21] show that the single-frame model performs equally well as the multi-frames model. Simonyan and Zisserman [51] trained two CNNs, one for RGB images and one for optical flow, to learn spatio-temporal features. Gkioxari and Malik [15] extended this approach for action localization. Donahue *et al*. [9] proposed an end-to-end trainable recurrent convolutional network which processes video frames with a CNN, whose outputs are passed through a recurrent neural network. These methods are not designed for cross-view action

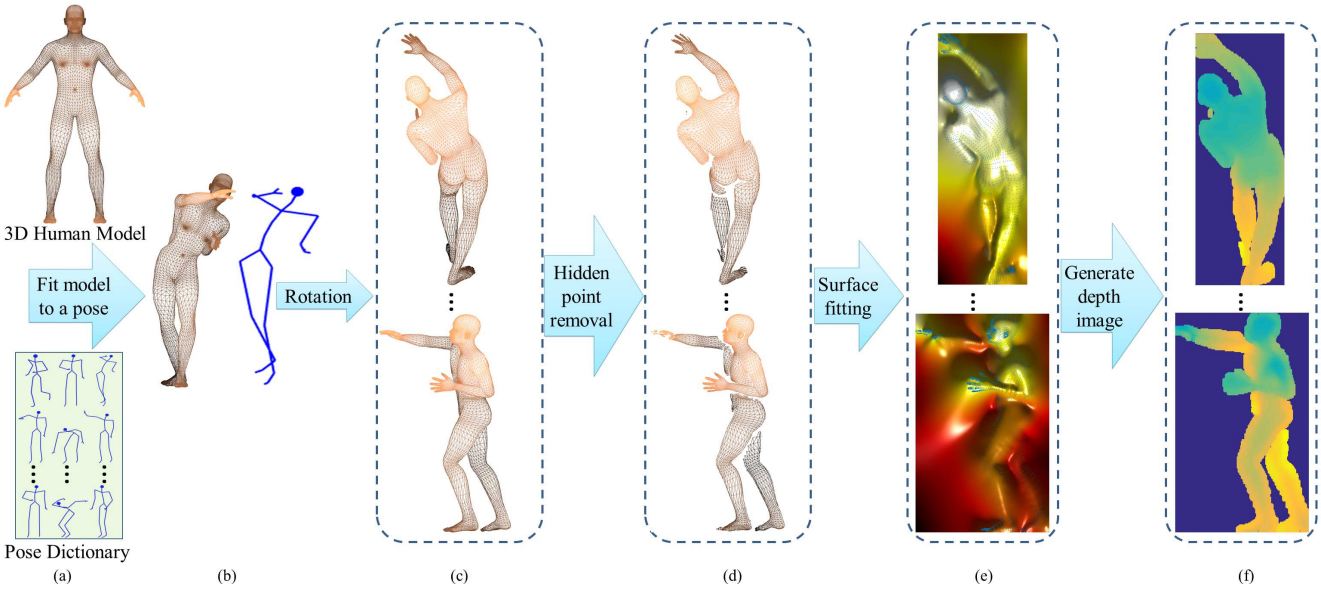

Figure 1: Proposed pipeline for generating synthetic depth images. (a)-(b) A 3D human body is fitted to each mocap skeleton, (c) rendered from 180 different viewing directions (see Fig. 2), (d) processed for hidden point removal, (e) fitted with smooth surfaces [8] and finally (f) processed for removal of extrapolated points and normalized in the $0-255$ range to generate depth images.

recognition in videos acquired from novel views. For cross-view action recognition, Rahmani and Mian [41] proposed a deep network which learns a set of non-linear transformations from multiple source views to a single canonical view. However, this method uses a fixed canonical view (frontal) as target view and learns the transfer model from hand-crafted features *i.e*. motion trajectories.

All the above deep models are designed for RGB videos and learning these models requires a large corpus of action video training data which is unavailable in the case of depth videos. Furthermore, motion trajectory and optical flow features, besides being hand crafted, are unreliable in the case of depth videos [34]. These limitations motivate us to propose methods for learning a view-invariant human pose model and for reliable encoding of the temporal structure of depth videos for cross-view action recognition.

3. Generating synthetic training data

We propose a pipeline (see Fig. 1) for generating synthetic depth images of different human body shapes in a large number of poses rendered from numerous viewing directions. Details of each step are given below.

3.1. Building a pose dictionary

The set of all possible human body poses is extremely large. Therefore, we build a dictionary that contains the most representative ones. We use the CMU Motion Capture database [2] which contains over 2600 mocap sequences (over $200K$ poses) of subjects performing a variety of actions. The 3D joint positions in the dataset are quite accurate as they were captured using a high-precision camera array and body joint markers. However, many poses look similar. Using the skeletal distance function [49], we apply k-means clustering to $50K$ randomly selected mocap poses and select 339 representative ones to form a pose dictionary which is later used to generate synthetic depth images and train the CNN. Note that we do not use the action labels provided with the CMU mocap data [2].

3.2. Full 3D human body models

Bogo *et al*. [6] developed the FAUST dataset containing full 3D human body scans of 10 individuals in 30 poses. However, skeleton data is not provided for the scans. Another way to generate 3D human model is to use the open source MakeHuman software [3] which can generate different synthetic human shapes in a predefined pose and provide the joint positions which can be used for changing the human pose. We use this technique for generating the 3D human body models in our work.

3.3. Fitting 3D human models to mocap data

Several methods [4, 30] have been proposed to fit a human model to motion capture skeleton data of a person. For instance, the SCAPE method [4] learns pose and body-shape deformation models from scans of different human bodies in a few poses. Given a set of markers, SCAPE constructs a full mesh which is consistent with the SCAPE models and best matches with the given markers. These methods aim to generate fine-grain human bodies in a variety of poses. However, real-time depth cameras generally have low resolution. Therefore, we use the open source Blender package [1] to fit 3D human models to mocap data. Given a 3D human model generated by the MakeHuman software and a mocap frame, Blender normalizes the mo-

cap skeleton with respect to the skeleton data of the human model and then fits the model to the normalized mocap data. This process results in a synthetic full 3D human body pose corresponding to the given mocap skeleton (Fig. 1-(b)).

3.4. Rendering from multiple viewpoints

We deploy a total of 180 synthetic cameras (at distinct latitudes and longitudes) on a hemisphere surrounding the subject as shown in Fig. 2. For each camera, we remove self-occluded points. First, we perform back-face culling by removing points whose normals face away from the camera and then perform hidden point removal [22] on the remaining points. Figure 1-(c) shows the full human model from two different views and Fig. 1-(d) shows the corresponding 3D pointclouds after removing the hidden points.

3.5. Surface fitting

So far, we have generated 3D pointclouds of different 3D human models in different poses. To generate their corresponding depth images, we fit a surface of the form $z(x, y)$ to each 3D pointcloud using *gridfit* [8] which approximates the 3D pointcloud as closely as possible. Figure 1-(e) shows two surfaces constructed using *gridfit* for two views of a human pose. The extrapolated points that do not belong to the human body are set to zero using a neighborhood test with the pointcloud that was used for surface fitting. The z values are normalized in the $0-255$ range to get the final depth image. Figure 1-(f) shows two depth images corresponding to the surfaces in Fig. 1-(e). It is worth mentioning that surface fitting is not required for real data at test time as real data is already in the form of depth images.

4. View-invariant human pose representation

Realistic action videos lie on non-linear manifolds, especially when actions are captured from different views. However, most cross-view action recognition methods [17, 27, 39, 40, 60, 72] represent the connection between action videos captured from two different views as a sequence of linear transformations of action descriptors. Moreover, such methods do not scale well to new action classes because they must repeat the computationally expensive model learning process. To overcome these problems, we propose a general view-invariant human pose representation model that learns to transfer human poses from any view to a shared view-invariant high-level space.

4.1. Model architecture and learning

Our proposed model is a deep convolutional neural network (CNN) whose architecture is similar to [18] except that we replace the last fully-connected layer with a 339-neurons layer. Let $C(k, n, s)$ denote a convolutional layer with kernel size $k \times k$, n filters and a stride of s, $P(k, s)$ a max pooling layer of kernel size $k \times k$ and stride s, N a normalization layer, RL a rectified linear unit, $FC(n)$ a fully connected layer with n filters and $D(r)$ a dropout layer with dropout ratio r. The architecture of our CNN follows: $C(11, 96, 4) \rightarrow RL \rightarrow P(3, 2) \rightarrow N \rightarrow C(5, 256, 1) \rightarrow RL \rightarrow P(3, 2) \rightarrow N \rightarrow C(3, 384, 1) \rightarrow RL \rightarrow C(3, 384, 1) \rightarrow RL \rightarrow C(3, 256, 1) \rightarrow RL \rightarrow P(3, 2) \rightarrow FC(4096) \rightarrow RL \rightarrow D(0.5) \rightarrow FC(4096) \rightarrow RL \rightarrow D(0.5) \rightarrow FC(339)$. We refer to the fully-connected layers as fc_6, fc_7, and fc_8, respectively. During learning, a softmax loss layer is added at the end of the network.

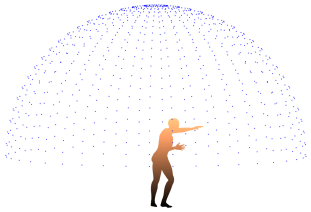

Figure 2: Each point on the hemisphere corresponds to a virtual camera looking towards the center of the sphere.

For each pose $i = 1, \cdots, 339$ in the dictionary, the corresponding synthetic depth images from all 180 synthetic cameras are generated using our proposed pipeline and assigned the same class label i. Thus, our training dataset consists of 339 human pose classes. We use the synthetic depth images from 162 randomly selected cameras as the training set and those from the remaining 18 cameras as the validation set. Proper initialization is a key for successful training of CNNs and for avoiding over-fitting. We initialize the CNN with a model that was trained on approximately 1.2 million RGB images from the 2012 ImageNet challenge and then fine-tuned on depth images from NYUD2 [18]. We train our CNN with back-propagation and use an initial learning rate of 0.01 for the convolution layers and 0.01 for the fully-connected layers. We use a momentum of 0.9 and a weight decay of 0.0005. We train the network for $21K$ iterations. During training, the input images are flipped horizontally with a probability of 0.5.

4.2. Inference

So far, we have learned a deep CNN model whose input is a human pose depth image and output is the corresponding pose class. The proposed CNN is able to classify only 339 pose classes which do not cover all possible human poses. However, the fully-connected layers (*e.g.* fc_6 and fc_7) of the learned model encode the view-invariant high-level representation of human poses. To use this model for extracting view-invariant features from real depth videos, we perform the following two steps.

Pre-processing: The synthetic depth dataset used for training the proposed CNN model contains depth images of only human body poses. Therefore, for extracting features from a real depth image, we pass the segmented human body image through our learned model. Fortunately, the Kinect camera is able to discern the human body from the rest of the scene and provide a segmented image (*i.e.* human body)

in real-time (see Fig. 4). The segmented depth image is then cropped to the bounds of the region of interest *i.e.* human body, and converted to a form that is compatible with the learned CNN model. More precisely, the depth values of the region of interest are normalized in the range $0-255$ and the image is resized to 227×227. The average depth image calculated from training images is then subtracted from it. In case human body segmentations are not available, our method still achieves state-of-the-art recognition accuracy e.g. see our result on the MSR Action Pairs3D [34] dataset in the supplementary material.

Feature extraction: For each depth video frame, view-invariant features are computed by forward propagating the mean-subtracted 227×227 depth image through the CNN and the outputs of the fc_7 layer are used as the view-invariant frame descriptor. Our experiments show that using the outputs of this layer achieves better recognition accuracy than fc_6 (see supplementary material for results).

5. Temporal modeling and classification

To represent an action sequence with our CNN model, we feed forward the depth images sequentially through the network and temporally align the fc_7 layer features. Recall that no surface fitting is required for real depth images. Depth images captured by low cost real-time cameras have high levels of noise [64]. Moreover, the correct region of interest, containing only the human, extracted in real-time by Kinect cameras is not always accurate and may contain some parts of the background as shown in Fig. 4. Finally, to match two video segments, they must be temporally aligned. Therefore, we need a representation that is robust to noisy depth images, inaccurate segmentations and temporal misalignments between different video segments.

The Fourier Temporal Pyramid (FTP) [59] is shown to be successful for encoding temporal variations of noisy data. We employ the FTP representation since it is robust to noise and temporal misalignments. In addition to the global Fourier coefficients, we recursively partition the actions into a pyramid, and use the short time Fourier transform for all the segments to better capture the temporal structure of the action videos. The final action video descriptor is the concatenation of the Fourier coefficients from all the segments.

Let f denote the number of frames in a given action video and $m = 4096$ the number of neurons in the fully-connected fc_7 layer of the proposed model. Let us denote each neuron output of the i-th video sample by $B^i_{j,t}$, where $j = 1 \dots m$ is the neuron number and $t = 1 \dots f$ is the frame number. We apply the Short Fourier Transform [33] to $B^i_j = \left[B^i_{j,1}\ B^i_{j,2}\ \cdots\ B^i_{j,f}\right]$ and keep the first q low frequency coefficients. Next, we divide B^i_j into two segments and apply the Short Fourier Transform again to each individual segment to obtain its low frequency coefficients. We repeat this process l times and compute a Fourier Temporal Pyramid descriptor, A^i_j, for each neuron j by concatenating the low-frequency coefficients at all levels of the pyramid. Thus $A_j \in \mathbb{R}^\gamma$ where $\gamma = 2^l \times q$. We refer to the concatenated descriptor $A^i = \left[A^i_1\ A^i_2\ \cdots A^i_j\ \cdots\ A^i_m\right]^\top$ as the spatio-temporal features for the i-th video sample.

Each neuron in the fully-connected fc_7 layer contributes differently to different pose classes and hence, different actions. This is because each neuron in the fc_7 layer is connected to the penultimate layer, fc_8, with different weights. We define a *neurons-set* as a conjunction of neurons whose outputs are more discriminative for a particular action. If a neuron is considered for a particular action, then all its output FTP features must be selected and if a neuron is not selected then all its output features must be discarded. We discover the discriminative neurons-sets by solving an ℓ_1/ℓ_2-norm regularized least squares problem [70]:

$$\min_X \frac{1}{2}||AX - Y||_2^2 + \lambda \sum_{j=1}^{m} ||X_{G_j}||_2, \qquad (1)$$

where $A = \left[A^1\ A^2\ \cdots\ A^n\right]^\top \in \mathbb{R}^{n\times v}$, $Y \in \mathbb{R}^{n\times 1}$, $X \in \mathbb{R}^{v\times 1}$ is divided into m non-overlapping groups $X_{G_1}, X_{G_2}, \cdots, X_{G_m}$, and $v = \gamma \times 4096$ denotes the dimension of feature vector of each video sample. Such a solution incorporates a grouping structure by inducing sparsity at the neuron level and smoothness in the individual neuron output feature vector A_j. We solve this optimization function using the one-vs-all strategy for all action classes which gives us a sparse discriminative *neurons-set* for each action class. This process also reduces the complexity of the action specific classifiers and leads to better generalization of the learning [24, 32]. Figure 3 shows an overview of the proposed temporal modeling and classification method.

6. Experiments

We evaluated our proposed algorithm on two multiview and three single-view benchmark datasets. The former includes the Northwestern-UCLA Multiview Action3D [60] and UWA3D Multiview Activity II [40] datasets whereas the latter includes MSR Action Pairs3D [34], MSR Daily Activity3D [58], and MSR Gesture3D [26,57] datasets. The results on single-view datasets are provided in the supplementary material.

We report action recognition results of our method for unseen (novel) and unknown views, *i.e.* we assume that no videos, labels or correspondences from the target view are available at training time. More importantly, we use the same CNN model, learned from synthetic data, for all five datasets to show the generalization strength of our model and to show that our model can be applied to any depth action video without the need for re-training or fine-tuning. We used the MatConvNet toolbox [53] for implementing convolutional neural networks. In our experiments, we set the number of Fourier Pyramid levels $l = 3$, and the number

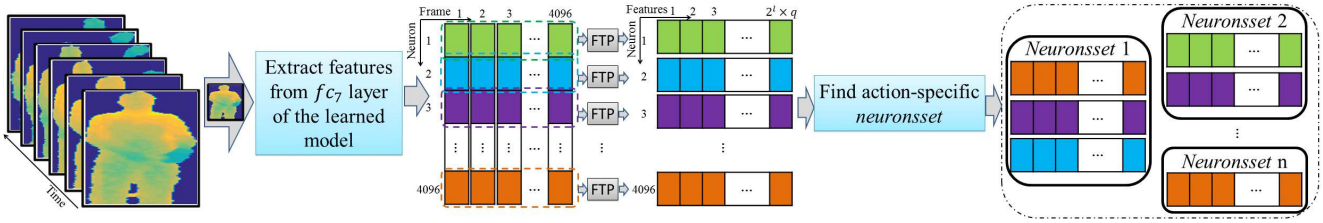

Figure 3: Overview of the proposed temporal modeling and classification. Video frames are individually passed through the CNN model and the Fourier Temporal Pyramid features are extracted from the time series of each neuron output of the CNN representation layer.

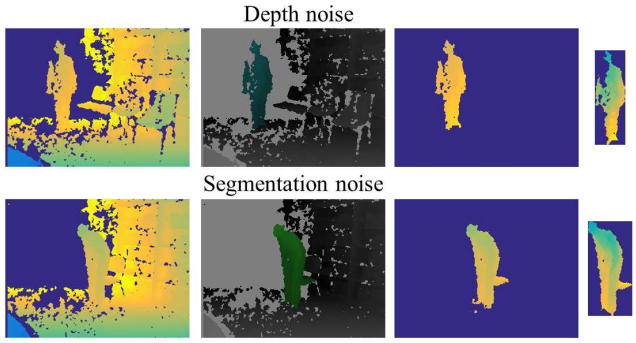

Figure 4: Visualizing the data and segmentation noise. Column wise: Raw depth images from Kinect; after background removal by Kinect (the green pixels); after background removal; the normalized depth image which feeds to our proposed model.

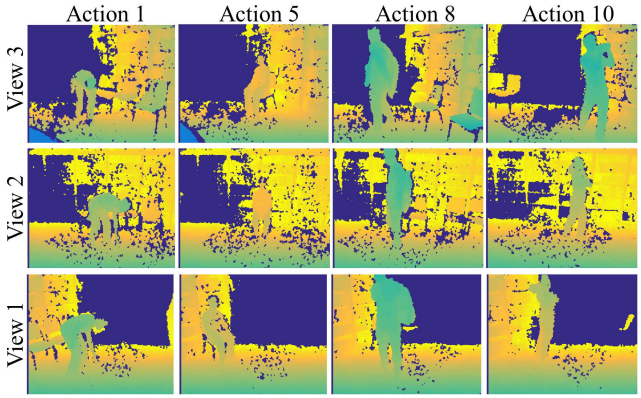

Figure 5: Sample depth images from the Northwestern-UCLA dataset [60] captured simultaneously by 3 Kinect cameras.

of low frequency Fourier coefficients $q = 4$ using cross-validation on training samples. The learned CNN model and MATLAB code of our method are freely available [1].

From here on, we refer to our proposed view-invariant human pose representation model (Section 4) and our proposed temporal modeling (Section 5) as HPM and TM, respectively. In addition to other compared methods, we report the accuracy of our defined baseline method which uses a similar approach to [15] but with the CNN model that was fine-tuned on depth images from NYUD2 [18]. We report the recognition accuracy of our method in two different settings: (1) HPM where we apply average pooling on the CNN features of all frames of a video to obtain its representation, and (2) HPM+TM where we employ the proposed temporal modeling approach on the CNN features to capture the temporal structure of the videos.

6.1. Northwestern-UCLA dataset [60]

This dataset contains RGB, depth and human skeleton data captured simultaneously by 3 Kinect cameras from different views. It consists of 10 action classes including: (1) *pick up with one hand*, (2) *pick up with two hands*, (3) *drop trash*, (4) *walk around*, (5) *sit down*, (6) *stand up*, (7) *donning*, (8) *doffing*, (9) *throw*, and (10) *carry*. Each action was performed by 10 subjects 1 to 6 times. Fig. 5 shows sample depth images of four actions captured by the three cameras.

We follow [60] and use the samples from the first two cameras for training and the samples from the third camera for testing. Comparative results are shown in Table 1. The recognition accuracy of the proposed method in the first setting (HPM) significantly outperforms our defined baseline and all existing methods excluding HOPC [39, 40]. This demonstrates the effectiveness of the proposed training approach. However, average pooling is unable to fully encode the temporal structure of actions. Combining our temporal modeling algorithm with the HPM significantly improves the recognition accuracy by 14% and achieves 92.0% accuracy. Moreover, it dramatically outperforms state-of-the-art methods irrespective of the modality they use.

Figure 6 compares the action specific recognition accuracies of our method in the two settings. The proposed temporal modeling (HPM+TM) achieves significantly higher accuracies than average pooling (HPM) for most action classes. The recognition accuracies of the *stand up* and *sit down* actions significantly improve, because these actions result in similar descriptors through average pooling.

It is important to emphasize that the proposed view-invariant pose model was learned from synthetic depth images generated from a small number of human poses, *i.e.* size of the pose dictionary was 339. A search for many human poses such as *drop trash, donning* and *doffing* from the Northwestern-UCLA dataset returns no results in the pose dictionary or mocap data. Moreover, some activities in this dataset (*e.g. donning, doffing, carry*) involve human-object interactions. Yet, the proposed model is able to achieve high recognition accuracies for these actions.

[1] http://www.csse.uwa.edu.au/∼ajmal/code.html

Table 1: Comparison of action recognition accuracy (%) on the Northwestern-UCLA Multiview Action3D dataset.

Method	Recognition accuracy(%)
Input: RGB images	
AOG [60]	73.3
Action Tube [15]	61.5
LRCN [9]	64.7
NKTM [41]	75.8
Input: Depth images+Skeleton data	
Actionlet [59]	76.0
LARP [54]	74.2
Input: Depth images	
CCD [7]	34.4
DVV [27]	52.1
CVP [72]	53.5
HON4D [34]	39.9
SNV [67]	42.8
HOPC [39]	80.0
baseline	70.2
Ours (HPM)	78.1
Ours (HPM+TM)	**92.0**

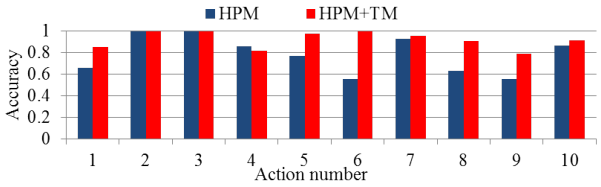

Figure 6: Class specific action recognition accuracies of our proposed method in two settings: 1) HPM and 2) HPM+TM on the Northwestern-UCLA Multiview Action3D dataset.

6.2. UWA3DII dataset [40]

This dataset consists of 30 human actions performed by 10 subjects with different scales: (1) *one hand waving*, (2) *one hand Punching*, (3) *two hand waving*, (4) *two hand punching*, (5) *sitting down*, (6) *standing up*, (7) *vibrating*, (8) *falling down*, (9) *holding chest*, (10) *holding head*, (11) *holding back*, (12) *walking*, (13) *irregular walking*, (14) *lying down*, (15) *turning around*, (16) *drinking*, (17) *phone answering*, (18) *bending*, (19) *jumping jack*, (20) *running*, (21) *picking up*, (22) *putting down*, (23) *kicking*, (24) *jumping*, (25) *dancing*, (26) *moping floor*, (27) *sneezing*, (28) *sitting down (chair)*, (29) *squatting*, and (30) *coughing*. Each subject performed 30 actions 4 times. Each time the action was captured from a different viewpoint (front, top, left and right). This dataset is challenging because the videos were acquired at different times from varying viewpoints and the data contains self-occlusions, more action classes and high similarity across action classes. Moreover, in the top view, the lower part of the body was not properly captured because of occlusion. Figure 7 shows sample depth images of four actions observed from the four viewpoints.

We follow [40] and use videos from two views for training and videos from the remaining views as test data. Table 2 summarizes our results. Our HPM significantly outperforms the state-of-the-art methods excluding NKTM [41] on all view pairs. However, NKTM [41] must extract hand-crafted dense motion trajectories prior to using the model. The combination of HPM and our proposed temporal modeling (HPM+TM) dramatically improves the average recognition accuracy to 76.9% which is over 13.4% higher than the nearest competitor (NKTM). It is interesting to note that our method achieves 76.5% average recognition accuracy when view 4 is used as the test view. As shown in Fig. 7, view 4 is the top view where the lower part of the body was not properly captured by the videos.

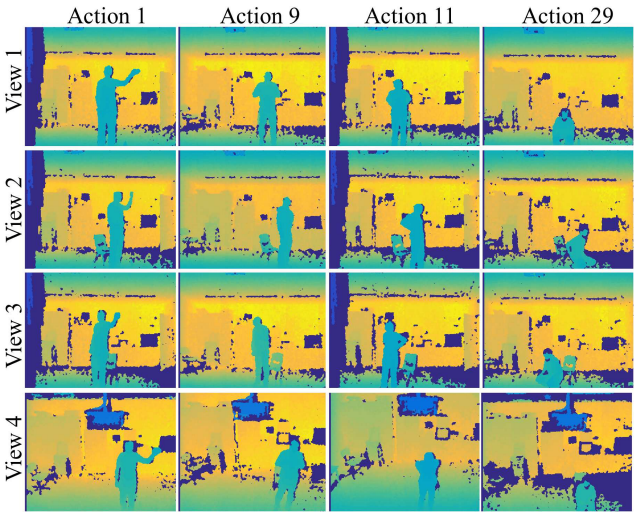

Figure 7: Sample depth images from the UWA3D Multiview ActivityII dataset [40] captured by one camera from 4 different views.

Figure 8 compares the class specific action recognition accuracies of our proposed method in the two settings. HPM+TM achieves significantly higher accuracies than using average pooling for most action classes. The recognition accuracies of the *stand up* and *sit down* actions dramatically improve which again demonstrates the effectiveness of our proposed temporal modeling method.

It is important to emphasize that for many human poses in the UWA3DII dataset such as *two hand waving, holding chest, holding head, holding back, sneezing* and *coughing*, a similar pose does not exist in the CMU mocap data and hence the pose dictionary used to learn our model. However, our method still achieves high recognition accuracies for these actions.

6.3. Computation time

Our model can be used in real-time applications as it does not involve complex feature processing or computationally expensive training and testing phases. With a Matlab implementation, our method can process 25 frames per second on a 3.4GHz machine with 24GB RAM. The nearest competitor, in terms of accuracy, HOPC [39, 40] is 50 times slower than our method. Table 3 compares the speed of our

Table 2: Comparison of action recognition accuracy (%) on the UWA3D Multiview ActivityII dataset. Each time two views are used for training and the remaining two views are individually used for testing.

Training views	V_1 & V_2		V_1 & V_3		V_1 & V_4		V_2 & V_3		V_2 & V_4		V_3 & V_4		Mean
Test view	V_3	V_4	V_2	V_4	V_2	V_3	V_1	V_4	V_1	V_3	V_1	V_2	
Input: RGB images													
AOG [60]	47.3	39.7	43.0	30.5	35.0	42.2	50.7	28.6	51.0	43.2	51.6	44.2	42.3
Action Tube [15]	49.1	18.2	39.6	17.8	35.1	39.0	52.0	15.2	47.2	44.6	49.1	36.9	37.0
LRCN [9]	53.9	20.6	43.6	18.6	37.2	43.6	56.0	20.0	50.5	44.8	53.3	41.6	40.3
NKTM [41]	60.1	61.3	57.1	65.1	61.6	66.8	70.6	59.5	73.2	59.3	72.5	54.5	63.5
Input: Depth images+Skeleton data													
Actionlet [59]	45.0	40.4	35.1	36.9	34.7	36.0	49.5	29.3	57.1	35.4	49.0	29.3	39.8
LARP [54]	49.4	42.8	34.6	39.7	38.1	44.8	53.3	33.5	53.6	41.2	56.7	32.6	43.4
Input: Depth images													
CCD [7]	10.5	13.6	10.3	12.8	11.1	8.3	10.0	7.7	13.1	13.0	12.9	10.8	11.2
DVV [27]	23.5	25.9	23.6	26.9	22.3	20.2	22.1	24.5	24.9	23.1	28.3	23.8	24.1
CVP [72]	25.0	25.6	25.5	28.2	24.7	24.0	23.0	24.5	26.6	23.3	30.3	26.8	25.6
HON4D [34]	31.1	23.0	21.9	10.0	36.6	32.6	47.0	22.7	36.6	16.5	41.4	26.8	28.9
SNV [67]	31.9	25.7	23.0	13.1	38.4	34.0	43.3	24.2	36.9	20.3	38.6	29.0	29.9
HOPC [39]	52.7	51.8	59.0	57.5	42.8	44.2	58.1	38.4	63.2	43.8	66.3	48.0	52.2
baseline	53.1	47.3	50.1	49.2	35.5	42.3	52.2	31.6	65.2	51.6	67.8	50.9	49.7
Ours (HPM)	71.3	58.4	58.3	64.4	38.7	51.5	58.0	42.7	69.5	64.6	71.7	57.1	58.9
Ours (HPM+TM)	**80.6**	**80.5**	**75.2**	**82.0**	**65.4**	**72.0**	**77.3**	**67.0**	**83.6**	**81.0**	**83.6**	**74.1**	**76.9**

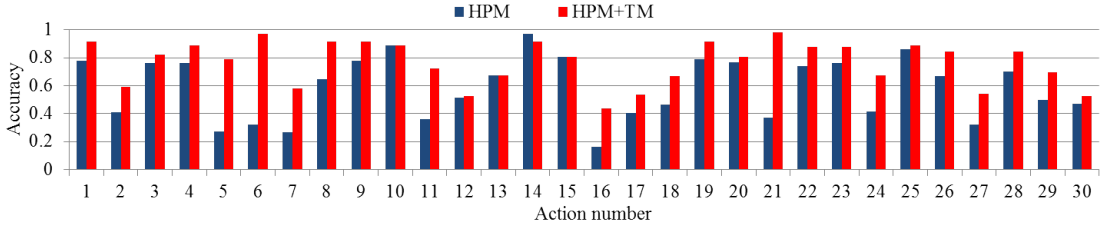

Figure 8: Per class recognition accuracy of the proposed HPM and HPM+TM on the UWA3D Multiview ActivityII dataset.

method to the nearest competitors from each modality.

It is interesting to note that our technique outperforms the current state-of-the-art on both cross-view datasets while using the same CNN model learned from synthetic data. This shows the generalization ability of our CNN model and its ability to be deployed for online action recognition because the cost of adding a new action class is equal to training the action specific classifiers. On the other hand, adding more action classes is computationally expensive for existing techniques [17, 39–41, 54, 59, 60]. NKTM [41] must extract computationally expensive motion trajectories. LARP [60] requires to compute a nominal curve for the new action and warp all the training curves to this nominal curve using DTW. Similarly, HOPC [60] computes computationally expensive spatio-temporal features.

Table 3: Average computation speed (fps: frames per second). On-line training speed is that of adding a new action class.

Method	On-line training	Testing
NKTM [41]	12 fps	16 fps
LARP [54]	0.1 fps	10 fps
HOPC [39]	0.04 fps	0.5 fps
Ours	**22** fps	**25** fps

7. Conclusion

We proposed a deep CNN model that represents depth images of different human poses acquired from multiple views in a view-invariant high-level space. To train the model, we proposed a framework for generating a large corpus of training data synthetically by fitting realistic human models to real mocap data and rendering it from multiple viewpoints. We also introduced a temporal modeling and classification method which encodes the temporal structures of actions and discovers a discriminative set of neurons corresponding to each action class. The proposed method is scalable as it requires to be trained only once using synthetic depth images and generalizes well to real data. Experiments on benchmark multiview datasets show that the proposed approach outperforms existing state-of-the-art. Our method performs equally well on single-view benchmark datasets (see supplementary material) and generalizes to hand gestures even though the CNN model was trained on full human body poses.

Acknowledgment We thank the authors of [9, 15, 27, 34, 54, 58, 67] for making their codes publicly available. We thank NVIDIA for their K40 GPU donation. This research was supported by ARC grant DP110102399 and DP160101458.

References

[1] Blender: a 3D modelling and rendering package. http://www.blender.org/. 3

[2] CMU motion capture database. http://mocap.cs.cmu.edu/. 1, 3

[3] MakeHuman: an open source 3D computer graphics software. http://www.makehuman.org/. 3

[4] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: shape completion and animation of people. *ACM SIGGRAPH*, 2005. 3

[5] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005. 2

[6] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *CVPR*, 2014. 3

[7] Z. Cheng, L. Qin, Y. Ye, Q. Huang, and Q. Tian. Human daily action analysis with multi-view and color-depth data. In *ECCVW*, 2012. 7, 8

[8] J. R. D'Errico. Surface fitting using gridfit. In *MATLAB Central File Exchange*, 2008. 3, 4

[9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2, 7, 8

[10] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015. 2

[11] A. Farhadi and M. K. Tabrizi. Learning to recognize activities from the wrong view point. In *ECCV*, 2008. 2

[12] A. Farhadi, M. K. Tabrizi, I. Endres, and D. A. Forsyth. A latent model of discriminative aspect. In *ICCV*, 2009. 2

[13] D. Gavrila and L. Davis. 3D model-based tracking of humans in action: a multi-view approach. In *CVPR*, 1996. 2

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[15] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015. 1, 2, 6, 7, 8

[16] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, 2014.

[17] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham. 3D pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In *CVPR*, 2014. 1, 2, 4, 8

[18] S. Gupta, R. Girshick, P. Arbelez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*, 2014. 4, 6

[19] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *PAMI*, 2013. 2

[20] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *PAMI*, 2011. 1

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1, 2

[22] S. Katz, A. Tal, and R. Basri. Direct visibility of point sets. *ACM Transactions on Graphics*, 2007. 4

[23] Y. Kong and Y. Fu. Bilinear heterogeneous information machine for RGB-D action recognition. In *CVPR*, 2015. 1

[24] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomial logistic regression: fast algorithms and generalization bounds. *PAMI*, 2005. 5

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[26] A. Kurakin, Z. Zhang, and Z. Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *EUSIPCO*, 2012. 5

[27] R. Li and T. Zickler. Discriminative virtual views for cross-view action recognition. In *CVPR*, 2012. 2, 4, 7, 8

[28] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011. 2

[29] J. Liu, M. Shah, B. Kuipersy, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011. 2

[30] M. Loper, N. Mahmood, and M. J. Black. MoSh: motion and shape capture from sparse markers. *ACM Transactions on Graphics*, 2014. 3

[31] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR*, 2011. 1

[32] A. Maurer and M. Pontil. Structured sparsity and generalization. *Journal of Machine Learning Research*, 2012. 5

[33] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. Discrete time signal processing. *Prentice Hall Signal Processing Series*, 1999. 5

[34] O. Oreifej and Z. Liu. HON4D: histogram of oriented 4D normals for activity recognition from depth sequences. In *CVPR*, 2013. 1, 2, 3, 5, 7, 8

[35] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *IJCV*, 2006. 2

[36] H. Rahmani, D. Q. Huynh, A. Mahmood, and A. Mian. Discriminative human action classification using locality-constrained linear coding. *Pattern Recognition Letters*, 2015. 1

[37] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian. Action classification with locality-constrained linear coding. In *ICPR*, 2014. 1

[38] H. Rahmani, A. Mahmood, A. Mian, and D. Huynh. Real time action recognition using histograms of depth gradients and random decision forests. In *WACV*, 2014. 1

[39] H. Rahmani, A. Mahmood, D. Q Huynh, and A. Mian. HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition. In *ECCV*, 2014. 1, 2, 4, 6, 7, 8

[40] H. Rahmani, A. Mahmood, D. Q Huynh, and A. Mian. Histogram of oriented principal components for cross-view action recognition. *PAMI*, 2016. 1, 2, 4, 5, 6, 7, 8

[41] H. Rahmani and A. Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *CVPR*, 2015. 1, 2, 3, 7, 8

[42] H. Rahmani, A. Mian, and M. Shah. Learning a deep model for human action recognition from novel viewpoints. In *arXiv:1602.00828*, 2016. 1

[43] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *IJCV*, 2002. 2

[44] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012. 1

[45] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016. 2

[46] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang. Deep multimodal feature analysis for action recognition in RGB+D videos. In *arXiv*, 2016. 2

[47] A. Shahroudy, T.-T. Ng, Q. Yang, and G. Wang. Multimodal multipart learning for action recognition in depth videos. *PAMI*, 2016. 1

[48] A. Shahroudy, G. Wang, T.-T. Ng, and Q. Yang. Multi-modal feature fusion for action recognition in RGB-D sequences. In *International Symposium on Communications, Control and Signal Processing (ISCCSP14)*, 2014. 1

[49] G. Shakhnarovich. *Learning Task-Specific Similarity*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2005. 3

[50] 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 *CVPR*, 2011. 2

[51] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2

[52] T. Syeda-Mahmood, A. Vasilescu, and S. Sethi. Action recognition from arbitrary views using 3D exemplars. In *ICCV*, 2007. 2

[53] A. Vedaldi and K. Lenc. MatConvNet - Convolutional Neural Networks for MATLAB. In *ACM International Conference on Multimedia*, 2015. 5

[54] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a Lie group. In *CVPR*, 2014. 1, 2, 7, 8

[55] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 1

[56] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1

[57] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3D action recognition with random occupancy patterns. In *ECCV*, 2012. 1, 2, 5

[58] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012. 2, 5, 8

[59] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Learning actionlet ensemble for 3D human action recognition. *PAMI*, 2013. 1, 2, 5, 7, 8

[60] J. Wang, X. Nie, Y. Xia, Y. Wu, and S. Zhu. Cross-view action modeling, learning and recognition. In *CVPR*, 2014. 1, 2, 4, 5, 6, 7, 8

[61] D. Weinland, M. Özuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV*, 2010. 2

[62] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 2006. 2

[63] S. Wu, O. Oreifej, and M. Shah. Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories. In *ICCV*, 2011. 1

[64] L. Xia and J. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recongition using depth camera. In *CVPR*, 2013. 1, 2, 5

[65] L. Xia, C. C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *CVPRW*, 2012. 2

[66] P. Yan, S. M. Khan, and M. Shah. Learning 4D action feature models for arbitrary view action recognition. In *CVPR*, 2008. 2

[67] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. In *CVPR*, 2014. 1, 2, 7, 8

[68] X. Yang, C. Zhang, and Y. Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *ACM ICM*, 2012. 1

[69] A. Yilmaz and M. Shah. Action sketch: a novel action representation. In *CVPR*, 2005. 2

[70] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, 2006. 5

[71] W. Zaremba and I. Sutskever. Learning to execute. *arXiv:1410.4615*, 2014.

[72] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi. Cross-view action recognition via a continuous virtual path. In *CVPR*, 2013. 2, 4, 7, 8

[73] J. Zheng, Z. Jiang, J. Phillips, and R. Chellappa. Cross-view action recognition via a transferable dictionary pair. In *BMVC*, 2012. 2

[74] Y. Zhou, B. Ni, R. Hong, M. Wang, and Q. Tian. Interaction part mining: A mid-level approach for fine-grained action recognition. In *CVPR*, 2015. 1