

Learning Human Pose Models from Synthesized Data for Robust RGB-D Action Recognition

Jian Liu · Hossein Rahmani · Naveed Akhtar · Ajmal Mian

Received: date / Accepted: date

Abstract We propose Human Pose Models that represent RGB and depth images of human poses independent of clothing textures, backgrounds, lighting conditions, body shapes and camera viewpoints. Learning such universal models requires training images where all factors are varied for every human pose. Capturing such data is prohibitively expensive. Therefore, we develop a framework for synthesizing the training data. First, we learn representative human poses from a large corpus of real motion captured human skeleton data. Next, we fit synthetic 3D humans with different body shapes to each pose and render each from 180 camera viewpoints while randomly varying the clothing textures, background and lighting. Generative Adversarial Networks are employed to minimize the gap between synthetic and real image distributions. CNN models are then learned that transfer human poses to a shared high-level invariant space. The learned CNN models are then used as invariant feature extractors from real RGB and depth frames of human action videos and

the temporal variations are modelled by Fourier Temporal Pyramid. Finally, linear SVM is used for classification. Experiments on three benchmark human action datasets show that our algorithm outperforms existing methods by significant margins for RGB only and RGB-D action recognition.

Keywords Human Action Recognition · Cross-view · Cross-subject · Depth Sensor · CNN · GAN

1 Introduction

Human action recognition has many applications in security, surveillance, sports analysis, human computer interaction and video retrieval. However, automatic human action recognition algorithms are still challenged by noises due to action irrelevant factors such as changing camera viewpoints, clothing textures, body shapes, backgrounds and illumination conditions. In this paper, we address these challenges to perform robust human action recognition in conventional RGB videos and RGB-D videos obtained from range sensors.

A human action can be defined as a collection of sequentially organized human poses where the action is encoded in the way the human pose transitions from one pose to the other. However, for action classification, a human pose must be represented in a way that is invariant to the above conditions. Since some human poses are common between multiple actions and the space of possible human poses is much smaller compared to that of possible human actions, we first model the human pose independently and then model the actions as the temporal variations between human poses.

To suppress action irrelevant information in videos, many techniques use dense trajectories (Gupta et al, 2014; Wang et al, 2011, 2013a; Wang and Schmid, 2013)

Jian Liu is with the School of Computer Science and Software Engineering, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009
E-mail: jian.liu@research.uwa.edu.au

Hossein Rahmani is with School of Computing and Communications, Lancaster University, Lancashire, England
E-mail: h.rahmani@lancaster.ac.uk

Naveed Akhtar is with the School of Computer Science and Software Engineering, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009
E-mail: naveed.akhtar@uwa.edu.au

Ajmal Mian is with the School of Computer Science and Software Engineering, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009
E-mail: ajmal.mian@uwa.edu.au
Tel.: +61-8-64882702
Fax: +61-8-64881089



Fig. 1 Block diagram of the proposed synthetic RGB data generation. Representative human poses are learned from CMU MoCap skeleton database and a 3D human model is fitted to each skeleton. Four different 3D human body shapes are used. Clothing textures are randomly selected from a choice of 262 textures for shirts and 183 for trousers. Each model is placed in a random background that is either a 2D scene image, or a 360° spherical HDR image. We use 360° spherical HDR images obtained from Google Images. We use illumination from three random intensity lamps and render from 180 camera viewpoints to generate RGB training images with known pose labels. The shown final image is without further refinement with GAN.

or Hanklets (Li et al, 2012) which encode only the temporal cues that are essential for action classification. Such methods have shown good performance for human action recognition in conventional videos. However, they are still sensitive to viewpoint variations and do not fully exploit the appearance (human pose) information. Dense trajectories are also noisy and contain self occlusion artefacts.

While appearance is an important cue for action recognition, human poses appear very differently from different camera viewpoints. Research efforts have been made to model these variations. For example, synthetic 2D human poses from many viewpoints and their transitions were used for action recognition in (Lv and Nevatia, 2007). However, 3D viewpoint variations cannot be modelled accurately using 2D human poses. Spatio-temporal 3D occupancy grids built from multiple viewpoints were used in (Weinland et al, 2007) to achieve view-invariant action recognition. However, occupancy grids rely on silhouettes which are noisy in real videos. In this paper, we use full 3D human models to learn a representation of the human pose that is not only invariant to viewpoint but also to other action irrelevant factors such as background, clothing and illumination.

Our contributions can be summarized as follows. Firstly, we propose a method for generating RGB and Depth images of human poses using Computer Graphics and Generative Adversarial Network (GAN) training. We learn representative human poses by clustering real human joint/skeleton data obtained with motion capture technology (CMU MoCap database¹). Each representative pose skeleton is fitted with synthetic 3D human models and then placed in random scenes, given different clothes, illuminated from multiple directions and rendered from 180 camera viewpoints to generate RGB and depth images of the human poses with known labels. Figure 1 illustrates the proposed RGB

data generation pipeline. Depth images are generated in a similar way except that they are devoid of texture and background. We employ GANs to minimize the gap between the distributions of synthetic and real images. Although used as an essential component of network training in this work, the proposed synthetic data generation technique is generic and can be used to produce large amount of synthetic human poses for deep learning in general.

Secondly, we propose Human Pose Models (HPM) that are Convolutional Neural Networks and transfer human poses to a shared high level invariant space. The HPMs are trained with the images that are refined with GAN and learn to map input (RGB or Depth) images to one of the representative human poses irrespective of the camera viewpoint clothing, human body size, background and lighting conditions. The layers prior to the Softmax label in the CNNs serve as high-level invariant human pose representations. Lastly, we propose to temporally model the invariant human pose features with the Fourier Temporal Pyramid and use SVM for classification. The proposed methods work together to achieve robust RGB-D human action recognition under the modeled variations.

Experiments on three benchmark human action datasets show that our method outperforms existing state-of-the-art for action recognition in conventional RGB videos as well as RGB-D videos obtained from depth sensors. For cross-view evaluation protocol, the proposed method improves RGB-D human action recognition accuracies by 15.4%, 11.9% and 8.4% on the UWA3D-II (Rahmani et al, 2016), NUCLA (Wang et al, 2014) and NTU (Shahroudy et al, 2016a) datasets respectively. For cross-subject evaluation protocol, the proposed method improves recognition accuracy by 6.0% for the NTU dataset. Our method also improves RGB-only human action recognition accuracy by 9% for the UWA3D-II dataset.

¹ <http://mocap.cs.cmu.edu>

This work is an extension of (Rahmani and Mian, 2016) where only depth image based human pose model and action recognition results were presented. However, depth images are almost always accompanied with RGB images. Therefore, we present the following extensions: (1) We present a method² for synthesizing realistic RGB human pose images containing variations in clothing textures, background, lighting conditions and more variations in body shapes. This method can also synthesize depth images more efficiently compared to (Rahmani and Mian, 2016). (2) We adopt Generative Adversarial Networks (GANs) to refine the synthetic RGB images and depth images so as to reduce their distribution gaps from real images and achieve improved accuracy. (3) We present a Human Pose Model HPM_{RGB} for human action recognition in conventional RGB videos which has wider applications. The proposed HPM_{RGB} achieves state-of-the-art human action recognition accuracy in conventional RGB videos. (4) We re-train the depth model HPM_{3D} using GoogleNet (Szegedy et al, 2015) architecture which performs similar to the AlexNet (Krizhevsky et al, 2012) model in (Rahmani and Mian, 2016) but with four times smaller feature dimensionality. (5) We perform additional experiments on the largest RGB-D human action dataset (NTU (Shahroudy et al, 2016a)) and report state-of-the-art results for action recognition in RGB videos and RGB-D videos in the cross-view and cross-subject settings. From here on, we refer to the Human Pose Models as HPM_{RGB} and HPM_{3D} for RGB and depth modalities respectively.

2 Related Work

2.1 Multi-view Action Recognition

Viewpoint variation is a major challenge in robust action recognition. Thus, the existing literature has seen multiple attempts of addressing multi-view action recognition. The closest work to our method is the key pose matching technique proposed by Lv and Nevatia (2007). In their approach, actions are modelled as series of synthetic 2D human poses rendered from many viewpoints and the transition between the synthetic poses is represented by an Action Net graph. However, the rendered images are not realistic as they do not model variations in clothing, background and lighting as in our case. Moreover, our method directly learns features from the rendered images rather than [hand-crafted](#) features.

Another closely related work to our method is the 3D exemplars for action recognition proposed by Wein-

land et al (2007). In their framework, actions are modelled with 3D occupancy grids built from multiple viewpoints. The learned 3D exemplars are then used to produce 2D images that are compared to the observations during recognition. This method essentially relies on silhouettes which may not be reliably extracted from the test videos especially under challenging background/lighting conditions.

Li et al (2012) proposed Hankelet which is a viewpoint invariant representation that captures the dynamic properties of short tracklets. Hankelets do not carry any spatial information and their viewpoint invariant properties are limited. Very early attempts for view invariant human action recognition include the following methods. Yilmaz and Shah (2005) proposed action sketch, an action representation that is a sequence of the 2D contours of an action in the x, y, t space-time. Such a representation is not completely viewpoint invariant. Parameswaran and Chellappa (2006) used 2D projections of 3D human motion capture data as well on manually segmented real image sequences to perform viewpoint robust action recognition. Rao et al (2002) used the spatio-temporal 2D trajectory curvatures as a compact representation for view-invariant action recognition. However, the same action can result in very different 2D trajectories when observed from different viewpoints. Weinland et al (2006) proposed Motion History Volumes (MHV) as a viewpoint invariant representation for human actions. MHVs are aligned and matched using Fourier Transform. This method requires multiple calibrated and background-subtracted video cameras which is only possible in controlled environments.

View knowledge transfer methods transfer features of different viewpoints to a space where they can be directly matched to achieve viewpoint invariant action recognition. Early methods in this category learned similar features between different viewpoints. For example, Farhadi and Tabrizi (2008) represented actions with histograms of silhouettes and optical flow and learned features with maximum margin clustering that are similar in different views. Source views are then transferred to the target view before matching. Given sufficient multiview training instances, it was shown later that a hash code with shared values can be learned (Farhadi et al, 2009). Gopalan et al (2011) used domain adaptation for view transfer. Liu et al (2011) used a bipartite graph to model two view-dependent vocabularies and applied bipartite graph partitioning to co-cluster two vocabularies into visual-word clusters called bilingual-words that bridge the semantic gap across view-dependent vocabularies. More recently, Li and Zickler (2012) proposed the idea of virtual views

² The code for this method will be made public.

that connect action descriptors from one view to those extracted from another view. Virtual views are learned through linear transformations of the action descriptors. Zhang et al (2013) proposed the idea of continuous virtual path that connects actions from two different views. Points on the virtual path are virtual views obtained by linear transformations of the action descriptors. They proposed a virtual view kernel to compute similarity between two infinite-dimensional features that are concatenations of the virtual view descriptors leading to kernelized classifiers. Zheng and Jiang (2013) learned a view-invariant sparse representation for cross-view action recognition. Rahmani and Mian (2015) proposed a non-linear knowledge transfer model that mapped dense trajectory action descriptors to canonical views. However, this method does not exploit the appearance/shape features. Kerola et al (2017) used spatio-temporal key points (STKP) and skeletons to represent an action as a temporal sequence of graphs, and then applied the spectral graph wavelet transform to create the action descriptors for cross-view action recognition.

2.2 Deep Learning in Action Recognition

Deep learning has been widely adopted for robust action recognition, either in the form of end-to-end trainable frameworks, or hybridized with conventional approaches. Simonyan and Zisserman (2014) and Feichtenhofer et al (2016) proposed two stream CNN architectures using appearance and optical flow to perform action recognition. Wang et al (2015) proposed a two stream structure to combine hand crafting features and deep learned features. They used trajectory pooling for one stream and deep learning for the second and combined the features from the two streams to form trajectory-pooled deep-convolutional descriptors. Nevertheless, the method did not explicitly address viewpoint variations. Pfister et al (2015) proposed a CNN architecture to estimate human poses. Their architecture directly regresses pose heat maps and combines them with optical flow. This architecture relies on neighbouring frames for pose estimations. Ji et al (2013) proposed a 3D Convolutional Neural Network (3D-CNN) for human action recognition. They used a set of hard-wired kernels to generate multiple information channels corresponding to the gray pixel values, (x, y) gradients, and (x, y) optical flow from seven input frames. This was followed by three convolution layers whose parameters were learned through back propagation. The 3D-CNN model did not explicitly address invariance to viewpoint or other factors.

Wang et al (2016b) proposed joint trajectory maps, projections of 3D skeleton sequences to multiple 2D

images, for human action recognition. Karpathy et al (2014) suggested a multi-resolution foveated architecture for speeding up CNN training for action recognition in large scale videos. Varol et al (2017a) proposed long-term temporal convolutions (LTC) and showed that LTC-CNN models with increased temporal extents improve action recognition accuracy. Tran et al (2015) treated videos as cubes and performed convolutions and pooling with 3D kernels. Recent methods (Li et al, 2016; Zhu et al, 2016; Wang and Hoai, 2016; Zhang et al, 2016; Su et al, 2016; Wang et al, 2016a) emphasize on action recognition in large scale videos where the background context is also taken into account.

Shahroudy et al (2016b) divided the actions into body parts and proposed a multimodal-multipart learning method to represent their dynamics and appearances. They selected the discriminative body parts by integrating a part selection process into the learning and proposed a hierarchical mixed norm to apply sparsity between the parts, for group feature selection. This method is based on depth and skeleton data and uses LOP (local occupancy patterns) and HON4D (histogram of oriented 4D normals) as features. Yu et al (2016) proposed a Structure Preserving Projection (SPP) to represent RGB-D video data fusion. They described the gradient fields of RGB and depth data with a new Local Flux Feature (LFF), and then fused the LFFs from RGB and depth channels. With structure-preserving projection, the pairwise structure and bipartite graph structure are preserved when fusing RGB and depth information into a Hamming space, which benefits the general action recognition.

Huang et al (2016) incorporated the Lie group structure into deep learning, to transform high-dimensional Lie group trajectory into temporally aligned Lie group features for skeleton-based action recognition. The incorporated learning structure generalizes the traditional neural network model to non-Euclidean Lie groups. Luo et al (2017) proposed to use Recurrent Neural Network based Encoder-Decoder framework to learn video representation in capturing motion dependencies. The learning process is unsupervised and it focuses on encoding the sequence of atomic 3D flows in consecutive frames.

Jia et al (2014a) proposed a latent tensor transfer learning method to transfer knowledge from the source RGB-D dataset to the target RGB only dataset such that the missing depth information in the target dataset can be compensated. The learned 3D geometric information is then coupled with RGB data in a cross-modality regularization framework to align them. However, to learn the latent depth information for RGB data, a RGB-D source dataset is required to perform the transfer learning, and for different source

datasets, the learned information may not be consistent which could affect the final performance. Kong and Fu (2017) proposed max-margin heterogeneous information machine (MMHIM) to fuse RGB and depth features. The histograms of oriented gradients (HOG) and histograms of optical flow (HOF) descriptors are projected into independent shared and private feature spaces, and the features are represented in matrix forms to build a low-rank bilinear model for the classification. This method utilizes the cross-modality and private information, which are also de-noised before the final classification.

2.3 Synthetic Data Generation

The impracticality of careful data annotation at large scales has led researchers to also seek efficient ways of synthesizing large datasets for robust action recognition. Varol et al (2017b) recently proposed SURREAL to synthesize human images for the task of body segmentation and depth estimation. The generated dataset includes RGB images, together with depth maps and body parts segmentation information. They then learned a CNN model from the synthetic dataset, and then conduct pixel-wise classification for the real RGB pose images. This method made efforts in diversifying the synthetic data, however, it didn't address the distribution gap between synthetic and real images. Moreover, this method only performs human body segmentation and depth estimation.

Our survey shows that none of the existing techniques explicitly learn invariant features through a training dataset that varies all irrelevant factors for the same human pose. This is partly because such training data is very difficult and expensive to generate. We resolve this problem by developing a method to generate such data synthetically. The proposed data generation technique is a major contribution of this work that can synthesize large amount of data for training data-hungry deep network models. By easily introducing a variety of action irrelevant variations in the synthetic data, it is possible to learn effective models that can extract invariant information. In this work, we propose Human Pose Models for extracting such information from both RGB and depth images.

3 Generating Synthetic Training Data

The proposed synthetic data generation steps are explained in the following subsections. Unless specified, the steps are shared by RGB and depth data generation.

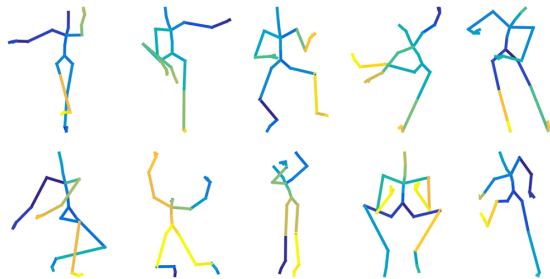


Fig. 2 Samples from the 339 representative human poses learned from the CMU MoCap skeleton data

3.1 Learning Representative Human Poses

Since the space of possible human poses is extremely large, we learn a finite number of representative human poses in a way that is not biased by irrelevant factors such as body shapes, appearances, camera viewpoints, illumination and backgrounds. Therefore, we learn the representative poses from 3D human joints (skeletons), because each skeleton can be fitted with 3D human models of any size/shape. The CMU MoCap database is ideal for this purpose because it contains joint locations of real humans performing different actions resulting in a large number of different poses. This data consists of over 2500 motion sequences and over 200,000 human poses. We randomly sample 50,000 frames as the pose candidates and cluster them with HDBSCAN algorithm (McInnes et al, 2017) using the skeletal distance function (Shakhnarovich, 2005)

$$D(\theta_1, \theta_2) = \max_{i \leq j \leq L} \sum_{d \in \{x, y, z\}} |\theta_{d,1}^i - \theta_{d,2}^j| \quad (1)$$

where θ_1 and θ_2 are the x, y, z joint locations of two skeletons, and L denotes the total number of joints in CMU MoCap skeleton. By setting the minimum cluster size to 20, the HDBSCAN algorithm outputs 339 clusters and we choose the pose with highest HDBSCAN score in each cluster to form the representative human poses. Figure 2 shows a few of the learned representative human poses.

3.2 Generating 3D Human Models

The 339 representative human pose skeletons are fitted with full 3D human models. We use the open source MakeHuman³ software to generate 3D human models because it has three attractive properties. Firstly, the 3D human models created by MakeHuman contain information for fitting the model to the MoCap skeletons

³ <http://www.makehuman.org>



Fig. 3 3D human (male, heavy male, female, child) models generated with the MakeHuman software. 3D mesh and sample clothing by the MakeHuman software are also shown for each model.

to adopt that pose. Secondly, it is possible to vary the body shape, proportion and gender properties to model different shape humans. Figure 3 shows the four human body shapes we used in our experiments. Thirdly, MakeHuman allows for selecting some common clothing types as shown in Figure 3. Although, the MakeHuman offers limited textures for the clothing, we write a Python script to apply many different types of clothing textures obtained from the Internet.

3.3 Fitting 3D Human Models to MoCap Skeletons

MakeHuman generates 3D human models in the same canonical pose as shown in Figure 3. We use the open source Blender⁴ software to fit the 3D human models to the 339 representative human pose skeletons. Blender loads the 3D human model and re-targets its rigs to the selected MoCap skeleton. As a result, the 3D human model adopts the pose of the skeleton and we get the representative human poses as full 3D human models with different clothing types and body shapes. Clothing textures are varied later.

3.4 Multiview Rendering to Generate RGB Images

We place each 3D human model (with a representative pose) in different backgrounds and lighting conditions using Blender. In the following, we explain how different types of variations were introduced in the rendered images.

Camera Viewpoint: We place 180 virtual cameras on a hemisphere over the 3D human model to render RGB images. These cameras are 12 degrees apart along the latitude and longitude and each camera points to the center of the 3D human model. Figure 4 illustrates the virtual cameras positioned around a 3D human model where no background has been added yet to make the cameras obvious. Figure 5 shows a few images rendered

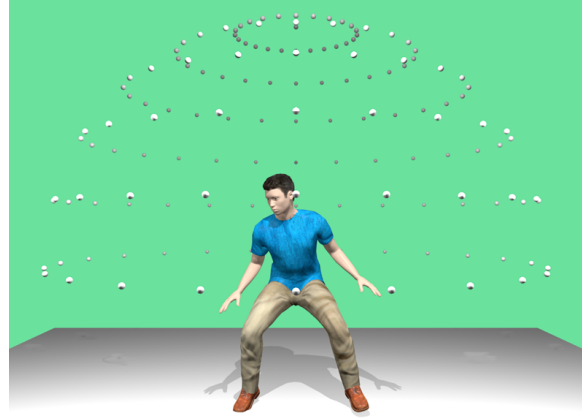


Fig. 4 Each 3D human model is rendered from 180 camera viewpoints on a hemisphere. All cameras point to the center of the human model



Fig. 5 Human pose images rendered from multiple viewpoints

from multiple viewpoints after adding the background and lighting.

Background and Lighting: We incorporate additional rich appearance variations in the background and lighting conditions to synthesize images that are as realistic as possible. Background variation is performed in two modes. One, we download thousands of 2D indoor scenes from Google Images and randomly select one as Blender background during image rendering. Two, we download 360° spherical High Dynamic Range Images (HDRI) from Google Images and use them as the environmental background in Blender. In the latter case, when rendering images from different viewpoints, the background changes accordingly. Figure 5 shows some illustrations. It is also possible to incorporate 3D background layout in our method using Blender. Interaction between 3D human pose and surroundings can result in further improving the images. However, it requires the laborious task of 3D modelling of all items in a given scene, which also results in computational overhead. From the perspective of data augmentation for learning deep models, the proposed pipeline considers both the image quality and data generation efficiency. Our method is able to *efficiently* incorporate large variations of multiple factors in the generated data, includ-

⁴ <http://www.blender.org>

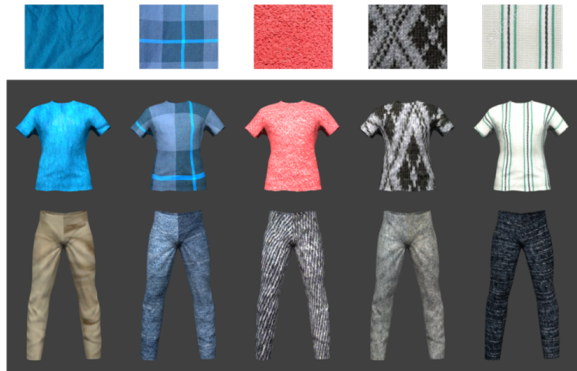


Fig. 6 We use 262 textures for shirts and 183 for the trousers/shorts. The clothing textures of the human models are varied randomly through a Python script in Blender

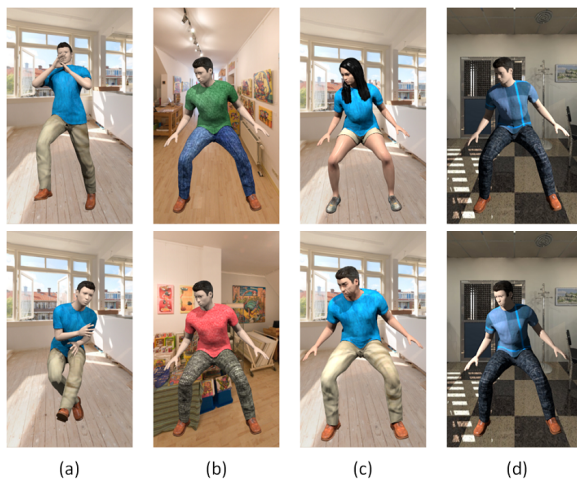


Fig. 7 Synthetic RGB images where variations are introduced in (a) human pose, (b) background and clothing texture, (c) body shapes and (d) lighting conditions

ing poses, viewpoints, backgrounds, textures and illuminations. In total, we use 2000 different backgrounds that are mostly indoor scenes, building lobbies with natural lighting and a few outdoor natural scenes. We place three lamps at different locations in the scene and randomly change their energy to achieve lighting variations.

Clothing Texture: Clothing texture is varied by assigning different textures, downloaded from Google Images, to the clothing of the 3D human models. In total, we used 262 different textures of shirts and 183 textures for trousers/shorts to generate our training data. Figure 6 shows some of the clothing textures we used and Figure 7 shows some rendered images containing all types of variations.

3.5 Multiview Rendering to Generate Depth Images

Depth images simply record the distance of the human from the camera without any background, texture or lighting variation. Therefore, we only vary the camera viewpoint, clothing types (not textures) and body shapes when synthesizing depth images. The virtual cameras are deployed in a similar way to the RGB image rendering. Figure 8 shows some depth images rendered from different viewpoints. In the Blender rendering environment, the bounding box of the human model is recorded as a group of vertices, which can be converted to an xy bounding box around the human in the rendered image coordinates. This bounding box is used to crop the human in the rendered depth images as well as RGB images. The cropped images are used to learn the Human Pose Models.

3.6 Efficiency of Synthetic Data Generation

To automate the data generation, we implemented a Python script⁵ in Blender on a 3.4GHz machine with 32GB RAM. The script runs six separate processing threads and on the average, generates six synthetic pose images per second. Note that this is an off-line process and can be further parallelized since each image is rendered independently. Moreover, the script can be implemented on the cloud for efficiency without the need to upload any training data. The training data is only required for learning the model and can be deleted afterwards. For each of the 339 representative poses, we generate images from all 180 viewpoints while applying a random set of other variations (clothing, background, body shape and lighting). In total, about 700,000 synthetic RGB and depth images are generated to train the proposed HPM_{RGB} and HPM_{3D} .

4 Synthetic Data Refinement with GANs

The synthetic images are labelled with 339 different human poses and cover common variations that occur in real data. They can be used to learn Human Pose Models that transfer human poses to a high level view-invariant space. However, it is likely that the synthetic images are sampled from a distribution which is different from the distribution of real images. Given this distribution gap, the Human Pose Models learned from synthetic images may not generalize well to real images. Therefore, before learning the models, we minimize the gap between the distributions. For this purpose, we

⁵ The data synthesis script will be made public.

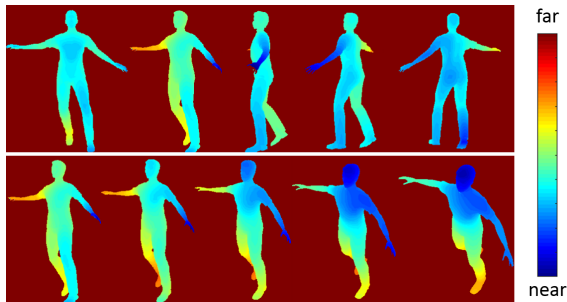


Fig. 8 Depth human pose images rendered from multiple viewpoints. The first row of the figure illustrates horizontal viewpoint change, and the second row illustrates vertical viewpoint change. Color coding is used for depth image in this figure for better visibility.

adopt the simulated and unsupervised learning framework (SimGAN) (Shrivastava et al, 2016) proposed by Shrivastava *et al.*. This framework uses an adversarial network structure similar to the Generative Adversarial Network (GAN) (Goodfellow et al, 2014), but the learning process is based on synthetic images, instead of random noises as in the original GAN method. The SimGAN framework learns two competing networks, refiner $R_\theta(x)$ and discriminator $D_\phi(\tilde{x}, y)$, where x is synthetic image, y is unlabelled real image, $\tilde{x} = R_\theta(x)$ is refined image. In SimGAN implementation, the refiner $R_\theta(x)$ is a fully convolutional neural network without striding or pooling, and the discriminator $D_\phi(\tilde{x}, y)$ is a ConvNet whose last layer outputs the probability of the sample being a refined image. The loss function of these two networks are defined as $\mathcal{L}_R(\theta)$ and $\mathcal{L}_D(\phi)$ (Shrivastava et al, 2016)

$$\mathcal{L}_R(\theta) = -\sum \log(1 - D_\phi(R_\theta(x_i))) + \lambda \|R_\theta(x_i) - x_i\|_1, \quad (2)$$

$$\mathcal{L}_D(\phi) = -\sum_i \log(D_\phi(\tilde{x}_i)) - \sum_j \log(1 - D_\phi(y_j)), \quad (3)$$

where x_i is the i^{th} synthetic image, \tilde{x}_i is its corresponding refined image, y_j is the j^{th} real image, $\|\cdot\|_1$ is ℓ_1 norm, and $\lambda \in [0, 1]$ is the regularization factor.

5 Learning the Human Pose Models

Every image in our synthetic data has a label corresponding to one of the 339 representative human poses. For a given human pose, the label remains the same irrespective of the camera viewpoint, clothing texture, body shape, background and lighting conditions. We learn CNN models that map the rendered images to their respective human pose labels. We learn HPM_{RGB} and HPM_{3D} for RGB and depth images independently and test three popular CNN architectures, i.e. AlexNet

(Krizhevsky et al, 2012), GoogLeNet (Szegedy et al, 2015), and ResNet-50 (He et al, 2016a), to find the most optimal architecture through controlled experiments. These CNN architectures performed well in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, 2014, and 2015 respectively, and come with increasing number of layers. We fine tune the ILSVRC pre-trained models using our synthetic data and compare their performance on human action recognition.

5.1 Extracting Features from Real Videos

To extract features from real videos, the region containing the human is first cropped from each frame and then the cropped region is resized to match the input of the network. The cropped-resized regions from each frame are passed individually through the learned Human Pose Model (HPM_{RGB} for RGB frames, and HPM_{3D} for depth frames), and a layer prior to the labels is used as invariant representation of the human pose. Specifically, we use fc7 layer for AlexNet, pool5/7x7_s1 layer for GoogLeNet and pool5 for ResNet-50. This representation is unique, compact, invariant to the irrelevant factors and has the added advantage that it aligns the features between multiple images. While the pixels of one image may not correspond to the pixels of another image, the individual variables of the CNN features are aligned. Therefore, we can perform temporal analysis along the individual variables.

6 Temporal Representation and Classification

For temporal representation, we use the Fourier Temporal Pyramid (FTP) (Wang et al, 2013b) on the features extracted from the video frames. Temporal representation for HPM_{3D} and HPM_{RGB} features is done in a similar way and explained in general in the next paragraph.

Let V_t^i denote the t -th frame of i -th video, $t = 1, 2, \dots, f$ where f is the total number of frames. Take HPM with GoogleNet structure as an example, denote the pool5/7x7_s1 layer activations of frame V_t^i as $A_t^i \in \mathbb{R}^{1024 \times 1}$ and the frame-wise pose features of the i -th video as $A^i = [A_1^i, A_2^i, \dots, A_f^i]^T$. FTP is applied on A^i for temporal encoding using a pyramid of three levels where A^i is divided in half at each level giving $1+2+4 = 7$ feature groups. Short Fourier Transform is applied to each feature group, and the first four low-frequency coefficients (i.e. $4 \times 7 = 28$) are used to form a spatio-temporal action descriptor $B^i \in \mathbb{R}^{1024 \times 28}$. Finally, B^i is stretched to $D^i \in \mathbb{R}^{1 \times 28672}$ to get the final spatio-temporal representation of the i -th video. When the

dimension of frame-wise pose feature changes, the dimension of spatio-temporal descriptor changes accordingly, for example, $B^i \in \mathbb{R}^{4096 \times 28}$ for AlexNet, and $B^i \in \mathbb{R}^{2048 \times 28}$ for ResNet-50.

Note that the FTP encodes the temporal variations of the RGB action videos in the HPM_{RGB} feature space. The video frames are first aligned in the A_t^i HPM_{RGB} feature space which makes it possible to preserve the spatial location of the features while temporal encoding with FTP. On the other hand, dense trajectories model temporal variations in the pixel space (of RGB videos) where pixels corresponding to the human body pose are not aligned. This is the main reason why dense trajectory features are encoded with Bag of Visual Words (BoVW) which facilitates direct matching of dense trajectory features from two videos. However, this process discards the *spatial* locations of the trajectories. Thus, similar trajectories from different locations in the frame will vote to the same bin in BoVW feature.

An advantage of performing temporal encoding in different feature spaces is that the features are non-redundant. Thus our HPM_{RGB} and dense trajectories capture complementary information. Although dense trajectories cannot capture the appearance information, they are somewhat robust to viewpoint changes as shown in (Rahmani et al, 2017). Therefore, we augment our HPM_{RGB} features with dense trajectory features before performing classification. We use the improved dense trajectories (iDT) (Wang and Schmid, 2013) implementation which provides additional features such as HOG (Dalal and Triggs, 2005), HOF and MBH (Dalal et al, 2006). However, we only use the trajectory part and discard HOG/HOF/MBH features for two reasons. Firstly, unlike our HPMS, HOG/HOF/MBH features are not view-invariant. Secondly, our HPMS already encode the appearance information. We use the NKTM (Rahmani and Mian, 2015) codebook to encode the trajectory features and denote the encoded BoVW as $D^i \in \mathbb{R}^{2000}$ for video V^i .

We use SVM (Fan et al, 2008) for classification and report results in three settings i.e. RGB, depth and RGB-D. In the RGB setting, we represent the HPM_{RGB} features temporally encoded with FTP and then combine them with the trajectory BoVW features since both types of features can be extracted from RGB videos. In the depth setting, we represent the $\text{HPM}_{3\text{D}}$ features with FTP but do not combine trajectory features because trajectories cannot be reliably extracted from the depth videos. In the RGB-D setting, we combine the FTP features from both HPM models with the trajectory BoVW features.

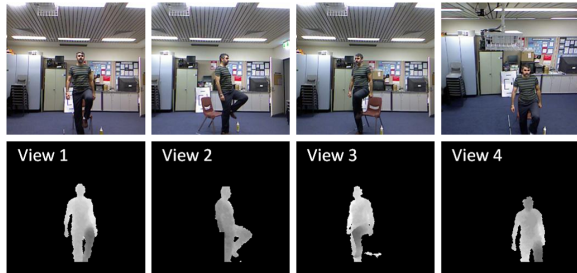


Fig. 9 Sample frames from the UWA3D Multiview Activity II dataset (Rahmani et al, 2016)

7 Implementation

7.1 GAN Refinement Implementation

For synthetic data refinement with GAN, we modify the Tensorflow implementation⁶ of SimGAN to make it suitable for our synthetic RGB images. For the refiner network, we extend the input data channel from 1 to 3. The input images are first convolved with 7×7 filters and then converted into 64 feature maps. The 64-channel feature maps are passed through multiple ResNet blocks. The setting of ResNet blocks and the structure of discriminator network are the same as (Shrivastava et al, 2016).

To get benchmark distribution for the synthetic images, we randomly select 100,000 unlabelled real images from the NTU RGB+D Human Activity Dataset (Shahroudy et al, 2016a). Each image is cropped to get the human body as the region of interest and then resized to 224×224 . Through adversarial learning, SimGAN framework (Shrivastava et al, 2016) will force the distribution of synthetic images to approach this benchmark distribution. Although, we use samples from the NTU RGB+D Human Activity Dataset as benchmark to train the SimGAN network, this is not mandatory as any other dataset containing real human images can be used. This is because the SimGAN learning is an unsupervised process, which means no action labels are required. Our experiments in later sections also illustrate that the performance improvement gained from GAN-refinement has no dependence on the type of real images used for SimGAN learning.

7.2 HPM Training Implementation

We implement HPM_{RGB} and $\text{HPM}_{3\text{D}}$ training with three popular CNN architectures, i.e. AlexNet (Krizhevsky et al, 2012), GoogLeNet (Szegedy et al, 2015), and ResNet-50 (He et al, 2016a). The three pre-trained models have a last *fully connected* layer with 1000 neurons.

⁶ <https://github.com/carpedm20/>

For fine tuning, we replace the last layer with a 339 neuron layer representing the number of classes in our synthetic human pose dataset. All synthetic images are cropped to include only the human body and then resized to 256×256 pixels. During training, these images are re-cropped to the required input dimension for the specific network with default data augmentation, and are also mirrored with a probability of 0.5. We use the synthetic pose images from 162 randomly selected camera viewpoints for training, and the images from the remaining 18 cameras for validation.

The Caffe library (Jia et al, 2014b) is used to learn the proposed HPM_{RGB} and HPM_{3D} models. The initial learning rate of the model was set to 0.01 for the last fully-connected layers and 0.001 for all other layers. We used a batch size of 100 and trained the model for 3 epochs. We decreased the learning rate by a factor of 10 after every epoch. Training was done using a single NVIDIA Tesla K-40 GPU.

8 Experiments

8.1 Datasets

Experiments are performed in the following three benchmark datasets for cross-view human action recognition.

8.1.1 UWA3D Multiview Activity-II Dataset

Figure 9 shows sample frames from this dataset. The dataset (Rahmani et al, 2016) consists of 30 human actions performed by 10 subjects and recorded from 4 different viewpoints at different times using the Kinect v1 sensor. The 30 actions are: (1) one hand waving, (2) one hand punching, (3) two hands waving, (4) two hands 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. The four viewpoints are: (a) front, (b) left, (c) right, (d) top.

This dataset is challenging because of the large number of action classes and because the actions are not recorded simultaneously leading to intra-action differences besides viewpoint variations. The dataset also contains self-occlusions and human-object interactions in some videos.

We follow the protocol of (Rahmani et al, 2016) where videos from two views are used for training and the videos from the remaining views are individually

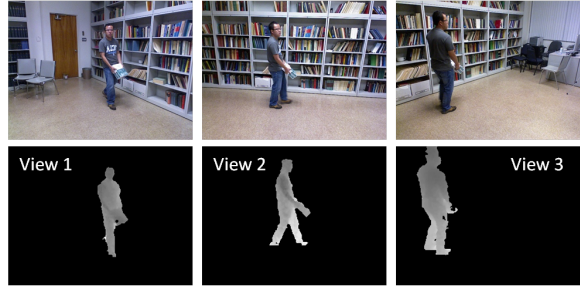


Fig. 10 Sample frames from the Northwestern-UCLA Multiview Action dataset (Wang et al, 2014)

used for testing leading to 12 different cross-view combinations in this evaluation protocol.

8.1.2 Northwestern-UCLA Multiview Dataset

This dataset (Wang et al, 2014) contains RGB-D videos captured simultaneously from three different viewpoints with the Kinect v1 sensor. Figure 10 shows sample frames of this dataset from the three viewpoints. The dataset contains RGB-D videos of 10 subjects performing 10 actions: (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. The three viewpoints are: (a) left, (b) front, and (c) right. This dataset is very challenging because many actions share the same “walking” pattern before and after the actual action is performed. Moreover, some actions such as “pick up with one hand” and “pick up with two hands” are hard to distinguish from different viewpoints.

We use videos captured from two views for training and the third view for testing making three possible cross-view combinations.

8.1.3 NTU RGB+D Human Activity Dataset

The NTU RGB+D Human Activity Dataset (Shahroudy et al, 2016a) is a large-scale RGB+D dataset for human activity analysis. This dataset was collected with the Kinect v2 sensor and includes 56,880 action samples each for RGB videos, depth videos, skeleton sequences and infra-red videos. We only use the RGB and depth parts of the dataset. There are 40 human subjects performing 60 types of actions including 50 single person actions and 10 two-person interactions. Three sensors were used to capture data simultaneously from three horizontal angles: -45° , 0° , 45° , and every action performer performed the action twice, facing the left or right sensor respectively. Moreover, the height of sensors and their distance to the action performer were

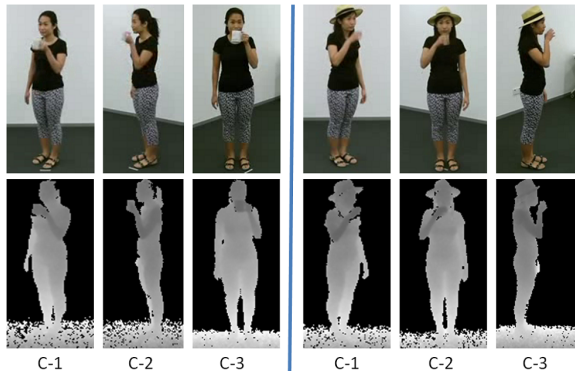


Fig. 11 RGB and depth sample frames from the NTU RGB+D Human Activity Dataset (Shahroudy et al, 2016a). Three sensors C-1, C-2 and C-3 are used to record this dataset. The left group of images in this figure shows the actions recorded with the performer facing the sensor C-3, and the right group of images are recorded when the action performer faces the sensor C-2

further adjusted to get more viewpoint variations. The NTU RGB+D dataset is the largest and most complex cross-view action dataset of its kind to date. Figure 11 shows RGB and depth sample frames in NTU RGB+D dataset.

We follow the standard evaluation protocol proposed in (Shahroudy et al, 2016a), which includes cross-subject and cross-view evaluations. For cross-subject protocol, 40 subjects are split into training and testing groups, and each group consists of 20 subjects. For cross-view protocol, the videos captured by sensor C-2 and C-3 are used as training samples, and the videos captured by sensor C-1 are used as testing samples.

8.2 Ablation Study

We first use raw synthetic images to train HPM_{RGB} and HPM_{3D} for the three different CNN architectures (AlexNet, GoogLeNet, and ResNet-50), and compare their performance on the UWA and NUCLA datasets. The best performing architecture is then selected and re-trained on the GAN-refined synthetic images. Next, we compare the HPM models trained on raw synthetic images to those trained on the GAN refined synthetic images. Finally, we perform comprehensive experiments to compare our proposed models trained on GAN refined synthetic images to existing methods on all three datasets.

8.2.1 HPM Performance with Different Architectures

We determine the best CNN architecture that maximizes generalization power of HPM_{RGB} and HPM_{3D} for RGB and depth images respectively. We use the raw

Network	Layer	Dimension	HPM_{RGB}	HPM_{3D}
UWA3D Multiview Activity-II				
AlexNet	fc7	4096	61.2	72.1
ResNet-50	pool5	2048	65.4	74.0
GoogLeNet	pool5	1024	64.7	74.1
Northwestern-UCLA Multiview				
AlexNet	fc7	4096	69.9	78.7
ResNet-50	pool5	2048	75.7	77.3
GoogLeNet	pool5	1024	76.4	79.8

Table 1 Comparison of feature dimensionality and action recognition accuracy(%) for HPM_{RGB} and HPM_{3D} trained using different network architectures

synthetic pose images to fine tune AlexNet, GoogLeNet, and ResNet-50, and then test them on the UWA and NUCLA datasets. Since the trained model is to be used as a frame-wise feature extractor for action recognition, we take recognition accuracy and feature dimensionality both into account. Table 1 compares the average results on all possible cross-view combinations for the two datasets. The results show that for RGB videos, GoogLeNet and ResNet-50 perform much better than AlexNet. GoogLeNet also performs the best for depth videos and has the smallest feature dimensionality. Therefore, we select GoogLeNet as the network architecture for both HPM_{RGB} and HPM_{3D} in the remaining experiments.

8.2.2 Qualitative Analysis of GAN Refined Images

We compare the real images, raw synthetic images, and GAN-refined synthetic images, to analyse the effect of GAN refinement on our synthetic RGB and depth human pose datasets.

Figure 12 compares the real, raw synthetic and GAN-refined synthetic RGB images. One obvious difference between real and synthetic RGB images is that the synthetic ones are sharper and have more detail than the real images. This is because the synthetic images are generated under ideal conditions and the absence of realistic image noises makes them different from real images. However, with GAN learning, the refined RGB images lose some of the details (i.e. they are not as sharp) and become more realistic. The last row of Figure 12 shows the difference between the raw and refined synthetic images. Notice that the major differences (bright pixels) are at the locations of the humans and especially at their boundaries whereas the backgrounds have minimal differences (dark pixels). The reason for this is that the synthetic images are created using synthetic 3D human models but real background images i.e. rather than building Blender scene models (walls, floors, furnitures, etc.) from scratch, we used real background



Fig. 12 Comparing real, raw synthetic and GAN-refined synthetic RGB images. Last row shows the difference between raw and GAN-refined synthetic images. Since the backgrounds used are already real, differences are mostly on the synthetic human body especially at their edges

images for efficiency and diversity of data. Moreover, the Blender lighting function causes shading variation on the human models only, and the shading effects of backgrounds always remain the same. All these factors make the human model stand out of the background. On the other hand, the human subjects are perfectly blended with the background in the real images. The GAN refinement removes such differences in synthetic images and makes the human models blend into the background. Especially, the bright human boundaries (last row) shows that the GAN refinement process is able to sense and remove the difference between human model and background images.

Figure 13 shows a similar comparison for depth images. The most obvious difference between real and synthetic depth images is the noise along the boundary. The edges in the real depth images are not smooth whereas they are very smooth in the synthetic depth images. Other differences are not so obvious to the naked eye but nevertheless, these differences might limit the generalization ability of the Human Pose Model learned from synthetic images. The third row of Fig. 13 shows the refined synthetic depth images and the last column shows the difference between the images in more detail. Thus the GAN refinement successfully learns to model boundary noise and other non-obvious differences of real images and applies them to the synthetic depth maps narrowing down their distribution gap.

In next section, we will show quantitative results indicating that the Human Pose Models learned from the GAN refined synthetic images outperform those that are learned from raw synthetic images.

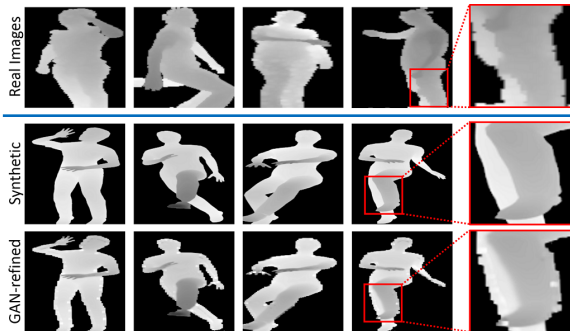


Fig. 13 Comparing real, raw synthetic and GAN-refined synthetic depth images

Training Data	HPM _{RGB}	HPM _{3D}
UWA3D Multiview Activity-II		
Raw synthetic images	64.7	73.8
GAN-refined synthetic images	68.0	74.8
Northwestern-UCLA Multiview		
Raw synthetic images	76.4	78.4
GAN-refined synthetic images	77.8	79.7

Table 2 Comparison of action recognition accuracy (%) for GoogLeNet based HPM_{RGB} and HPM_{3D} when trained on raw synthetic images and GAN-refined synthetic images

8.2.3 Quantitative Analysis of GAN Refinement

We quantitatively compare the effect of GAN-refinement using the UWA and NUCLA datasets by comparing the action recognition accuracies when the HPM_{RGB} and HPM_{3D} are fine tuned once on raw synthetic images and once on GAN-refined synthetic images.

Table 2 shows the average accuracies for all cross-view combinations on the respective datasets. We can see that the HPM_{RGB} fine tuned on the GAN-refined synthetic RGB images achieves 3.3% and 1.4% improvement over the one fine tuned with raw synthetic images on the UWA and NUCLA datasets respectively. For HPM_{3D}, GAN-refined synthetic data also improves the recognition accuracy for the two datasets by 1% and 1.3% respectively. The improvements are achieved because the distribution gap between synthetic and real images is narrowed by GAN refinement.

Recall that the real images used as a benchmark distribution for GAN refinement are neither from UWA nor NUCLA dataset. We impose no dependence on the type of real images used for SimGAN learning, because it is an unsupervised process and no pose labels are required. In the remaining experiments, we use HPM_{RGB} and HPM_{3D} fine tuned with GAN refined synthetic images for comparison with other techniques.

Training Data	HPM _{RGB}	HPM _{3D}	HPM _{RGB+HPM_{3D}}
UWA3D Multiview Activity-II			
1st camera circle	65.5	74.4	75.5
3rd camera circle	64.9	72.3	74.3
5th camera circle	64.3	71.4	73.9
Full dataset	68.0	74.8	77.4
Northwestern-UCLA Multiview			
1st camera circle	75.1	77.6	78.5
3rd camera circle	73.0	76.8	78.3
5th camera circle	71.5	75.3	75.6
Full dataset	77.8	79.7	81.1

Table 3 Comparison of action recognition accuracy (%) for HPM_{RGB}, HPM_{3D} trained on different subset of synthetic images.

8.2.4 Advantage of Hemispherical Sampling

As shown in Figure 4, we sample 3D human models with 180 virtual cameras deployed in a hemisphere around the models. The hemispherical sampling provides comprehensive viewpoint variations to help HPMS in recognizing multiview actions. To analyze the effectiveness of hemispherical sampling, we also performed experiments by sub-sampling the hemisphere using three circles of fixed heights. Starting from the bottom, we selected the first, third, and fifth circle of cameras in Figure 4, and individually trained three HPMS using the resulting data. We compare performance of the resulting HPMS on RGB, Depth and RGB+D modalities in Table 3. The models are evaluated on UWA3D and NUCLA datasets, and compared with the original HPM that is trained using the full synthetic dataset. The results demonstrate that data from the lower circles result in higher model performance. However, data from the full hemisphere provides the best performance.

8.3 Evaluation and Comparison

8.3.1 Comparison on the UWA3D Multiview-II Dataset

Table 4 compares our method with existing state-of-the-art. The proposed HPM_{RGB} alone achieves 68.0% average recognition accuracy for RGB videos, which is higher than the nearest RGB-only competitor RNKTM (Rahmani et al, 2017), and for 8 out the 12 train-test combinations, our proposed HPM_{RGB} features provide significant improvement in accuracy. This shows that the invariant features learned by the proposed HPM_{RGB} are effective.

Combining HPM_{RGB} and dense trajectory features (Traj) gives a significant improvement in accuracy. It

improves the RGB recognition accuracy to 76.4%, which is 9% higher than the nearest RGB competitor. It is also higher than the depth only method HPM_{3D}. This shows that our method exploits the complementary information between the two modes of spatio-temporal representation, and enhances the recognition accuracy especially when there are large viewpoint variations. State-of-the-art RGB-D action recognition accuracy of 82.8% on UWA dataset is achieved when we combine HPM_{RGB}, HPM_{3D} and dense trajectory features.

Table 4 mentions the best results of the existing method taken directly from the literature, except for C3D, LRCN and JOULE. For C3D and LRCN (and C3D†, LRCN†), we fine-tuned the original models using the target dataset. For JOULE, we used the author recommended parameter values with the public code. Unless mentioned otherwise, the results of the existing methods in the remaining article also follow the same convention. Notice that, the deep learning based existing methods achieve their best performance after fine tuning with the target datasets. In contrast, HPMS are not fine tuned to any dataset once their training on the proposed synthetic data is completed. These models are used out-of-the-box for the test data. These settings hold for all the experiments conducted in this work. Indeed, fine tuning HPMS on datasets leads to further improvements in the results. For instance, HPM_{RGB} achieves 68.9% accuracy with fine tuning on UWA3D dataset. A 0.6% improvement was also observed for NUCLA dataset. However, the results reported in the Tables do not take advantage of fine tuning to emphasize the generalization power of HPMS trained on the proposed synthetic data.

Our data generation technique endows HPMS with inherent robustness to viewpoint variations along robustness to changes in background, texture and clothing etc. The baseline methods lack in these aspects which is one of the reasons for the improvement achieved by our approach over those methods. Note that, HPMS are unique in the sense that they model individual human poses in frames instead of actions. Therefore, our synthetic data generation method, which is an essential part of HPM training, also focuses on generating human pose frames. One interesting enhancement of our data generation technique is to produce synthetic videos instead. We can then analyze the performance gain of (video-based) baseline methods trained with our synthetic data. To explore this direction, we extended our technique to generate synthetic videos and applied it to C3D (Tran et al, 2015) and LRCN (Donahue et al, 2015) methods as follows.

For transparency, we selected all ‘atomic’ action sequences from CMU MoCap. Each of these sequences

Method	Data	$V_{1,2}^3$	$V_{1,2}^4$	$V_{1,3}^2$	$V_{1,3}^4$	$V_{1,4}^2$	$V_{1,4}^3$	$V_{2,3}^1$	$V_{2,3}^4$	$V_{2,4}^1$	$V_{2,4}^3$	$V_{3,4}^1$	$V_{3,4}^2$	Mean
Baseline														
DVV (Li and Zickler, 2012)	Depth	35.4	33.1	30.3	40.0	31.7	30.9	30.0	36.2	31.1	32.5	40.6	32.0	33.7
Action Tube (Gkioxari and Malik, 2015)	RGB	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
CVP (Zhang et al, 2013)	Depth	36.0	34.7	35.0	43.5	33.9	35.2	40.4	36.3	36.3	38.0	40.6	37.7	37.3
LRCN (Donahue et al, 2015)	RGB	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
AOG (Wang et al, 2014)	RGB	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
LRCN† (Wang et al, 2014)	RGB	55.2	31.5	50.0	30.7	33.5	39.2	52.8	31.5	55.4	47.8	56.1	42.5	43.8
Hankelets (Li et al, 2012)	RGB	46.0	51.5	50.2	59.8	41.9	48.1	66.6	51.3	61.3	38.4	57.8	48.9	51.8
2-stream (Simonyan and Zisserman, 2014)	RGB	63.0	47.1	55.8	60.6	53.4	54.2	66.0	50.9	65.3	55.5	68.0	51.9	57.6
JOULE (Hu et al, 2015)	RGB-D	54.3	71.6	43.5	68.9	52.7	58.8	72.4	58.3	78.7	37.1	80.6	45.0	60.2
DT (Wang et al, 2011)	RGB	57.1	59.9	54.1	60.6	61.2	60.8	71.0	59.5	68.4	51.1	69.5	51.5	60.4
C3D (Tran et al, 2015)	RGB	59.5	59.6	56.6	64.0	59.5	60.8	71.7	60.0	69.5	53.5	67.1	50.4	61.0
nCTE (Gupta et al, 2014)	RGB	55.6	60.6	56.7	62.5	61.9	60.4	69.9	56.1	70.3	54.9	71.7	54.1	61.2
C3D† (Tran et al, 2015)	RGB	62.7	57.3	59.2	68.0	63.2	64.6	71.0	54.7	68.8	52.6	74.3	62.8	63.3
NKTM (Rahmani and Mian, 2015)	RGB	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
R-NKTM (Rahmani et al, 2017)	RGB	64.9	67.7	61.2	68.4	64.9	70.1	73.6	66.5	73.6	60.8	75.5	61.2	67.4
Proposed														
HPM _{RGB}	RGB	72.4	73.4	64.3	71.9	50.8	62.3	69.9	61.8	75.5	69.4	78.4	66.2	68.0
HPM _{RGB} +Traj	RGB	81.0	78.3	72.9	76.8	67.7	75.7	79.9	67.0	85.1	77.2	85.5	69.9	76.4
HPM _{3D}	Depth	80.2	80.1	75.6	78.7	59.0	69.0	72.1	65.2	84.8	79.1	82.5	71.1	74.8
HPM _{RGB} +HPM _{3D}	RGB-D	79.9	83.9	76.3	84.6	61.3	71.3	77.0	68.9	85.1	78.7	87.0	74.8	77.4
HPM _{RGB} +HPM _{3D} +Traj	RGB-D	85.8	89.9	79.3	85.4	74.4	78.0	83.3	73.0	91.1	82.1	90.3	80.5	82.8

Table 4 Action recognition accuracy (%) on the UWA3D Multiview-II dataset. $V_{1,2}^3$ means that view 1 and 2 were used for training and view 3 alone was used for testing. The symbol † indicates that the original model was fine-tuned with our synthetic data before applying the testing protocol. **The mentioned results are taken directly from the literature, except for C3D, LRCN and JOULE. For C3D and LRCN (and C3D†, LRCN†), we fine-tuned the original models using the target datasets. For JOULE, we used the authors recommended parameter settings with public code that resulted in the best performance.**

presents a single action, which also serves as the label of the video clip. To generate synthetic videos, the frames in the training clips were processed according to the procedure described in Section 3 and 4 with the following major differences. (1) No clustering was performed to learn representative poses because it was not required. (2) The parameters (i.e. camera viewpoints, clothing etc.) were kept the same within a single synthetic video but different random settings were adopted for each video. The size of the generated synthetic data was matched to our “pose” synthetic data. We took the original C3D model (Tran et al, 2015) that is pre-trained on the large scale dataset Sports-1M (Karpathy et al, 2014) and the original LRCN model that is pre-trained on UCF-101 (Soomro et al, 2012) and fine-tuned these models using our synthetic videos. The fine-tuned models were then employed under the used protocol.

We report the results of these experiments in Table 4 by denoting our enhancements of C3D and LRCN as C3D† and LRCN†. The results demonstrate that our synthetic data can improve the performance of baseline models for multi-view action recognition. The results also ascertain that the proposed approach exploits the proposed data very effectively to achieve significant performance improvement over the existing meth-

ods. We note that C3D and LRCN enhancements discussed above do not strictly follow the same evaluation as HPM due to major differences between the training procedure requirements of these techniques. Nevertheless, the provided results clearly demonstrate that these techniques can also benefit from the proposed synthetic data. We provide further discussion on the role of synthetic data in the overall performance of our approach in Section 9.

8.3.2 Comparison on the Northwestern-UCLA Dataset

Table 5 shows comparative results on the NUCLA dataset. The proposed HPM_{RGB} alone achieves 77.8% average accuracy which is 8.4% higher than the nearest RGB competitor NKTM (Rahmani and Mian, 2015). HPM_{RGB}+Traj further improves the average accuracy to 78.5%. Our RGB-D method (HPM_{RGB}+HPM_{3D}+Traj) achieves 81.3% accuracy which is the highest accuracy reported on this dataset.

8.3.3 Comparison on the NTU RGB+D Dataset

Table 6 compares our method with existing state-of-the-art on the NTU dataset. The proposed HPM_{RGB} uses RGB frames only and achieves 68.5% cross-subject

Table 5 Action recognition accuracy (%) on the NUCLA Multiview dataset. $V_{1,2}^3$ means that view 1 and 2 were used for training and view 3 was used for testing. The symbol † indicates that the original model was fine-tuned with our synthetic data before applying the testing protocol.

Method	Data	$V_{1,2}^3$	$V_{1,3}^2$	$V_{2,3}^1$	Mean
Baseline					
Hankelets (Li et al, 2012)	RGB	45.2	-	-	-
LRCN (Donahue et al, 2015)	RGB	64.0	36.2	51.7	50.6
DVV (Li and Zickler, 2012)	Depth	58.5	55.2	39.3	51.0
LRCN†	RGB	62.6	39.6	53.3	51.8
CVP (Zhang et al, 2013)	Depth	60.6	55.8	39.5	52.0
C3D (Tran et al, 2015)	RGB	71.2	53.7	54.5	59.8
AOG (Wang et al, 2014)	Depth	73.3	-	-	-
JOULE (Hu et al, 2015)	RGB-D	76.5	53.1	53.5	61.0
C3D†	RGB	68.4	64.6	53.2	62.1
nCTE (Gupta et al, 2014)	RGB	68.6	68.3	52.1	63.0
NKTM (Rahmani and Mian, 2015)	RGB	75.8	73.3	59.1	69.4
R-NKTM (Rahmani et al, 2017)	RGB	78.1	-	-	-
Proposed					
HPM _{RGB}	RGB	91.5	69.0	73.1	77.8
HPM _{RGB} +Traj	RGB	89.3	75.2	71.0	78.5
HPM _{3D}	Depth	91.9	75.2	71.9	79.7
HPM _{RGB} +HPM _{3D}	RGB-D	92.4	74.1	76.8	81.1
HPM _{RGB} +HPM _{3D} +Traj	RGB-D	91.7	73.0	79.0	81.3

recognition accuracy, which is comparable to that of the best joints-based method ST-LSTM(Liu et al, 2016) 69.2% even though joints have been estimated from depth data and do not contain action irrelevant noises. This demonstrates that the HPM_{RGB} effectively learns features that are invariant to action irrelevant noises such as background, clothing texture and lighting etc.

Comparison of RGB Results:

Note that this paper is the first to provide RGB only human action recognition results on the challenging NTU dataset (see Table 6). Our method (HPM_{RGB}+Traj) outperforms all others by a significant margin while using only RGB data in both cross-subject and cross-view settings. In the cross-subject setting, our method achieves 75.8% accuracy which is higher than state-of-the-art DSSCA-SSLM (Shahroudy et al, 2017) even though DSSCA-SSLM uses both RGB and depth data whereas our method HPM_{RGB}+Traj uses only RGB data. DSSCA-SSLM does not report cross-view results as it did not perform well in that setting (Shahroudy et al, 2017) whereas our method achieves 83.2% accuracy for the cross-view case which is 7.7% higher than the nearest competitor ST-LSTM (Liu et al, 2016) which uses Joints data that is estimated from depth images. In summary, our 2.5D action recognition method outperforms existing 3D action recognition methods.

Comparison of RGB-D Results:

From Table 6, we can see that our RGB-D method (HPM_{RGB}+HPM_{3D}+Traj) achieves state-of-the-art results in both cross-subject and cross-view settings outperforming the nearest competitors by 6% and 8.4% respectively.

8.4 Timing

Table 7 shows the computation time for the major steps of our proposed method. Using a single core of a 3.4GHz CPU and the Tesla K-40 GPU, the proposed RGB (HPM_{RGB}+Traj) and RGB-D (HPM_{RGB}+HPM_{3D}+Traj) methods run at about 20 frames per second whereas the depth only method runs at about 46 frames per second.

9 Discussion

When uncropped video frames are used to learn a neural network model, the background context is more dominant as it occupies more pixels. A recent study showed that by masking the human in the UCF-101 dataset, a 47.4% “human” action recognition accuracy could still be achieved which, using the same algorithm, is only 9.5% lower than when the humans are included (He et al, 2016b). Our HPM_{RGB} learns human poses rather than the background context which is important for surveillance applications where the background is generally static and any action can be performed in the same background. Moreover, HPM_{RGB} and HPM_{3D} are not fine tuned on any of the datasets on which they are tested. Yet, our models outperform all existing methods by a significant margin. For applications such as robotics and video retrieval where the background context is important, our HPM models can be used to augment the background context. For optimal performance, the cropped human images must be passed through the HPM_{RGB} and HPM_{3D}. However, both HPMS are robust to cropping errors as many

Table 6 Action recognition accuracy (%) on the NTU RGB+D Human Activity Dataset. Our RGB only ($\text{HPM}_{\text{RGB}}+\text{Traj}$) accuracies are higher than the nearest competitors which use RGB-D or Joints data. Our RGB-D method ($\text{HPM}_{\text{RGB}}+\text{HPM}_{\text{3D}}+\text{Traj}$) outperforms all methods by significant margins in both settings

Method	Data type	Cross Subject	Cross View
Baseline			
HON4D (Oreifej and Liu, 2013)	Depth	30.6	7.3
SNV (Yang and Tian, 2014)	Depth	31.8	13.6
HOG-2 (Ohn-Bar and Trivedi, 2013)	Depth	32.4	22.3
Skeletal Quads (Evangelidis et al, 2014)	Joints	38.6	41.4
Lie Group (Vemulapalli et al, 2014)	Joints	50.1	52.8
Deep RNN (Shahroudy et al, 2016a)	Joints	56.3	64.1
HBRNN-L (Du et al, 2015)	Joints	59.1	64.0
Dynamic Skeletons (Hu et al, 2015)	Joints	60.2	65.2
Deep LSTM (Shahroudy et al, 2016a)	Joints	60.7	67.3
LieNet (Huang et al, 2016)	Joints	61.4	67.0
P-LSTM (Shahroudy et al, 2016a)	Joints	62.9	70.3
LTMD (Luo et al, 2017)	Depth	66.2	-
ST-LSTM (Liu et al, 2016)	Joints	69.2	77.7
DSSCA-SSLM (Shahroudy et al, 2017)	RGB-D	74.9	-
Proposed			
HPM_{RGB}	RGB	68.5	72.9
$\text{HPM}_{\text{RGB}}+\text{Traj}$	RGB	75.8	83.2
HPM_{3D}	Depth	71.5	70.5
$\text{HPM}_{\text{RGB}}+\text{HPM}_{\text{3D}}$	RGB-D	75.8	78.1
$\text{HPM}_{\text{RGB}}+\text{HPM}_{\text{3D}}+\text{Traj}$	RGB-D	80.9	86.1

Data	HPM Feature	Trajectory	FTP	SVM	Total	Rate(fps)
RGB	2.13E-02	2.78E-02	4.39E-05	2.95E-04	4.94E-02	20.2
Depth	2.13E-02	-	4.64E-05	3.32E-04	2.17E-02	46.1
RGB-D	2.13E-02	2.78E-02	4.64E-05	6.80E-04	4.98E-02	20.1

Table 7 Execution time in seconds for the proposed method

frames in the UWA dataset (especially view 4 in Fig. 9) and the NTU dataset have cropping errors.

9.1 Comparison with existing synthetic data

One of the major contributions of this work is synthetic data generation for robust action recognition. We note that SURREAL (Synthetic hUmans foR REAL tasks) (Varol et al, 2017b) is another recent method to generate synthetic action data that can be used to train the proposed HPMS. However, there are some major differences between SURREAL and the proposed synthetic data. (1) SURREAL was originally proposed for body segmentation and depth estimation whereas our dataset aims at modeling distinctive human poses from multiple viewpoints. While both datasets provide sufficient variations in clothing, human models, backgrounds, and illuminations; our dataset systematically covers 180° of view to enable viewpoint invariance, which is not the case for SURREAL. (2) To achieve realistic viewpoint variations, our dataset uses 360° spherical High Dynamic Range Images whereas SURREAL uses the LSUN dataset (Yu et al, 2015) for backgrounds. Hence,

our approach is more suitable for large viewpoint variations. (3) Finally, we use Generative Adversarial Network to reduce the distribution gap between synthetic and real data. Our results in Table 2 already verified that this provides additional boost to the action recognition performance.

To demonstrate the use of SURREAL with our pipeline and quantitatively analyze the advantages of the proposed dataset for robust action recognition, we compare the performance of our underlying approach using the two datasets on UWA3D and NUCLA databases. [We repeated our experiments with SURREAL using the same 339 representative poses as used by our data. This is possible because both datasets use CMU MoCap as the pose source. We trained HPMS with the SURREAL frames using the representative poses, followed by the same temporal encoding and classification as our data.](#)

Table 8 reports the mean recognition accuracies of HPMS on UWA3D and NUCLA datasets when trained using the SURREAL dataset and the proposed data. The table also includes results for the most challenging viewpoints in the datasets. For UWA3D, View 4

Method	Training Data	$V^{\text{challenge}}$	Mean
UWA3D Multiview Activity-II			
HPM _{RGB}	SURREAL	61.6	67.4
HPM _{RGB}	Proposed data	69.0	68.0
HPM _{3D}	SURREAL	65.8	72.1
HPM _{3D}	Proposed data	74.7	74.8
Northwestern-UCLA Multiview			
HPM _{RGB}	SURREAL	69.9	74.4
HPM _{RGB}	Proposed data	73.1	77.8
HPM _{3D}	SURREAL	68.1	77.4
HPM _{3D}	Proposed data	71.9	79.7

Table 8 Action recognition accuracy (%) for HPM_{RGB} and HPM_{3D} when trained with SURREAL dataset (Varol et al, 2017b) and the proposed synthetic data. $V^{\text{challenge}}$ represents the most challenging viewpoint in the multi-view test data, i.e. View 4 for UWA3D and View 3 for NUCLA dataset.

is challenging due to the large variations in both azimuth and elevation angles (see Fig. 9). For NUCLA, View 3 is particularly challenging as compared to the other viewpoints (see Fig. 10). From the results, we can see that the proposed data is able to achieve significant performance gain over SURREAL for these viewpoints. In our opinion, systematic coverage of 180° of view in our data is the main reason behind this fact. Our data also achieves a consistent overall gain for both RGB and depth modalities of the test datasets.

9.2 Improvements with synthetic data

Our experiments in Section 8 demonstrate a significant performance gain over the current state-of-the-art. Whereas the contribution of the network architecture, data modalities and GAN to the overall performance is clear from the presented experiments, we further investigate the performance gain contributed by the proposed synthetic data. To that end, we first compare HPM_{RGB}, which has been fine-tuned with the proposed data, to the original GoogLeNet model - not fine-tuned with our synthetic data. To ensure a fair comparison, all the remaining steps in the proposed pipeline, including temporal encoding and classification, are kept exactly the same for the two cases. The first two rows of Table 9 compare the mean recognition accuracies of HPM_{RGB} and GoogLeNet for UWA3D and NUCLA datasets. These results ascertain a clear performance gain with the proposed synthetic dataset.

The last two rows of Table 9 examine the performance gain of two popular baseline methods when fine-tuned on our synthetic data. Although significant, the average improvement in the accuracies of these methods is rather small compared to that of our method on

Method	Training	UWA3D	NUCLA
GoogLeNet	without synthetic data	62.8	66.7
HPM _{RGB}	with synthetic data	68.0	77.8
C3D†	with synthetic data	↑2.3	↑2.3
LRCN†	with synthetic data	↑3.5	↑1.2

Table 9 Analysis of performance gain due to the proposed data and method. The symbols † and ‡ denote enhancement of the existing techniques with our data and the improvement in accuracy (%), respectively.

our synthetic data (first two rows). Recall that our data generation method generates synthetic “poses” rather than videos and we had to extend our method to generate synthetic videos for the sake of this experiment. Details on synthetic video generation and training of the baseline methods are already provided in Section 8.3.1. From the results in Table 9, we conclude that our proposed method exploits our synthetic data more effectively, and both the proposed method and our synthetic data contribute significantly to the overall performance gain.

9.3 Role of synthetic RGB data in action recognition

Although our approach deals with RGB, depth and RGB-D data; we find it necessary to briefly discuss the broader role of synthetic RGB data in human action recognition. In contrast to depth videos, multiple large scale RGB video action datasets are available to train deep action models. Arguably, this diminishes the need of synthetic data in this domain. However, synthetic data generation methods such as ours and (Varol et al, 2017b) are able to easily ensure a wide variety of action irrelevant variations in the data, e.g. in camera viewpoints, textures, illuminations; up to any desired scale. In natural videos, such variety and scale of variations can not be easily guaranteed even in large scale datasets. For instance, in our experiments in Sections 8.3.1 and 8.3.2, both C3D and LRCN were originally pre-trained on large scale RGB video datasets in Table 4 and 5, yet our RGB synthetic data was able to boost their performance. Our synthetic data method easily and efficiently captures as many variations of the exact same action as desired, a real-world analogous to which is extremely difficult.

We also tested the performance of our approach on UCF-101 dataset (Soomro et al, 2012) to analyze the potential of synthetic data and HPMS for the standard action recognition benchmarks in the RGB domain. UCF-101 is a popular RGB-only action dataset, which includes video clips of 101 action classes. The ac-

Method	Human Body Motion
C3D	84.8
HPM _{RGB}	82.5
HPM _{RGB} †	84.6

Table 10 Action recognition accuracy (%) on the Human Body Motion subset of UCF-101 dataset. For a transparent analysis, the results do not include augmentation by trajectory features and/or ensemble features for any of the approaches. The symbol † denotes larger (2×) synthetic training data size.

tions covers 1) Human-Object Interaction, 2) Human Body Motion, 3) Human-Human Interaction, 4) Playing Musical Instruments, and 5) Sport Actions. Since we trained our HPMS to model human *poses*, the appearances of human poses in the test videos are important for a transparent analysis. Therefore, we selected 1910 videos from the dataset with 16 classes of Human Body Motion, and classified them using the proposed HPMS. Table 10 reports the performance of our approach along the accuracy of C3D on the same subset for comparison. The table also reports the accuracy of HPM_{RGB}†, for which we used twice as many pose labels and training images as used for training HPM_{RGB}. This improved the performance of our approach, indicating the advantage of easily producible synthetic data. Notice that, whereas the performance of HPMS remains comparable to C3D, the latter is trained on Millions of ‘videos’ as compared to the few hundred thousand ‘frames’ used for training our model. Moreover, our model is nearly 7 times smaller than C3D in size. These facts clearly demonstrate the usefulness of the proposed method and synthetic data generation technique for standard RGB action recognition.

10 Conclusion

We proposed Human Pose Models for human action recognition in RGB, depth and RGB-D videos. The proposed models uniquely represent human poses irrespective of the camera viewpoint, clothing textures, background and lighting conditions. We proposed a method for synthesizing realistic RGB and depth training data for learning such models. The proposed method learns 339 representative human poses from MoCap skeleton data and then fits 3D human models to these skeletons. The human models are then rendered as RGB and depth images from 180 camera viewpoints where other variations such as body shapes, clothing textures, backgrounds and lighting conditions are applied. We adopted Generative Adversarial Networks (GAN) to reduce the distribution gap between the synthetic and

real images. Thus, we were able to generate millions of realistic human pose images with known labels to train the Human Pose Models. The trained models contain complementary information between RGB and depth modalities, and also show good compatibility to the hand-crafted dense trajectory features. Experiments on three benchmark RGB-D datasets show that our method outperforms existing state-of-the-art on the challenging problem of cross-view and cross-person human action recognition by significant margins. The HPM_{RGB}, HPM_{3D} and Python script for generating the synthetic data will be made public.

Acknowledgements This research was sponsored by the Australian Research Council grant DP160101458. The Tesla K-40 GPU used for this research was donated by the NVIDIA Corporation.

References

- Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*, vol 1, pp 886–893
- Dalal N, Triggs B, Schmid C (2006) Human detection using oriented histograms of flow and appearance. In: *European conference on computer vision*, pp 428–441
- Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 2625–2634
- Du Y, Wang W, Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1110–1118
- Evangelidis G, Singh G, Horaud R (2014) Skeletal quads: Human action recognition using joint quadruples. In: *International Conference on Pattern Recognition*, pp 4513–4518
- Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9:1871–1874
- Farhadi A, Tabrizi MK (2008) Learning to recognize activities from the wrong view point. In: *European Conference on Computer Vision*, pp 154–166
- Farhadi A, Tabrizi MK, Endres I, Forsyth D (2009) A latent model of discriminative aspect. In: *IEEE International Conference on Computer Vision*, pp 948–955

- Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1933–1941
- Gkioxari G, Malik J (2015) Finding action tubes. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 759–768
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: *Advances in neural information processing systems*, pp 2672–2680
- Gopalan R, Li R, Chellappa R (2011) Domain adaptation for object recognition: An unsupervised approach. In: *IEEE International Conference on Computer Vision*, pp 999–1006
- Gupta A, Martinez J, Little JJ, Woodham RJ (2014) 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 2601–2608
- He K, Zhang X, Ren S, Sun J (2016a) Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 770–778
- He Y, Shirakabe S, Satoh Y, Kataoka H (2016b) Human action recognition without human. In: *European Conference on Computer Vision Workshops*, pp 11–17
- Hu JF, Zheng WS, Lai J, Zhang J (2015) Jointly learning heterogeneous features for rgb-d activity recognition. In: *IEEE conference on Computer Vision and Pattern Recognition*, pp 5344–5352
- Huang Z, Wan C, Probst T, Van Gool L (2016) Deep learning on lie groups for skeleton-based action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*
- Ji S, Xu W, Yang M, Yu K (2013) 3d convolutional neural networks for human action recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence* 35(1):221–231
- Jia C, Kong Y, Ding Z, Fu YR (2014a) Latent tensor transfer learning for rgb-d action recognition. In: *ACM international conference on Multimedia*, pp 87–96
- Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014b) Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:14085093*
- Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: *IEEE conference on Computer Vision and Pattern Recognition*, pp 1725–1732
- Kerola T, Inoue N, Shinoda K (2017) Cross-view human action recognition from depth maps using spectral graph sequences. *Computer Vision and Image Understanding* 154:108–126
- Kong Y, Fu Y (2017) Max-margin heterogeneous information machine for rgb-d action recognition. *International Journal of Computer Vision* 123(3):350–371
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105
- Li B, Camps OI, Sznaiar M (2012) Cross-view activity recognition using hankets. In: *IEEE conference on Computer Vision and Pattern Recognition*, pp 1362–1369
- Li R, Zickler T (2012) Discriminative virtual views for cross-view action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 2855–2862
- Li Y, Li W, Mahadevan V, Vasconcelos N (2016) Vlad3: Encoding dynamics of deep features for action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1951–1960
- Liu J, Shah M, Kuipers B, Savarese S (2011) Cross-view action recognition via view knowledge transfer. In: *IEEE conference on Computer Vision and Pattern Recognition*, pp 3209–3216
- Liu J, Shahroudy A, Xu D, Wang G (2016) Spatio-temporal lstm with trust gates for 3d human action recognition. In: *European Conference on Computer Vision*, pp 816–833
- Luo Z, Peng B, Huang DA, Alahi A, Fei-Fei L (2017) Unsupervised learning of long-term motion dynamics for videos. In: *IEEE Conference on Computer Vision and Pattern Recognition*
- Lv F, Nevatia R (2007) Single view human action recognition using key pose matching and viterbi path searching. In: *IEEE conference on Computer Vision and Pattern Recognition*, pp 1–8
- McInnes L, Healy J, Astels S (2017) hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*
- Ohn-Bar E, Trivedi M (2013) Joint angles similarities and hog2 for action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp 465–470
- Oreifej O, Liu Z (2013) Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 716–723
- Parameswaran V, Chellappa R (2006) View invariance for human action recognition. *International Journal of Computer Vision* 1(66):83–101

- Pfister T, Charles J, Zisserman A (2015) Flowing convnets for human pose estimation in videos. In: *IEEE International Conference on Computer Vision*, pp 1913–1921
- Rahmani H, Mian A (2015) Learning a non-linear knowledge transfer model for cross-view action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 2458–2466
- Rahmani H, Mian A (2016) 3d action recognition from novel viewpoints. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1506–1515
- Rahmani H, Mahmood A, Huynh D, Mian A (2016) Histogram of oriented principal components for cross-view action recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence* 38(12):2430–2443
- Rahmani H, Mian A, Shah M (2017) Learning a deep model for human action recognition from novel viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*
- Rao C, Yilmaz A, Shah M (2002) View-invariant representation and recognition of actions. *International Journal of Computer Vision* 50(2):203–226
- Shahroudy A, Liu J, Ng TT, Wang G (2016a) Ntu rgb+d: A large scale dataset for 3d human activity analysis. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1010–1019
- Shahroudy A, Ng TT, Yang Q, Wang G (2016b) Multimodal multipart learning for action recognition in depth videos. *IEEE transactions on Pattern Analysis and Machine Intelligence* 38(10):2123–2129
- Shahroudy A, Ng TT, Gong Y, Wang G (2017) Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*
- Shakhnarovich G (2005) Learning task-specific similarity. PhD thesis, Massachusetts Institute of Technology
- Shrivastava A, Pfister T, Tuzel O, Susskind J, Wang W, Webb R (2016) Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:161207828*
- Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: *Advances in neural information processing systems*, pp 568–576
- Soomro K, Zamir AR, Shah M (2012) Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:12120402*
- Su B, Zhou J, Ding X, Wang H, Wu Y (2016) Hierarchical dynamic parsing and encoding for action recognition. In: *European Conference on Computer Vision*, pp 202–217
- Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1–9
- Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: *IEEE international conference on Computer Vision*, pp 4489–4497
- Varol G, Laptev I, Schmid C (2017a) Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*
- Varol G, Romero J, Martin X, Mahmood N, Black MJ, Laptev I, Schmid C (2017b) Learning from Synthetic Humans. In: *IEEE Conference on Computer Vision and Pattern Recognition*
- Vemulapalli R, Arrate F, Chellappa R (2014) Human action recognition by representing 3d skeletons as points in a lie group. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 588–595
- Wang H, Schmid C (2013) Action recognition with improved trajectories. In: *IEEE International Conference on Computer Vision*, pp 3551–3558
- Wang H, Kläser A, Schmid C, Liu CL (2011) Action recognition by dense trajectories. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 3169–3176
- Wang H, Kläser A, Schmid C, Liu CL (2013a) Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision* 103(1):60–79
- Wang J, Liu Z, Wu Y, Yuan J (2013b) Learning action-let ensemble for 3d human action recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence*
- Wang J, Nie X, Xia Y, Wu Y, Zhu SC (2014) Cross-view action modeling, learning and recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 2649–2656
- Wang L, Qiao Y, Tang X (2015) Action recognition with trajectory-pooled deep-convolutional descriptors. In: *IEEE conference on Computer Vision and Pattern Recognition*, pp 4305–4314
- Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016a) Temporal segment networks: Towards good practices for deep action recognition. In: *European Conference on Computer Vision*, pp 20–36
- Wang P, Li Z, Hou Y, Li W (2016b) Action recognition based on joint trajectory maps using convolutional neural networks. In: *ACM on Multimedia Conference*, pp 102–106