Recognizing Human Actions as the Evolution of Pose Estimation Maps

Mengyuan Liu1  Junsong Yuan2,∗
1 School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798
2 Department of CSE, SUNY at Buffalo, Buffalo NY 14260
liumengyuan@ntu.edu.sg  jsyuan@buffalo.edu

Abstract

Most video-based action recognition approaches choose to extract features from the whole video to recognize actions. The cluttered background and non-action motions limit the performances of these methods, since they lack the explicit modeling of human body movements. With recent advances of human pose estimation, this work presents a novel method to recognize human action as the evolution of pose estimation maps. Instead of relying on the inaccurate human poses estimated from videos, we observe that pose estimation maps, the byproduct of pose estimation, preserve richer cues of human body to benefit action recognition. Specifically, the evolution of pose estimation maps can be decomposed as an evolution of heatmaps, e.g., probabilistic maps, and an evolution of estimated 2D human poses, which denote the changes of body shape and body pose, respectively. Considering the sparse property of heatmap, we develop spatial rank pooling to aggregate the evolution of heatmaps as a body shape evolution image. As body shape evolution image does not differentiate body parts, we design body guided sampling to aggregate the evolution of poses as a body pose evolution image. The complementary properties between both types of images are explored by deep convolutional neural networks to predict action label. Experiments on NTU RGB+D, UTD-MHAD and PennAction datasets verify the effectiveness of our method, which outperforms most state-of-the-art methods.

1. Introduction

1.1. Motivation and Objective

Human action recognition from videos has been researched for decades, since this task enjoys various applications in intelligent surveillance, human-robot interaction and content-based video retrieval. The intrinsic property of existing methods [22,43,37,24,1] is to learn mapping functions which transform videos to action labels. Since they do not directly distinguish human body from videos, these methods are easily affected by clutters and non-action motions from backgrounds.

To address this limitation, an alternative solution is to detect human [39] and estimate the body pose in each frame. This approach works well in the field of human action recognition from depth videos, e.g., Microsoft Kinect [55,27]. By detecting 3D pose from each depth frame with an accurate body pose estimation method [36], human movements in depth videos can be simplified as 3D pose sequences [52]. Recent deep learning models, e.g., CNN [17,20], RNN [9] and LSTM [26,25], have achieved high performances on the extracted 3D poses, which outperform methods [32,50] that rely on raw depth video sequences.

The success of 3D human pose inspires us to estimate 2D human poses from videos for action recognition. However, despite the significant advances of 2D pose estima-
tion in images and videos [51, 5, 46, 2, 4], the performance is still inferior to the 3D pose estimation in depth videos. Fig. 1 illustrates the estimated poses from video frames by a state-of-the-art pose estimation method [4]. Due to complex background and self-occlusion of human body parts, the estimated poses are not fully reliable and may misinterpret the configuration of human body. In the first row of Fig. 1 (b), the multi-modal pose estimation map in the white bounding box indicates the location of the person’s hand. The map contains two peaks, where the ground truth location does not correspond to the highest peak, thus providing a wrong estimation of the hand’s location.

To better utilize the pose estimation maps, instead of relying on the inaccurate 2D pose estimated from the pose estimation maps, we propose to directly model the evolution of pose estimation maps for action recognition. In Fig. 1 (c), heatmaps (averaged pose estimation maps) provide richer information to reflect human body shape.

1.2. Method Overview and Contributions

Our method is shown in Fig. 2. Given each frame of a video, we use convolutional pose machines to predict pose estimation map for each body part. The goal of representing these pose estimation maps is to preserve both global cues, which reflect whole shapes that suffer less from the noise and local cues, which detail the locations of body parts.

To this end, we average pose estimation maps of all body parts to generate an averaged pose estimation map (heatmap) for each frame. The temporal evolution of heatmaps can reflect the movements of body shape. Different from the original RGB image, the heatmap is sparse. Considering the huge spatial redundancy, we develop a spatial rank pooling method to compress the heatmap as a compact yet informative feature vector. The merit of spatial rank pooling is that it can effectively suppress spatial redundancy, without significantly losing spatial distribution information of the heatmap. The temporal concatenation of feature vectors constructs a 2D body shape evolution image, which reflects the temporal evolution of body shapes.

As body shape evolution image cannot differentiate body parts, we further predict joint location from pose estimation map of each body part, generating a pose for each frame. Since the number of estimated pose joints is limited, we use body structure to guide the sampling of more abundant pose joints to represent human body. The temporal concatenation of all pose joints constructs a body pose evolution image, which reflects the temporal evolution of body parts. Intuitively, the body shape evolution image and body pose evolution image benefit the recognition of general movements of body shape and elaborate movements of body parts. Thereby, both images are explored by CNNs to generate discriminative features, which are later fused to predict action label. Generally, our contributions are three-fold.

- Given inaccurate 2D poses estimated from videos, we boost the performance of human action recognition by recognizing actions as the evolution of pose estimation maps instead of the unreliable 2D body poses.
- The evolution of pose estimation maps are described as body shape evolution image and body pose evolution image, which capture the movements of both whole body and specific body parts in a compact way.
- With CNNs and late fusion scheme, our method achieves state-of-the-art performances on NTU RGB-D, UTD-MHAD and PennAction datasets.

2. Related Work

2.1. 3D Pose-based Action Recognition

3D pose provides direct physical interpretation for human actions from depth videos. Hand-crafted features [42, 47, 13] were designed for describing evolution of 3D poses. Recently, deep neural networks were introduced to model the spatial structures and temporal dynamics of poses. For example, Du et al. [9] firstly used hierarchical RNN for pose-based action recognition. Liu et al. [25] extended this idea and proposed spatio-temporal LSTM to learning spatial and temporal domains. To enhance the attention capability of LSTM, Global Context-Aware Attention LSTM [26] was developed with the assistance of global context.

2.2. Video-based Action Recognition

Local features are motion-related and are robust to cluttered background to some extent. Spatial temporal interest points (STIPs) [22] and dense trajectory [43] were applied to extract and describe local spatial temporal patterns. Based on these basic features, multi-feature max-margin hierarchical Bayesian model [49] and a novel feature enhancement technique called Multi-skIp Feature Stacking [21] were proposed to learn more distinctive features. Since local features ignore global relationships, holistic features were encoded by two-stream convolutional network [37], which learns spatial-temporal features by fusing convolutional networks spatially and temporally. Based on this network, the relationships between the spatial and temporal structures were further explored [11, 45]. Different from two-stream network, the spatial and temporal information of actions can be fused before they are input to CNNs. Fernando et al. [12] proposed rank pooling method to aggregate all video frames to a compact representation. Bilen et al. [1] deeply merged rank pooling method with CNN to generate an efficient dynamic image network.

Human actions are inherently structured patterns of body movements. Recent studies [56, 31, 14, 38, 30] extracted whole human body or body parts instead of whole video for action analysis. Meanwhile, human action recognition and
pose estimation tasks have been integrated to extract pose guided features for recognition. Wang et al. [41] improved an existing pose estimation method, and then designed pose features to capture both spatial and temporal configurations of body parts. Xiaohan et al. [48] proposed a framework to integrate training and testing of action recognition and pose estimation. They decomposed actions into poses which are further divided to mid-level ST-parts and then parts. Most recently, Du et al. [8] proposed an end-to-end recurrent network which can exploit important spatial-temporal evolutions of human pose to assist action recognition in a unified framework. Different from pose features [41] or pose-guided color features [48, 8], this paper recognizes human actions from only pose estimation maps, which have not been explored for action recognition task before.

### 3. Generation of Pose Estimation Maps

This section predicts pose estimation maps from each frame of a video (Fig. 3 (a)), and then generates a heatmap (Fig. 3 (b)) and a pose (Fig. 3 (c)) to denote each frame.

**Pose Estimation Maps:** The task of human pose estimation from a single image can be modeled as a structure prediction problem. In [34], a pose machine is proposed to sequentially predict pose estimation maps for body joints, where previous predicted pose estimation maps iteratively improve the estimates in following stages. Let $Y_k \in \{x, y\}$ denote the set of coordinates from body joint $k$. The structural output can be formulated as $Y = \{Y_1, ..., Y_k, ..., Y_K\}$, where $K$ is the total number of body joints. Multi-class classifier $g_l^t$ is trained to predict the $k_{th}$ body joint in the $t_{th}$ stage. For an image position $z$, the pose estimation map for assigning it to the $k_{th}$ body joint is formulated as:

$$B_l^k(z) = g_l^k \left( f_l^t \left( \bigcup_{i=1,...,K} \psi(z, B_{l-1}^i) \right) \right),$$

(1)

where $f_l$ is the color feature at position $z$, $B_{l-1}^i$ is the pose estimation map predicted by $g_{l-1}^i$, $\bigcup$ is the operator for vector concatenation, $\psi$ is the feature function for computing contextual features from previous pose estimation maps. After $T$ stages, the generated pose estimation maps are used to predict locations of body joints. The pose machine [34] uses boosted classifier with random forests for the weak learners. Instead, this paper applies the convolutional pose machine [46, 4] to combine pose machine with convolutional architectures, which does not need graphical-model style inference and boosts the performances of pose machine.

**Heatmaps & Poses:** For the $n_{th}$ frame of a video, $K$ types of pose estimation maps, namely $\{B_{T,n}^1, ..., B_{T,n}^K\}$, are generated. To reduce the redundancy of pose estimation maps, we describe them as a heatmap $G_n$ and a pose $L_n$. The heatmap $G_n$ can be expressed as:

$$G_n = \frac{1}{K} \sum_{k=1}^{K} B_{T,n}^k,$$

(2)

which reflects the global body shape. The pose $L_n$ can be expressed as $\{z_{k,n}\}_{k=1}^{K}$, where $z_{k,n}$ is often estimated via Maximum A Posterior (MAP) criterion [4]:

$$z_{k,n} = \arg \max_{z \in \mathbb{Z}} \{B_{T,n}^k(Y_k = z)\},$$

(3)

where $\mathbb{Z} \in \mathbb{R}^2$ denote all positions on the image. Till now, each frame of a video is described as a heatmap and a pose. In other words, the video is converted to the evolution of heatmaps and the evolution of poses.
4. Evolution of Pose Estimation Maps

This section describes the evolution of heatmaps as a body shape evolution image using temporal rank pooling (Fig. 3 (d)), based on which spatial rank pooling (Fig. 3 (e)) is developed. Further, body guided sampling is developed to describe the evolution of poses as a body pose evolution image (Fig. 3 (f)). The complementary properties between two images are learned by CNNs.

**Temporal Rank Pooling:** As a robust and compact video representation method, temporal rank pooling [12, 1] has the ability to aggregate the temporal relevant information throughout a video via a learning to rank approach. The encoded temporal information denotes the temporal order among frames, which is a robust feature showing less sensitive to different types of input data. As heatmaps are distinct from natural images, we treat heatmaps as a new type of data and apply temporal rank pooling to encode the evolution of heatmaps. Suppose a sequence \( V_G = \{G_1, ..., G_n, ..., G_N\} \) contains \( N \) heatmaps, and \( G_n \in \mathbb{R}^{P \times Q} \) denotes the \( n_{th} \) frame with \( P \) rows and \( Q \) columns. \( G_{1:n} \) can be mapped to a vector defined as:

\[
v_n = V \left( \frac{1}{n} \sum_{i=1}^{n} G_i \right), \tag{4}
\]

where the function \( V \) reshapes a matrix into a vector and \( v_n \in \mathbb{R}^{(P \times Q) \times 1} \). Let \( v_{n+1} \succ v_n \) denote the temporal ordering relationship between \( v_{n+1} \) and \( v_n \). A natural constraint among frames is \( v_N \succ ... \succ v_{n+1} \succ v_n \succ ... \succ v_1 \). Temporal rank pooling (t-rk) [12] optimizes parameters \( u \in \mathbb{R}^{(P \times Q) \times 1} \) of a linear function \( \psi(v; u) \) to ensure that \( \forall n_1, n_2, v_{n_1} \succ v_{n_2} \Leftrightarrow u^T \cdot v_{n_1} \succ u^T \cdot v_{n_2} \). The parameter \( u \) is used as the representation of temporal rank pooling method, as it implicitly encodes the appearance evolution information of the sequence.

**Spatial Rank Pooling:** We reshape \( u \) as the same size of input frame to facilitate the observation. As shown in Fig. 3 (d), the temporal rank pooling method mainly preserves spatial information while ignores most of the temporal information. To improve this, we propose a novel spatial rank pooling method (s-rk) which takes both spatial and temporal information into account. The observation is that there exists huge spatial redundancy in each heatmap. Therefore, we take advantage of the learning to rank method to reduce each heatmap to a compact feature, which has the ability of preserving spatial order. Concatenating all feature vectors according to the temporal order will generate a body shape evolution image, which can preserve both spatial and temporal information of heatmaps in a compact way. The pipeline of generating body shape evolution image is shown in Fig. 4. Specifically, we partition the \( n_{th} \) frame of the sequence \( V \) into \( P \) rows, i.e., \( G_n = [(p_1)^T, ..., (p_j)^T, ..., (p_P)^T]^T \), or \( Q \) columns, i.e., \( G_n = [q_1, ..., q_s, ..., q_Q] \). Similar to Eq. 4, function \( V \) is applied to map \( (p_1:s)^T \) and \( q_{1:s} \) to \( v^p_\eta \) and \( v^q_\eta \), respectively. Using the structural risk minimization and max-margin framework, the objective is defined as:

\[
\arg \min_{\eta} \frac{1}{2} \| u^\eta \|^2 + W \sum_{v_{i,j} \prec v^p_{i,j}} \epsilon_{i,j}, \tag{5}
\]

\[
s.t. \; (u^\eta)^T \cdot (v^p_{i,j} - v^q_{i,j}) \geq 1 - \epsilon_{i,j} \; \epsilon_{i,j} \geq 0
\]

where \( \eta \in \{p, q\} \), \( u^p \in \mathbb{R}^Q \) and \( u^q \in \mathbb{R}^P \). For all \( N \) frames, we concatenate vectors according to the temporal order, and obtain \( U^p \in \mathbb{R}^{Q \times N} \) and \( U^q \in \mathbb{R}^{P \times N} \). The final matrix \( U \) via spatial rank pooling is defined as \( \left[(U^p)^T, (U^q)^T\right]^T \), where \( U \) is called body shape evolution image.

**Body Guided Sampling:** Since body shape evolution image only considers the shape of accumulated pose estimation map while does not differentiate different joints, we need body pose evolution image to consider the information from each specific joint. To this end, this section builds body pose evolution image from joints, which are densely sampled by body guided sampling to denote the specific pose of human body. Suppose a sequence \( \mathcal{L}_\eta = \{L_1, ..., L_n, ..., L_N\} \) contains \( N \) poses, where \( L_n = (z^{x,n})_k \) for \( k = 1 \) and \( z^{x,n} = (x^{k,n}, y^{k,n}) \), which denote the horizontal and vertical coordinates of the \( k_{th} \) joint. When
recording pose sequences, the distances between human bodies and the depth video are not strictly the same. In other words, different pose sequences have their own scales, which bring intra-varieties to same types of actions. To this end, \( \{ x^k_n \}_{k=1}^K \) and \( \{ y^k_n \}_{k=1}^K \) are normalized to change from 0 to 1, respectively.

To encode the spatial and temporal evolution of pose joints in a compact way, we represent the pre-processed pose sequence \( Y_c \) as a body pose evolution image. Since original estimated pose joints are too sparse to represent the human body, we sort the order of joint labels according to the body structure, and use linear interpolation to sample abundant points from pose limbs. The pipeline of building one channel of body pose evolution image from horizontal coordinates is shown in Fig. 5. Another channel of body pose evolution image from vertical coordinates can be similarly built. Mathematically, let \( x^n = [x^1,n, ..., x^K,n]^T \) and \( y^n = [y^1,n, ..., y^K,n]^T \) denote the coordinate vector of the generated joints, where \( K' \) is the total number of sampled joints. In our experiments, the pose limbs can be roughly denoted by sampling five joints on each limb. Suppose \( K = 14 \) and the number of limb is \( K - 1 \), then the \( K' \) can be calculated as \( K + (K - 1) \times 5 = 79 \). Two channels of body pose evolution image are formed as \( [x^1, ..., x^N] \) and \( [y^1, ..., y^N] \), denoting horizontal and vertical coordinates, respectively. Both channels reflect the temporal evolution of joints, which are densely sampled to denote the pose of human body.

**Late Fusion:** A video \( I^c \) has been denoted as a body shape evolution image and a body pose evolution image, where \( c \) means the \( c_{th} \) sample from a batch that is used for training. As CNN has achieved success in image classification task, we use CNN model that is pre-trained on Imagenet [7] for transfer learning. Since these two images contain significantly different spatial structure, we use separate CNN to explore deep features from them. To accommodate existing CNN models, the single channel of body shape evolution image is repeated three times to form a 3-channel image, and two channels of body pose evolution image are combined with a zero-valued channel to form a 3-channel image. Let \( I_m \) denote these two images. Mean removal is adopted for input images to improve the convergence speed. Then, each color image is processed by a CNN. For the image \( I_m \), the output \( \Upsilon \) of the last fully-connected (fc) layer is normalized by the softmax function to obtain the posterior probability: \( \text{prob}(r \mid I_m) = e^{\Upsilon_m} / \sum_{c=1}^C e^{\Upsilon^c} \), which indicates the probability of image \( I_m \) belonging to the \( r_{th} \) action class. \( R \) is the number of total action classes. The objective function of our model is to minimize the maximum-likelihood loss function \( \mathcal{L}(I_m) = -\sum_{r=1}^R \sum_{c=1}^C \delta(r - s_c) \text{prob}(r \mid I_m) \), where function \( \delta \) equals to one if \( r = s_c \) and equals to zero otherwise, \( s_c \) is the ground truth label of \( I_m \), and \( C \) is the batch size. For a sequence \( I \), its final class score is the average of the two posteriors: \( \text{score}(r \mid I) = \frac{1}{2} \sum_{m=1}^2 \text{prob}(r \mid I_m) \), where \( \text{prob}(r \mid I_m) \) is the probability of \( I_m \) belonging to the \( r_{th} \) action class.

### 5. Experiments

#### 5.1. Datasets and Settings

**PennAction dataset** [54] contains 15 action categories and 2326 sequences in total. Since all sequences are collected from the internet, complex body occlusions, large appearance and motion variations make it challenging for
pose-related action recognition [48, 8]. We follow [48] to split the data into half and half for training and testing. Snaps with estimated poses 1 are shown in Fig. 6.

**NTU RGB+D dataset** [35] contains 60 actions performed by 40 subjects from various views, generating 56880 sequences. Following the cross subject protocol in [35], we split the 40 subjects into training and testing groups. Each group contains samples captured from different views performed by 20 subjects. For this evaluation, the training and testing sets have 40320 and 16560 samples, respectively. Following the cross view protocol in [35], we use all the samples of camera 1 for testing and samples of cameras 2, 3 for training. The training and testing sets have 37920 and 18960 samples, respectively.

**UTD-MHAD dataset** [6] was collected using a Microsoft Kinect sensor and a wearable inertial sensor in an indoor environment. It contains 27 actions performed by 8 subjects. Each subject repeated each action 4 times, generating 861 sequences. We use this dataset to compare the performances of methods using different data modalities. Cross subject protocol [6] is used for the evaluation.

**Implementing details:** In our model, each CNN contains five convolutional layers and three fc layers. The first and second fc layers contain 4096 neurons, and the number of neurons in the third one is equal to the total number of action classes. Filter sizes are set to $11 \times 11$, $5 \times 5$, $3 \times 3$, $3 \times 3$ and $3 \times 3$, respectively. Local Response Normalization (LRN), max pooling and ReLU neuron are adopted and the dropout regularization ratio is set to 0.5. The network weights are learned using the mini-batch stochastic gradient descent with the momentum value set to 0.9 and weight decay set to 0.0005. Learning rate is set to 0.001 and the maximum training cycle is set to 60. When the CNN model achieves 99% accuracy on the training set, the training procedure is stopped beforehand. In each cycle, a mini-batch of $C$ samples is constructed by randomly sampling images from training set. For NTU RGB+D dataset, UTD-MHAD dataset and PenAction dataset, $C$ is set to 64, 16 and 16, considering the size of training set. To reduce the effect of random parameter initialization and random sampling, we repeat the training of CNN model for five times and report the average results. The implementation is based on PyTorch with one TITAN X card and 16G RAM.

### 5.2. Discussions

**2D pose & 3D pose from depth video:** As poses estimated from videos lose depth cues, we begin with evaluating the significance of depth information for pose-based human action recognition. In Table 1, a 3D pose sequence extracted from a depth video is described as a three channel body pose evolution image, which is further encoded by CNN to predict action label. This method is called “S2”. By setting all values of depth channel to zero, 2D pose sequences from depth videos are used instead. This method is called “S1”. Without using depth channel, the accuracy drops from 89.44% to 85.53% on UTD-MHAD dataset. While, accuracies drop by only 1.86% from 82.38% to 80.52% for cross subject setting and 0.90% from 86.65%

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensor</th>
<th>Data</th>
<th>Feature</th>
<th>Action</th>
<th>NTU RGB+D</th>
<th>UTD-MHAD</th>
<th>PennAction</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Kinect</td>
<td>2D Pose</td>
<td>BPI</td>
<td>0.52%</td>
<td>85.53%</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>Kinect</td>
<td>3D Pose</td>
<td>BPI</td>
<td>0.53%</td>
<td>84.61%</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>H1</td>
<td>RGB</td>
<td>2D Pose</td>
<td>BPI</td>
<td>72.96%</td>
<td>85.63%</td>
<td>84.08%</td>
<td></td>
</tr>
<tr>
<td>H2</td>
<td>RGB</td>
<td>Heatmap</td>
<td>BSI (t-rk)</td>
<td>53.91%</td>
<td>58.88%</td>
<td>84.61%</td>
<td></td>
</tr>
<tr>
<td>H3</td>
<td>RGB</td>
<td>Heatmap</td>
<td>BSI (s-rk)</td>
<td>72.75%</td>
<td>74.88%</td>
<td>87.02%</td>
<td></td>
</tr>
<tr>
<td>H1 + H3</td>
<td>RGB</td>
<td>2D Pose + Heatmap</td>
<td>BPI + BSI (s-rk)</td>
<td>78.80%</td>
<td>92.51%</td>
<td>91.39%</td>
<td></td>
</tr>
<tr>
<td>S1 + H3</td>
<td>Kinect</td>
<td>2D Pose + Heatmap</td>
<td>BPI + BSI (s-rk)</td>
<td>90.90%</td>
<td>92.84%</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>S2 + H3</td>
<td>Kinect</td>
<td>3D Pose + Heatmap</td>
<td>BPI + BSI (s-rk)</td>
<td>91.71%</td>
<td>94.51%</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

1 These poses are generated by pose estimation method [4], which can be found from https://github.com/tensorboy/pytorch_Realtime_Multi-Person_Pose_Estimation. Four pose joints on the head are not used since they are redundant for denoting actions. For scenes with multi-person, the coordinates of joints are averaged for feature extraction.
Figure 8: Confusion matrices of body pose evolution image-based method (first row), body shape evolution image-based method (second row), body pose and body shape evolution images-based method (third row) on NTU RGB+D dataset using cross subject protocol. Confusion matrices of ten actions are enlarged. These ten actions are: touch back (backache) (46), touch neck (neck-ache) (47), nausea or vomiting condition (48), use a fan (with hand or paper/feeling warm (49), punching/slapping other person (50), kicking other person (51), pushing other person (52), pat on back of other person (53), point finger at the other person (54), hugging other person (55). Red boxes verify that the body pose and body shape evolution images compensate for each other and improve the recognition of action (49).

2D pose from video: With accurate pose estimation method and additional depth cues, 3D poses from depth videos are more reliable than 2D poses from videos. This part evaluates the performance of noisy 2D poses from videos by comparing with 2D poses and 3D poses from depth videos. “H1” denotes our proposed method using body pose evolution image. On NTU RGB+D dataset, “H1” performs worse than “S1”. The reason is that this dataset contains multi-view samples, which brings more ambiguities to 2D pose from video than 3D pose from depth video. The performance of “H1” is comparable with “S1” on UTD-MHAD dataset. The reason is that this dataset contains samples observed from single view, which helps the pose estimation from both RGB and depth videos. Generally, 2D pose from video can only compete with that from depth video in simple scenes; 2D pose from video can barely achieve the performance of 3D pose from depth video.

Heatmap from video: Instead of using sole 2D pose, we evaluate the performance of combining heatmap with 2D pose for recognition from video. The method called “H3” describes heatmap as body shape evolution image using spacial rank pooling. For comparisons, the method called “H2” is implemented by temporal rank pooling. “H3” outperforms “H2” by more than 15% on both NTU RGB+D and UTD-MHAD datasets, which verifies the advantage of spatial rank pooling method in preserving both spatial and temporal cues. The method called “H1 + H3” denotes the combination of both 2D pose and heatmap. “H1 + H3” outperforms at least 5% than “H1”. Detailed improvements on NTU RGB+D dataset using cross subject protocol are shown in Fig. 7. These results indicate the complementary property between 2D pose and heatmap. In Fig. 8, we analyze the confusion matrices among 10 types of actions. The red boxes highlight the improvement on action “use a fan (with hand or paper/feeling warm (49))”, by combining 2D pose and heatmap. As shown in Fig. 9, body pose evolution image only captures 2D pose joints, which are noisy due to occlusions. However, body shape evolution image, which captures global shape of heatmap, can provide cues for inferring accurate locations of 2D pose joints.

Table 2: Comparisons between our proposed method and state-of-the-art methods on NTU RGB+D dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Year</th>
<th>CS</th>
<th>CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>HON4D [32]</td>
<td>2013</td>
<td>30.56%</td>
<td>7.26%</td>
</tr>
<tr>
<td>Super Normal Vector [50]</td>
<td>2014</td>
<td>31.82%</td>
<td>13.61%</td>
</tr>
<tr>
<td>Skeletal Quads [10]</td>
<td>2014</td>
<td>38.60%</td>
<td>41.40%</td>
</tr>
<tr>
<td>Lie Group [40]</td>
<td>2014</td>
<td>50.10%</td>
<td>52.80%</td>
</tr>
<tr>
<td>HBRNN-L [9]</td>
<td>2015</td>
<td>59.07%</td>
<td>63.97%</td>
</tr>
<tr>
<td>FTP Dynamic Skeletons</td>
<td>2015</td>
<td>60.23%</td>
<td>65.22%</td>
</tr>
<tr>
<td>Deep RNN [35]</td>
<td>2016</td>
<td>59.29%</td>
<td>64.09%</td>
</tr>
<tr>
<td>Deep LSTM [35]</td>
<td>2016</td>
<td>60.69%</td>
<td>67.29%</td>
</tr>
<tr>
<td>2Layer P-LSTM [35]</td>
<td>2016</td>
<td>62.93%</td>
<td>70.27%</td>
</tr>
<tr>
<td>ST-LSTM + Trust Gate</td>
<td>2016</td>
<td>69.20%</td>
<td>77.70%</td>
</tr>
<tr>
<td>Unsupervised Learning</td>
<td>2017</td>
<td>56.00%</td>
<td>-</td>
</tr>
<tr>
<td>LieNet-3Blocks [17]</td>
<td>2017</td>
<td>61.37%</td>
<td>66.95%</td>
</tr>
<tr>
<td>GCA-LSTM network [26]</td>
<td>2017</td>
<td>74.40%</td>
<td>82.80%</td>
</tr>
<tr>
<td>Body-part appearance + skeleton [33]</td>
<td>2017</td>
<td>75.20%</td>
<td>83.10%</td>
</tr>
<tr>
<td>Clips + CNN + MTLN [20]</td>
<td>2017</td>
<td>79.57%</td>
<td>84.83%</td>
</tr>
<tr>
<td>View-invariant [28]</td>
<td>2017</td>
<td>80.03%</td>
<td>87.21%</td>
</tr>
<tr>
<td><strong>Proposed Method: H1 + H3</strong></td>
<td>-</td>
<td><strong>78.80%</strong></td>
<td><strong>84.21%</strong></td>
</tr>
<tr>
<td><strong>Proposed Method: S2 + H3</strong></td>
<td>-</td>
<td><strong>91.71%</strong></td>
<td><strong>95.26%</strong></td>
</tr>
</tbody>
</table>

Table 3: Comparisons between our proposed method and state-of-the-art methods on UTD-MHAD dataset

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Method</th>
<th>Year</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinect</td>
<td>Cor3D3 [18]</td>
<td>2013</td>
<td>85.58%</td>
</tr>
<tr>
<td>Kinect</td>
<td>Kinect [6]</td>
<td>2015</td>
<td>66.10%</td>
</tr>
<tr>
<td>Inertial</td>
<td>Inertial [6]</td>
<td>2015</td>
<td>67.20%</td>
</tr>
<tr>
<td>Kinect + Inertial</td>
<td>Kinect&amp;Inertial [6]</td>
<td>2015</td>
<td>79.10%</td>
</tr>
<tr>
<td>Kinect</td>
<td>PFTM [44]</td>
<td>2016</td>
<td>85.81%</td>
</tr>
<tr>
<td>Kinect</td>
<td>Optical Spectra [15]</td>
<td>2016</td>
<td>86.97%</td>
</tr>
<tr>
<td>Kinect</td>
<td>3DHOT-MBC [53]</td>
<td>2017</td>
<td>84.40%</td>
</tr>
<tr>
<td>Kinect</td>
<td>JDM [23]</td>
<td>2017</td>
<td>88.10%</td>
</tr>
<tr>
<td>RGB</td>
<td><strong>Proposed Method: H1 + H3</strong></td>
<td>-</td>
<td><strong>92.84%</strong></td>
</tr>
<tr>
<td>Kinect</td>
<td><strong>Proposed Method: S2 + H3</strong></td>
<td>-</td>
<td><strong>94.51%</strong></td>
</tr>
</tbody>
</table>
5.3. Comparisons with State-of-the-arts

Ours versus 3D Pose-based methods: Even with inaccurate 2D pose estimation method, our method outperforms 3D pose, which is extracted by 3D pose estimation method from depth sensors. In Table 2 and Table 3, “H1 + H3” outperforms most state-of-the-art methods using 3D poses. Specifically, “H1 + H3” achieves 78.80% and 84.21% on the currently largest NTU RGB+D dataset. “H1 + H3” also outperforms LSTM-based method, i.e., GCA-LSTM [26]. Although slightly worse, “H1 + H3” approaches to similar performance of the most recent CNN-based method, i.e., View-invariant [28]. On UTD-MHAD dataset, “H1 + H3” outperforms all 3D pose-based methods, e.g., 3DHT-MBC [53] and JDM [23]. Instead of using 2D poses, it is interesting to combine heatmap with more accurate poses, namely 2D poses from depth video. Table 1 shows that “S1+H3” outperforms “H1+H3”. With additional depth information, “S2+H3” achieves the best performances. These results verify that our proposed heatmaps benefit both 2D poses from videos and 3D poses from depth videos.

Ours versus video-based methods: Among approaches using videos, our method is compared with most related 2D pose-based action recognition methods. In Table 4, poselet detected by Action Bank [54] achieves accuracy of 83.90% on PennAction dataset. AOG [48] and Pose + IDT-FV [19] benefit from treating pose estimation and action recognition as a uniform framework, and achieve accuracy of 85.50% and 92.00%. RPAN [8] goes beyond previous studies by training an end-to-end RNN network, and achieves accuracy of 97.40%. Our method jointly learns 2D poses and heatmaps, and the complementary property between them alleviate the effect of noisy 2D poses. We further select one frame for each video and use CNN to extract deep features. Also, we encode annotated poses, which are provided by original dataset. Fused with these additional information, our method “(H1 + H3)*” achieves accuracy of 98.22% and outperforms all recent methods. The confusion matrix of our method is shown in Fig. 10, where most of ambiguities among actions are suppressed.

6. Conclusions

This paper recognizes actions from videos as the evolution of pose estimation maps. Compared with unreliable estimated 2D poses, pose estimation maps provide richer cues for inferring body parts and their movements. By describing the evolution of pose estimation maps as compact body shape evolution image and body pose evolution image, our method can effectively capture movements of both body shape and body parts, thereby outperforming all 2D pose or 3D pose-based methods on benchmark datasets. It is noted that our features only rely on the estimated pose estimation maps rather than original videos, from which the pose estimation maps are estimated. This property indicates the generalization ability of our method by estimating pose estimation maps from various types of input video, e.g., depth or infrared video, for action recognition task.

7. Acknowledgement

This work is supported in part by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015-T2-2-114 and start-up funds of University at Buffalo.
References


