3D Action Recognition Using Multi-temporal Depth Motion Maps and Fisher Vector

Chen Chen\textsuperscript{1,*}, Mengyuan Liu\textsuperscript{2,*}, Baochang Zhang\textsuperscript{3†}, Jungong Han\textsuperscript{4}, Junjun Jiang\textsuperscript{5}, Hong Liu\textsuperscript{2}

\textsuperscript{1} Department of Electrical Engineering, University of Texas at Dallas
\textsuperscript{2} Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University
\textsuperscript{3} School of Automation Science and Electrical Engineering, Beihang University
\textsuperscript{4} Department of Computer Science and Digital Technologies, Northumbria University
\textsuperscript{5} School of Computer Science, China University of Geosciences

Abstract

This paper presents an effective local spatio-temporal descriptor for action recognition from depth video sequences. The unique property of our descriptor is that it takes the shape discrimination and action speed variations into account, intending to solve the problems of distinguishing different pose shapes and identifying the actions with different speeds in one goal. The entire algorithm is carried out in three stages. In the first stage, a depth sequence is divided into temporally overlapping depth segments which are used to generate three depth motion maps (DMMs), capturing the shape and motion cues. To cope with speed variations in actions, multiple frame lengths of depth segments are utilized, leading to a multi-temporal DMMs representation. In the second stage, all the DMMs are first partitioned into dense patches. Then, the local binary patterns (LBP) descriptor is exploited to characterize local rotation invariant texture information in those patches. In the third stage, the Fisher kernel is employed to encode the patch descriptors for a compact feature representation, which is fed into a kernel-based extreme learning machine classifier. Extensive experiments on the public MSRAction3D, MSRGesture3D and DHA datasets show that our proposed method outperforms state-of-the-art approaches for depth-based action recognition.

1 Introduction

Action recognition plays a significant role in a number of computer vision applications such as context-based video retrieval, human-computer interaction and intelligent surveillance systems, e.g., [Chen et al., 2014a; 2014b; Bloom et al., 2012]. Previous works focus on recognizing actions captured by conventional RGB video cameras, e.g., [Wang and Schmid, 2013]. Based on compact local descriptors, state-of-the-art results have been achieved on benchmark RGB action datasets. However, these works suffer from several common problems such as various lighting conditions and cluttered backgrounds, due to the limitations of conventional RGB video cameras.

Recent progresses witness the change of action recognition from conventional RGB cameras to depth cameras. Compared with RGB cameras, depth cameras have several advantages: 1) depth data is more robust to changes in lighting conditions and depth cameras can even work in dark environment; 2) color and texture are ignored in depth images, which makes the tasks of human detection and foreground extraction from cluttered backgrounds easier [Yang and Tian, 2014]; 3) depth cameras provide depth images with appropriate resolution and accuracy, which capture the 3D structure information of subjects/objects in the scene [Ni et al., 2011]; 4) human skeleton information (e.g., 3D joints positions and rotation angles) can be efficiently estimated from depth images providing additional information for action recognition [Shotton et al., 2011].

Since the release of cost-effective depth cameras (in particular Microsoft Kinect), more recent works on action recognition have been conducted using depth images. Various representations of depth sequences have been explored including bag of 3D points [Li et al., 2010], spatio-temporal depth cuboid [Xia and Aggarwal, 2013], depth motion maps (DMMs) [Yang et al., 2012; Chen et al., 2013; 2015], surface normals [Oreifej and Liu, 2013; Yang and Tian, 2014] and skeleton joints [Vemulapalli et al., 2014]. Among those, DMMs-based representations effectively transform the action recognition problem from 3D to 2D and have been successfully applied to depth-based action recognition. Specifically, DMMs [Yang et al., 2012] are obtained by projecting the depth frames onto three orthogonal Cartesian planes and accumulating the difference between projected maps over the entire sequence. They can be used to describe the shape and motion cues of a depth action sequence.

Motivation and contributions However, DMMs based on an entire depth sequence may not be able to capture detailed temporal motion in a subset of depth images. Old motion his-
In the field of 3D object retrieval, surface normal vectors can efficiently reflect local shapes of 3D objects [Tang et al., 2013]. By extending the same concept to temporal dimension, [Oreifej and Liu, 2013] described a depth sequence by a Histogram of Oriented Normal vectors in the 4D space of depth, spatial coordinates and time (HON4D). To increase the descriptive power of HON4D, [Rahmani et al., 2014a] characterized each 3D point by encoding Histogram of Oriented Principal Components (HOPC) within a volume around that point, which is more informative than HON4D as it captures the spread of data in three principal directions. To alleviate the loss of information in quantization part of constructing HON4D, [Kong et al., 2015] adopted the concept of surface normal and proposed kernel descriptors to convert pixel-level 3D gradient into patch-level features. Rather than describing a depth sequence by using surface normal vectors, [Rahmani et al., 2015] divided a depth sequence into equally spatio-temporal cells, which were represented by a Histogram of Oriented 3D Gradients (HOG3D) and encoded by locality-constrained linear coding. [Lu et al., 2014] developed a binary descriptor by conducting $\tau$ test to encode relative depth relationships among pairwise 3D points. [Chen et al., 2016] presented a weighted fusion framework of combining 2D and 3D auto-correlation of gradients features from depth images for action recognition. [Zhang and Tian, 2015] proposed an effective descriptor, the Histogram of 3D Facets (H3DF), to explicitly encode the 3D shape and structures of various depth images by coding and pooling 3D Facets from depth images.

Estimating skeleton joints from depth images [Shotton et al., 2011] provides a more intuitive way to perceive human actions. Existing skeleton-based approaches can be broadly grouped into joint-based and body part-based approaches. [Wang et al., 2014] selected an informative subset of joints for one specific action type, and extracted pairwise relative position features to represent each selected joint. Instead of using joint locations as features, [Vemulapalli et al., 2014] represented skeletons as points in the Lie group $SE(3) \times \ldots \times SE(3)$, which explicitly models the 3D geometric relationships among human body parts.

Obviously, skeleton joints only reflect the state of human bodies, therefore skeleton-based methods gain limited recognition rates in human object interaction scenarios. To improve the recognition performance using skeleton joints, [Wang et al., 2014] proposed an ensemble model which associates local occupancy pattern features from depth images with skeleton joints. [Ohn-Bar and M. Trivedi, 2013] utilized joint angles pairwise similarities to represent skeletons and extracted HOG features involving depth information around joints. These two can be considered as representatives of combining skeleton joints and depth information. Although multimodal fusion methods generally achieve good recognition performance, running a depth descriptor on top of a complicated skeleton tracker makes such algorithms computationally expensive, limiting their use in real-time applications.

**2 Related Work**

In this section, we briefly review recent methods on action recognition using depth information, which can be broadly categorized into depth images-based, skeleton-based, and depth and skeleton fusion-based methods. A comprehensive review on action recognition from 3D data is provided in [Aggarwal and Lu, 2014].
3 Proposed Depth Video Representation

3.1 Multi-temporal Depth Motion Maps

According to [Chen et al., 2013], the DMMs of a depth sequence with \( N \) frames are computed as follows:

\[
DMM_{(f,s,t)} = \sum_{i=2}^{N} |map_i^{(f,s,t)} - map_{i-1}^{(f,s,t)}| \tag{1}
\]

where \( map^f \), \( map^s \), and \( map^t \) indicate three projected maps of the \( i \)-th depth frame on three orthogonal Cartesian planes corresponding to the front view (f), side view (s) and top view (t). As mentioned before, the DMMs based on the entire depth sequence may not be able to capture the detailed motion cues. Therefore, to overcome this shortcoming, we divide a depth sequence into a set of overlapping 3D depth segments with equal number of frames (i.e., same frame length for each depth segment) and compute three DMMs for each depth segment. Since different people may perform an action in different speeds, we further employ multiple frame lengths to represent multiple temporal resolutions to cope with action speed variations. The proposed multi-temporal DMMs representation framework is shown in Fig. 2. Take this figure as an example, generating DMMs using the entire depth sequence (i.e., all the frames in the sequence) is considered as a default level of temporal resolution (denoted by Level 0 in Fig. 2). In the second level (Level 1 in Fig. 2), the frame length (\( L_1 \)) of a depth segment is set to 5 (i.e., 5 frames in a depth segment). In the third level (Level 2 in Fig. 2), the frame length (\( L_2 \)) of a depth segment is set to 10. Note that \( L_1 \) and \( L_2 \) can be changed. Obviously, the computational complexity increases with the increase of temporal levels. Therefore, we limit the maximum number of levels to be 3 including a default level, i.e., Level 0 which considers all the frames. The frame interval (\( R, R < L_1 \) and \( R < L_2 \)) in Fig. 2 is the number of frames between the first frames (or the starting frames) respectively in two neighboring depth segments, indicating how much overlapping between the two segments. For simplicity, we use the same \( R \) in Level 1 and Level 2.

3.2 Patch-based LBP Features

DMMs can effectively capture the shape and motion cues of a depth sequence. However, DMMs are pixel-level features. To enhance the discriminative power of DMMs, we adopt the patch-based LBP feature extraction approach in [Chen et al., 2015] to characterize the rich texture information (e.g., edges, contours, etc.) in the LBP coded DMMs. Fig. 3 shows the process of patch-based LBP feature extraction. The overlap between two patches is controlled by the pixel shift (\( ps \)) illustrated in Fig. 3. Under each projection view, a set of patch-based LBP histogram features are generated to describe the corresponding multi-temporal DMMs. Therefore, three feature matrices \( H_f \), \( H_s \), and \( H_t \) are generated associated with front view DMMs, side view DMMs and top view DMMs, respectively. Each column of the feature matrix (e.g., \( H_f \)) is a histogram feature vector of a local patch.

3.3 A Fisher Kernel Representation

Fisher kernel representation [Perronnin et al., 2010] is an effective patch aggregation mechanism to characterize a set of low-level features, which shows superior performance over the popular Bag-of-Visual-Words (BoVW) model. Therefore, we employ the Fisher kernel to build a compact and descriptive representation of the patch-based LBP features.

Let \( H = \{h_i \in \mathbb{R}^D, 1 \leq i \leq M \} \) be a set of \( M \) \( D \)-dimensional patch-based LBP feature vectors extracted from the multi-temporal DMMs of a particular projection view (e.g., front view) for a depth sequence. By assuming statistical independence, \( H \) can be modeled by a \( K \)-component Gaussian mixture model (GMM):

\[
p(H|\theta) = \prod_{i=1}^{M} \sum_{k=1}^{K} \omega_k N(h_i|\mu_k, \Sigma_k), \tag{2}
\]
where \( \theta = \{ \omega_k, \mu_k, \Sigma_k \}, k = 1, \ldots, K \) is the parameter set with mixing parameters \( \omega_k \), means \( \mu_k \) and diagonal covariance matrices \( \Sigma_k \) with the variance vector \( \sigma_k^2 \). These GMM parameters can be estimated by using the Expectation-Maximization (EM) algorithm based on a training dataset (or feature set).

Two \( D \)-dimensional gradients with respect to the mean vector \( \mu_k \) and standard deviation \( \sigma_k \) of the \( k \)th Gaussian component are defined as

\[
\rho_k = \frac{1}{M \sqrt{\pi_k}} \sum_{i=1}^{M} \frac{h_i - \mu_k}{\sigma_k},
\]

\[
\tau_k = \frac{1}{M \sqrt{2\pi_k}} \sum_{i=1}^{M} \frac{\gamma_{k,i}}{\sigma_k^2} \left( \frac{h_i - \mu_k}{\sigma_k} \right)^2 - 1, \tag{3}
\]

where \( \gamma_{k,i} \) is the posterior probability that \( q_i \) belongs to the \( k \)th Gaussian component. The Fisher vector (FV) of \( H \) is represented as \( \Phi(H) = (\rho_1^T, \tau_1^T, \ldots, \rho_K^T, \tau_K^T)^T \). The dimensionality of the FV is \( 2KD \).

A power-normalisation step introduced in [Perronnin et al., 2010], i.e., signed square rooting (SSR) and \( \ell_2 \) normalization, is applied to eliminate the sparseness of the FV as follows:

\[
T(\Phi(H)) = \text{sgn}(\Phi(H)) * |\Phi(H)|^{\alpha}, \quad 0 < \alpha \leq 1. \tag{4}
\]

Let \( H_f \), \( H_s \) and \( H_t \) denote three sets of patch-based LBP feature vectors from three projection views, each depth sequence is then represented by concatenating three FVs \( \{ \Phi(H_f); \Phi(H_s); \Phi(H_t) \} \) as the final feature representation.

4 Experiments

In this section we extensively evaluate our proposed method on three public benchmark datasets: MSRAction3D [Li et al., 2010], MSRGesture3D [Wang et al., 2012] and DHA [Lin et al., 2012]. We employ kernel-based extreme learning machine (KELM) [Huang et al., 2006] with a radial basis function (RBF) kernel as the classifier due to its general good classification performance and efficient computation.

4.1 Datasets

MSRAction3D dataset [Li et al., 2010] is one of the most popular depth datasets for action recognition as reported in the literature. It contains 20 actions: “high arm wave”, “horizontal arm wave”, “hammer”, “hand catch”, “forward punch”, “high throw”, “draw x”, “draw tick”, “draw circle”, “hand clap”, “two hand wave”, “side boxing”, “bend”, “forward kick”, “side kick”, “jogging”, “tennis swing”, “tennis serve”, “golf swing”, “pick up & throw”. Each action is performed 2 or 3 times by 10 subjects facing the depth camera. It is a challenging dataset due to similarity of actions and large speed variations in actions.

MSRGesture3D dataset [Wang et al., 2012] is a benchmark dataset for depth-based hand gesture recognition. It consists of 12 gestures defined by American Sign Language: “bathroom”, “blue”, “finish”, “green”, “hungry”, “milk”, “past”, “pig”, “store”, “where”, “i”, “x”. Each action is performed 2 or 3 times by each subject, resulting in 336 depth sequences.


4.2 Experimental Settings

Several action snaps from the three datasets are shown in Figs. 4-6, where inter-similarity among different types of actions is observed. In the MSRAction3D dataset, actions such as “drawX” and “drawTick” are similar except for a slight difference in the movement of one hand. In the MSRGesture3D dataset, actions such as “milk” and “hungry” are alike, since both actions involve the motion of bending palm. What’s more, self-occlusion is also a challenge for this dataset. In the DHA dataset, “golf-swing” and “rod-swing” actions share similar motions by moving hands from one side up to the other side. More similar pairs can be found in “leg-curl” and “leg-kick”, “run” and “walk”, etc.

In order to keep the reported results consistent with other works, we follow the same evaluation protocols in [Wang et al., 2014], [Wang et al., 2012] and [Lin et al., 2012] respectively for the three datasets.

We adopt the same parameter values in [Chen et al., 2015] for the patch sizes and parameters for the LBP operator in our method. The other parameters are determined empirically. The overall accuracies on three datasets with different parameters are shown in Figure 7, where frame length \( L_1 \), frame length \( L_2 \), frame interval \( R \), pixel shift \( ps \) and the number of Gaussians \( K \) respectively change from 3 to 11, 10 to 18, 1
Figure 7: Recognition accuracies with changing parameters.

Table 1: Recognition accuracy and average feature computation time of our method with different numbers of temporal levels on the MSRAction3D dataset.

<table>
<thead>
<tr>
<th>Temporal levels</th>
<th>Accuracy</th>
<th>Time/sequence (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 level (Level 0)</td>
<td>89.95%</td>
<td>0.35</td>
</tr>
<tr>
<td>2 levels (Levels 0, 1)</td>
<td>93.34%</td>
<td>2.51</td>
</tr>
<tr>
<td>3 levels (Levels 0, 1, 2)</td>
<td>95.97%</td>
<td>4.49</td>
</tr>
</tbody>
</table>

to 5, 3 to 7 and 20 to 100 at equal intervals. Experiments are conducted with one parameter changes and the other parameters in default values: $L_1 = 7$, $L_2 = 14$, $R = 3$, $ps = 5$ and $K = 60$. From Figure 7, we can see that more than 90% accuracies are achieved with different parameters, which reflect the robustness of our method to parameter settings. Since default parameters work well for all three datasets, the following experiments are conducted with these values in default.

In our method, we use three levels for the multi-temporal DMMs representation. We test the algorithm on the MSRAction3D dataset using different numbers of temporal levels. The recognition accuracy and average feature computation time are reported in Table 1. It is worth mentioning that our algorithm is implemented in MATLAB and executed on CPU platform with an Intel(R)Core(TM)i7 CPU @2.60GHz and 8GB of RAM. It can easily gain the efficiency by converting the code to C++ and running the multi-temporal DMMs representation in parallel.

4.3 Results on the MSRAction3D Dataset

In Figure 8 (a), we show the confusion matrix of the MSRAction3D dataset with the accuracy of 95.97%. It is observed that large ambiguities exist between similar action pairs, for example “handCatch” and “highThrow”, and “drawX” and “drawTick”, due to the similarities of their DMMs. We also compare our method with the state-of-the-art methods in Table 2. “Moving Pose” [Zanfir et al., 2013], “Skeletons in a Lie group” [Vemulapalli et al., 2014] and “Skeletal Quads” [Evangelidis et al., 2014] belong to skeleton-based features, and “Actionlet Ensemble” [Wang et al., 2014] belongs to skeleton+depth based features. Our method outperforms these methods for two reasons: first, skeleton joints used by these methods contain a lot of noises, which bring ambiguities to distinguish similar actions; second, our method directly using DMMs, which provide more affluent motion information. Our result is also better than the recent depth-based features such as Super Normal Vector [Yang and Tian, 2014] and Range-Sample [Lu et al., 2014], which demonstrates the superior discriminatory power of our multi-temporal DMMs representation.

4.4 Results on the MSRGesture3D Dataset

In Figure 8 (b), we show the confusion matrix of the MSRGesture3D dataset with the accuracy of 98.19%. It is observed
that similar action pairs like “milk” and “hungry” can be distinguished with high accuracy. We compare our method with several existing methods in Table 3. As can be seen from this table, our method outperforms Histogram of Oriented Principal Components (HOPC) [Rahmani et al., 2014a] by 1.96%, leading to a new state-of-the-art result.

4.5 Results on the DHA Dataset

In Figure 8 (c), we present the confusion matrix of our method on the DHA dataset. The DHA dataset is originally collected by [Lin et al., 2012], which only contains 17 action categories. We use an extended version of the DHA dataset where extra 6 action categories are involved. [Lin et al., 2012] split depth sequences into space-time volume and constructed 3bit binary patterns as depth features, which achieved an accuracy of 86.80% on the original dataset. By incorporating multi-temporal information to the DMMs, our proposed method achieves higher accuracy even on the extended DHA dataset. In Table 4, we observe that our method outperforms D-DMHI-PHOG [Gao et al., 2015] by 3.04% and outperforms DMPP-PHOG [Gao et al., 2015] by 0.44%. These improvements show that operating LBP on multi-temporal DMMs can produce more informative features than operating PHOG on depth difference motion history image (D-MHI).

4.6 Execution Rate and Frame Rate Invariance

Regarding to the execution rate invariance, we have calculated the statistics for the MSRAction3D dataset, which contains the actions executed by different subjects with different execution rates. To be more precise, there are 20 actions, each being executed by 10 subjects for 2 or 3 times. The standard derivation of the sequence lengths (numbers of frames) across the actions is 9.21 frames (max: 13.30 frames; min: 4.86 frames), which means that execution rate difference is actually quite large. In view of the achieved 95.97% recognition rate, we would say that our algorithm is resistant to the execution rate.

To test the effect by frame rate difference, we carry out an experiment using the MSRAction3D dataset. Specifically, we use the sequences performed by subjects 1, 3, 5, 7, 9, (the original action samples) as training data. We select half number of frames (odd numbers of frames, e.g., 1, 3, 5 ...) of the sequences performed by subjects 2, 4, 6, 8, 10 to form a set of new testing samples with 1/2 of the original frame rate. The achieved recognition result of our method is 93.27%. Therefore, our proposed algorithm is capable of dealing with frame rate changes considering the fact that 1/2 frame rate reduction is actually unrealistic.

5 Conclusion

This paper presents an effective feature representation for action recognition from depth sequences. A multi-temporal DMMs representation is proposed to capture more temporal motion information in depth sequences for better distinguishing similar actions. Multiple temporal resolutions in the proposed representation can also cope with the speed variations in actions. Patch-based LBP features are extracted from dense patches in the DMMs and the Fisher kernel representation is utilized to aggregate local patch features into a compact and discriminative representation. The proposed method is extensively evaluated on three benchmark datasets. Experimental results show that our method outperforms the state-of-the-art methods in all datasets.

References


[Chen et al., 2014a] Chen Chen, Nasser Kehtarnavaz, and Roozbehz Jafari. A medication adherence monitoring system for pill bottles


