How Good Is Kernel Descriptor on Depth Motion Map for Action Recognition

Thanh-Hai Tran[1](✉) and Van-Toi Nguyen[1,2,3]

[1] International Research Institute MICA, HUST-CNRS/UMI-2954-INP Grenoble,
Hanoi, Vietnam
{thanh-hai.tran,van-toi.nguyen}@mica.edu.vn
[2] L3i Laboratory, University of La Rochelle, La Rochelle, France
[3] University of Information and Communication Technology Under
Thai Nguyen University, Thai Nguyen, Vietnam

Abstract. This paper presents a new method for action recognition using depth data. Each depth sequence is represented by depth motion maps from three projection views (front, side and top) to exploit different aspects of the motion. However, different from state of the art works extracting local binary pattern or histogram of oriented gradients, we describe an action based on gradient kernel descriptor. The proposed method is evaluated on two benchmark datasets (MSRAction3D and MSRGestures3D) and obtains very competitive performances with the best state of the arts methods. Our best recognition rate is 91.57 % on MSRAction3D and 100 % on MSRGestures3D dataset whereas [1] achieved 93.77 % and 94.60 % respectively.

Keywords: Action recognition · Depth motion map · Kernel descriptor

1 Introduction

Action recognition is an active topic in computer vision because of its wide range of practical applications, more specifically, home abnormal activity, sport activity, human gestures, human interaction, pedestrian traffic, healthcare, gaming. Research on human action recognition initially employed video sequences provided by conventional RGB camera. With the development of new and low-cost depth sensors such as Microsoft Kinect, new opportunities for action recognition have emerged.

Kinect sensor provides multi-modal data for processing such as RGB, Depth, Skeleton. RGB data is strongly affected by illumination changing. Skeleton is usually computed from a long training on a very large data [2]. Sometimes, the skeleton is not available or not precise due to the (self-)occusion of the human. As a result, conventional approaches based on color information could not perform well. Currently, numerous approaches for action recognition usually exploit the depth data [3] with different aspects: point cloud, surface normals, etc.

In this paper, we propose a novel method based upon depth motion map and kernel descriptor. Depth motion map (DMM) is a technique to compress depth

L. Nalpantidis et al. (Eds.): ICVS 2015, LNCS 9163, pp. 137–146, 2015.
DOI: 10.1007/978-3-319-20904-3_13

sequence into one map representing the motion history of the action. It has been applied successfully in [4] and [1]. However, instead of extracting histogram of oriented gradients (HOG) in [4] or local binary pattern (LBP) in [1], we use a new gradient descriptor based on kernel.

Kernel descriptor has been initially introduced by [5] for general visual recognition problem. Kernel descriptor provides an unified framework to define different descriptors such as SIFT, HOG, LBP. Kernel descriptor computed on RGB images has been shown to be one of the best descriptors for object recognition on several public datasets. However, original kernel descriptor has some limitations that is it is not invariant to rotation and scale changes. In addition, it has never been proved on motion depth data.

In this paper, we improve the original kernel descriptor in [5] to make it more robust to scaling and rotation. We then study on how the proposed kernel descriptor is good on depth motion maps for action recognition. The proposed method is extensively evaluated with different configurations of machine learning techniques such as Support Vector Machine (SVM) and Kernel based Extreme Machine Learning (KEML) on each projection view of the motion map. The experiments show that our method outperforms state of the art works on MSRGesture3D dataset until now and obtains comparable results on MSRAction3D dataset in term of accuracy (Table 3).

2 Related Works

Human action recognition has been mentioned since more than twenty years ago. There are many methods that have been proposed to aim this goal [6]. In the section, we are not ambitious to update the survey but we focus on methods that employ depth data for action representation and recognition.

In [4], the authors proposed to represent the depth sequence by depth motion map. To make use of the additional body shape and motion information from depth maps, each depth frame is projected onto three orthogonal Cartesian planes. Then region of interest (ROI) corresponding to the bounding box of the human is extracted and normalized to a fixed size to avoid the intra-class variation. Then HOG feature is computed on the ROI which is the input to a linear SVM classifier for human action recognition. Experiments have been done with MSRAction3D dataset. The accuracy is computed with different sub-sets of data. The method achieves the best result (96.2 %) on the third subset with cross validation.

Inspired from the idea of Spatio temporal Interest Point (STIP) computed on RGB sequence, L. Xia and J.K.Aggarwal extended to depth data by extracting STIPs on each depth map of the sequence (so called DSTIP) [7]. Then they built a depth cuboid similarity feature (DCSF) to describe the local 3D depth cuboid around the DSTIPs with an adaptable supporting size. To model an action, Bag of Word (BoW) model was employed. Each action sequence is represented by a distribution of code-words computed on all depth maps of the sequence. Finally, SVM with histogram intersection kernel is applied for classification. This method has been tested on two public datasets (MSRAction3D and MSRActivity3D) and obtained 89.3 % and 88.2 % respectively in term of accuracy with a half data for training and the rest for testing.

In [8], the authors claimed that the existing features for action representation are usually based on shape or motion independently. These features fail to capture the complex joint shape-motion cues at pixel-level. Therefore, in the paper, the authors consider the depth sequence as a function from R^3 (spatial coordinates, time) to R^1 (depth) that constitutes a surface in 4D space (time, depth and spatial coordinates). They then proposed to describe depth sequence by a histogram capturing the distribution of surface normal orientations in 4D space (HON4D). Following the author, HON4Ds capture richer information than 3D gradient orientation (HOG3D) [4] therefore the representation is more discriminant. The proposed method has been evaluated on three public datasets (MSRAction3D, MSRGesture3D, MSRDailyActivity3D). The best recognition rate on MSRAction3D is 88.89 % while the best on MSRGesture3D is 92.45 %.

In [9], the authors in [4] proposed a new method for human action recognition which based on the polynormal which is a group of hypersurface normals in depth sequences. For representing a depth video, firstly, the depth video is subdivided into a set of spate-time grids. An adaptive spatio-temporal pyramid is proposed to capture the spatial layout and temporal order in a global way. Then they concatenate the vector extracted from all the space-time grids as the final representation of super normal vector (SNV). The method has been tested on four datasets (MSRAction3D, MSRGesture3D, MSRActionPairs, MSRDaily-Activity3D) and shown to ourperform all published works at that time (93.09 %, 94.72 %, 98.89 %, 86.25 % respectively).

Currently, L. Bo *et al.* have introduced kernel descriptor for visual recognition problem [5] that shown to be the best descriptor for visual recognition on some challenging datasets. In this paper, we improved kernel descriptor to be more robust to scaling and rotation. We would like to investigate how the improved kernel descriptor is good for action recognition based on depth motion map.

3 Proposed Approach

3.1 General Framework

We propose a framework for action recognition which composes of three main steps (Fig. 1):

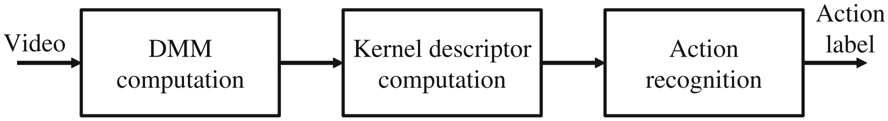

Fig. 1. Main steps of action recognition

- Motion representation: Given a video, we compute depth motion map from three projection views (front, side, top).
- Action modeling: For each depth motion map, we compute gradient based kernel descriptor to output the final feature vector.
- Action recognition: The feature vector inputs to a multiclass classifier (SVM, KEML) to decide the class that the action belongs to. At this step, two fusion solutions (feature level fusion and classifier level fusions) will be studied.

In the next sections, we will describe in detail each step of the framework.

3.2 Depth Motion Map

Depth Motion Map was firstly introduced in [4]. Given a sequence of N depth maps $D_1, D_2, ..., D_N$, the depth motion map is defined as follows:

$$DMM = \sum_{i=1}^{N-1} (|D^{i+1} - D^{i}| > \epsilon) \tag{1}$$

where ϵ is a threshold to make binary the difference between two consecutive maps D^{i+1} and D^{i}. The binary map of motion energy indicates motion regions or where movement happens in each temporal interval. So the DMM represents sum of motions through entire video sequences.

Different from [4], in [1], the authors modified the procedure to obtain DMM. Specifically, instead of computing the sum on binary maps, [1] take the absolute difference:

$$DMM = \sum_{i=1}^{N-1} |D^{i+1} - D^{i}| \tag{2}$$

In [4], the authors proposed to project depth frames onto three orthogonal Cartesian planes to characterize the motion of an action. Specifically, each 3D depth frame is used to generate three 2D projected maps corresponding to front, side and top views, denoted by D_f, D_s, D_t respectively. By this way, we obtain three DMMs corresponding to three views.

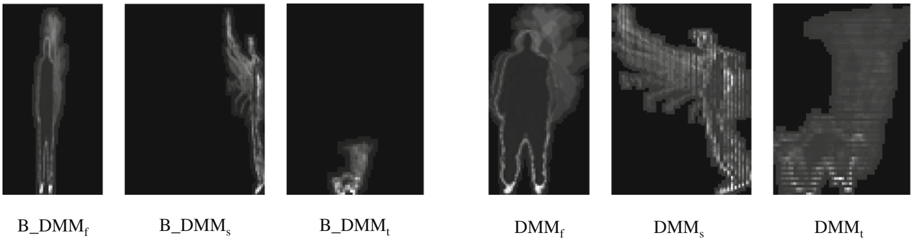

Fig. 2. Three DMMs computed from front, side, top projection views of depth sequence

We apply also a bounding box to extract the non-zero region as the foreground in each DMM. Figure 2 shows three DMMs computed following (1) (we call B_DMM with B means Binary) and (2) respectively from an action sequence in MSRAction3D dataset. Obviously, we see that (2) gets richer motion information than (1). We have tested both procedures of computing DMM and found that the binary DMM gives worse performance. Therefore, in the following, we will use DDM computed according to (2).

3.3 Gradient Based Kernel Descriptor

Kernel descriptor was initially introduced by [5]. This method for object representation has been shown to outperform all state of the art descriptors on several published datasets.

When working with the original kernel descriptor presented in [5], we observe some problems. Firstly, the gradient based kernel considers the current gradient vector of a pixel on the patch, it is therefore not invariant to rotation. In addition, the size of patch is fixed for all images. As consequent, the description is not invariant to scale change.

We have studied deeply on kernel descriptor and propose two improvements to make the original kernel descriptor more robust to rotation and scale changes. The computation of kernel descriptor is presented in Fig. 3. It comprises three main steps:

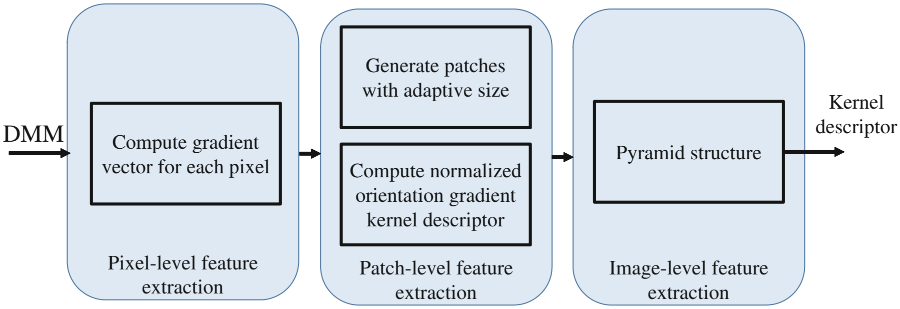
DMM
Compute gradient vector for each pixel
Pixel-level feature extraction
Generate patches with adaptive size
Compute normalized orientation gradient kernel descriptor
Patch-level feature extraction
Pyramid structure
Image-level feature extraction
Kernel descriptor

Fig. 3. Computation of kernel descriptor

- Pixel-level feature extraction: At this level, a normalized gradient vector is computed for each pixel of the image. The normalized gradient vector at a pixel z is defined by its magnitude $m(z)$ and normalized orientation $\omega(z) = \theta(z) - \overline{\theta}(P)$, where $\theta(z)$ is orientation of gradient vector at the pixel z, and $\overline{\theta}(P)$ is the dominant orientation of the patch P that is the vector sum of all the gradient vectors in the patch. This normalization will make patch-level features invariant to rotation. In practice, the normalized orientation of a gradient vector will be $\widetilde{\omega}(z) = [sin(\omega(z))\ cos(\omega(z))]$. Note that in the original kernel descriptor proposed in [5], the gradient orientation was not normalized.

- Patch-level feature extraction: A set of patches is generated with adaptive size. The size of the patch is directly proportional to the size of the image. This adaptive size ensures the number of patches to be considered unchanged. This adaptive patch makes the patch descriptor more robust to scale change. Note that in [5], the patch size was fixed for all images. Therefore the number of generated patches from two images was very different that leads to two different representations of the same scene observed at different two scales. Roughly speaking, such representation was not invariant to scale change.
  For each patch, we compute a patch feature based on a given definition of match kernel. The gradient match kernel is constructed from three kernels that are gradient magnitude kernel $k_{\widetilde{m}}$, orientation kernel k_o and position kernel k_p.

$$K_{gradient}(P, Q) = \sum_{z \in P} \sum_{z' \in Q} k_{\widetilde{m}}(z, z') k_o(\widetilde{\omega}(z), \widetilde{\omega}(z')) k_p(z, z') \quad (3)$$

  where P and Q are patches of two different images that we need to measure the similarity. z and z' denote the 2D position of a pixel in the image patch P and Q respectively. Let $\varphi_o(.)$ and $\varphi_p(.)$ the feature maps for the gradient orientation kernel k_o and position kernel k_p respectively. Then, the approximate feature over image patch P is constructed as:

$$\overline{F}_{gradient}(P) = \sum_{z \in P} \widetilde{m}(z) \phi_o(\widetilde{\omega}(z)) \otimes \phi_p(z) \quad (4)$$

  where $\otimes$ is the Kronecker product, $\phi_o(\widetilde{\omega}(z))$ and $\phi_p(z)$ are approximate feature maps for the kernel k_o and k_p, respectively. The approximate feature maps are computed based on a basic method of kernel descriptor. The basic idea of representation based on kernel methods is to compute the approximate explicit feature map for kernel match function [5].
- Image-level feature extraction: At this step, as in [5], a pyramid structure is used to combine patch features. Given an image, the final representation is built based on features extracted from lower levels using efficient match kernels (EMK). First, the feature vector for each cell of the pyramid structure is computed. The final descriptor is the concatenation of feature vectors of all cells.
  Let C be a cell that has a set of patch-level features $X = \{x_1, ..., x_p\}$ then the feature map on this set of vectors is defined as:

$$\overline{\phi}_S(X) = \frac{1}{|X|} \sum_{x \in X} \phi(x) \quad (5)$$

  Where $\phi(x)$ is approximate feature maps for the kernel $k(x, y)$. The feature vector on the set of patches, $\overline{\phi}_S(X)$, is extracted explicitly.
  Given an image, let L be the number of spatial layers to be considered. In our experiment $L = 3$. The number of cells in layer l-th is (n_l). $X(l, t)$ is set of patch-level features falling within the spatial cell (l, t) (cell t-th in the l-th

level). A patch is fallen in a cell when its centroid belongs to the cell. The feature map on the pyramid structure is:

$$\overline{\phi}_P(X) = [w^{(1)}\overline{\phi}_S(X^{(1,1)}); ...; w^{(l)}\overline{\phi}_S(X^{(l,t)}); ...; w^{(L)}\overline{\phi}_S(X^{(L,n_L)})] \tag{6}$$

In (6), $w^{(l)} = \frac{\frac{1}{n_l}}{\sum_{l=1}^{L}\frac{1}{n_l}}$ is the weight associated with level l.

3.4 Action Classification

Once kernel descriptor is computed, the classification could be simplified by a linear classifier. In this paper, we use multi-class SVM classifier. However, to compare the efficiency of the descriptors, we employ also KELM method as in [1]. The input of these classifier is the action descriptor vector that is computed in the previous steps.

Feature Level Fusion. As we consider three project views of the depth map, we obtain three depth motion maps corresponding to front, side and top view. A straightforward solution to combine these information is to concatenate kernel descriptors computed from three views to make the final representation of the action sequence.

Decision Level Fusion. The second solution is to build three independent classifiers for each descriptor and then fuse the result from three classifiers. We follow the same approach for decision fusion as presented in [1]. More specifically, the SVM/KEML classifier outputs a value $f_L(x)$ which is the distance between a given feature x and the model. This value is normalized to [0, 1] and the posterior probability is approximated using sigmoid function according to Platt's empirical analysis.

$$p(y_k|x) = \frac{1}{1 + exp(Af_L(x)_k + B)} \tag{7}$$

In our experiment, A = -1, B = 0. This probability is used to estimate a global membership function:

$$logP(y_k|x) = \sum_{q=1}^{Q} \alpha_i p_q(y_k|x) \tag{8}$$

where Q is the number of classifiers and $\{\alpha_q\}_{q=1}^{Q}$ are uniformly distributed classifier weights. The final class label $y*$ is selected as follows:

$$y* = argmaxP(y_k|x) \tag{9}$$

4 Experimental Results

We evaluate the proposed method on two published datasets: MSRAction3D [10] and MSRGesture3D [11]. Both datasets are built by depth camera.

4.1 MRSAction3D Dataset

The MSRAction3D dataset includes 20 action types realized by 10 subjects, each subject performs each action 2 or 3 times. The resolution is 320 × 240. There are 567 depth map sequences in total. However, as reported in [12], 10 sequences are not used in experiment because the skeletons are either missing or too erroneous and to be comparible with the current work, we will use only 557 sequences. The actions are: *high wave, horizontal 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 pick, jogging, tennis swing, tennis serve, golf swing, and pickup throw.*

We follow the experiment setting in [13] in which one half of the subjects (1, 3, 5, 7, 9) are used for training and the remaining are used for testing. This dataset is challenging because of the number of action classes is large while the samples for training is not numerous. To normalize the size of DMM, we follow the setting in [1]. Specifically, the size of DMM_f, DMM_s, DMM_t are 102×54, 102×75, 75×54 respectively.

4.2 MSRGesture3D Dataset

The MSRGesture3D dataset is a dynamic hand gesture dataset that contains a subset of gestures defined by American Sign Language (ASL). It includes 12 gestures: *bathroom, blue, finish, green, hungry, milk, past, pig, store, where, j, z.* The dataset comprises 333 depth sequences. We follows the same experimental setting as in [12] that uses leave-one-subject-out cross-validation. The size of DMM_f, DMM_s, DMM_t are 118×133, 118×29, 29×133.

4.3 Analysis

Different features, classifiers have been combined to make the comparison. We label feature level fusion approach as FF, decision level fusion as DF. Tables 1 and 2 show the comparative performance of the original kernel descriptor (OKD), local binary pattern (LBP) and the proposed kernel descriptor (PKD) computed on three depth motion maps. Globally, we make some conclusions:

Table 1. Comparison of recognition accuracy (%) on MSRAction3D dataset

DepthMap	OKD-SVM [5]	PKD-SVM	PKD-KELM	LPB-KELM [1]
DMM_f	79.85	83.15	**83.88**	78.75
DMM_s	71.06	73.62	**73.99**	68.13
DMM_t	67.39	72.16	**71.79**	64.10
DMM_{FF}	81.68	88.64	89.01	**91.94**
DMM_{DF}	81.97	88.65	91.57	**93.77**

Table 2. Comparison of recognition accuracy (%) on MSRGesture3D dataset

DepthMap	OKD-SVM [5]	PKD-SVM	PKD-KELM	LPB-KELM [1]
DMM_f	96.67	100	**100**	84.58
DMM_s	81.11	85.56	**93.33**	68.47
DMM_t	68.37	70.55	**76.66**	64.30
DMM_{FF}	96.66	93.34	**96.67**	93.40
DMM_{DF}	93.33	94.44	90.00	**94.60**

- The proposed kernel descriptor outperforms the original one in all tests with each projection view independently. It shows the robustness of out descriptor w.r.t scaling and rotation.
- The proposed kernel descriptor outperforms LPB features for each projection view. The use of KELM classification instead of SVM helps to improve lightly the performance. Once again, we show the efficiency of kernel descriptor on depth motion map.
- The combination of projection views according to the decision fusion solution does not improve the performance as in case of LPB descriptor. The reason for this is this the kernel descriptor gives stable performances for all classes on each project view. For MSRGesture3D dataset, the feature level fusion obtain better accuracy than the case of LPB.

Table 3. Recognition accuracy (%) on two datasets

Dataset	Best in [8]	Best in [9]	Best in [1]	Our best
MSRAction3D	88.89	93.09	93.77	91.57
MSRGesture3D	92.45	94.72	94.60	100

5 Conclusion

In this paper, we have presented a novel method for action recognition. The method compresses the video sequence into one image using Depth Motion Map technique then describes the DMM using kernel descriptor. In comparison with the original kernel descriptor [5], the proposed kernel descriptor is more robust to scaling and rotation because we have performed a normalization in gradient orientation as well as selection of adaptive patch size. Using the new descriptor help to improve significantly the classification rate on each projection view of the depth map. In the future, we will analyse in more detail how the kernel descriptor acts on each projection view and propose a new solution to efficiently fuse these information.

Acknowlegment. This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number FWO.102.2013.08.

References

1. Chen, C., Jafari, R., Kehtarnavaz, N.: Action recognition from depth sequences using depth motion maps-based local binary patterns. In: WACV, pp. 1092–1099 (2015)
2. Shotton, J., Fitzgibbon., A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from a single depth image. In: PAMI (2012)
3. Ye, M., Zhang, Q., Wang, L., Zhu, J., Yang, R., Gall, J.: A survey on human motion analysis from depth data. In: Grzegorzek, M., Theobalt, C., Koch, R., Kolb, A. (eds.) Time-of-Flight and Depth Imaging. LNCS, vol. 8200, pp. 149–187. Springer, Heidelberg (2013)
4. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: ACM Multimedia (MM), pp. 1057–1060 (2012)
5. Bo, L., Ren, X., Fox, D.: Kernel descriptors for visual recognition. In: Advances in Neural Information Processing Systems, pp. 244–252 (2010)
6. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: a survey. IEEE Trans. Circ. Syst. Video Technol. **18**(11), 1473–1488 (2008)
7. Xia, L., Aggarwal, J.K.: Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: CVPR, pp. 2834–2841 (2013)
8. Omar, O., Liu, Z.: HON4D: histogram of oriented 4D normals for: activity recognition from depth sequences. In: CVPR, pp. 716–723 (2013)
9. Yang, X., Tian, Y.: Super normal vector for activity recognition using depth sequences. In: CVPR, pp. 804–811 (2014)
10. Li, W., Zhang, Z., Liu, Y.: Action recognition based on a bag of 3D points. In: CVPR Workshop, pp. 9–14 (2010)
11. Kurakin, A., Zhang, Z., Liu, Z.: A real time system for dynamic hand gesture recognition with a depth sensor. In: EUSIPCO, pp. 1975–1979 (2012)
12. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: CVPR (2012)
13. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3D action recognition with random occupancy patterns. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 872–885. Springer, Heidelberg (2012)