Polynormal Fisher Vector for Activity Recognition from Depth Sequences

Xiaodong Yang and YingLi Tian

Department of Electrical Engineering
City College, City University of New York

1. Abstract

The advent of depth sensors has facilitated a variety of visual recognition tasks including human activity understanding. This paper presents a novel feature representation to recognize human activities from video sequences captured by a depth camera. We assemble local neighboring hypersurface normals from a depth sequence to form the polynormal which jointly encodes local motion and shape cues. Fisher vector is employed to aggregate the low-level polynormals into the Polynormal Fisher Vector. In order to capture the global spatial layout and temporal order, we employ a spatio-temporal pyramid to subdivide a depth sequence into a set of space-time cells. Polynormal Fisher Vectors from these cells are combined as the final representation of a depth video. Experimental results demonstrate that our method achieves the state-of-the-art results on the two public benchmark datasets, i.e., MSRAction3D and MSRGesture3D.

CR Categories: I.4.8 [Image Processing and Computer Vision]: Scene Analysis

Keywords: Polynormal, Fisher Vector, RGB-D Camera, Activity Recognition

2. Introduction

Human activity recognition has a number of real-world applications including telepresence, video surveillance, human-computer interaction, etc. As the imaging techniques advance, the availability of low-cost depth sensors (e.g., Microsoft Kinect) has facilitated various visual recognition tasks including object recognition, indoor place segmentation, as well as human gesture and action recognition. Compared to conventional color images in activity recognition, depth maps have the following merits: (1) additional shape cues to provide more informative geometric description; (2) precluded color and texture to ease human detection and segmentation; and (3) independence of visible lighting to benefit monitoring in dark environments. These advantages have motivated recent research to explore a set of representations of depth sequences ranging from skeleton joints, cloud points, depth projections, local depth points, to surface normals.

Additional body shape information provided by depth maps has been successfully applied to recover skeleton joints [Shotton et al. 2011]. EigenJoints [Yang and Tian 2014] employed joint differences to characterize static posture, consecutive motion, and overall dynamics. In order to reduce errors of joint estimation, the pose set in [Wang et al. 2013] selected the top joint configurations by using segmentation cues and temporal constraints. Actionlet ensemble was proposed in [Wang et al. 2012] to model a subset of skeleton joints associated with a specific action. However, the estimated skeleton joints could be quite noisy or even completely wrong if severe occlusion occurs. Compared to skeleton joints, cloud points are more robust to noise and occlusion. The local and random occupancy patterns proposed in [Wang et al. 2012; Wang et al. 2012] represented depth appearance by counting the number of cloud points falling into a local spatio-temporal grid.

Depth projection transforms the recognition problem from 3D to 2D. Li et al. [Li et al. 2010] sampled points from silhouettes of projected depth maps on three orthogonal planes and employed an action graph to model action dynamics. The depth motion maps [Yang et al. 2012] stacked differences between consecutive projected depth maps from three orthogonal views and computed HOG from the stacked motion maps. Following the low-level feature extraction in traditional videos, several local interest point detectors specifically designed for depth videos were recently proposed. Hadfield and Bowden [Hadfield and Bowden 2013] extended the algorithms of Harris and Hessian detectors and separable filters to 3.5D and 4D space in depth sequences. DSTIP introduced in [Xia and Aggarwal 2013] localized more robust and stable motion-related foreground interest points through suppressing the frequent flip noise in depth sequences.

It was recently shown that surface normals can provide more informative geometric cues of an object in 3D. Oreifej and Liu [Oreifej and Liu 2013] proceeded along with this observation to impose the temporal gradient on surface normals in a depth sequence. The extended normal was quantized by regular and discriminative learned polychorons. Our Polynormal Fisher Vector (PFV) also follows this direction. It is based on the polynormal proposed in [Yang and Tian 2014] which clustered the local neighboring extended surface normal. We employ Fisher vector to aggregate the low-level polynormals and spatio-temporal pyramid to globally capture the spatial layout and temporal order. We concatenate the feature vectors from all the space-time cells as the final representation of a depth sequence.

2. Polynormal

To make this paper more self-contained, we briefly introduce the concept of polynormal in this section. A normal to a surface in 3-dimensional space can be extended to a hypersurface in m-dimensional space. We define a hypersurface in m-dimensional space as a function $\mathbb{R}^{m-1} \rightarrow \mathbb{R}^1: x_m = f(x_1, \ldots, x_{m-1})$, which describes a set of local points satisfying $F(x_1, \ldots, x_m) = f(x_1, \ldots, x_{m-1}) - x_m = 0$. A normal vector to the hypersurface at a m-dimensional point can be computed by the gradient $\nabla F(x_1, \ldots, x_m) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_{m-1}}, -1\right)$. In a depth sequence, the hypersurface is with $m = 4$ dimensions, i.e., each cloud point

e-mail: {xyang02, ytian}@ccny.cuny.edu

satisfies $F(x,y,t,z) = f(x,y,t) - z = 0$. So we define the hypersurface normal in a depth sequence as

$$\boldsymbol{n} = \nabla F = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial t}, -1\right)^T. \quad (1)$$

It was recently shown in [Oreifej and Liu 2013] that the distribution of normal orientations is more informative in terms of describing object shapes than the distribution of gradient orientations. In addition to the geometric cues, the motion information is also incorporated in the normal vector of Eq. (1). Polynormal is a cluster of normals from a local hypersurface. This is motivated by the jointly encoded spatial neighborhood of low-level features in macrofeatures [Boureau et al. 2010]. Compared to each individual normal, polynormal is more robust to noise and retains the correlation between neighboring normals.

We concatenate the hypersurface normals from the local neighborhood $\mathcal{L}$ of a cloud point as the polynormal associated with this point, i.e., $\boldsymbol{p} = \left(\boldsymbol{n}_1^T, \dots, \boldsymbol{n}_{|\mathcal{L}|}^T\right)^T, \boldsymbol{n}_1, \dots, \boldsymbol{n}_{|\mathcal{L}|} \in \mathcal{L}$. The local neighborhood $\mathcal{L}$ is a spatio-temporal subvolume determined by $\mathcal{L}_x \times \mathcal{L}_y \times \mathcal{L}_t$, where $\mathcal{L}_x$, $\mathcal{L}_y$, and $\mathcal{L}_t$ denote the number of neighboring points in x, y, and t axes, respectively.

3. Depth Video Representation

We employ PFV combined with a spatio-temporal pyramid to represent depth videos. Fisher vector has been successfully applied in the large-scale image classification and retrieval [Perronnin et al. 2010; Sanchez et al. 2013]. Here we employ Fisher vector to aggregate polynormals based on the Fisher kernel which shares the benefits of both generative and discriminative models. Each polynormal is represented by its deviation with respect to the parameters of a generative model. The spatio-temporal pyramid is then used to globally capture the spatial layout and temporal orders.

3.1 Polynormal Fisher Vector

Polynormal Fisher Vector employs the Gaussian mixture model (GMM) to describe the distribtion of polynomals as $U_\lambda(\boldsymbol{p}) = \sum_{k=1}^{K} \pi_k u_k(\boldsymbol{p})$, and u_k denotes the kth Gaussian component:

$$u_k(\boldsymbol{p}) = \frac{1}{2\pi_k^{\frac{D}{2}} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{p} - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{p} - \boldsymbol{\mu}_k)\right\}, \quad (2)$$

$$\forall k\colon\ \pi_k \geq 0,\ \textstyle\sum_{k=1}^{K} \pi_k = 1,$$

where the polynormal $\boldsymbol{p} \in \mathbb{R}^D$; K is the number of Gaussian components; π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ are the prior mode probability, mean vector, and covariance matrix. We assume $\boldsymbol{\Sigma}_k$ to be a diagonal matrix with the variance vector $\boldsymbol{\sigma}_k^2$. To better fit the diagonal covariance matrix assumption, we first apply PCA to decorrelate polynormals and reduce the dimensions. The GMM parameters $\lambda = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k,\ k = 1, \dots, K\}$ can be estimated by using the Expectation-Maximization (EM) algorithm to optimize the Maximum Likelihood (ML) from a large number of polynormals. For a set of polynormals $\boldsymbol{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_N\}$ extracted from a depth video or a spatio-temporal cell, the soft assignment of $\boldsymbol{p}_i$ to the kth Gaussian component is:

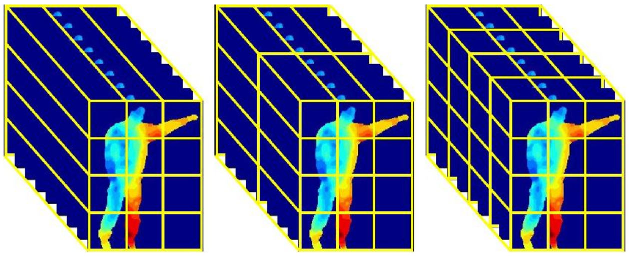

Figure 1: Illustration of the spatio-temporal pyramid with 4×3×7 space-time cells.

$$\gamma_i^k = \frac{\pi_k u_k(\boldsymbol{p}_i)}{\sum_{j=1}^{K} \pi_j u_j(\boldsymbol{p}_i)}. \quad (3)$$

The PFV of $\boldsymbol{P}$ is represented as $\psi(\boldsymbol{P}) = (\boldsymbol{\rho}_1^T, \boldsymbol{\tau}_1^T, \dots, \boldsymbol{\rho}_K^T, \boldsymbol{\tau}_K^T)^T$, where $\boldsymbol{\rho}_k$and $\boldsymbol{\tau}_k$ are D-dimensional gradients with respect to the mean vector $\boldsymbol{\mu}_k$ and standard deviation $\boldsymbol{\sigma}_k$ of the kth Gaussian component:

$$\boldsymbol{\rho}_k = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} \gamma_i^k \left(\frac{\boldsymbol{p}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k}\right), \quad (4)$$

$$\boldsymbol{\tau}_k = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} \gamma_i^k \left[\frac{(\boldsymbol{p}_i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1\right]. \quad (5)$$

In comparison to the Bag-of-Words (BoW) based methods, Fisher vector representation has the following advantages. (1) Gradients with respect to mean vectors and standard deviations provide extra feature distribution information in the polynormal space. (2) Fisher vector can be computed from a much smaller vocabulary which significantly reduces the computational complexity. (3) Simple linear classifiers perform very well with Fisher vectors which are efficient in both training and testing.

We follow the two normalization schemes introduced in [Perronnin et al. 2010], i.e., signed square rooting (SSR) and ℓ_2 normalization. As the number of Gaussian components increases, Fisher vector could become quite peaky around zero in a certain dimension. In order to reduce this sparsified effect, SSR $f(z) = \operatorname{sign}(z)|z|^\alpha$ with $0 < \alpha \leq 1$ is applied to each dimension z in the vector. ℓ_2 normalization is employed to mitigate the dependence on the proportion of activity specific cues contained in each video. In other words, it is used to reduce the effect of different amount of foreground/background information contained in different depth videos.

3.2 Spatio-Temporal Pyramid

We employ the spatio-temporal pyramid to roughly capture the spatial geometry and temporal order of a depth video. In the spatial dimensions, we use a $s_h \times s_w$ grid to capture the geometry layout as illustrated in Fig. 1. By making use of the depth information for human segmentation, we enforce the spatial grid onto the largest bounding box of the human body from a depth sequence, which is different from the spatial division on the entire frame as widely used in [Oreifej and Liu 2012; Wang et al. 2012]. We further adopt the temporal pyramid to incorporate the global temporal order. Here a video sequence is repeatedly and evenly subdivided into a set of temporal segments where PFVs are pooled. In this paper, we use a 3-level temporal pyramid as shown in Fig. 1. In together with the spatial grid, the spatio-temporal pyramid generates $s_h \times s_w \times 7$ space-time cells in total.

4. Experiments and Discussions

The proposed method is evaluated on two public benchmark datasets: MSRAction3D [Li et al. 2010] and MSRGesture3D [Wang et al. 2012]. In all experiments, we set a 3×3×3 local neighborhood $\mathcal{L}$ of each cloud point to form the polynormal and 100 components in GMM if not specified. The spatio-temporal pyramid consists of 4×3×7 space-time cells in height, width, and time, respectively. LIBLINEAR [Fan et al. 2008] is used as the linear SVM solver. The proposed PFV is extensively compared to a set of depth-based recognition methods. Experimental results show that our approach significantly outperforms the previous methods on the two datasets.

4.1 MSRAction3D Dataset

MSRAction3D is a human action dataset of depth sequences captured by a RGB-D camera. It contains 20 actions (Fig. 3) performed by 10 subjects facing the camera. Each action was performed 2 or 3 times by each subject. The 20 actions are chosen in the context of gaming and cover a wide range of motions related to arms, legs, torso, etc. In order to facilitate a fair comparison, we follow the same experimental settings as [Wang et al. 2012].

We first evaluate the size of local neighborhood $\mathcal{L}$ to form a polynormal. As discussed in Section 2, the size of $\mathcal{L}$ is determined by $\mathcal{L}_x \times \mathcal{L}_y \times \mathcal{L}_t$. Fig. 2 shows the recognition accuracy of PFV with different sizes of $\mathcal{L}$ under K Gaussian components. If no local temporal cue is encoded ($\mathcal{L}_t = 1$), increasing the spatial size of $\mathcal{L}$ lowers the recognition accuracy, e.g., from 1×1×1, 3×3×1, to 5×5×1. When $\mathcal{L}_x$ and $\mathcal{L}_y$ are fixed, the accuracy based on $\mathcal{L}_t >$ 1 is much higher than the one with $\mathcal{L}_t = 1$, e.g., the results of 3×3×3 significantly outperforms the ones of 3×3×1. In addition, the overall performance of $\mathcal{L}_t > 1$ is superior to that of 1×1×1. This shows the local temporal information embedded in polynormal helps to characterize the low-level motion cues. In the following experiments, we use the 3×3×3 local neighborhood $\mathcal{L}$ to form the polynormal and 100 components in computing GMM.

We compare the performance of PFV with other results in Table 1. Our method obtains the recognition accuracy of 92.73%, which is comparable to the most recent state-of-the-art SNV and significantly outperforms previous approaches. The joint-based methods [Xia et al. 2012; Yang and Tian 2014] are prone to errors of joint estimation when serious self-occlusion presents. In order to mitigate the joint error, models in [Wang et al. 2012; Wang et al. 2013] are proposed to learn a set of joint configurations which largely remove inaccurate estimations and therefore significantly improve the results. Compared to joints-based methods, the approaches [Vieira et al. 2012; Wang et al. 2012; Wang et al; 2012] using cloud points to model body shapes are more robust to occlusions and noises. While both methods are based upon hypersurface normals, PFV outperforms HON4D [Oreifej and Liu 2013] by 3.84%. This is mainly because polynormals are more informative than individual normals and the encoding scheme of Fisher vector is more representative than the polychoron or learned projectors. The confusion matrix of our method is demonstrated in Fig. 3. PFV performs pretty well on most actions. The confusions occur in recognizing quite similar actions, e.g., *hand catch* to *high throw* and *draw circle* to *draw tick.*

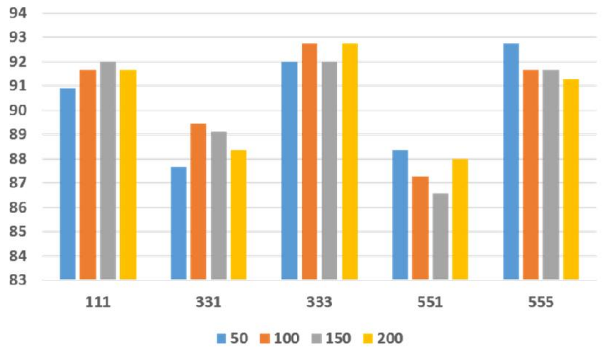

Figure 2. Recognition accuracy (%) of PFV with different sizes $\mathcal{L}_x\mathcal{L}_y\mathcal{L}_t$ of $\mathcal{L}$ under various numbers of Gaussian components K = 50, 100, 150, 200.

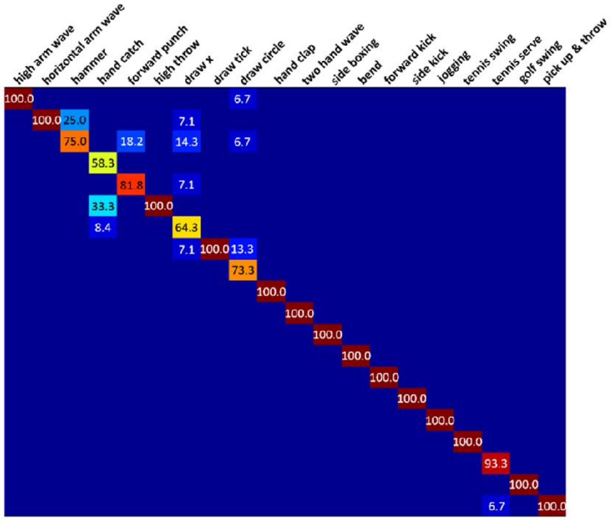

Figure 3. The confusion matrix of our proposed PFV on the MSRAction3D dataset.

Table 1. Comparison of recognition accuracy of our proposed PFV and other methods on the MSRAction3D dataset.

Method	**Accuracy**
Bag of 3D Points [Li et al. 2010]	74.70%
HOJ3D [Xia et al. 2012]	79.00%
EigenJoints [Yang and Tian 2014]	82.30%
STOP [Vieria et al. 2012]	84.80%
ROP [Wang et al. 2012]	86.50%
Actionlet Ensemble [Wang et al. 2012]	88.20%
Depth Motion Maps [Yang et al. 2012]	88.73%
HON4D [Oreifej and Liu 2013]	88.89%
DSTIP [Xia and Aggarwal 2013]	89.30%
Pose Set [Wang et al. 2013]	90.00%
SNV [Yang and Tian 2014]	**93.09%**
Ours	**92.73%**

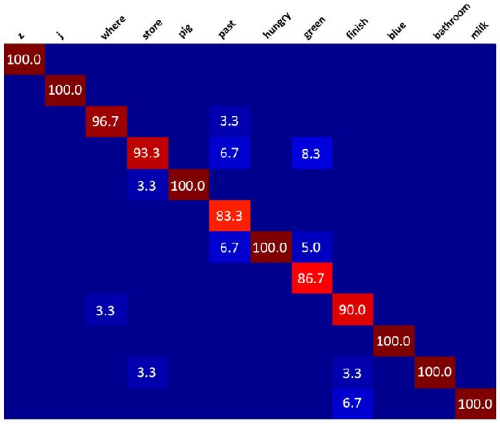

Figure 4. The confusion matrix of our proposed PFV on the MSRGesture3D dataset.

4.2 MSRGesture3D Dataset

MSRGesture3D is a dynamic hand gesture dataset captured by a RGB-D camera. There are 12 dynamic hand gestures (Fig. 4) defined by the American Sign Language (ASL) in this dataset. Each dynamic gesture was performed 2 or 3 times by each one of 10 subjects. We use the leave-one-out cross-validation as in [Wang et al. 2012].

PFV achieves the state-of-the-art accuracy of 95.83% which significantly outperforms all previous methods in Table 2. The confusion matrix of our method is shown in Fig. 4. PFV works very well on most dynamic gestures. The recognition errors concentrate on quite similar gestures, e.g., *green* and *j*. The two gestures share similar hand motions but only with different fingers. The joint-based methods cannot be used in this application because the joint estimation of hands is not available.

Table 2. Comparison of recognition accuracy of our proposed PFV and other methods on the MSRGesture3D dataset.

Method	**Accuracy**
Action Graph on Occupancy [Kurakin et al. 2012]	80.50%
Action Graph on Silhouette [Kurakin et al. 2012]	87.70%
ROP [Wang et al. 2012]	88.50%
Depth Motion Maps [Yang et al. 2012]	89.20%
HON4D [Oreifej and Liu 2013]	92.45%
SNV [Yang and Tian 2014]	94.72%
Ours	**95.83%**

5. Conclusions

We have presented an effective feature representation PFV for activity recognition from depth sequences. The polynormal assembled by hypersurface normals jointly characterizes local shape and motion information. We employ Fisher vector to aggregate polynormals into a representative feature. The spatio-temporal pyramid roughly captures the global geometric layout and temporal order of an activity sequence. Our proposed method is evaluated on two public benchmark datasets and achieves the state-of-the-art performance by using efficient linear classifiers. This also makes PFV well suited for large-scale activity recognition tasks. The future work will focus on a comprehensive evaluation of polynormal-based feature representations computed by different feature coding and pooling schemes.

Acknowledgement

This Work was supported in part by NSF EFRI-1137172, IIP-1343402, and IIS-1400802.

References

Boureau, Y., Bach, F., and LeCun, Y. 2010. Learning Mid-Level Features for Recognition. In *CVPR*.

Fan, R., Chang, K., Hsieh, C., Wang, X., and Lin, C. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*.

Hadfield, S. and Bowden, R. 2013. Hollywood3D: Recognizing Actions in 3D Natural Scenes. In *CVPR*.

Kurakin, A., Zhang, Z., and Liu, Z. 2012. A Real-Time System for Dynamic Hand Gesture Recognition with a Depth Sensor. In *EUSIPCO*.

Li, W., Zhang, Z., and Liu, Z. 2010. Action Recognition based on A Bag of 3D Points. In *CVPR Workshop*.

Perronnin, F., Sanchez, J., and Mensink, T. 2010. Improving the Fisher Kernel for Large-Scale Image Classification. In *ECCV*.

Oreifej, O. and Liu, Z. 2013. HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In *CVPR*.

Sanchez, J., Perronnin, F., Mensink, T., and Verbeek, J. 2013. Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. 2011. Real-Time Pose Recognition in Parts from Single Depth Images. In *CVPR*.

Vieira, A., Nascimento, E., Oliveria, G., Liu, Z., and Campos, M. 2012. STOP: Space-Time Occupancy Patterns for 3D Action Recognition from Depth Map Sequences. In *CIARP*.

Wang, J., Liu, Z., Wu, Y., and Yuan, J. 2012. Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In *CVPR*.

Wang, J., Liu, Z., Chorowski, J., Chen, Z., and Wu, Y. 2012. Robust 3D Action Recognition with Random Occupancy Patterns. In *ECCV*.

Wang, C., Wang, Y., and Yuille, A. 2013. An Approach to Pose based Action Recognition. In *CVPR*.

Xia, L. and Aggarwal, J. 2013. Spatio-Temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera. In *CVPR*.

Xia, L. Chen, C., and Aggarwal, J. 2012. View Invariant Human Action Recognition Using Histogram of 3D Joints. In *CVPR Workshop*.

Yang, X. and Tian, Y. 2014. Effective 3D Action Recognition Using EigenJoints. *Journal of Visual Communication and Image Representation*.

Yang, X., Zhang, C., and Tian, Y. 2012. Recognizing Actions Using Depth Motion Maps based Histogram of Oriented Gradients. In *ACM Multimedia*.

Yang, X. and Tian, Y. 2014. Super Normal Vector for Activity Recognition Using Depth Sequences. In *CVPR*, 2014.