Super Normal Vector for Activity Recognition Using Depth Sequences

Xiaodong Yang and YingLi Tian
Department of Electrical Engineering
City College, City University of New York
{xyang02, ytian}@ccny.cuny.edu

Abstract

This paper presents a new framework for human activity recognition from video sequences captured by a depth camera. We cluster hypersurface normals in a depth sequence to form the polynormal which is used to jointly characterize the local motion and shape information. In order to globally capture the spatial and temporal orders, an adaptive spatio-temporal pyramid is introduced to subdivide a depth video into a set of space-time grids. We then propose a novel scheme of aggregating the low-level polynormals into the super normal vector (SNV) which can be seen as a simplified version of the Fisher kernel representation. In the extensive experiments, we achieve classification results superior to all previous published results on the four public benchmark datasets, i.e., MSRAction3D, MSRDailyActivity3D, MSRGesture3D, and MSRActionPairs3D.

1. Introduction

Activity recognition has been widely applied in a number of real-world applications, e.g., video surveillance, human-computer interaction, sign language recognition, and medical health care. In the past decades, research on activity recognition mainly focused on recognizing actions from videos captured by conventional visible light cameras. As the imaging techniques advance, the recent emergence of low-cost and easy-operation depth sensors such as Kinect [15] facilitates a variety of visual recognition tasks including activity recognition.

Depth maps have several advantages with respect to traditional color images in the context of activity recognition. First, they provide additional body shape and structure information, which has been successfully applied to recover skeleton joints from a single depth map. Second, color and texture are precluded in depth maps, which makes the problems of human detection and segmentation easier. Third, depth sensors are insensitive to lighting change, which brings great benefits to the system monitoring in the dark environment.

It was recently shown in [12, 22] that conventional approaches based upon color sequences could not perform well on depth maps due to a large amount of false point detections fired on the spatio-temporally discontinuous regions. On the other hand, depth maps and color sequences have quite different properties. The descriptors based on brightness, gradient, and optical flow in traditional color sequences might be unsuited to represent depth maps. It is therefore intuitive to design action features according to the specific characteristics of depth sequences, e.g., cloud points [20, 21] and surface normals [12].

In this paper, we propose a novel activity recognition framework based upon the polynormal which is a group of hypersurface normals in depth sequences. A polynormal clusters the extended surface normals [12] from a local space-time subvolume. It can be used to jointly capture the local motion and geometry cues. A sparse coding approach [11] is employed to compute the polynormal dictionary and coefficients. We record the differences between polynormals and visual words. The coefficient-weighted difference vectors are aggregated through spatial average pooling and temporal max pooling for each visual word. The vectors of all visual words are in the end concatenated as a feature vector, which can be viewed as a non-probabilistic simplification of the Fisher kernel representation [13]. We further subdivide a depth video into a set of space-time grids. An adaptive spatio-temporal pyramid is proposed to capture the spatial layout and temporal order in a global way. We concatenate the vectors extracted from all the space-time grids as the final representation of super normal vector (SNV).

3D Activity Dataset	Best Results	Our Results
MSRAction3D	91.70% [26]	**93.09%**
MSRGesture3D	92.45% [12]	**94.72%**
MSRActionPairs	96.67% [12]	**98.89%**
MSRDailyActivity3D	85.75% [21]	**86.25%**

Table 1. Our results compared to the best published results so far on the four datasets (more detailed comparisons in Table 2-5).

We evaluate our method according to the standard experimental protocols on the four public benchmark datasets: MSRAction3D [9], MSRDailyActivity3D [21], MSRGesture3D [20], and MSRActionPairs3D [12]. Our results outperform all published ones as shown in Table 1.

The main contributions of this paper can be summarized as follows. First, we group hypersurface normals from a local space-time depth subvolume into polynormal which reserves the correlation between local normals and is more robust against noise than the individual normal [12]. Second, a novel approach is proposed to aggregate low-level polynormals into the discriminative representation SNV. Third, our adaptive spatial-temporal pyramid is better adapted to retain the spatial and temporal orders than the widely used uniform cells [7, 12, 19, 21]. Moreover, our framework is flexible to combine with skeleton joints and compute SNV for each joint trajectory.

The remainder of this paper is organized as follows. Section 2 introduces the related work on activity recognition using depth sequences. Section 3 describes the concept of polynormal. In Section 4, we provide detailed procedures of computing SNV. A variety of experimental results and discussions are presented in Section 5. Finally, Section 6 summarizes the remarks of this paper.

2. Related Work

Research on activity recognition has explored a number of representations of depth sequences which range from skeleton joints [26], cloud points [20], projected depth maps [25], local interest points [22], to surface normals [12].

Biological observations have indicated human actions can be modeled by the movements of skeleton joints. The moving pose descriptor was recently proposed in [26] by using the configuration, speed, and acceleration of joints. To reduce joint estimation errors, the pose set [18] selected the best-k joint configurations by segmentation and temporal constraints. The relative positions of pairwise joints were also used in [21] as a complementary feature to characterize the motion information. Compared to skeleton joints, cloud points are more robust to noise and occlusion. Wang et al. [20, 21] introduced local and random occupancy patterns to describe depth appearance. In local occupancy patterns [21], they subdivided the local 3D subvolumes associated with skeleton joints into a set of spatial grids and counted the number of cloud points falling into each grid. Similar representation based on cloud points was also applied to the 4D subvolumes sampled by a weighted sampling scheme in random occupancy patterns [20].

The approaches based on projected depth maps usually transform the problem in 3D to 2D. Yang et al. [25] stacked differences between projected depth maps as the depth motion maps where HOG was extracted as the global representation of a depth video. Several local interest point detectors specifically designed for the depth data were recently proposed. DSTIP was introduced in [22] to localize activity-related interest points from depth videos by suppressing flip noise. Hadfield et al. [4] extended the detection algorithms of Harris corners, Hessian points, and separable filters to the 3.5D and 4D for depth sequences. As demonstrated in [16], the surface normal provides most shape and structure information of an object in 3D. HON4D [12] followed this observation to extend the surface normal to the 4D space and quantized them by the regular and discriminative learned polychorons.

Our method presented in this paper proceeds along with this direction. It relies on the polynormal which is a local cluster of extended surface normals. We propose a novel approach to aggregate the low-level polynormals in each adaptive spatio-temporal grid. The concatenation of feature vectors extracted from all space-time grids forms the final depth video representation.

3. Polynormal

The concept of a normal to a surface in 3-dimensional space can be extended to a hypersurface in m-dimensional space. The hypersurface can be viewed as a function $\mathbb{R}^{m-1} \rightarrow \mathbb{R}^1 : x_m = f(x_1, \ldots, x_{m-1})$, which is represented by a set of m-dimensional points that locally satisfy $F(x_1, \ldots, x_m) = f(x_1, \ldots, x_{m-1}) - x_m = 0$. The normal vectors to the hypersurface at these points are 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 the context of depth sequences, i.e., $m = 4$, each point satisfies $F(x, y, t, z) = f(x, y, t) - z = 0$. We therefore obtain the extended surface normal by

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

The distribution of normal orientations is able to provide more informative geometric cues than the traditional gradient orientations [12]. Moreover, the motion cues are also embedded in the normal vector of Eq. (1). In order to retain the correlation between neighboring normals and make them more robust to noise, we propose polynormal to cluster normals from a local spatio-temporal neighborhood. Similar schemes have been validated in other fields. For example, the spatial neighborhoods of low-level features are jointly encoded in deep learning [8] and macrofeatures [1].

A polynormal $\boldsymbol{p}$ associated with each cloud point in a depth video concatenates L normals in the local neighborhood $\mathcal{L}$ of this point:

$$\boldsymbol{p} = \left(\boldsymbol{n}_1^T, \ldots, \boldsymbol{n}_L^T\right)^T, \quad \boldsymbol{n}_1, \ldots, \boldsymbol{n}_L \in \mathcal{L}. \quad (2)$$

The neighborhood $\mathcal{L}$ is a spatio-temporal depth subvolume determined by two parameters n_s and n_t, where n_s

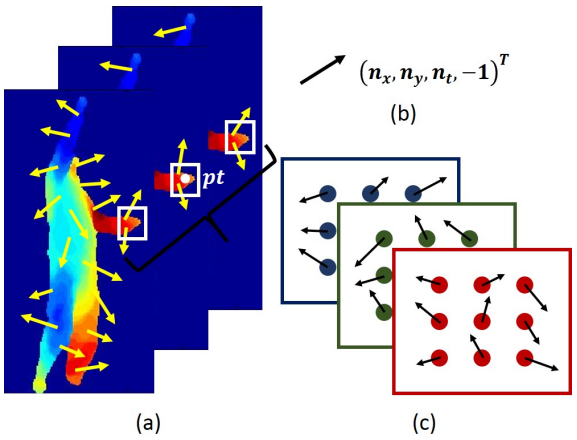

Figure 1. Illustration of generating polynormal of the point *pt*. (a) A depth sequence of tennis serve and normals associated with cloud points. For figure clarity, only a few normals are visualized. The three white squared regions correspond to the neighborhood $\mathcal{L}$. (b) The extended surface normal vector. (c) If $n_s = 9$ and $n_t = 3$, the polynormal of *pt* is consisted of the 27 neighboring normals.

denotes the number of neighboring points in spatial and n_t indicates the number of neighboring maps in temporal. Fig. 1 illustrates the concept of polynormal. A short sequence of the tennis serve action is shown in Fig. 1(a). If we set $n_s = 9$ and $n_t = 3$, then the polynormal of the white point *pt* concatenates the 27 normals from the three adjacent depth maps as shown in Fig. 1(c).

4. Computing Super Normal Vector (SNV)

In this section, we describe the detailed procedures of computing SNV based on the low-level polynormals. We utilize the sparse coding to learn a dictionary and code polynormals. Instead of directly pooling the coefficients of coded polynormals, we aggregate the weighted differences between polynormals and visual words into a vector. A depth video is subdivided into a set of space-time grids by our proposed adaptive spatio-temporal pyramid. The feature vectors extracted from each grid are then concatenated as the final SNV representation.

4.1. Aggregating Polynormals

In visual recognition, the global representation of an image or a video is usually obtained by extracting low-level features, coding them over a learned dictionary, and then pooling the distribution of the codes in some well-chosen support regions. After the coding step, low-level features are discarded in the recognition pipeline. In our framework, we keep the low-level features by recording the differences between them and visual words. As shown in [5, 13, 27], the relative displacements can provide the extra distribution information of low-level features.

We employ sparse coding to learn the dictionary and code polynormals. It is well known that the ℓ_1 penalty yields a sparse solution. Given a training set of whitened polynormals $\boldsymbol{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N]$ in $\mathbb{R}^{M\times N}$, the sparse coding problem can be solved by

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{2}\|\boldsymbol{p}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1\right), \quad (3)$$

$$\text{subject to } \mathbf{d}_k^T\mathbf{d}_k \leq 1, \forall k = 1, \ldots, K,$$

where $\mathbf{D}$ in $\mathbb{R}^{M\times K}$ is the dictionary, each column $(\mathbf{d}_k)_{k=1}^K$ representing a visual word; $\boldsymbol{\alpha}$ in $\mathbb{R}^{K\times N}$ is the coefficients of sparse decomposition; λ is the sparsity inducing regularizer.

We ℓ_1-normalize each column $(\boldsymbol{\alpha}_i)_{i=1}^N$ to obtain the soft assignment $\alpha_{k,i}$ of polynormal $\boldsymbol{p}_i$ to the kth visual word. The size of the volume (depth sequences) where we perform the aggregation is $H \times W$ pixels and T frames. The volume corresponds to either the entire video sequence or a subsequence defined by a space-time grid. We denote by N_t the set of indices within the frame t. For each visual word, the spatial average pooling is first applied to aggregate the coefficient-weighted differences:

$$\boldsymbol{u}_k(t) = \frac{1}{|N_t|}\sum_{i\in N_t}\alpha_{k,i}\left(\boldsymbol{p}_i - \mathbf{d}_k\right), \quad (4)$$

where $\boldsymbol{u}_k(t)$ represents the pooled difference vector of the kth visual word in the tth frame. The temporal max pooling is then used to aggregate the vectors from T frames:

$$\boldsymbol{u}_{k,i} = \max_{t=1,\ldots,T}\boldsymbol{u}_{k,i}(t), \text{ for } i = 1, \ldots, M, \quad (5)$$

where $\boldsymbol{u}_k$ is the vector representation of the kth visual word in the whole volume; i indicates the ith component in corresponding vectors. The final vector representation $\boldsymbol{U}$ is the concatenation of the $\boldsymbol{u}_k$ vectors from the K visual words and is therefore of KM dimensions:

$$\boldsymbol{U} = \left(\boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_K^T\right)^T. \quad (6)$$

In order to capture the global spatial layout and temporal order, a depth sequence is subdivided into a set of space-time grids. We extract a feature vector U from each grid and concatenate them as SNV. This representation has several remarkable properties. (1) The displacements to visual words retain some information lost in feature quantization process. (2) We can compute SNV upon a much smaller dictionary (e.g., 100) which reduces computational cost. (3) SNV performs quite well with simple linear classifiers which are efficient in terms of both training and testing.

4.2. Relationship with Fisher Kernel

We now demonstrate our proposed SNV is a simplified non-probabilistic version of the Fisher kernel representation which has been successfully applied in the image classification tasks [13]. Fisher kernel assumes low-level features are distributed according to a generative model, e.g., Gaussian Mixture Model (GMM).

In the framework of Fisher kernel, each feature descriptor is described by its deviations with respect to the GMM parameters $\beta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, k = 1, \ldots, K\}$, where π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\sigma}_k$ are the mixture weight, mean vector, and variance matrix (diagonal) of the kth Gaussian component φ_k. The soft assignment of the descriptor $\boldsymbol{p}_i$ to the component φ_k is defined as:

$$\gamma_{k,i} = \frac{\pi_k \varphi_k\left(\boldsymbol{p}_i\right)}{\sum_{j=1}^{K} \pi_j \varphi_j\left(\boldsymbol{p}_i\right)}, \tag{7}$$

We denote by $\boldsymbol{p}_i$ a general descriptor and N_t a general pooling region in this context. We focus on the gradient $\boldsymbol{g}_k$ with respect to the mean vector $\boldsymbol{\mu}_k$ of the kth Gaussian:

$$\boldsymbol{g}_k = \frac{1}{|N_t|\sqrt{\pi_k}} \sum_{i \in N_t} \gamma_{k,i} \boldsymbol{\sigma}_k^{-1} \left(\boldsymbol{p}_i - \boldsymbol{\mu}_k\right). \tag{8}$$

If making the two hypotheses: (1) mixture weights are equal, i.e., $\pi_k = 1/K$ and (2) covariance matrices are isotropic, i.e., $\boldsymbol{\sigma}_k = \epsilon \mathbb{I}$ with $\epsilon > 0$, we can simplify Eq. (8) to

$$\boldsymbol{g}_k \propto \frac{1}{|N_t|} \sum_{i \in N_t} \gamma_{k,i} \left(\boldsymbol{p}_i - \boldsymbol{\mu}_k\right), \tag{9}$$

where $\gamma_{k,i}$ is simplified to $\varphi_k\left(\boldsymbol{p}_i\right) / \sum_{j=1}^{K} \varphi_j\left(\boldsymbol{p}_i\right)$. The two representations in Eq. (4) and Eq. (9) have the same form except the ways to obtain the weight ($\alpha_{k,i}$ and $\gamma_{k,i}$) and the center ($\mathbf{d}_k$ and $\boldsymbol{\mu}_k$). We utilize sparse coding to compute the weight and center, while GMM clustering is used in the Fisher kernel.

We choose sparse coding over GMM in our aggregation scheme because it is cheaper to compute the centers (dictionary), especially it was recently shown in [2] that a reasonably good dictionary can be created by some simple methods, e.g., random sampling a training set. In addition, our empirical evaluations show our method based on sparse coding improves the recognition accuracy.

4.3. Adaptive Spatio-Temporal Pyramid

In the spatial dimensions, we use a $n_H \times n_W$ grid to capture the geometry layout as shown in the top of Fig. 2. As the depth information greatly facilitates human segmentation, we enforce the spatial grid on the largest bounding box of the human body, instead of on the entire frame as widely used in [7, 12, 19].

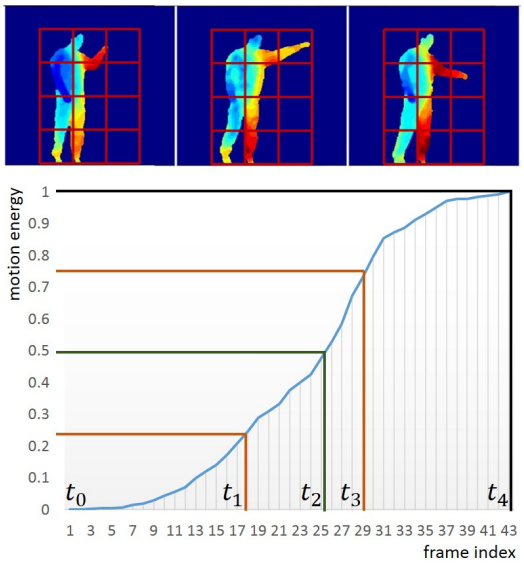

Figure 2. Adaptive spatio-temporal pyramid. Top: a 4×3 spatial grid. The spatial grid is implemented on the largest bounding box of human body rather than on the entire frame. Bottom: the frame index and associate motion energy used to build the adaptive temporal pyramid. The temporal segments are obtained by repeatedly and evenly subdividing the normalized motion energy vector instead of the time axis.

The temporal pyramid was introduced by Laptev et al. [7] to take into account the rough temporal order of a video. It was also employed in depth sequences [12, 21] to incorporate cues from the temporal context. In these methods, a video sequence (either color or depth) is repeatedly and evenly subdivided into a set of temporal segments where descriptor-level statistics are pooled. However, different people could have varied motion speed or frequency when they are performing the same activity. It is therefore inflexible to handle this variance by evenly subdividing a video along the time axis. In addition, it is more desirable to pool low-level features within the similar activity status, e.g., neutral, onset, apex, and offset. In order to handle these difficulties, we propose an adaptive temporal pyramid based on the motion energy.

Given a depth sequence, we first project the ith frame $\boldsymbol{I}^i$ onto three orthogonal planes to obtain the projected maps $\boldsymbol{I}_v^i, v \in \{1, 2, 3\}$. The difference between two consecutive maps is then thresholded to generate a binary map. We compute the motion energy by accumulating summations of non-zero elements of binary maps as:

$$\varepsilon(i) = \sum_{v=1}^{3} \sum_{j=1}^{i-1} \text{sum}\left(\left|\boldsymbol{I}_v^{j+1} - \boldsymbol{I}_v^{j}\right| > \epsilon\right), \tag{10}$$

where $\varepsilon(i)$ is the motion energy of the ith frame; ϵ is the threshold; $\text{sum}(\cdot)$ returns the number of non-zero elements

Algorithm 1: Computation of SNV

Input: a depth sequence
a dictionary $\mathbf{D} = (\mathbf{d}_k)_{k=1}^{K}$
a set of space-time grids $V = \{v_i\}$
Output: SNV

compute polynormals $\{\boldsymbol{p}_i\}$ from the depth sequence
compute coefficients $\{\boldsymbol{\alpha}_i\}$ of $\{\boldsymbol{p}_i\}$ by sparse coding
for grid $i = 1$ **to** $|V|$ **do**
  for visual word $k = 1$ **to** K **do**
    $\boldsymbol{u}_i^k$:= spatial average pooling and temporal max pooling of $\alpha_{k,i}\,(\boldsymbol{p}_i - \mathbf{d}_k)$, where $\boldsymbol{p}_i \in v_i$
  end
  $\boldsymbol{U}_i := \left(\boldsymbol{u}_i^1, \ldots, \boldsymbol{u}_i^K\right)$
end
SNV := $\left(\boldsymbol{U}_1, \ldots, \boldsymbol{U}_{|V|}\right)$

in a binary map. The motion energy of a frame reflects its relative motion status with respect to the entire activity.

Our proposed adaptive temporal pyramid is built on this measurement as shown in the bottom of Fig. 2. We evenly subdivide the normalized motion energy vector into a set of segments, whose corresponding frame indices are used to partition a video. In this paper, we use a 3-level temporal pyramid as illustrated in this figure: $\{t_0t_4\}$, $\{t_0t_2, t_2t_4\}$, and $\{t_0t_1, t_1t_2, t_2t_3, t_3t_4\}$. In together with the spatial grid, our adaptive spatio-temporal pyramid in total generates $n_H \times n_W \times 7$ space-time cells.

We summarize the outline of computing SNV of a depth video in Algorithm 1. The set of space-time grids V are chosen by the proposed adaptive spatio-temporal pyramid.

4.4. Joint Trajectory Aligned SNV

While the framework discussed above operates on the entire depth sequence, our method is flexible to combine with skeleton joints [15] to compute SNVs based on joint trajectories. This is useful in the scenarios where people significantly change their spatial locations in a depth video. The aggregation process is the same as the earlier discussion, except the pooling region is based on the space-time volume aligned around each joint trajectory. It was also shown in dense trajectories [19] that descriptors aligned with trajectories were superior to those computed from straight cuboids.

As shown in Fig. 3, the volume aligned with a joint trajectory can be viewed as a single video sequence with $H \times W$ pixels and T frames. We apply the adaptive spatio-temporal pyramid on this volume to obtain $n_H \times n_W \times 7$ space-time cells. In each cell, we use the same aggregation scheme, i.e., spatial average pooling and temporal max pooling of the weighted difference vectors as in Eq. (4-5). The vectors from all the space-time cells are concatenated as the joint trajectory aligned SNV. We in the end combine the SNVs aligned with all the joint trajectories as the final representation of a depth sequence.

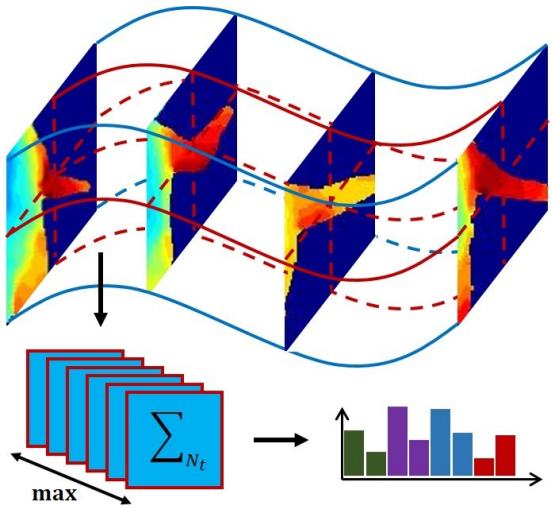

Figure 3. SNV based on the skeleton joint trajectory. The trajectory-aligned volume is subdivided into a set of space-time grids according to the adaptive spatio-temporal pyramid. Each cell generates a feature vector by the spatial average pooling and temporal max pooling.

5. Experiments

In this section we extensively evaluate our proposed method on four public benchmark datasets: MSRAction3D [9], MSRGesture3D [20], MSRActionPairs3D [12], and MSRDailyActivity3D [21]. In all experiments, we set a 9×3 neighborhood for each cloud point to form the polynormal. We use 100 visual words in the sparse coding. The adaptive spatio-temporal pyramid is typically of $4 \times 3 \times 7$ space-time grids in height, width, and time, respectively. We employ LIBLINEAR [3] as the linear SVM solver. Our method is extensively compared to the existing depth-based approaches. The methods designed for color videos are not included in our comparisons because they have been widely shown to be unsuited for depth maps [12, 21, 22]. Experimental results show that our algorithm significantly outperforms the state-of-the-art methods on these datasets. Our source code for computing SNV is available online.[1]

5.1. MSRAction3D Dataset

The MSRAction3D [9] is an action dataset of depth sequences captured by a depth camera. It contains 20 actions performed by 10 subjects facing the camera. Each action is performed 2 or 3 times by each subject. The 20 actions

[1] http://yangxd.org/code

Method	Accuracy
Bag of 3D Points [9]	74.70%
HOJ3D [23]	79.00%
EigenJoints [24]	82.30%
STOP [17]	84.80%
Random Occupancy Pattern [20]	86.50%
Actionlet Ensemble [21]	88.20%
Depth Motion Maps [25]	88.73%
HON4D [12]	88.89%
DSTIP [22]	89.30%
Pose Set [18]	90.00%
Moving Pose [26]	91.70%
Ours	**93.09%**

Table 2. Recognition accuracy comparison of our method and previous approaches on the MSRAction3D dataset.

are chosen in the context of gaming and cover a variety of movements related to arms, legs, torso, etc.

In order to facilitate a fair comparison, we follow the same experimental setting as [21]. SNV achieves an accuracy of 93.09% which significantly outperforms the existing methods. If we only keep the first level (i.e., $\{t_0 t_4\}$ in Fig. 2) of the temporal pyramid, the accuracy goes down to 91.64%. This shows the recognition benefits from the cues in the global temporal context. We also compare to the polynormal based Fisher kernel representation which obtains 92.00% accuracy, 1.09% inferior to SNV. The confusion matrix of our method is demonstrated in the left of Fig. 4. Our method works very well on most actions. The recognition errors concentrate on quite similar actions, e.g., *hand catch* to *high throw* and *draw circle* to *draw tick*.

We compare the performance of SNV with other results in Table 2. The methods based on joints are vulnerable to joint errors due to severe self-occlusions. So the model in [18] selects the best-k joint configurations which largely remove inaccurate joints. The approach in [26] utilizes pose, speed, and acceleration of joints. While still inferior to our method, the approaches in [17, 20, 21] improve the results in [23, 24] because cloud points are more resistant to occlusions and provide additional shape cues compared to skeleton joints. SNV outperforms HON4D [12] by 4.20%, though both methods are based upon hypersurface normals. This is probably because (1) polynormals obtain more discriminative local motion and shape information than individual normals; (2) sparse coding is more robust than the polychoron and learned projectors; (3) our aggregation scheme, i.e., spatial average pooling and temporal max pooling of weighted difference vectors, is more representative than the sum pooling of inner production values; (4) the adaptive pyramid is more flexible than the uniform cells to capture the global spatio-temporal cues.

Method	Accuracy
Action Graph on Occupancy [6]	80.50%
Action Graph on Silhouette [6]	87.70%
Random Occupancy Pattern [20]	88.50%
Depth Motion Maps [25]	89.20%
HON4D [12]	92.45%
Ours	**94.74%**

Table 3. Recognition accuracy comparison of our method and previous approaches on the MSRGesture3D dataset.

5.2. MSRGesture3D Dataset

The MSRGesture3D [20] is a dynamic hand gesture dataset of depth sequences captured by a depth camera. It contains 12 dynamic hand gestures defined by the American Sign Language (ASL). There are 10 subjects, each one performing each dynamic gesture 2 or 3 times. This dataset presents more self-occlusions than MSRAction3D.

The leave-one-out cross-validation scheme as [20] is used in our evaluation. SNV obtains the state-of-the-art accuracy of 94.74% which outperforms all previous methods as shown in Table 3. The confusion matrix of SNV is shown in the middle of Fig. 4. Our method performs pretty well on most dynamic gestures. The most confusion occurs in recognizing the gestures *green* which shares similar motion to *j* but with different fingers. As the joint estimation is not available for human hands, the joint-based methods [18, 21, 23, 24, 26] cannot be used in this application.

5.3. MSRActionPairs3D Dataset

The MSRActionPairs3D [12] is a paired-activity dataset of depth sequences captured by a depth camera. It contains 12 activities (i.e., 6 pairs) of 10 subjects with each subject performing each activity 3 times. This dataset is collected to investigate how the temporal order affects activity recognitions.

The same evaluation setup as [12] is used in our experiment. SNV achieve the state-of-the-art accuracy of 98.89%. The detailed comparison to other approaches is demonstrated in Table 4. The skeleton feature [21] only involves pair-wise difference of joint positions within each frame. The LOP feature [21] is used to characterize the depth appearance. It counts the number of cloud points falling into each spatial grid of a depth subvolume. There is no temporal information encoded in the two features. In the depth motion maps [25], depth sequences are collapsed onto three projected maps where temporal orders are eliminated. These methods therefore suffer the inner-paired confusion. The skeleton and LOP features equipped with a uniform temporal pyramid improves the recognition result as the global temporal order is incorporated. However, this result is still significantly inferior to ours.

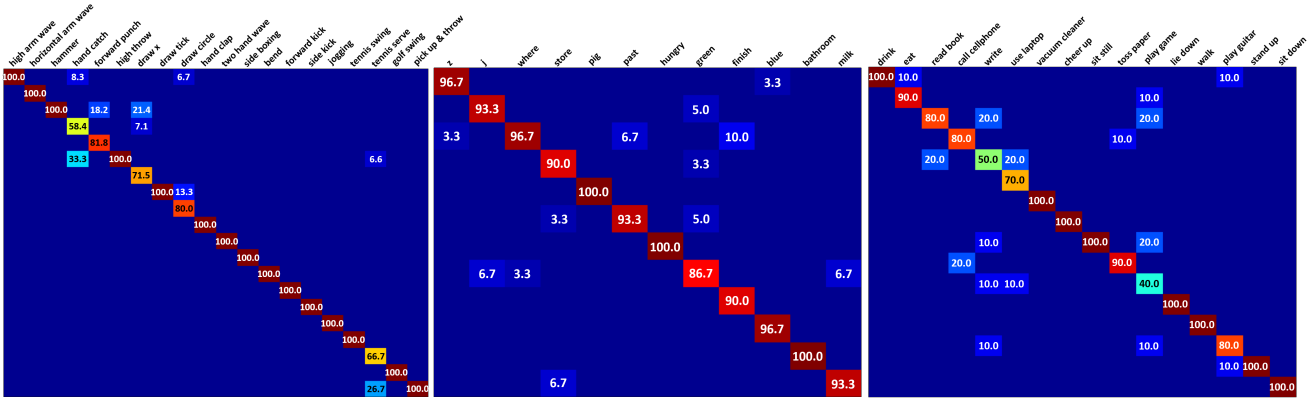

Figure 4. Confusion matrices of our method on the MSRAction3D (left), MSRGesture3D (middle), and MSRDailyActivity3D (right) datasets. This figure is better viewed on screen.

Method	Accuracy
Skeleton + LOP [21]	63.33%
Depth Motion Maps [25]	66.11%
Skeleton + LOP + Pyramid [21]	82.22%
HON4D [12]	96.67%
Ours	**98.89%**

Table 4. Recognition accuracy comparison of our method and previous approaches on the MSRActionPairs3D dataset.

It is therefore crucial to capture the spatio-temporal orders to distinguish the activities with similar motion and shape cues. In our method, the space-time orders are embedded in two levels, i.e., polynormals and the adaptive pyramid, which characterize the local and global spatio-temporal orders, respectively. It is interesting to observe that SNV achieves an accuracy of 97.78% if no temporal pyramid is used. This promising result demonstrates the local motion cues enclosed in the polynormals reflect the temporal orders pretty well. Because of the high recognition accuracy, the confusion matrix on this dataset is omitted in Fig. 4.

5.4. MSRDailyActivity3D Dataset

The MSRDailyActivity3D [21] is a daily activity dataset of depth sequences captured by a depth camera. There are 16 daily activities which are performed by 10 subjects. Each subject performs each activity twice, one in standing position and the other in sitting position. Compared to the other three datasets, actors in this dataset present large spatial and scaling changes. Moreover, most activities involve human-object interactions.

In order to handle the significant spatial and scaling changes, we employ the joint trajectory aligned SNV on this dataset. Each joint is tracked through the entire depth sequence. A patch is associated with each joint in each frame.

Method	Accuracy
LOP [21]	42.50%
Depth Motion Maps [25]	43.13%
EigenJoints [24]	58.10%
Joint Position [21]	68.00%
NBNN + Parts + Time [14]	70.00%
RGGP [10]	72.10%
Moving Pose [26]	73.80%
Local HON4D [12]	80.00%
Actionlet Ensemble [21]	85.75%
Ours	**86.25%**

Table 5. Recognition accuracy comparison of our method and previous approaches on the MSRDailyActivity3D dataset.

Because depth values inversely vary with an object size, we set an adaptive size s/z to each patch, where $s = 300$K is a scale factor and z is the depth value of a joint in the current frame. Unlike the fixed patch size used in [12], the adaptive size is more robust to handle the scaling change. So the patch size in Fig. 3 is not necessary to be consistent. We compute SNV and joint position difference feature for each joint trajectory. The actionlet ensemble model [21] is then used to combine the features from multiple joints.

We follow the same experimental setting as [21] and obtain the accuracy of 86.25%. The confusion matrix is shown in the right of Fig. 4. Most recognition errors occur in the almost still activities, e.g., *read book*, *write*, and *use laptop*. Since most activities involve human-object interactions, this dataset can be used to evaluate how the motion and shape information are correlated. It could be insufficient to capture motion and shape independently because some activities share quite similar motion cues but present distinct shape properties. SNV jointly encodes local motion and shape information in polynormals which in the high level reflect the co-occurrence of hand motion and object shape.

Table 5 shows the performance of our method compared to the previous approaches. Note: an accuracy of 88.20% was reported in [22], however, four activities with less motion (i.e., *sit still*, *read books*, *write on paper*, and *use laptop*) were removed in their experiment. The holistic approach [25] suffers the non-aligned sequences. The methods [10, 14, 21, 24, 26] based on either motion or shape information alone are significantly inferior to our method and the ones [12, 21] that jointly model the two cues.

6. Conclusion

We have presented a novel framework to recognize human activities from depth sequences. The polynormal based on extended surface normals jointly encodes local motion and shape cues. A new aggregation scheme is proposed by sparse coding polynormals, and spatial average pooling and temporal max pooling of coefficient-weighted difference vectors between polynormals and visual words. We have introduced the adaptive spatial-temporal pyramid which is shown to be better adapted to retain the spatial and temporal orders. Our proposed framework is flexible to be used in the joint trajectory aligned depth sequence, which is well suited in the scenarios where significant spatial and scaling changes present. Our method is extensively evaluated on four public benchmark datasets and compared to a number of state-of-the-art approaches. Experimental results demonstrate that our method outperforms all previous approaches on these datasets.

Acknowledgement

This work was supported in part by NSF grant EFRI-1137172 and FHWA grant DTFH61-12-H-00002.

References

[1] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning Mid-Level Features for Recognition. In *CVPR*, 2010.

[2] A. Coates and A. Ng. The Importance of Encoding versus Training with Sparse Coding and Vector Quantization. In *ICML*, 2011.

[3] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A Library for Large Linear Classification. *JMLR*, 2008.

[4] S. Hadfield and R. Bowden. Hollywood 3D: Recognizing Actions in 3D Natural Scenes. In *CVPR*, 2013.

[5] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating Local Descriptors into A Compact Image Representation. In *CVPR*, 2010.

[6] A. Kurakin, Z. Zhang, and Z. Liu. A Read-Time System for Dynamic Hand Gesture Recognition with A Depth Sensor. In *EUSIPCO*, 2012.

[7] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In *CVPR*, 2008.

[8] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representation. In *ICML*, 2009.

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

[10] L. Liu and L. Shao. Learning Discriminative Representations from RGB-D Video Data. In *IJCAI*, 2013.

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Dictionary Learning for Sparse Coding. In *ICML*, 2009.

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

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

[14] L. Seidenari, V. Varano, S. Berretti, A. Bimbo, and P. Pala. Recognizing Actions from Depth Cameras as Weakly Aligned Multi-Part Bag-of-Poses. In *CVPR Workshop*, 2013.

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

[16] S. Tang, X. Wang, T. Han, J. Keller, M. Skubic, S. Lao, and Z. He. Histogram of Oriented Normal Vectors for Object Recognition with A Depth Sensor. In *ACCV*, 2012.

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

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

[19] H. Wang, A. Klaser, C. Schmid, and C. Liu. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *IJCV*, 2013.

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

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

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

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

[24] X. Yang and Y. Tian. EigenJoints based Action Recognition Using Naive Bayes Nearest Neighbor. In *CVPR Workshop*, 2012.

[25] X. Yang and Y. Tian. Recognizing Actions Using Depth Motion Maps based Histograms of Oriented Gradients. In *ACM Multimedia*, 2012.

[26] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection. In *ICCV*, 2013.

[27] X. Zhou, K. Yu, T. Zhang, and T. Huang. Image Classification Using Super-Vector Coding of Local Image Descriptors. In *ECCV*, 2010.