Super Normal Vector for Human Activity Recognition with Depth Cameras

Xiaodong Yang, *Member, IEEE,* and YingLi Tian, *Senior Member, IEEE*

Abstract—The advent of cost-effectiveness and easy-operation depth cameras has facilitated a variety of visual recognition tasks including human activity recognition. This paper presents a novel framework for recognizing human activities from video sequences captured by depth cameras. We extend the surface normal to polynormal by assembling local neighboring hypersurface normals from a depth sequence to jointly characterize local motion and shape information. We then propose a general scheme of super normal vector (SNV) to aggregate the low-level polynormals into a discriminative representation, which can be viewed as a simplified version of the Fisher kernel representation. In order to globally capture the spatial layout and temporal order, an adaptive spatio-temporal pyramid is introduced to subdivide a depth video into a set of space-time cells. In the extensive experiments, the proposed approach achieves superior performance to the state-of-the-art methods on the four public benchmark datasets, i.e., MSRAction3D, MSRDailyActivity3D, MSRGesture3D, and MSRActionPairs3D.

Index Terms—Human activity recognition, depth camera, feature representation, spatio-temporal information.

1 Introduction

Recognizing human activities has been widely applied to a number of real-world applications, e.g., human-computer interaction [26], surveillance event detection [48], content-based video search and summarization [52], etc. Human activity recognition can be performed at various abstract levels. A *movement* is a primitive motion pattern that can be depicted at the limb level, e.g., right leg forward [30]. An *action* contains a series of atomic movements, e.g., running [16]. An *activity* consists of complex sequence of actions, e.g., a football team scoring a goal [1]. The three levels roughly correspond to the low-level, mid-level, and high-level vision tasks. However, there is no hard boundary but a significant gray area among the three levels. In this paper, we use human activity to indicate the general categories at all three levels, which involve hand gestures, single persons, multiple people, and human-object interactions.

In the past decades, research on human activity recognition mainly focused on recognizing human activities from videos captured by conventional visible light cameras. Along with the advance of imaging techniques, the recently emerged depth sensor brings great advantages to the task of activity recognition. In comparison to conventional color frames in human activity recognition, depth maps have the following merits: (1) additional shape cues to provide more informative geometric description, which has been successfully applied to recover skeleton joints from a single depth map; (2) precluded color and texture, which significantly ease the problems of human detection and segmentation; (3) robustness to variable lightings, which greatly brings benefits to the systems working in a dark environment.

- *X. Yang is with NVIDIA Research, Santa Clara, CA 95050 USA. E-mail: xiaodongy@nvidia.com.*
- *Y. Tian is with Department of Electrical Engineering, City College and Graduate Center, City University of New York, New York, NY 10031 USA. E-mail: ytian@ccny.cuny.edu.*

Manuscript received Apr. 1, 2015; revised Nov. 29, 2015; accepted Apr. 22, 2016.

It was recently shown in [27] [47] 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 frames have quite different properties. The traditional descriptors based on brightness, gradient, and optical flow in color frames might be unsuited to represent the depth maps. It is therefore more desirable to design new feature representations according to the specific characteristics of depth maps, e.g., cloud points [42] and surface normals [27].

In this paper, we propose a novel human activity recognition framework based on the polynormal which is a group of hypersurface normals from a depth sequence. A polynormal clusters the extended surface normals from a local spatio-temporal subvolume. It can be used to jointly capture the local motion and geometry cues. A general feature coding approach [4] [8] [22] [25] is then employed to compute the visual polynormal dictionary and corresponding coefficients. We record the coefficient-weighted differences between polynormals and visual words. These difference vectors are aggregated through spatial average pooling and temporal max pooling for each visual word. The aggregated vectors of all visual words are finally concatenated as a feature vector, which can be seen as a non-probabilistic simplification of the Fisher kernel representation [29]. A further step is to subdivide a depth video into a set of space-time cells. An adaptive spatio-temporal pyramid is proposed to capture the spatial layout and temporal order in a global and flexible manner. The final representation of super normal vector (SNV) is formed by combining the vectors extracted from all the space-time cells.

In an extension to [51], the main contributions of the proposed approach are summarized as follows. First, we introduce the polynormal through assembling hypersurface normals from a local depth subvolume to reserve the correlation between neighboring normals and make them

more resistant against noise than the individual normal [27]. Second, a novel and generalized scheme is proposed to aggregate low-level polynormals into the discriminative representation of SNV. Third, we present the adaptive spatial-temporal pyramid which is better adapted to retain the spatial layout and temporal order than the widely used uniform cells [17] [27] [38]. Fourth, we systematically evaluate the individual components and parameter selections in our framework. Moreover, our approach 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 human activity recognition with depth cameras. Section 3 describes the concept of polynormal. In Section 4, we provide the 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

It is of great challenge to recognize human activities in unconstrained videos due to large intra-class variations caused by factors such as viewpoint, occlusion, motion style, performance duration, etc. It is therefore critical to extract robust representations of spatio-temporal patterns to these variations. Since most representations proposed for color videos have been widely shown to be unsuited for depth sequences, here we focus attention on the related work that are specifically developed for depth videos. A number of representations of human activity in depth sequences have been explored, ranging from skeleton joints [36] [50], cloud points [40] [42], projected depth maps [20] [49], local interest points [10], [47] to surface normals [27] [51].

Biological observations [14] indicate that human activities can be modeled by the motions of skeleton joints. In [50], joint differences were employed to capture the activity cues of static postures, consecutive motions, and overall dynamics. Zanfir et al. [53] proposed the moving pose by using speed, configuration, and acceleration of joints. A skeletal representation was introduced in [36] to model the geometric relationships among various body parts. A dictionary learning and temporal pyramid matching method was proposed in [24] to compute skeleton joint based representations. Relative positions of pairwise joints were adopted in [41] as a complementary feature to characterize the motion information.

Compared to skeleton joints, cloud points are more robust to noise and occlusion. Vieira et al. [37] presented the spatio-temporal occupancy patterns by partitioning a depth sequence to space-time cells and computing corresponding occupancy values. Wang et al. [40] [42] proposed the local and random occupancy patterns to describe depth appearances. In local occupancy patterns [42], they subdivided a local subvolume associated with each skeleton joint into a set of spatial cells and counted the number of cloud points falling into each cell. Similar representations based on cloud points were applied to the subvolumes sampled by a weighted sampling scheme in the random occupancy patterns [40]. A binary range-sample feature based on τ test of cloud points was introduced in [23] to achieve reasonable invariance to geometric change in scale, rotatoin and viewpoint.

Approaches based on projected depth maps usually transform the problem from 4D to 2D. Li et al. [20] sampled a number of 2D points along the contours of projected silhouettes and retrieved 3D points on depth maps according to the selected contour points. Each depth map was finally represented as a bag of 3D points. Our previous work on depth motion maps [49] stacked differences between consecutive projected depth maps from three orthogonal views. HOG was then extracted from the depth motion maps as the global representation of a depth video.

Traditional methods based on local interest points developed for color videos [5] [16] performed poorly on depth sequences. Several local interest point approaches specifically designed for depth maps were recently proposed. DSTIP was introduced in [47] to localize activity-related interest points from depth sequences by suppressing flip noise. Hadfield et al. [10] extended the detection algorithms of Harris corners, Hessian points, and separable filters to the 3.5D and 4D for depth videos.

As demonstrated in [35], the surface normal provides informative shape and structure cues to describe an object in 3D. HON4D [27] followed this observation to extend the surface normal to the 4D space and quantized them by the regular and discriminative learned polychorons. Our approach presented in this paper proceeds along with this direction. It is built upon the polynormal which is a cluster of neighboring hypersurface normals from a local spatio-temporal depth volume. A novel scheme is designed to aggregate the low-level polynormals in each adaptive spatio-temporal cell. The concatenation of feature vectors extracted from all spatio-temporal cells forms the final representation of depth sequences.

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 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 the context of depth sequences, i.e., $m = 4$, each cloud point satisfies $F(x, y, t, z) = f(x, y, t) - z = 0$. Therefore the extended surface normal can be obtained 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 more informative in term of describing object shapes than the distribution of gradient orientations [27]. In addition to the geometric properties encoded in the first two terms, the motion cues are also incorporated in the third term of the normal vector in Eq. (1). In order to retain the correlation between neighboring normals and make them more robust to noise, we propose polynormal to assemble the normals from a local

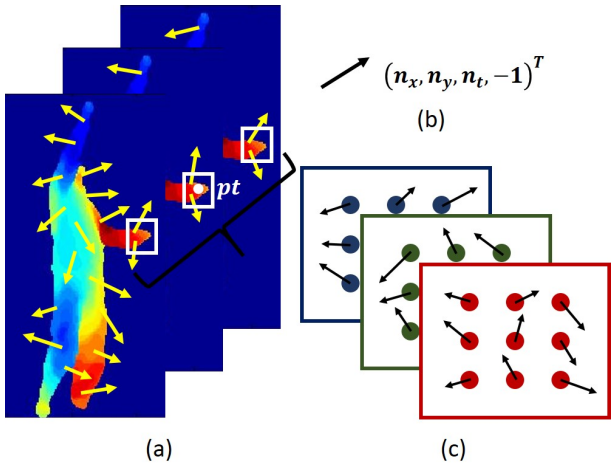


Fig. 1. Illustration of generating the polynormal for a cloud point *pt*. (a) shows a depth sequence of *tennis serve* and normals associated with cloud points. For figure clarity, only a few normals are visualized. The three white squared regions correspond to the neighborhood $\mathcal{L}$. (b) denotes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

space-time neighborhood. Similar ideas have been validated in other fields. For instance, the spatial neighborhood of low-level features are jointly encoded in macrofeatures [2] and convolutional neural networks [19].

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

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

The neighborhood $\mathcal{L}$ is a local 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. Fig. 1 illustrates the concept of polynormal. A short sequence of the activity *tennis serve* is shown in Fig. 1(a). If we set $\mathcal{L}_x = \mathcal{L}_y = \mathcal{L}_t = 3$, then the polynormal of the cloud point *pt* concatenates the 27 normals from the three adjacent depth maps as shown in Fig. 1(c).

4 Computing Super Normal Vector

In this section, we describe the detailed procedures of computing SNV based on the low-level polynormals. Most of recent activity recognition approaches hinge on computing and aggregating statistics of low-level features [39] [40] [47]. In these approaches, a video representation can be obtained by extracting low-level features, coding them over a visual dictionary, and pooling the codes in some well-chosen support regions. In our proposed framework, we compute a visual dictionary and code polynormals by a generalized coding operator. Rather than directly pooling the coefficients of the coded polynormals, we aggregate the weighted differences between polynormals and visual words into a vector. A depth sequence is subdivided into a set of space-time cells by an adaptive spatio-temporal pyramid. The feature vectors extracted from each cell are concatenated as the final representation of SNV.

4.1 Coding and Pooling Polynormals

Our framework is general to various coding and pooling approaches. Here we introduce the notations and outline the related methods on coding and pooling used throughout this paper. We represent a depth video $\mathcal{V}$ by a set of low-level polynormals $\boldsymbol{\mathcal{P}} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$ in $\mathbb{R}^{M \times N}$. $\boldsymbol{\mathcal{D}} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K\}$ is a visual polynormal dictionary with K visual words $\boldsymbol{d}_k \in \mathbb{R}^M$. C indicates a set of spatio-temporal cells with C_j denoting the j-th cell.

Let $\mathcal{G}$ and $\mathcal{C}$ indicate the general coding and pooling operator, respectively. A traditional representation of $\mathcal{V}$ is the vector $\boldsymbol{z}$ obtained by sequentially coding, pooling, and concatenating over all spatio-temporal cells:

$$\boldsymbol{\alpha}_i = \mathcal{G}(\boldsymbol{p}_i), \quad i = 1, \ldots, N, \tag{3}$$

$$\boldsymbol{h}_j = \mathcal{C}\left(\{\boldsymbol{\alpha}_i\}_{i \in C_j}\right), \quad j = 1, \ldots, |C|, \tag{4}$$

$$\boldsymbol{z}^T = \left[\boldsymbol{h}_1^T, \ldots, \boldsymbol{h}_{|C|}^T\right]. \tag{5}$$

In the basic bag-of-visual-words framework [4], hard assignment or vector quantization $\mathcal{G}$ minimizes the distance of $\boldsymbol{p}_i$ to $\boldsymbol{\mathcal{D}}$ which is commonly learned by K-means. $\mathcal{C}$ performs averaging over each spatio-temporal pooling cell C_j:

$$\boldsymbol{\alpha}_i \in \{0,1\}^K, \boldsymbol{\alpha}_{i,j} = 1 \text{ iff } j = \arg\min_k \|\boldsymbol{p}_i - \boldsymbol{d}_k\|_2^2, \tag{6}$$

$$\boldsymbol{h}_j = \frac{1}{|C_j|} \sum_{i \in C_j} \boldsymbol{\alpha}_i. \tag{7}$$

In order to enhance the probability density estimation, soft assignment was introduced in [8]. It codes a polynormal $\boldsymbol{p}_i$ by multiple visual words in $\boldsymbol{\mathcal{D}}$ using a kernel function (e.g., the Gaussian function) of the distance between $\boldsymbol{p}_i$ and $\boldsymbol{d}_k$. Liu et al. proposed local soft assignment in [22] to further improve the membership estimation to all visual words. By taking account of the underlying manifold structure of low-level features, the coding operator $\mathcal{G}$ in local soft assignment only employs the $\mathcal{K}$ nearest visual words $N_{\mathcal{K}}(\boldsymbol{p}_i)$ to code a polynormal $\boldsymbol{p}_i$ and sets its distances of the remaining visual words to infinity:

$$\boldsymbol{\alpha}_{i,k} = \frac{\exp\left(-\beta \hat{d}\left(\boldsymbol{p}_i, \boldsymbol{d}_k\right)\right)}{\sum_{j=1}^{K} \exp\left(-\beta \hat{d}\left(\boldsymbol{p}_i, \boldsymbol{d}_j\right)\right)}, \tag{8}$$

$$\hat{d}\left(\boldsymbol{p}_i, \boldsymbol{d}_k\right) = \begin{cases} \|\boldsymbol{p}_i - \boldsymbol{d}_k\|^2 & \text{if } \boldsymbol{d}_k \in N_{\mathcal{K}}\left(\boldsymbol{p}_i\right), \\ \infty & \text{otherwise,} \end{cases} \tag{9}$$

where β is a smoothing factor to control the softness of assignment. As for the pooling operator $\mathcal{C}$ in local soft assignment, it is observed that max pooling in the following equation outperforms average pooling:

$$\boldsymbol{h}_{j,k} = \max_{i \in C_j} \boldsymbol{\alpha}_{i,k}, \text{ for } k = 1, \ldots, K. \tag{10}$$

On the other hand, parsimony has been widely employed as a guiding principle to compute a sparse representation with respect to an overcomplete visual dictionary.

$\mathcal{G}$ in sparse coding [25] approximates $\boldsymbol{p}_i$ by using a linear combination of a limited number of visual words. It is well known that the ℓ_1-norm penalty yields a sparse solution for $\boldsymbol{\alpha}$. So the sparse coding problem can be solved by:

$$\min_{\boldsymbol{\mathcal{D}},\boldsymbol{\alpha}} \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{2}\|\boldsymbol{p}_i - \boldsymbol{\mathcal{D}}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1\right), \quad (11)$$
$$\text{subject to } \boldsymbol{d}_k^T\boldsymbol{d}_k \leq 1, \forall k = 1,\ldots,K,$$

where λ is the sparsity-inducing regularizer to control the number of non-zero coefficients in $\boldsymbol{\alpha}_i$. It is customary to combine sparse coding with max pooling in Eq. (10).

Fisher vector [29] extends the bag-of-visual-words representation by recording the deviation of $\boldsymbol{p}_i$ with respect to the parameters of a generative model, e.g., the Gaussian mixture model (GMM): $G_\xi(\boldsymbol{p}_i) = \sum \pi_k G_k(\boldsymbol{p}_i)$. We denote $\xi = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, k = 1,\ldots,K\}$ where π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\sigma}_k$ are the prior mode probability, mean vector, and covariance matrix (diagonal), respectively. Let $\boldsymbol{\gamma}_{i,k}$ be the soft assignment of $\boldsymbol{p}_i$ to the k-th Gaussian component:

$$\boldsymbol{\gamma}_{i,k} = \frac{\pi_k G_k(\boldsymbol{p}_i)}{\sum_{j=1}^{K}\pi_j G_j(\boldsymbol{p}_i)}. \quad (12)$$

We obtain the Fisher vector of $\mathcal{V}$ by concatenating the gradient vectors from each Gaussian components:

$$\boldsymbol{\rho}_k = \frac{1}{N\sqrt{\pi_k}}\sum_{i=1}^{N}\gamma_{i,k}\left(\frac{\boldsymbol{p}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k}\right), \quad (13)$$

$$\boldsymbol{\tau}_k = \frac{1}{N\sqrt{2\pi_k}}\sum_{i=1}^{N}\gamma_{i,k}\left[\frac{(\boldsymbol{p}_i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1\right], \quad (14)$$

where $\boldsymbol{\rho}_k$ and $\boldsymbol{\tau}_k$ are M-dimensional gradients with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ of the k-th Gaussian component. The relative displacements of low-level features to the mean and covariance in Eq. (13-14) retain more information lost in the traditional coding process. The superiority of Fisher vector was recently identified in both image classification [32] and human activity recognition [43].

4.2 Aggregating Polynormals

After coding low-level features over a visual dictionary, they are usually removed in the representation pipeline [4] [22] [25]. In our framework, we keep the low-level features by recording the differences between polynormals and visual words. As demonstrated in [13] [29] [54], the relative displacements are able to provide extra distribution information of the low-level features.

Given the low-level polynormals $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{M\times N}$ of a depth sequence $\mathcal{V}$, they are first transformed by a generalized coding operator $\mathcal{G}$ to the decomposition coefficients $\mathcal{G}(\boldsymbol{p}_i)_{i=1}^N$ with $\mathcal{G}(\boldsymbol{p}_i) \in \mathbb{R}^K$. Each $\mathcal{G}(\boldsymbol{p}_i)$ is then ℓ_1-normalized to obtain the assignment $\mathcal{G}(\boldsymbol{p}_i)_k$ of the polynormal $\boldsymbol{p}_i$ to the k-th visual word $\boldsymbol{d}_k$. The size of a volume (depth sequence) where we perform the aggregation is $H \times W$ pixels and T frames. This volume can correspond to either the entire video sequence or a subsequence defined by a space-time cell. We denote by N_t the set of indices of polynormals within the frame t. For each visual word $\boldsymbol{d}_k$, the spatial average pooling is first applied to aggregate the coefficient-weighted differences:

$$\boldsymbol{u}_k(t) = \frac{1}{|N_t|}\sum_{i\in N_t}\mathcal{G}(\boldsymbol{p}_i)_k\,(\boldsymbol{p}_i - \boldsymbol{d}_k), \quad (15)$$

where $\boldsymbol{u}_k(t)$ represents the pooled difference vector of the k-th visual word in the t-th frame. The temporal max pooling is then employed to aggregate the vectors from the entire volume containing T frames:

$$\boldsymbol{u}_{k,i} = \max_{t=1,\ldots,T}\boldsymbol{u}_{k,i}(t), \text{ for } i = 1,\ldots,M, \quad (16)$$

where $\boldsymbol{u}_k$ is the vector representation of the k-th visual word in the whole volume; i indicates the i-th 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:

$$\boldsymbol{U} = \left(\boldsymbol{u}_1^T,\ldots,\boldsymbol{u}_K^T\right)^T. \quad (17)$$

In order to globally capture the spatial layout and temporal order, a depth sequence is subdivided into a set of space-time cells by the proposed adaptive spatio-temporal pyramid (Section 4.4). We extract a feature vector $\boldsymbol{U}$ from each cell and concatenate them as SNV. This representation has several remarkable properties. (1) The displacements between low-level polynormals to visual words retain more information that could be lost in the feature quantization process. (2) We can compute SNV on a much smaller dictionary size (e.g., $K = 100$) which largely reduces the computational cost. (3) SNV utilizes a generalized coding operator which is more efficient and flexible. (4) The spatial average and temporal max pooling is more effective to aggregate low-level and mid-level features for videos. (5) SNV performs quite well with simple linear classifiers which are efficient in terms of both training and testing.

4.3 Relationship with Fisher Kernel

We now demonstrate that the proposed SNV is a simplified non-probabilistic version of the Fisher kernel representation which has been successfully applied in the image and video classification tasks [32] [43]. Fisher kernel assumes that low-level features are distributed according to a generative model such as GMM.

It is shown in [12] that the gradients of the log-likelihood of GMM with respect to the parameters describe the contributions of these parameters to the generation process of low-level features. Here we focus on the gradient with respect to the mean $\boldsymbol{\mu}_k$ to the k-th Gaussian in Eq. (13). Let N_t denote a general pooling region in this context and $|N_t|$ the number of low-level features in this region. If making the following approximations for each Gaussian component:

- the prior mode probabilities are uniformly distributed, i.e., $\pi_k = 1/K$,
- the soft assignment can be estimated by the coding coefficient, i.e., $\boldsymbol{\gamma}_{i,k} = \mathcal{G}(\boldsymbol{p}_i)_k$,
- the mean can be represented by the visual word, i.e., $\boldsymbol{\mu}_k = \boldsymbol{d}_k$,
- the covariance matrix is isotropic, i.e., $\boldsymbol{\sigma}_k = \epsilon\mathbb{I}$ and $\epsilon > 0$,

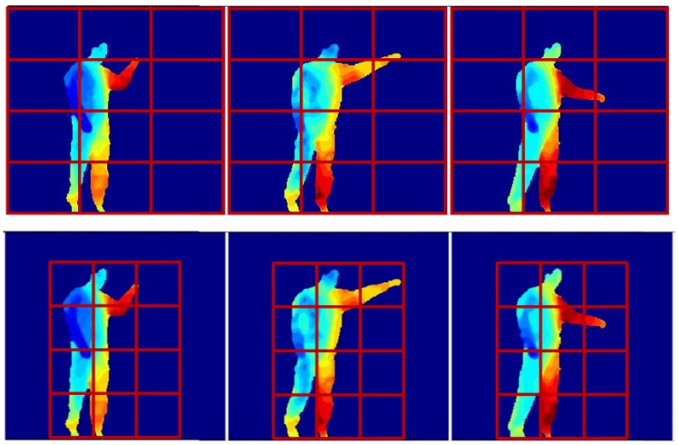


Fig. 2. Comparison between the traditional (top) and our proposed (bottom) spatial grids. We place the 4×3 spatial grid on the largest bounding box of the human body instead of the entire frame.

we can simplify Eq. (13) to

$$\boldsymbol{\rho}_k \propto \frac{1}{|N_t|} \sum_{i \in N_t} \mathcal{G}(\boldsymbol{p}_i)_k \left(\boldsymbol{p}_i - \boldsymbol{d}_k\right). \tag{18}$$

If comparing Eq. (15) and Eq. (18), we find that the two representations are in the same form. A generalized coding operator $\mathcal{G}$ can be utilized in the proposed framework, while GMM is used in the Fisher kernel representation. We choose a general coding approach over GMM in our aggregation scheme because it is more efficient to compute the centers (i.e., the visual dictionary), in particular, it was recently demonstrated in [3] that a reasonably good visual dictionary can be created by some simple methods, for instance, random sampling from a training set. In addition, our empirical evaluations demonstrate that the representation based on a general coding approach achieves improved or competitive recognition results.

In image search and retrieval, the vector of locally aggregated descriptors (VLAD) [13] was proposed to encode descriptors with respect to the visual words that they are quantized to. VLAD can be seen as a simplified version of the Fisher kernel representation as well. However, SNV differs from VLAD in the three main aspects. First, the aggregation scheme in SNV is based on a general coding approach which can be hard assignment, (local) soft assignment, or sparse coding. In contrast, VLAD only hinges on hard assignment. So the aggregation technique in VLAD can be treated as a special case in our framework. Second, the difference vectors of SNV in Eq. (15) are weighted by the coding coefficients, while no weighting is used in VLAD. Third, SNV applies both spatial average pooling and temporal max pooling in Eq. (15-16), conversely, difference vectors are simply summed up in VLAD.

4.4 Adaptive Spatio-Temporal Pyramid

The aforementioned representation of polynormals is orderless and therefore ignores the spatial layout and temporal order of a depth sequence, which could have conveyed discriminative information for human activity recognition. The dominant approach of incorporating the spatial and temporal information is the spatio-temporal pyramid [17], which repeatedly partitions a video sequence into a set of space-time cells through a coarse-to-fine manner. Each cell is then represented independently and the cell-level histograms $\boldsymbol{h}_j$ are finally concatenated into the video-level histogram $\boldsymbol{z}$ as in Eq. (4-5). Here we propose an adaptive spatio-temporal pyramid to retain the spatio-temporal cues in a more flexible way.

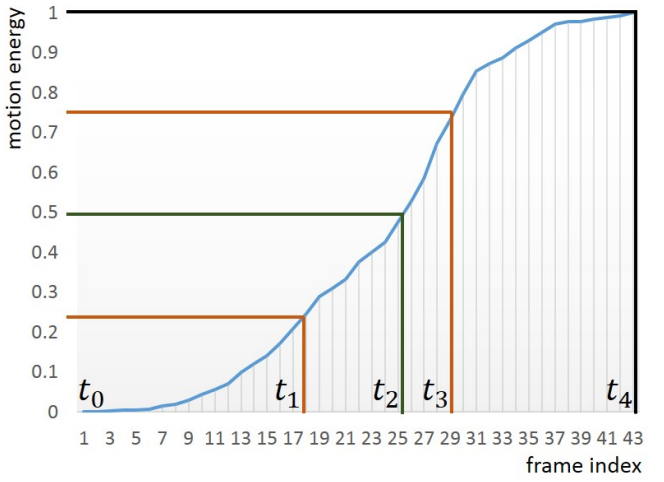


Fig. 3. The frame index and associated motion energy used to build the adaptive temporal pyramid. The temporal segments are generated by repeatedly and evenly subdividing the normalized motion energy axis instead of the time axis.

In the spatial dimensions, we place a $n_H \times n_W$ grid to capture the spatial layout. Since the depth maps greatly facilitate human detection and segmentation, we put the spatial grid on the largest bounding box of a human body across the whole video sequence, instead of on the entire frame as commonly used in traditional pyramid methods [17] [27] [38]. This makes each cell contain more foreground information as shown in Fig. 2.

In the temporal dimension, a pyramid based representation was introduced by Laptev et al. [17] to take into account the rough temporal order of a video sequence. It was also widely employed in depth sequences [27] [42] to incorporate cues from the temporal context. In these methods, a video sequence (either color or depth) is repeatedly and evenly subdivided into a series of temporal segments where the descriptor-level statistics are pooled. However, different people could have varied motion speed or frequency even when they are performing the same action. It is therefore inflexible to handle these variations by evenly subdividing a video along the time axis. In addition, it is more desirable to pool the low-level features within the similar action modes which usually contain the neutral, onset, apex, and offset statuses [9]. In order to deal with these difficulties, we propose the adaptive temporal pyramid based on the motion energy.

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

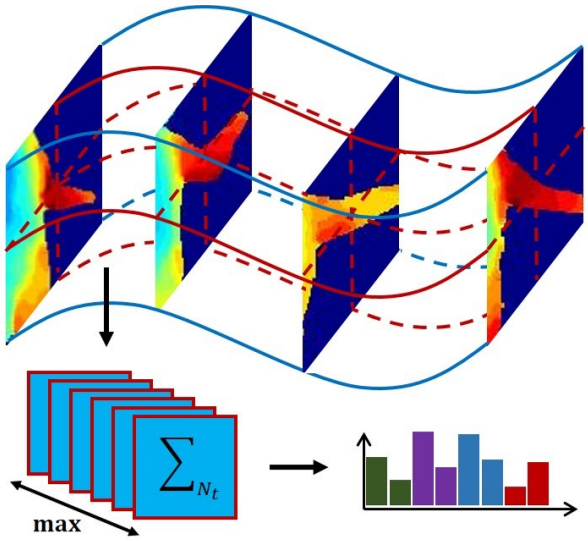


Fig. 4. SNV based on the skeleton joint trajectory. The trajectory-aligned volume is subdivided into a set of space-time cells according to the adaptive spatio-temporal pyramid. Each cell generates a feature vector of SNV by the spatial average pooling and temporal max pooling.

non-zero elements of the binary maps as:

$$\varepsilon(i) = \sum_{v=1}^{3} \sum_{j=1}^{i-1} \text{sum}\left(\left|\boldsymbol{I}_v^{j+1} - \boldsymbol{I}_v^{j}\right| > \eta\right), \tag{19}$$

where $\varepsilon(i)$ is the motion energy of the i-th frame; η is the threshold; $\text{sum}(\cdot)$ returns the summation of non-zero elements in a binary map. The motion energy of a frame reflects its relative motion status with respect to the entire activity.

As shown in Fig. 3, our proposed motion energy based adaptive temporal pyramid evenly subdivides the normalized motion energy axis into several temporal 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 $\{t_0t_1, t_1t_2, t_2t_3, t_3t_4\}$. Together with the spatial grid, our adaptive spatio-temporal pyramid in total generates $n_H \times n_W \times 7$ space-time cells.

4.5 Joint Trajectory Aligned SNV

While the framework discussed above operates on the entire depth sequence, our approach is also flexible to combine with the recovered skeleton joints [34] to compute SNV by using each joint trajectory. This is useful in the scenario 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 spatio-temporal volume aligned around each joint trajectory. It was also shown in the dense trajectories [38] that descriptors aligned with trajectories were superior to those computed from straight cuboids.

As shown in Fig. 4, the volume aligned with a joint trajectory can be viewed as a single video sequence with T frames and each frame has $H \times W$ pixels. 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 coefficient-weighted difference vectors as in Eq. (15-16). These vectors computed 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.

We summarize the outline of computing SNV of a depth sequence in Algorithm 1. The depth sequence $\mathcal{V}$ can be either a whole video sequence or a volume aligned with a joint trajectory. The coding operator $\mathcal{G}$ applied in the aggregation is a general coding approach which can be hard assignment in Eq. (6), soft assignment in the global version of Eq. (8), local soft assignment in Eq. (8-9), or sparse coding in Eq. (11). The set of space-time cells V are determined by the proposed motion energy based adaptive spatio-temporal pyramid.

Algorithm 1: Computation of SNV

Input: a depth sequence $\mathcal{V}$
a coding operator $\mathcal{G}$
a dictionary $\boldsymbol{\mathcal{D}} = (\boldsymbol{d}_k)_{k=1}^{K}$
a set of space-time cells $V = \{v_i\}$
Output: SNV

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

5 Experimental Results

In this section we extensively evaluate the proposed approach on four public benchmark datasets: MSRAction3D [20], MSRGesture3D [40], MSRActionPairs3D [27], and MSRDailyActivity3D [41]. In all experiments, we set the adaptive spatio-temporal pyramid to be of $4 \times 3 \times 7$ space-time cells in height, width, and time, respectively. We employ LIBLINEAR [7] as the linear SVM solver. Our method is extensively compared to the 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 sequences. Experimental results demonstrate that our algorithm significantly outperforms the state-of-the-art methods on the four benchmark datasets. Our source code of computing SNV is available online at http://media-lab.engr.ccny.cuny.edu/data-code/.

5.1 Evaluation of SNV Parameters

Our systematic evaluations of the parameters and settings in SNV are conducted on the MSRAction3D dataset. In these evaluations, the sparse coding approach in Eq. (11) is used as the coding operator $\mathcal{G}$, and the number of visual words is

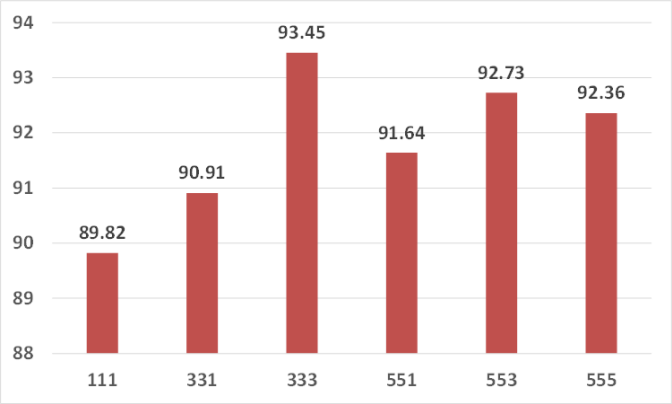


Fig. 5. Recognition accuracies (%) of SNV using different sizes $\mathcal{L}_x\mathcal{L}_y\mathcal{L}_t$ of a local spatio-temporal neighborhood $\mathcal{L}$ to form the polynormal.

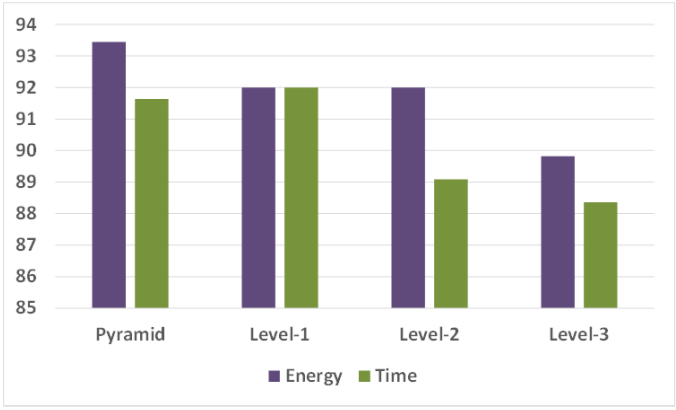


Fig. 7. Comparison of recognition accuracies (%) between the proposed adaptive spatio-temporal pyramid based on motion energy and the traditional pyramid based on time.

empirically set to $K = 100$. We observe that the recognition accuracy is quite stable with respect to K ranging from 50 to 200. Note this number is order of magnitude smaller than the ones used in most visual recognition systems [11] [38].

Polynormal Size: We first evaluate the effects of the size of local neighborhood $\mathcal{L}$ to form a polynormal. As discussed in Section 3, the size of $\mathcal{L}$ is determined by $\mathcal{L}_x \times \mathcal{L}_y \times \mathcal{L}_t$. Fig. 5 shows the recognition accuracies of SNV built by different sizes of $\mathcal{L}$. If no local temporal information is embedded in the polynormal, i.e., $\mathcal{L}_t = 1$, increasing the spatial size of $\mathcal{L}$ improves the performance, e.g., from $1 \times 1 \times 1$, $3 \times 3 \times 1$, to $5 \times 5 \times 1$. This benefits from the correlated local geometric cues provided by the spatial neighborhood of the extended normals. When $\mathcal{L}_x$ and $\mathcal{L}_y$ are fixed, the recognition results with $\mathcal{L}_t > 1$ outperforms the ones with $\mathcal{L}_t = 1$, e.g., the recognition accuracy of $3 \times 3 \times 3$ is much higher than the one of $3 \times 3 \times 1$. In addition, the overall performance of polynormal is superior to that of individual normal. This shows that the jointly encoded spatial and temporal cues in the polynormal are powerful to characterize the low-level motion and shape information. In the following experiments, we employ the $3 \times 3 \times 3$ local neighborhood $\mathcal{L}$ to form the polynormals.

Pooling Strategy: We apply the spatial average pooing and temporal max pooling to aggregate the low-level polynormals, where spatial pooling is local (from one frame) and temporal pooling is global (across whole video). The spatial average pooling in Eq. (15) summarizes the coefficient-weighted difference vectors within each depth map. This spatial pooling in local averages the low-level elementary information to mitigate noise. The temporal max pooling in Eq. (16) records the best response of the intermediate representations from each frame. This temporal pooling in global spotlights the mid-level leading cues to reinforce discrimination. Other pooling combinations will impair these effects, e.g., max pooling in spatial amplifies noise and average pooling in temporal undermines discrimination. This design can be also validated from the evaluation results. Fig. 6 compares the performances of different combinations of spatial/temporal and average/max poolings. As demonstrated in this figure, the proposed pooling strategy significantly outperforms others. Hence the appropriate choice of pooling strategy in spatial and temporal is vital to obtain effective feature representations for video sequences.

Adaptive Pyramid: We compare the performance of the proposed adaptive spatio-temporal pyramid to the traditional spatio-temporal pyramid in Fig. 7. Our adaptive pyramid subdivides a video sequence according to the motion energy in Eq. (19) which describes the relative status of a

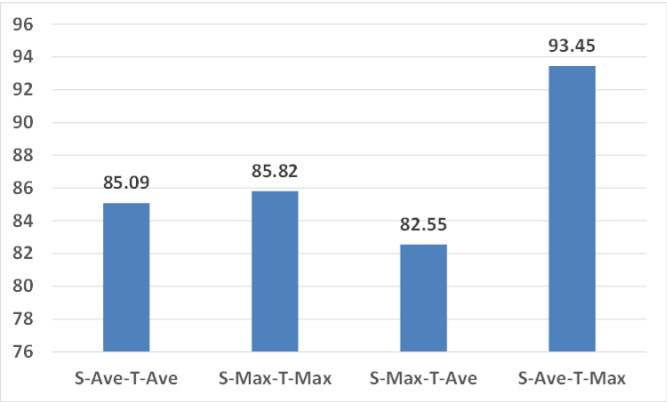


Fig. 6. Recognition accuracies (%) of SNV with different combinations of spatial (S) / temporal (T) and average (Ave) / max (Max) pooling.

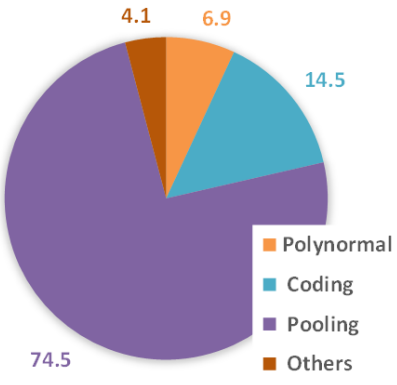


Fig. 8. Percentage of the time spent on each major step in computing SNV with the default parameter setting.

TABLE 1
Recognition Accuracy Comparison on the MSRAction3D Dataset.

Method	Accuracy
Bag of 3D Points [20]	74.70%
HOJ3D [46]	79.00%
EigenJoints [50]	82.30%
STOP [37]	84.80%
Random Occupancy Pattern [40]	86.50%
Actionlet Ensemble [42]	88.20%
Depth Motion Maps [49]	88.73%
Histogram of Depth Gradients [31]	88.82%
HON4D [27]	88.89%
DSTIP [47]	89.30%
Lie Group [36]	89.48%
Pose Set [44]	90.00%
Efficient Pose [6]	90.10%
Moving Pose [53]	91.70%
SNV-Hard	90.91%
SNV-Soft	90.55%
SNV-LocalSoft	**93.45%**
SNV-Sparse	**93.45%**

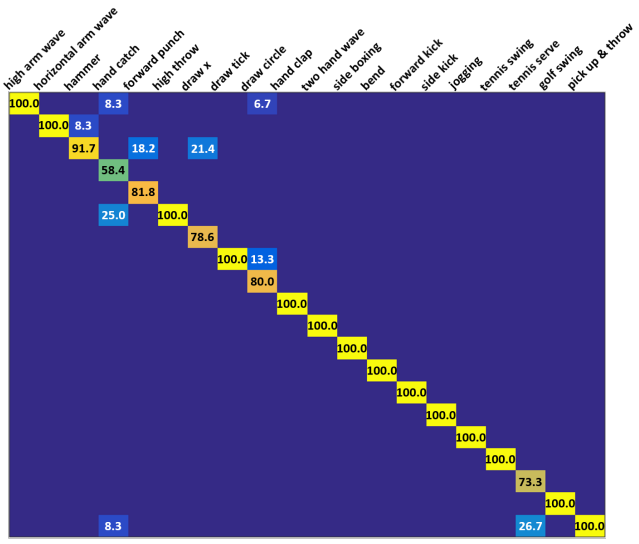


Fig. 9. Confusion matrix of SNV on the MSRAction3D dataset. This figure is better viewed on screen.

depth frame with respect to the whole activity, while the traditional pyramid makes segments based on the time. As demonstrated in the figure, the two methods have the same performance in level-1 since they both operate on the same temporal segment which is the entire video sequence. In level-2 and level-3, our motion energy based approach largely outperforms the traditional time based method. When the three levels are combined into the pyramid representation, our adaptive pyramid achieves over 1.8% improvement to the traditional pyramid. This evaluation result demonstrates that the proposed adaptive pyramid is better adapted to handle the motion variations and capture the global temporal order.

Computational Complexity: To analyze the computational complexity of SNV, we compute SNV from 191 depth sequences with the resolution of 320×240 pixels. These video clips correspond to a total of 7,776 frames. We report the run time using MATLAB on a desktop with 2.13GHz CPU and 24G RAM. With intention of having an explicit comparison of various steps, the adaptive pyramid is not considered in this evaluation. The average computational speed is 0.2 frames per second. Fig. 8 shows the percentage of time spent on each major step in the computation of SNV. The pooling process which involves spatial-average and temporal-max pooling takes most of the time with 74.5%. The sparse coding process is the second most time-consuming step with 14.5%. It only takes 6.9% to compute polynormals. The run time can be improved by reducing the densely sampled cloud points or replacing sparse coding in Eq. (11) with a more computationally efficient coding approach, e.g., hard assignment in Eq. (6) or local soft assignment in Eq. (8-9).

In the following, we extensively compare and analyze the performances of SNV and the state-of-the-art methods on the four benchmark datasets. We use SNV-Hard, SNV-Soft, SNV-LocalSoft, and SNV-Sparse in the experiments to indicate SNV based on hard assignment, soft assignment, local soft assignment, and sparse coding in the general coding operator $\mathcal{G}$, respectively.

5.2 Results on MSRAction3D Dataset

The MSRAction3D [20] is an action dataset of depth sequences captured by a depth camera. It contains 20 action categories: *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*, and *pick & throw*. Each action is performed 2 or 3 times by 10 subjects facing the camera. These 20 action categories are selected in the context of gaming and cover a variety of motions related to arms, legs, torso, etc.

In order to facilitate a fair comparison, we follow the same experimental setting as [42]. As shown in Table 1, SNV-Hard and SNV-Soft improves the accuracy over most existing methods. SNV-Sparse and SNV-LocalSoft both achieve the accuracy of 93.45% which significantly outperforms the previous methods. This demonstrates that local soft assignment and sparse coding provide more discriminative weighted coefficients in Eq. (15) than hard assignment and soft assignment on this task. If we only keep the first level, i.e., $\{t_0 t_4\}$ in Fig. 3, of the adaptive temporal pyramid, the accuracy decreases to 92.00%. This validates that the recognition performance could benefit from the temporal cues in the global context. We also compute the Fisher vector representation of our proposed polynormal. It achieves an accuracy of 92.73%, 0.72% inferior to SNV. We conjecture this improvement is due to the robust coding coefficient and simplified clustering estimation. The confusion matrix of SNV-Sparse is displayed in Fig. 9. Our approach performs very well on most action categories. The recognition errors occur on very similar actions, e.g., *hand catch* and *high throw*, *draw circle* and *draw tick*.

We compare the performance of SNV with other published results in Table 1. The methods solely based on skeleton joints are vulnerable to the errors of recovered joints due to severe self-occlusions. So the model in [44] selects the best-k joint configurations which largely remove inaccurate

TABLE 2
Recognition Accuracy Comparison on the MSRGesture3D Dataset.

Method	Accuracy
Action Graph on Occupancy [15]	80.50%
Action Graph on Silhouette [15]	87.70%
Random Occupancy Pattern [40]	88.50%
Depth Motion Maps [49]	89.20%
HON4D [27]	92.45%
HOG2 [28]	92.64%
Histogram of Depth Gradients [31]	92.76%
SNV-Hard	**94.58%**
SNV-Soft	91.50%
SNV-LocalSoft	**94.44%**
SNV-Sparse	**94.74%**

TABLE 3
Recognition Accuracy Comparison on the MSRActionPairs3D Dataset.

Method	Accuracy
Skeleton + LOP [41]	63.33%
Depth Motion Maps [49]	66.11%
Histogram of Depth Gradients [31]	74.40%
Skeleton + LOP + Pyramid [41]	82.22%
HON4D [27]	96.67%
MMTW [45]	97.22%
SNV-Hard	**100.00%**
SNV-Soft	93.33%
SNV-LocalSoft	96.67%
SNV-Sparse	98.89%

joints. The approach in [53] makes use of pose, speed, and acceleration of skeleton joints. It is able to perform one-shot learning and low-latency recognition. While still inferior to our approach, the methods in [37] [40] [42] improve the results over [46] [50] because cloud points are more resistant to occlusions and provide additional shape and appearance cues compared to skeleton joints. SNV outperforms HON4D [27] by 4.56%, though both methods are based on the extended surface normals. This is mainly because (1) polynormals obtain more discriminative local motion and appearance information than individual normals; (2) the coding operator 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 simple summation of inner production values; (4) the adaptive pyramid is more flexible than the uniform cells to capture the global spatio-temporal cues.

5.3 Results on MSRGesture3D Dataset

The MSRGesture3D [40] 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) including *where*, *store*, *pig*, *past*, *hungry*, *green*, *finish*, *blue*, *bathroom*, *z*, *j*, and *milk*. Most of these hand gestures are performed with one hand except *store* and *finish* with two hands. Each dynamic hand gesture is performed 2 or 3 times by 10 subjects. Compared to static hand gestures, both motion and shape convey semantics in dynamic hand gestures. Due to the articulated nature of hand structure, this dataset presents strong self-occlusions.

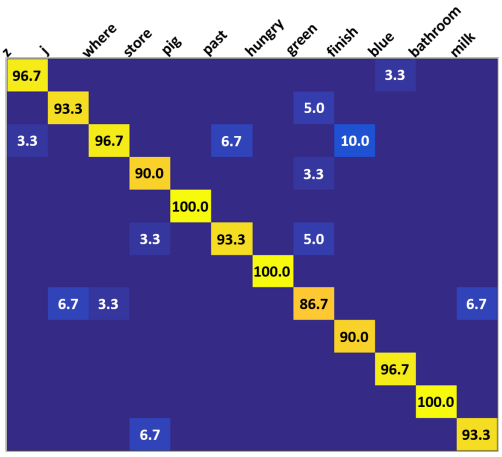


Fig. 10. Confusion matrix of SNV on the MSRGesture3D dataset. This figure is better viewed on screen.

The leave-one-out cross-validation scheme as [40] is used in our evaluation. As shown in Table 2, SNV-Hard, SNV-LocaSoft, and SNV-Sparse obtain the comparable and state-of-the-art results which outperform previous methods. We argue that the inferiority of SNV-Soft is due to the neglect of the underlying manifold structure of low-level polynormals in the global soft assignment of Eq. (8). The polynormal based Fisher vector achieves an accuracy of 95.83%, but with the cost of much higher computational complexity. GMM in Fisher vector is probably able to model polynormal distribution well in hand gestures which have less variations than human actions. The confusion matrix of SNV-Sparse is displayed in Fig. 10. Our approach performs well on most dynamic gestures. Most confusions occur in recognizing the gesture *green* which shares very similar motions to *j* only with different fingers. Since the estimation of hand skeleton joints is not available, the joint-based methods [36] [44] [50] [53] cannot be used in this application.

5.4 Results on MSRActionPairs3D Dataset

The MSRActionPairs3D [27] is a paired-activity dataset of depth sequences captured by a depth camera. It contains 12 activities (i.e., 6 pairs): *pick up a box*, *put down a box*, *pull a chair*, *push a chair*, *wear a hat*, *take off a hat*, *take on a backpack*, *take off a backpack*, *stick a poster*, *remove a poster*, *lift a box*, and *place a box*. Each activity is performed 3 times by 10 subjects. This dataset is collected to mainly investigate how the temporal information influences recognition results and how the motion and shape cues are correlated in human activities.

We follow the same evaluation setup as [27] in our experiment. As shown in Table 3, SNV-Hard and SNV-Sparse achieve the state-of-the-art accuracies. It is interesting to observe that SNV based on the simple coding operator of hard assignment in Eq. (6) obtains this excellent performance. The polynormal based Fisher vector obtains an accuracy of 98.33%. These promising results benefit from our robust modeling of the temporal information. In the framework of SNV, the chronological orders are embedded

TABLE 4
Recognition Accuracy Comparison on the MSRDailyActivity3D Dataset.

Method	Accuracy
LOP [42]	42.50%
Depth Motion Maps [49]	43.13%
EigenJoints [50]	58.10%
Joint Positions [42]	68.00%
NBNN + Parts + Time [33]	70.00%
RGGP [21]	72.10%
Efficient Pose [6]	73.10%
Moving Pose [53]	73.80%
Histogram of Depth Gradients [31]	74.45%
Local HON4D [27]	80.00%
Actionlet Ensemble [42]	85.75%
SNV-Hard	85.62%
SNV-Soft	81.25%
SNV-LocalSoft	85.00%
SNV-Sparse	**86.25%**

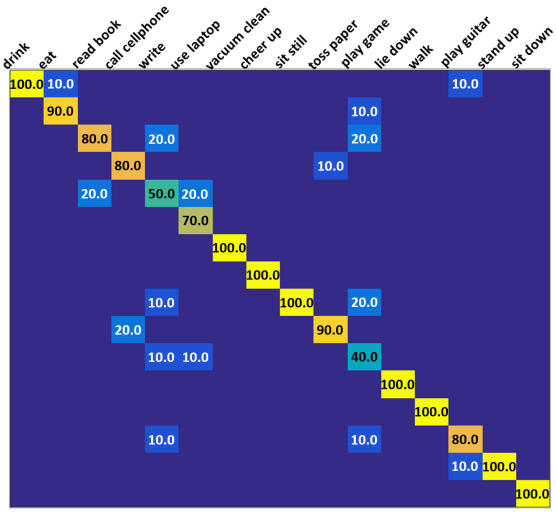


Fig. 11. Confusion matrix of SNV on the MSRDailyActivity3D dataset. This figure is better viewed on screen.

in three levels. In the low-level, the extended surface normal in Eq. (1) incorporates the temporal deviation between adjacent frames. In the mid-level, the polynormal in Fig. 1 further encodes the local temporal cues. In the high-level, the adaptive temporal pyramid in Fig. 3 captures the global temporal order. If no adaptive temporal pyramid is used, SNV-Sparse still achieves an accuracy of 97.78%. This further demonstrates that the local temporal cues enclosed in the polynormal already well reflect the chronological orders. It is therefore crucial to capture the temporal information in order to distinguish the activities with similar motions but different chronological orders.

Comparisons to other methods are shown in Table 3. The skeleton feature [42] involves pair-wise difference of joint positions within each frame. The LOP feature [42] is used to characterize the shape. It counts the number of cloud points falling into each spatial grid of a depth subvolume. No temporal information is encoded in the two features. In the depth motion maps [49], depth sequences are collapsed onto three projected maps where temporal orders are eliminated. These methods therefore suffer the inner-paired confusions. The skeleton and LOP features equipped with a uniform temporal pyramid improve the recognition result as the global temporal order is incorporated. However, this result is still significantly inferior to ours. Because of the high recognition accuracy, the confusion matrix on this dataset is omitted.

5.5 Results on MSRDailyActivity3D Dataset

The MSRDailyActivity3D [42] is a daily activity dataset of depth sequences captured by a depth camera. It includes 16 daily activities performed by 10 subjects: *drink*, *eat*, *read book*, *call cellphone*, *write*, *use laptop*, *vacuum clean*, *cheer up*, *sit still*, *toss paper*, *play game*, *lie down*, *walk*, *play guitar*, *stand up*, and *sit down*. Each subject performs each activity twice, one in standing position and the other in sitting position. Compared to the other three datasets, people in this dataset present large spatial and scaling changes. Moreover, most activities involve human-object interactions.

In order to handle the large spatial and scaling variations, we employ the joint trajectory aligned SNV on this dataset. Each joint is tracked through the whole depth sequence. A support region or depth patch is associated with each joint in each frame. Since the depth value inversely changes with the object size, we set an adaptive size s/z to each depth patch, where $s = 300,000$ is a scaling factor and z is the depth value of a joint in the current frame. Unlike the fixed patch size, the adaptive size is more robust to handle the spatial and scaling variations. So the patch size in Fig. 4 is not necessary to be consistent. We compute SNV and joint differences for each joint trajectory. The actionlet ensemble model [41] is then employed to combine the features from multiple joint trajectories.

We follow the same experimental setting as [42] and obtain the state-of-the-art accuracy of 86.25%. The confusion matrix of SNV-Sparse is displayed in Fig. 11. Most recognition errors occur in those almost still activities, e.g., *read book*, *write*, and *use laptop*, which contain very subtle motions. Since most daily activities involve human-object interactions, this dataset can be also used to evaluate how the motion and shape characteristics are correlated. It would be insufficient to capture the motion or shape information independently because some activities share quite similar motion cues but present distinct shape properties. SNV jointly encodes local motions and geometric shapes in the polynormals which further reflects the co-occurrence of human motions and object shapes. Accuracy of Fisher vector equipped with spatio-temporal pyramid is 76.88% which is 9.37% inferior to our best result. This demonstrates the advantages of our proposed feature aggregation method and the aligned joint trajectory strategy.

Table 4 shows the performance comparison of the proposed method to the previous ones. Note: an accuracy of 88.20% was reported in [47]. However, four activities with less motion (i.e., *sit still*, *read books*, *write on paper*, and *use laptop*) were excluded in their experiments. The holistic approach [49] suffers the non-aligned sequences. The methods [21] [33] [42] [50] [53] based on either motion or shape information alone are significantly inferior to our approach and the ones [27] [42] that jointly model the two cues.

6 Conclusion

We present a novel framework to recognize human activities from video sequences captured by depth cameras. We have proposed the low-level feature of polynormal to jointly model the local motion and shape cues. A new aggregation scheme is proposed by a general coding operator, as well as spatial average pooling and temporal max pooling of the coefficient-weighted differences between polynormals and visual words. We have introduced the motion energy based adaptive spatial-temporal pyramid which can be better adapted to retain the spatial layout and temporal orders. Our proposed framework is also flexible to be used in the joint trajectory aligned depth sequence. This is well suited in the scenarios where significant spatial and scaling variations present. Our approach is extensively evaluated on four public benchmark datasets and compared to a number of state-of-the-art methods. Experimental results demonstrate that our approach significantly outperforms previous methods on these datasets. The future work will focus on exploiting the complementary information and fusing multiple features from both color and depth channels for more advanced representations.

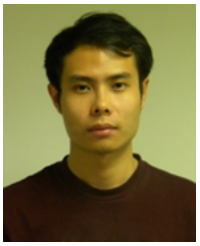
Acknowledgments

This work was supported in part by NSF Grants EFRI-1137172 and IIS-1400802.

References

[1] S. Bhattacharya, M. Kalayeh, R. Sukthankar, and M. Shah, "Recognition of Complex Events: Exploiting Temporal Dynamics between Underlying Concepts", *CVPR*, 2014.
[2] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning Mid-Level Features for Recognition", *CVPR*, 2010.
[3] A. Coates and A. Ng, "The Importance of Encoding versus Training with Sparse Coding and Vector Quantization", *ICML*, 2011.
[4] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, "Visual Categorization with Bags of Keypoints", *ECCV Workshop on Statistical Learning in Comptuer Vision*, 2004.
[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition via Sparse Spatio-Temporal Features", *PETS*, 2005.
[6] A. Eweiwi, M. Cheema, C. Bauckhage, and J. Gall, "Efficient Pose-based Action Recognition", *ACCV*, 2014.
[7] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A Library for Large Linear Classification", *Journal of Machine Learning Research*, 2008.
[8] J. Gemert, C. Veenman, A. Smeulders, J. Geusebroek, "Visual Word Ambiguity", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2009.
[9] H. Gunes and M. Piccardi, "Automatic Temporal Segment Detection and Affect Recognition from Face and Body Display", *IEEE Trans. Systems, Man, and Cybernetics - Part B: Cybernetics*, 2009.
[10] S. Hadfield and R. Bowden, "Hollywood 3D: Recognizing Actions in 3D Natural Scenes", *CVPR*, 2013.
[11] Y. Huang, Z. Wu, L. Wang, and T. Tan, "Feature Coding in Image Classification: A Comprehensive Study", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2014.
[12] T. Jaakkola and D. Haussler, "Exploiting Generative Models in Discriminative Classifiers", *NIPS*, 1998.
[13] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating Local Descriptors into a Compact Image Representation", *CVPR*, 2010.
[14] G. Johansson, "Visual Perception of Biological Motion and a Model for Its Analysis", *Perception & Psychophysics*, 1973.
[15] A. Kurakin, Z. Zhang, and Z. Liu, "A Read-Time System for Dynamic Hand Gesture Recognition with A Depth Sensor", *EUSIPCO*, 2012.
[16] I. Laptev, "On Space-Time Interst Points", *International Journal on Computer Vision*, 2005.
[17] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning Realistic Human Actions from Movies", *CVPR*, 2008.
[18] S. Lazebnik, C. Schmid, J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories", *CVPR*, 2006.
[19] H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representation", *ICML*, 2009.
[20] W. Li, Z. Zhang, and Z. Liu, "Action Recognition based on a Bag of 3D Points", *CVPR Workshop on Human Communicative Behavior Analysis*, 2010.
[21] L. Liu and L. Shao, "Learning Discriminative Representations from RGB-D Video Data", *2013*.
[22] L. Liu, L. Wang, and X. Liu, "In Defense of Soft-Assignment Coding", *ICCV*, 2011.
[23] C. Lu, J. Jia, and C. Tang, "Range-Sample Depth Feature for Action Recognition", *CVPR*, 2014.
[24] J. Luo, W. Wang, and H. Qi, "Group Sparsity and Geometry Constrained Dictionary Learning for Action Recognition from Depth Maps", *ICCV*, 2013.
[25] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Dictionary Learning for Sparse Coding", *ICML*, 2009.
[26] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. "Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks", *CVPR*, 2016.
[27] O. Oreifej and Z. Liu, "HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences", *CVPR*, 2013.
[28] E. Ohn-Bar and M. Trivedi, "Joint Angles Similarities and HOG2 for Action Recognition", *CVPR Workshop on Human Activity Understanding from 3D Data*, 2013.
[29] F. Perronnin, J. Sanchez, and T. Mensink, "Improving the Fisher Kernel for Large Scale Image Classification", *ECCV*, 2010.
[30] R. Poppe, "A Survey on Vision based Human Action Recognition", *Image and Vision Computing*, 2010.
[31] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian, "Real Time Action Recognition Using Histograms of Depth Gradients and Random Decision Forests", *WACV*, 2014.
[32] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image Classification with the Fisher Vector: Theory and Practice", *International Journal on Computer Vision*, 2013.
[33] L. Seidenari, V. Varano, S. Berretti, A. Bimbo, and P. Pala, "Recognizing Actions from Depth Cameras as Weakly Aligned Multi-Part Bag-of-Poses", *CVPR Workshop on Human Activity Understanding from 3D Data*, 2013.
[34] 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", *CVPR*, 2011.
[35] 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", *ACCV*, 2012.
[36] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group", *CVPR*, 2014.
[37] A. Vieira, E. Nascimento, G. Oliveira, Z. Liu, and M. Campos, "STOP: Space-Time Occupancy Patterns for 3D Action Recognition from Depth Map Sequences", *CIARP*, 2012.
[38] H. Wang, A. Klaser, C. Schmid, and C. Liu, "Dense Trajectories and Motion Boundary Descriptors for Action Recognition", *International Journal on Computer Vision*, 2013.
[39] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of Local Spatio-Temporal Features for Action Recognition", *BMVC*, 2009.
[40] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D Action Recognition with Random Occupancy Patterns", *ECCV*, 2012.
[41] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining Actionlet Ensemble for Action Recognition with Depth Cameras", *CVPR*, 2012.
[42] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning Actionlet Ensemble for 3D Human Action Recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2014.
[43] X. Wang, L. Wang, and Y. Qiao, "A Comparative Study of Encoding, Pooling, and Normalization Methods for Action Recognition", *ACCV*, 2012.
[44] C. Wang, Y. Wang, and A. Yuille, "An Approach to Pose based Action Recognition", *CVPR*, 2013.
[45] J. Wang and Y. Wu, "Learning Maximum Margin Temporal Warping for Action Recognition", *ICCV*, 2013.

[46] L. Xia, C. Chen, and J. Aggarwal, "View Invariant Human Action Recognition Using Histograms of 3D Joints", *CVPR Workshop on Human Activity Understanding from 3D Data*, 2012.
[47] L. Xia and J. Aggarwal, "Spatio-Temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera", *CVPR*, 2013.
[48] X. Yang, Z. Liu, E. Zavesky, D. Gibbon, B. Shahraray, and Y. Tian. "AT&T Research at TRECVID 2013: Surveillance Event Detection", *NIST TRECVID Workshop*, 2013.
[49] X. Yang, C. Zhang, and Y. Tian, "Recognizing Actions Using Depth Motion Maps based Histograms of Oriented Gradients", *ACM Multimedia*, 2012.
[50] X. Yang and Y. Tian, "Effective 3D Action Recognition Using EigenJoints", *Journal of Visual Communication and Image Representation*, 2014.
[51] X. Yang and Y. Tian, "Super Normal Vector for Activity Recognition Using Depth Sequences", *CVPR*, 2014.
[52] G. Ye, Y. Li, H. Xu, D. Liu, and S. Chang, "EventNet: A Large Scale Structured Concept Library for Complex Event Detection in Video", *ACM Multimedia*, 2015.
[53] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection", *ICCV*, 2013.
[54] X. Zhou, K. Yu, T. Zhang, and T. Huang, "Image Classification Using Super Vector Coding of Local Image Descriptors", *ECCV*, 2010.



Xiaodong Yang received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 2009, and the Ph.D. degree from the Department of Electrical Engineering at City College, City University of New York, New York, NY, USA, in 2015. He joined NVIDIA Research, Santa Clara, CA, USA, as a research scientist, in 2015. His current research interests include computer vision, machine learning, deep learning, and multimedia analytics. He has been working on large-scale image and video classification, hand gesture and human action recognition, video surveillance event detection, multimedia search, and computer vision based assistive technology.

YingLi Tian received the B.S. and M.S. degrees from Tianjin University, China, in 1987 and 1990, and the Ph.D. degree from Chinese University of Hong Kong, Hong Kong, in 1996. After holding a faculty position at National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, she joined Carnegie Mellon University in 1998, where she was a postdoctoral fellow at the Robotics Institute. She then worked as a research staff member in IBM T. J. Watson Research Center from 2001 to 2008. She is one of the inventors of the IBM Smart Surveillance Solutions. She is currently a professor in the Department of Electrical Engineering at City College and Graduate Center, City University of New York. Her current research focuses on a wide range of computer vision problems from motion detection and analysis, assistive technology, to human identification, facial expression analysis, and video surveillance.