International Journal of Multimedia and Ubiquitous Engineering
Vol. 10, No. 1 (2015), pp. 391-402
http://dx.doi.org/10.14257/ijmue.2015.10.1.36

Action Recognition Using Polyhedron Neighborhood Features

Jiangfeng Yang and Zheng Ma

School of Communication and Information Engineering
University of Electronic Science and Technology of China, Xiyuan Ave, No.2006,
West Hi-Tech Zone, 61173
wallsonyang@163.com, 369322023@qq.com

Abstract

To utilize the geometry structure information and similarity information within the neighborhood surrounding a spatio-temporal interest point for human action recognition task, we employ the axes of a regular polyhedron as a reference locating system, and build a novel local feature named polyhedron neighborhood feature (PNF). Then, to reduce quantization error in the coding stage, locality-constrained linear coding method is used to encode the obtained PNFs. Next, multi-temporal-scale PNFs (MPNFs) are created for handling the problem of various action speeds. In classification, support vector machine (SVM) based on linear kernel is used as classifier taking time consumption into account. The experiments on the KTH and UCF sports datasets show that the recognition system based on PNFs achieves better performance than the competing local spatio-temporal feature-based human action recognition methods.

Keywords: *Action recognition, action representation, polyhedron neighborhood feature, locality-constrained linear coding*

1. Introduction

Recently, many literatures on action recognition have shown promising results using local Spatio-Temporal (ST) descriptors together with bag-of-features (BoF) models, where the local features are quantized to form a visual vocabulary, and a video clip is summarized by the histogram of its feature occurrences [1-3]. The representation has a number of advantages: being local, the features have robustness to viewpoint changes and occlusions; being relatively sparse, they can be stored and manipulated efficiently.

However, a key limitation of Spatio-Temporal Interest Point (STIP) representations is that they can be too local, failing to capture adequate spatial or temporal relationships. In the extreme, the orderless bag-of-words lacks cues about motion trajectories, before-after relationships, or the relative layout of objects and actions. In an attempt to overcome this problem, several alternatives have been proposed to capture mid-level structure using space-time bins of points, with partitions formed either globally at the level of the entire clip (*e.g.*, a histogram for the upper third of the frames is recorded separately from one for the lower third) [4, 5], or else in a feature-centered manner where a cuboid with multiple sub-bins is used to describe a point's neighborhood [6, 7]. Unfortunately, a global binning makes the representation sensitive to position or time shifts in the clip segmentation, and using predetermined fixed-size space-time grid bins (whether global or feature-centered) assumes that the proper volume scale is known and uniform across action classes.

ISSN: 1975-0080 IJMUE

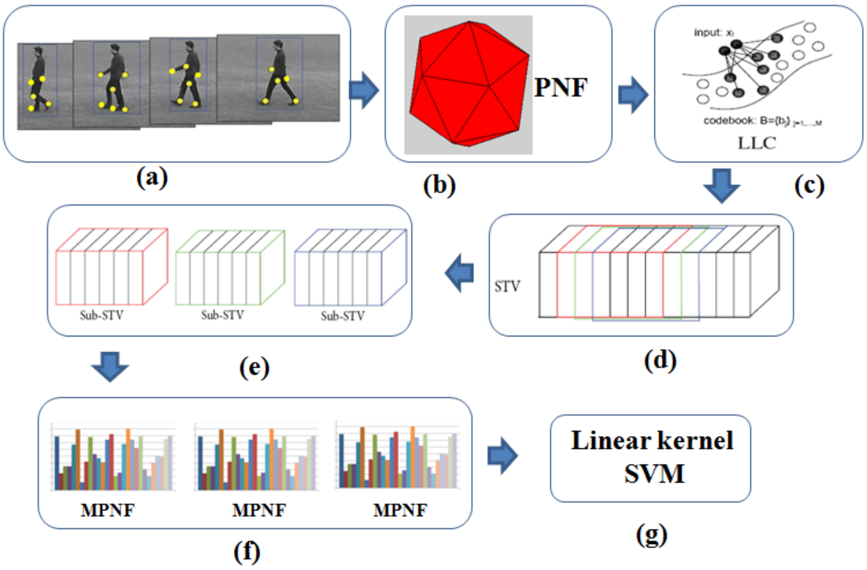


Figure 1. The Flowchart of the Action Recognition System based on PNFs. (a) Extracting STIPs from Action Video. (b) Employing the Axes of Regular Polyhedron as Reference Locating System, and Building PNFs. (c) Encoding PNFs with LLC Algorithm. (d) Regarding an Action Video as a STV. (e) Building Sub-STVs. (f) Computing MPNFs for every sub-STV. (g) Using SVM based on Linear Kernel

The relative position information between neighboring STIPs, including before/after, above/below, and left/right, can be distinguished by a reference coordinate system (e.g., X, Y, T axes in Cartesian coordinate system). However, traditional Cartesian system hardly provides more detailed information about the relative position, one needs a more complicated locating system to achieve this goal.

Inspirited by the work in [8], in which HOG3D feature descriptor was proposed by using a regular polyhedron as a local coordinate system. In the paper, a novel ST neighborhood feature named Polyhedron Neighborhood Feature (PNF) is proposed by using a regular polyhedron as a reference locating system (see Figure 2). We treat the axes running through the central positions of all faces and the gravity point of a regular polyhedron as a reference coordinate system.

There are two steps in building the PNFs. Firstly, the cosines between neighboring and central STIPs are projected onto the axes of polyhedron, and summed up all cosine values on each axis to form a component related to the axis. Secondly, like first step, the similarities between neighboring and central STIPs are projected on each axis, and summed up to form a component value related to the axis. PNFs are obtained by connecting the two types of information.

After that, action video clip is represented as a group of PNFs, and K-means clustering algorithm over the training PNFs is implemented to build codebook for encoding PNFs. Compared to the traditional coding schemes, such as Vector Quantization (VQ) and Sparse Coding (SC), recent proposed Locality-constrained Linear Coding (LLC) algorithm in [12] has attracted much attention due to its outstanding properties (see Section 3.2). In the paper, we employed LLC to do the task of encoding PNFs.

Undergoing LLC, an action video is represented as a group of reconstruction coefficient vectors, each of which is associated with a PNF feature. In action recognition in video, a challenging problem is that same actions can be carried out at different speeds and styles. To

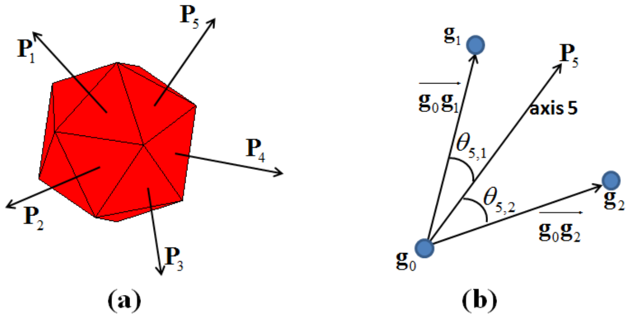


Figure 2. A Regular Polyhedron and PNF Computation. (a) An Icosahedron (20-sided) Polyhedron and its Axes (here only presents 5 axes). (b) The Cosines between two Vectors (formed by two neighboring STIPs g_1, g_2 and central STIP g_0, respectively) and Axis 5 of the Regular Polyhedron

deal with the problem, the temporal information between PNFs should be considered. Therefore, Multi-temporal-scale PNFs (MPNFs) are constructed by average-pooling over reconstruction coefficients within sub-STVs stacked by different frame numbers.

In classification, SVM based on linear kernel is employed as action classifier. The experimental results on KTH and UCF sports datasets show that our method achieves better performance than some classical methods published recently [15-19]. Figure1 shows the flowchart of the recognition system based on PNFs.

There are two contributions in this paper.

- To precisely describe the geometry structure information and similarity information in neighborhoods, axes of regular polyhedron is used as a reference locating system.
- To combine the geometry structure information and the similarities between STIPs, a novel feature named PNF is proposed.

The rest of this paper is organized as follows: PNF is proposed in Section 2, and encoded by LLC is provided in Section 3. Then, MPNF is constructed in Section 4. Experimental results and analysis are shown in Section 5. Finally, conclusions are drawn in Section 6.

2. Building Polyhedron Neighborhood Features

2.1. Detecting Spatio-temporal Interest Points (STIPs)

The inputs to our recognition system are the STIP positions and their associated local descriptors. We utilize Dollar detector [14] to extract STIPs from video sequences. The detector generally produces a high number of STIPs that is important for learning ST neighborhood feature. Using Dollar detector, action video sequence V is described by a set of position-descriptor tuples:

$$V = \{(x_i, y_i, t_i), \mathbf{d}_i : i = 1, \cdots, n_v\}, \tag{1}$$

where (x_i, y_i, t_i) records the position; $\mathbf{d}_i$ denotes a local feature vector of 3D support region around STIP i; n_v denotes the video's total number of STIPs.

2.2. Employing the Axes of Regular Polyhedron as Position Locating System

In action recognition, the ST position relationship between STIPs is powerful discriminative information. Many literatures [10, 11] studied in the representation of local neighborhood shape (*e.g.*, connected graph, only considering distance relation such as Gaussian distance, and no considering relative position relationship, such as before/behind, above/below, right/left).

To describe such relationship, in [10], the difference of coordinate of STIPs is used to judge their relative position (*e.g.*, $(x_1, y_1, t_1), (x_2, y_2, t_2)$ denote the positions of two STIPs, and the before/after, above/below position relationship can be decided by the signs of $(x_1 - x_2)$ and $(y_1 - y_2)$, respectively). These methods can distinguish only simple relative position between STIPs, and be difficult in offering more accuracy information about the relative position, such as upper-left, upper-right. To handle this limitation, ones need a sophisticated position locating system. To achieve this goal, we employed the axes of regular polyhedron as a reference coordinate system for locating the relative position accurately.

Mathematically, there are only five types of regular polyhedrons: the tetrahedron (4-sided), cube (6-sided), octahedron (8-sided), dodecahedron (12-sided), and icosahedrons (20-sided). As the octagon (8-sided polygon) is commonly used to quantize 2D gradients, in the paper, we consider the icosahedron (20-sided) for describing the relative position. (see Figure 2).

2.3. Building Polyhedron Neighborhood Feature

In this section, we show the procedure of building polyhedron neighborhood features (PNFs) based on the reference locating system.

Before building PNFs, the nearest neighbors of each STIP are selected. Each PNF is formed by the neighborhood around a central STIP $\mathbf{g}_1$. For a given STIP, we collect its N closest STIPs, where nearness is measured by Euclidean distance on its 3d position coordinates. Since STIPs are selected into the neighborhood according to their distance from a central STIP; using ranking on distance rather than fixed distances means that the nearest neighbors are robust to a scale change or internal shifts and stretching.

Let $N(\mathbf{g}_1) = \{\mathbf{g}_1, \cdots, \mathbf{g}_N\}$ denote a set of position of the N nearest neighbors for central point $\mathbf{g}_1$, $\mathbf{g}_i = (x_i, y_i, t_i)^T, i = 1; \cdots, N$. We employ the axes running through the central positions of all faces and the gravity point of a polyhedron as a reference locating system, and let the center of the polyhedron's gravity lie at the origin of a three dimensional Euclidean coordinate system. To depict the geometry structure of neighborhood (in the paper, the cosines between the polyhedron axes and the vectors, which are formed by neighboring STIPs and central STIP, are regarded as neighborhood geometry structure), we first project vector $\overrightarrow{\mathbf{g}_1\mathbf{g}_i} = (x_i - x_1, y_i - y_1, t_i - t_1), (i \succ 1)$ on the axes of polyhedron.

For the icosahedron (20-sided) polyhedron, the vectors corresponding to its 20 axes are stored as matrix $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_{20}]$, where $\mathbf{p}_j = (x_j, y_j, t_j)^T$ denotes vector of the j-th axis, and all vectors are:

$$(\pm 1, \pm 1, \pm 1), (0, \pm 1/\varphi, \pm \varphi), (\pm 1/\varphi, \pm \varphi, 0), (\pm \varphi, 0, \pm 1/\varphi), \tag{2}$$

where $\varphi = (1 + \sqrt{5})/2$ is called the golden ratio.

Each PNF feature $\mathbf{f} = [\mathbf{gm}, \mathbf{sl}]^T$ consists of two parts: geometry structure feature $\mathbf{gm}$ and similarity feature $\mathbf{sl}$. The procedure of computing PNF is as follows:

1) to compute geometry feature $\mathbf{gm}$, let $\theta_{i,j}$ denote the cosine between vectors $\overrightarrow{\mathbf{g}_1\mathbf{g}_i}$ and $\mathbf{p}_j$, and let $\boldsymbol{\theta}_i = [\theta_{i,1}, \cdots, \theta_{i,20}]^T$

$$\theta_{i,j} = \frac{\left\langle (\overrightarrow{\mathbf{g}_1\mathbf{g}_i}), \mathbf{p}_j \right\rangle}{\left\| (\overrightarrow{\mathbf{g}_1\mathbf{g}_i}) \right\|_2 \left\| \mathbf{p}_j \right\|_2}, \tag{3}$$

where $\langle ., . \rangle$ denotes inner product operation.

2) all cosine components on polyhedron axis j is summed up $gm_j = \sum_{i=1}^{N} \theta_{i,j}$, and normalized vector $\mathbf{gm}$ is obtained by

$$\mathbf{gm} = (1/c_{gm})[gm_1, \cdots, gm_{20}]^T, \quad c_{gm} = \sum_{j=1}^{20} \left|gm_j\right| + \varepsilon, \tag{4}$$

where c_{gm} is a normalization term, $\varepsilon \succ 0$ is small positive value so that $c_{gm} \neq 0$.

3) next, we compute similarity feature $\mathbf{sl}$. let $sim(\mathbf{d}_i, \mathbf{d}_1)$ denote the similarity between neighboring feature descriptors $\mathbf{d}_i$ and $\mathbf{d}_1$, and $\beta_{i,j}$ denote the value of $sim(\mathbf{d}_i, \mathbf{d}_1)$ projected on $\mathbf{p}_j$

$$\beta_{i,j} = sim(\mathbf{d}_i, \mathbf{d}_1).\theta_{i,j}, \tag{5}$$

4) all similarity components on polyhedron axis j is summed up $sl_j = \sum_{i=1}^{N} \beta_{i,j}$, and normalized vector $\mathbf{sl}$ is obtained by

$$\mathbf{sl} = (1/c_{sl})[sl_1, \cdots, sl_{20}]^T, \quad c_{sl} = \sum_{j=1}^{20} \left|sl_j\right| + \varepsilon, \tag{6}$$

where c_{sl} is a normalization value, $\varepsilon \succ 0$ is small positive value so that $c_{sl} \neq 0$.

Finally, action video sequence V is represented as a group of PNFs.

3. Encoding PNFs by LLC

Undergoing above processes, action video sequence V is represented as a group of PNFs. Our action recognition system is based on BoF model. BoF based method is often comprised of the following common steps: feature extraction, codebook (or dictionary) designing, feature encoding, and pooling. Of all the four steps, feature coding is the core component, which links feature extraction and feature pooling, and greatly influences classification performance in terms of both accuracy and speed.

Let $\mathbf{F} = \{\mathbf{f}_i \in R^D, i \in 1, \cdots, N\}$ be N local PNF feature descriptors with D -dimension, where PNF $\mathbf{f}_i = [\mathbf{gm}_i, \mathbf{sl}_i]^T$. Given a codebook with M bases $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_M] \in R^{D \times M}$, $\mathbf{f}_i$ is converted into M -dimensional code denoted as $\mathbf{c}_i \in R^M$ by feature coding schemes, such as VQ, SC, and LLC.

3.1. Vector Quantization and Sparse Coding

Let $\mathbf{F} = \{\mathbf{f}_i \in R^D, i \in 1, \cdots, N\}$ be N local PNF feature descriptors with D -dimension, where $\mathbf{f}_i = [\mathbf{gm}_i, \mathbf{sl}_i]^T$. Given a codebook with M bases $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_M] \in R^{D \times M}$, PNF feature $\mathbf{f}_i$ is converted into M -dimensional code denoted as $\mathbf{c}_i \in R^M$ by feature coding schemes, such as VQ and SC. For simplicity, Codebook $\mathbf{B}$ is generated by k-means clustering over training PNFs with Euclidean distance as metric.

In VQ, its coding strategy assigns just a single base to the feature, each local descriptor is assigned to the nearest visual word:

$$c_{i,j} = \begin{cases} 1, & \text{if } j = \arg\min_j \left\|\mathbf{f}_i - \mathbf{b}_j\right\|_2^2, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

This coding is simple but, as reported in [12], suffers from the reconstruction error due to the reason that it only assigns a single code word to the descriptor.

Another way to reduce the quantization loss of VQ is SC [12] that encodes a descriptor by using the coefficients of a linear combination of the codewords in $\mathbf{B}$, with a sparsity regularity term ℓ_1 -norm:

$$\mathbf{c}_i = \arg\min_{\mathbf{c}} (\|\mathbf{f}_i - \mathbf{B}\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1), \quad \lambda \in R, \tag{8}$$

where the first term represents the reconstruction error of $\mathbf{f}_i$ with respect to codebook $\mathbf{B}$. The second term denotes a sparse constraint regularization on code $\mathbf{c}$, and λ is a regularization factor to balance these terms. Although compared to VQ, SC significantly reduces the quantization loss, its computation complex is high, and not guarantee that same input features produce same encoding result.

3.2. Locality-constrained Linear Coding (LLC)

Compared to VQ and SC, locality-constrained linear coding (LLC) algorithm recently proposed in [13] has attracted much attention due to its outstanding properties:

- Better reconstruction. In VQ, each descriptor is represented by a single basis in the codebook. Due to the large quantization errors the VQ code for similar descriptors might be very different. Besides, the VQ process ignores the relationships between different bases. Hence non-linear kernel projection is required to make up such information loss. On the other side, as shown in (Figure 3.c) in LLC, each descriptor is more accurately represented by multiple bases, and LLC code captures the correlations between similar descriptors by sharing bases.
- Local smooth sparsity. Similar to LLC, SC also achieves less reconstruction error by using multiple bases. Nevertheless, the regularization term of ℓ_1 norm in SC is not smooth. As (shown in Figure 3.b), due to the over-completeness of the codebook, the SC process might select quite different bases for similar patches to favor sparsity, thus losing correlations between codes. On the other side, the explicit locality adaptor in LLC ensures that similar patches will have similar codes.
- Analytical solution. Solving SC usually requires computationally demanding optimization procedures. Unlike SC, the solution of LLC can be derived analytically such that LLC can be performed very fast in practice.

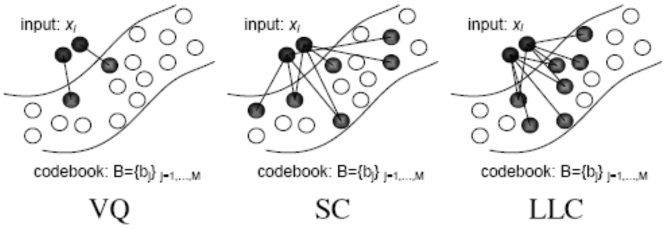

input: x_i
codebook: B={b$_j$} $_{j=1,...,M}$
VQ
input: x_i
codebook: B={b$_j$} $_{j=1,...,M}$
SC
input: x_i
codebook: B={b$_j$} $_{j=1,...,M}$
LLC

Figure3. Comparison between VQ, SC and LLC. The Selected Bases for Representation are Highlighted in Black

LLC can be formulated by

$$\mathbf{c}_i = \arg\min_{\mathbf{c}} (\|\mathbf{f}_i - \mathbf{B}\mathbf{c}\|_2^2 + \lambda \|\mathbf{d} \square \mathbf{c}\|_2^2), \quad \text{s.t.} \, \mathbf{1}^T \mathbf{c} = 1, \tag{9}$$

$$\mathbf{d} = \exp(\frac{\text{dist}(\mathbf{f}_i, \mathbf{B})}{\sigma}), \quad \text{dist}(\mathbf{f}_i, \mathbf{B}) = [\text{dist}(\mathbf{f}_i, \mathbf{b}_1), \cdots, \text{dist}(\mathbf{f}_i, \mathbf{b}_M)]^T, \tag{10}$$

where the first term is reconstruction error; the second term is the locality constraint regularization on code $\mathbf{c}$, and λ is a regularization factor; in the second term, denotes the element-wise multiplication, and $\mathbf{d} \in R^M$ is the locality adaptor that gives different weight for each base vector proportional to its similarity to the input feature $\mathbf{f}_i$; and $\mathrm{dist}(\mathbf{f}_i, \mathbf{b}_j)$ is the Euclidean distance between $\mathbf{f}_i$ and the j-th base $\mathbf{b}_j$. σ is used for adjusting the weight decay speed for the locality adaptor. $\mathbf{1}^T\mathbf{c} = 1$ is the shift invariant constraint according to [13].

LLC coding scheme bases on the hypothesis that descriptors approximately reside on a lower dimensional manifold in an ambient descriptor space; thus, it reduces the quantization error while preserving the consistent encoding ability.

In the paper, to reduce quantization error and keep the consistent coding, both LLC algorithm and a codebook with M bases $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_M] \in R^{D\times M}$ are employed to transform PNFs $\mathbf{F} = \{\mathbf{f}_i \in R^D, i \in 1, \cdots, N\}$ into their corresponding reconstruction coefficient vectors $\mathbf{C} = \{\mathbf{c}_i \in R^M, i \in 1, \cdots, N\}$.

4. Building Multi-temporal-scale PNFs (MPNFs)

Due to the different styles of human action, it is difficult to model the ST relationship of local features in a single space-time scale. The actions with different styles appear in different motion range (different spatial scale) and speed (different temporal scale). Compared with spatial position relationship between STIPs on XY plane, their temporal position relationship along T axis carries more discriminative information. In additional, PNFs contains the spatial position information among STIPs within neighborhood. Therefore, we proposed a multi-temporal-scale PNF by ignoring their spatial position information.

In implementation, we regard an action video as a ST volume (STV) stacked by all video frames. Because it is difficult to estimate action cycles in videos, the time coordinate system is difficult to establish. Fortunately, the feature ST relationships can be locally modeled by sub-STV. As a result, a STV are portioned into several temporal segments called sub-STVs, each of which centers on a STIP.

In our system, building MPNFs is composed of the following steps:

1) Given an action video STV V including n_v STIP according to (1), and V can be partitioned into n_v sub-STVs $\{sv_1, sv_2, \cdots, sv_{n_v}\}$ with spatial scale $\alpha(r)$. $\alpha(r)$ is the length of a sub-STV.
2) Assuming that sv_i contains n_i PNFs $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_{n_i}\}$. Then, we use LLC to encode each PNF and obtain corresponding codes $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{n_i}\}$.
3) After coding each PNF, we use average-pooling to get the descriptor $\mathbf{s}_{n_i}^{\alpha(r)}$ of sub-STV sv_{n_i}. $\mathbf{s}_{n_i}^{\alpha(r)}$ is the MPNF related to sv_{n_i}, and can be can be normalized by its ℓ_1-norm,

$$\mathbf{s}_{n_i}^{\alpha(r)} = (1/n_i)(\mathbf{c}_1 + \mathbf{c}_2 + \cdots + \mathbf{c}_{n_i}) \tag{11}$$

$$\overline{\mathbf{s}}_{n_i}^{\alpha(r)} = \mathbf{s}_{n_i}^{\alpha(r)} \Big/ \left\|\mathbf{s}_{n_i}^{\alpha(r)}\right\|_1 \tag{12}$$

4) Repeat steps 2, 3 to each sub-STV. STV V are represented as a group of MPNFs $\{\overline{\mathbf{s}}_1^{\alpha(r)}, \overline{\mathbf{s}}_2^{\alpha(r)}, \cdots, \overline{\mathbf{s}}_{n_v}^{\alpha(r)}\}$.

Next, to describe action video sequence V, statistical coefficient histogram $\mathbf{S}^{\alpha(r)}$ is computed as follows

$$\mathbf{S}^{\alpha(r)} = (1/n_v)\sum_{i=1}^{n_v} \overline{\mathbf{s}}_i^{\alpha(r)} \tag{13}$$

To handle the problem of different speeds of same action, multiple temporal scales are employed in our system, and a final histogram $\mathbf{S}$ for action recognition is obtained by

$$\mathbf{s} = [\mathbf{s}^{\alpha(1)}, \cdots, \mathbf{s}^{\alpha(R)}] \tag{14}$$

where $[\mathbf{s}^{\alpha(1)}, \cdots, \mathbf{s}^{\alpha(R)}]$ means concatenating coefficient histograms $\{\mathbf{s}^{\alpha(1)}, \cdots, \mathbf{s}^{\alpha(R)}\}$.

5. Experiment and Analysis

5.1. Experiment Setup

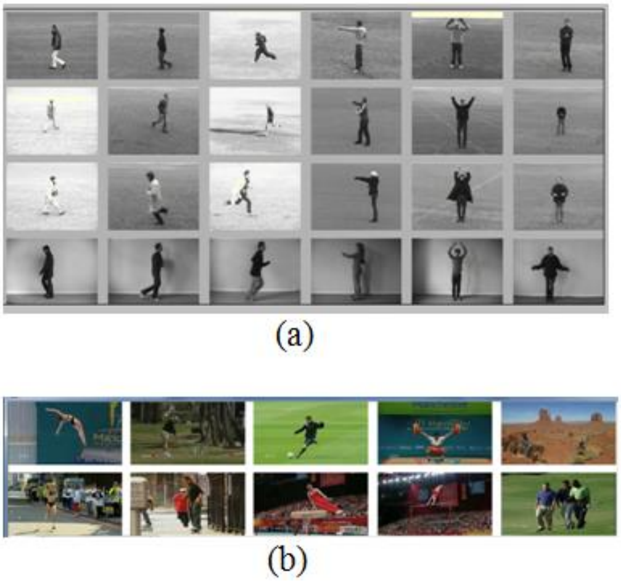
To evaluate the performance of our proposed algorithm, the KTH and UCF Sports datasets are used as benchmark.

In all experiments, HOG [14] (Histogram of Oriented Gradient) and HOF [14] (Histogram of Optical Flow) is utilized to depict local appearance and motion information, respectively. And HOG+HOF feature of i -th STIP is denoted as $\mathbf{d}_i$ in (1). The multi-scale version Dollar detector is used to extract STIPs, and its spatial scale [1.2, 1.3, 1.4, 1.5] and temporal scale [0.4, 0.45, 0.5, 0.55]. To capture multi-scale temporal relationship of local PNFs, the lengths of sub-STV are set as 10, 15, 20, and 25 frames (spatial scale $\alpha = [10, 15, 20, 25]$). In LLC coding stage, the dictionary size is set to 300. Since there are 4 spatial scales (the length of sub-STV), the dimensionality of final histogram $\mathbf{s}$ is $(300 \times 4) = 1200$.

For the KTH dataset, local features (HOG+HOF) from all videos of one subject are used to construct codebooks for LLC coding by k-means clustering algorithm. For each LOO run, we learn a model from the videos of 24 subjects, test the videos of the remaining subject. The recognition rate is the average value of the 25 runs.

For the UCF sports, local features (HOG+HOF) from 20 videos (2 videos selected from each action, 10 class actions) are used to build codebooks for LLC coding by k-means. In each LOO, one video of each class is randomly selected as test data, the other videos are treated as training data. 100 LOO runs are carried out. The recognition rate is the average value of the 100 runs.

In action classification stage, support vector machine (SVM) based on linear kernel is employed as action classifier.



(a)

(b)

Figure 4. Examples from the Two Public Datasets: (a) the KTH Dataset (b) the UCF Sports Dataset

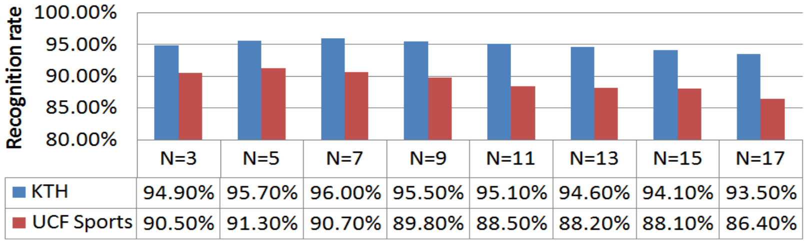
5.2. Datasets

The KTH dataset contains six types of human action examples (*i.e.*, boxing, hand clapping, hand waving, jogging, running, and walking) are performed by 25 different subjects. Each action is performed in four scenarios: indoors, outdoors, outdoors with scale variation, and outdoors with different clothes. It contains 600 low-resolution video sequences (160×120 pixels). Examples of this dataset are shown in Figure 4(a).

The UCF sports dataset includes a set of 150 videos, which are collected from various broadcast sports channels such as BBC and ESPN. It contains 10 different actions: diving, golf swing, horse riding, kicking, lifting, running, skating, swing bar, swing floor, and walking. This dataset is challenging for a wide range of scenarios and viewpoints. Examples of this datasets are presented in Figure 4(b).

5.3. Experimental Result and Analysis

To evaluate the influence of the neighborhood size of PNF upon action recognition accuracy, various neighborhood sizes is used, and related results are shown in Figure 5. It is observed that for the KTH dataset, when the number of the selected neighboring STIPs during building PNF feature is 7 (N=7), the highest recognition rate 96.0% is achieved; for the UCF Sports dataset, the highest recognition rate 91.3% is obtain, when its neighboring size is set as 5 (N=5). Moreover, we found that for the KTH, the recognition performance is distracted slightly, when N=3, 5, 7, 9; and for the UCF Sports, the system performance drops slightly, when N=3, 5, 7. On the whole, recognition performance on the UCF Sports is more sensitive to the neighborhood size than that on the KTH dataset.



	N=3	N=5	N=7	N=9	N=11	N=13	N=15	N=17
🟦 KTH	94.90%	95.70%	96.00%	95.50%	95.10%	94.60%	94.10%	93.50%
🟥 UCF Sports	90.50%	91.30%	90.70%	89.80%	88.50%	88.20%	88.10%	86.40%

Figure 5. Neighborhood Size (the number of the selected nearest STIPs) Influences on the Recognition Rates on the Two Datasets

Table 1 shows the performance comparison between our system and some classical system published recently. The competing methods include local representation-based approaches [15-18], global representation-based approach [19]. In detail, SC was used for feature coding together with BoF in [15], local feature distribution information was used in [16], ST context feature was employed in [17], sparse representation-based classification methods was applied in [18], and the global representation method was adopted in [19]. It demonstrates that our method achieves better performance than the competing methods. The confusion matrices for KTH and UCF sports datasets of our method are shown in Figures 2 and 3, respectively.

Table 1. Performance Comparison with other Systems

Methods	Year	KTH(%)	UCF Sports(%)
Zhu *et al.* [15]	2010	94.9	84.3
Wu *et al.* [16]	2011	94.5	91.3
Guha *et al.* [17]	2012	—	91.1
Bregonzio *et al.* [18]	2012	94.3	—
Saghafi *et al.* [19]	2012	92.6	—
Our method		96.0	91.3

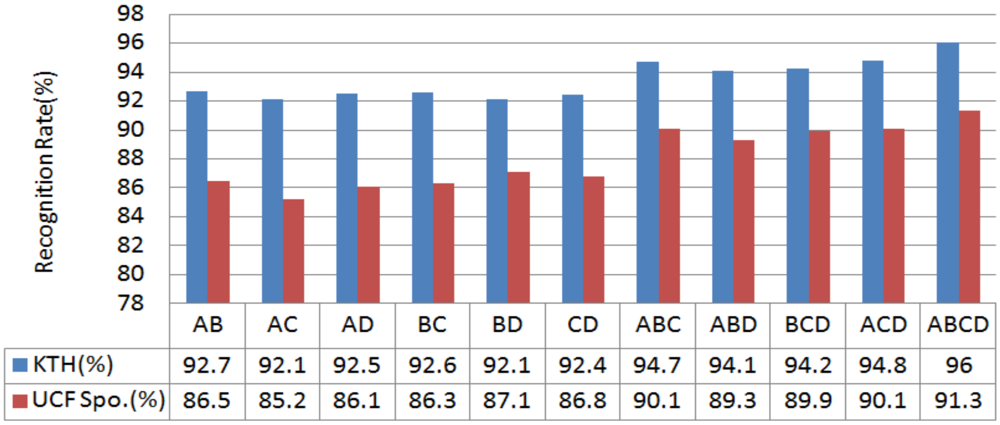
Table 2. Confusion Table on the KTH Dataset with our Methods.s1(boxing), s2(hand-waving),s3(hand-clapping),s4(walking),s5(jogging),s6(running)

	s1(%)	s2(%)	s3(%)	s4(%)	s5(%)	s6(%)
s1(%)	98.1	0.90	1.00			
s2(%)	0.80	98.4	0.80			
s3(%)	0.50	0.50	99.0			
s4(%)				93.2	3.20	3.60
s5(%)				3.80	92.8	3.40
s6(%)				4.00	2.40	93.6

Table 3. Confusion Table on the UCF Sports Dataset with our Methods.S1 (diving), S2 (golfing), S3 (kicking), S4 (lifting), S5 (horse-riding), S6 (running), S7 (skating), S8 (swing-bench), S9 (swing-high-bar), S10 (walking)

	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10
s1	0.96		0.02		0.02					
s2		0.91		0.06	0.03					
s3		0.05	0.90			0.05				
s4	0.04		0.04	0.92						
s5				0.03	0.92	0.05				
s6				0.03	0.06	0.91				
s7							0.91			0.09
s8								0.92	0.08	
s9								0.10	0.90	
s10								0.12		0.88

Figure 6 presents the recognition accuracies on the KTH and UCF Sports under varied combinations on temporal scale factors. It is clear that when four temporal scales are considered in classification, their highest performance is achieved. Furthermore, it can be seen that with more spatial scales involved in classification, the discriminative power of recognition system gets strong, the recognition rate increases.



	AB	AC	AD	BC	BD	CD	ABC	ABD	BCD	ACD	ABCD
🟦 KTH(%)	92.7	92.1	92.5	92.6	92.1	92.4	94.7	94.1	94.2	94.8	96
🟥 UCF Spo.(%)	86.5	85.2	86.1	86.3	87.1	86.8	90.1	89.3	89.9	90.1	91.3

A(5 frames); B(10 frames); C(15 frames); D(20 frames)

Figure 6. To Evaluate the Influence of the Temporal Scale Factor of MPNF on the Recognition Accuracy, Different Combinations of Four sub-STV Lengths (5, 10, 15 and 20 frames) are Generated

6. Conclusion

In the paper, in order to precisely describe both of geometry structure and similarity information within a neighborhood around a STIP for human action recognition task, we employed the axes of a regular polyhedron as reference locating system, and built PNF features that combines the above both information. The experimental results show that (1) the proposed reference locating system can effectively locate the relative position between STIPs. (2) The PNF, which contains geometry structure information and similarity information within neighborhood, is a salient local features for human action recognition tasks.

Acknowledgement

This research is supported by National Nature Science Foundation of China (Grant no. 61271288).

References

[1] J. Liu, J. Luo and M. Shah, “Recognizing realistic actions from videos “in the wild””, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), **(2009)**.

[2] A. Gilbert, J. Illingworth and R. Bowden, “Fast realistic multi-action recognition using mined dense spatio-temporal features”, In ICCV, **(2009)**.

[3] J. C. Niebles, H.Wang and L. Fei-Fei, “Unsupervised learning of human action categories using spatial-temporal words”, In IJCV, **(2008)**.

[4] J. Choi, W. Jeon and S.-C. Lee, “Spatio-temporal pyramid matching for sports videos”, In ACM Multimedia, **(2008)**.

[5] A. Fathi and G. Mori, “Action recognition by learning mid-level motion features”, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), **(2008)**.

[6] A. Gilbert, J. Illingworth and R. Bowden, “Fast realistic multi-action recognition using mined dense spatio-temporal features”, In ICCV, **(2009)**.

[7] Klaser A, Marszalek M, “A spatio-temporal descriptor based on 3d-gradients”, Proceedings of the British Machine Vision Conference, **(2008)**.

[8] I. Laptev, M. Marszalek, C. Schmid and B. Rozenfeld, “Learning realistic human actions from movies”, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), **(2008)**.

[9] M. Marszalek, I. Laptev and C. Schmid, “Actions in context,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), **(2009)**, pp.2929-2936.

[10] A. Kovashka, and K. Grauman, “Learning a hierarchy of discriminative space-time neighborhood features for human action recognition,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), **(2010)**.

[11] J. Yang, K. Yu, Y. Gong and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), **(2009)**, pp.1794-1801.

[12] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang and Y. Gong, “Locality-constrained linear coding for image classification”, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), **(2010)**, pp. 3360-3367.

[13] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie, “Behavior recognition via sparse spatio-temporal features”, Proceeding of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS ’05), October **(2005)**, pp. 65-72.

[14] Y. Zhu, X. Zhao and Y. Fu, “Sparse coding on local spatial-temporal volumes for human action recognition”, Proceedings of the Computer Vision (ACCV), **(2010)**, pp. 660-671, Springer, Berlin, Germany.

[15] X. Wu, D. Xu, L. Duan and J. Luo, “Action recognition using context and appearance distribution features”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), **(2011)**, pp.489-496.

[16] T. Guha and R. K. Ward, “Learning sparse representations for human action recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, **(2012)**, pp.1576-1588, vol.34, no.8.

[17] M. Bregonzio, T. Xiang and S. Gong, “Fusing appearance and distribution information of interest points for action recognition”, Pattern Recognition, **(2012)**, pp.1220-1234, vol.45, no.3.

[18] B. Saghafi and D. Rajan, “Human action recognition using pose-based discriminant embedding”, Signal Processing, **(2012)**, pp.96-111, vol.27, no.1.

Authors

Jiangfeng Yang, he is a Ph.D. student in University of electronic science and technology of China, China. He received his Master degree of Engineering from Kunming University of science and technology, China in computer software and theory in 2009. His research interests include machine vision and action recognition in video.

Zheng Ma, he has been working as a professor in School of communication and information engineering, University of electronic science and technology of China, China. His research interests include signal processing, machine vision and Internet security.