Modeling Geometric-Temporal Context with Directional Pyramid Co-occurrence for Action Recognition

Chunfeng Yuan[1], Xi Li[1], Weiming Hu[1], Haibin Ling[2], Steven Maybank[3]

[1] National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2] Department of Computer and Information Sciences, Temple University, Philadelphia, USA
[3] School of Computer Science and Information Systems, Birkbeck College, London, UK

In this paper, we present a new geometric-temporal representation for visual action recognition based on local spatio-temporal features. First, we propose a modified covariance descriptor under the log-Euclidean Riemannian metric to represent the spatio-temporal cuboids detected in the video sequences. Compared with previously proposed covariance descriptor, our descriptor can be measured and clustered in the Euclidian space. Second, to capture the geometric-temporal contextual information, we construct a *Directional Pyramid Co-occurrence Matrix* (DPCM) to describe the spatio-temporal distribution of the vector-quantized local feature descriptors extracted from a video. DPCM characterizes the co-occurrence statistics of local features as well as the spatio-temporal positional relationships among the concurrent features. These statistics provide strong descriptive power for action recognition. To use DPCM for action recognition, we propose a *Directional Pyramid Co-occurrence Matching Kernel* (DPCMK) to measure the similarity of videos. The proposed method achieves the state-of-the-art performance and largely improves the recognition performance over the bag of visual words (BOVW) models on six public datasets. For example, on the KTH dataset, it achieves 98.78% accuracy while the BOVW approach only achieves 88.06%. On both Weizmann and UCF CIL datasets, the highest possible accuracy of 100% is achieved.

***Index Terms*—Covariance cuboid descriptor, log-Euclidean Riemannian metric, spatio-temporal directional pyramid co-occurrence matrix, kernel machine, action recognition**

I. Introduction

Recognition of human actions in video sequences is an important yet challenging task in computer vision [1-3]. It has a wide range of applications, such as smart surveillance, video indexing and browsing, human-computer interaction, etc. A recent trend in action recognition has been the emergence of techniques based on volumetric video analysis, where a video is represented by a set of local features extracted from several spatio-temporal volumes of the video. Such local feature based descriptions are robust against noise, occlusion, and geometric variations. Many studies along this line [4-6, 16-17, 25] model spatio-temporal features using the bag of visual words (BOVW) framework, which is geometrically unconstrained and omits global spatial (or long-term temporal) information. Typically, these methods have two key components: i) extracting robust local spatio-temporal features, and ii) constructing effective representations of video sequences using these local features. Despite many previous studies, the spatio-temporal feature based representation in video analysis is still an open field of research.

In this paper, we propose effective algorithms for improving both components of the BOVW framework for action recognition. For local S-T features, we propose using the covariance descriptor to describe each detected 3D S-T cuboid. The covariance descriptor, as proposed by Tuzel et al [7], is used to represent 2D image regions for object recognition and target tracking. In this paper, we extend the covariance descriptor to 3D S-T volume and demonstrate its effectiveness for representing a spatio-temporal cuboid. Compared with other cuboid descriptors such as 3D-SIFT, HOG and HOF, our 3D covariance descriptor has several advantages: (1) it directly fuses different types of pixel-level features; (2) it is robust to noise in the pixel-level features and to changes in rotation and scale; and (3) it has low computation complexity.

To measure the similarity between covariance descriptors, we use the log-Euclidean Riemannian metric. Compared with the affine invariant Riemannian metric used in [47], the log-Euclidean Riemannian metric takes a much simpler form. With the log-Euclidean Riemannian metric, our covariance descriptors, after matrix logarithmic transformation, lie in the Euclidean space and can then be clustered by the *k*-means clustering method to generate the visual vocabulary.

One limitation of the traditional BOVW model is that it discards the rich spatial-temporal context information in video sequences. From Fig.1, the distribution of local features in the 3D space varies significantly for different action classes. This suggests that considering spatio-temporal geometrical information may improve the discriminative power of action representation.

In our action representation, we consider each local spatio-temporal feature and its neighborhood as a feature ensemble that is more discriminative than an individual feature. Moreover, we exploit the intrinsic spatio-temporal structural information in every feature ensemble by obtaining the directions from the feature to each of its neighboring features. The resulting spatio-temporal *Directional Pyramid Co-occurrence Matrix* (DPCM) represents a video sequence by counting the number of the directional feature pairs in all the feature ensembles in the video sequence. The directions of feature pairs are quantized in a coarse-to-fine fashion to yield a hierarchical pyramid of the directions. The proposed DPCM captures not only the appearance information but also the geometric-temporal information in the video.

To measure the similarity between the proposed DPCMs, we propose the *Directional Pyramid Co-occurrence Matching Kernel* (DPCMK), which is inspired by the pyramid matching kernel recently proposed by Grauman and Darrell [49]. The

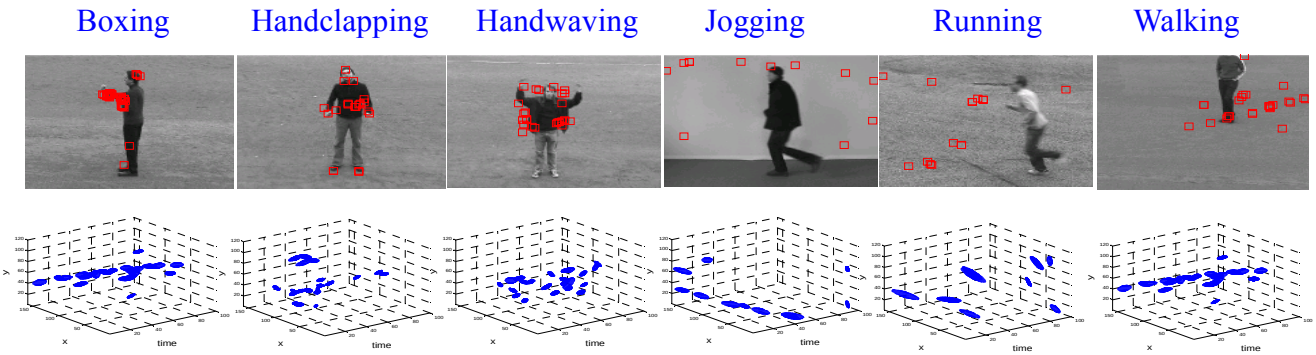


Fig. 1. Localization of interest points for six action videos in the KTH dataset. In the first row, one key frame for each video is shown and all interest points detected in that video are overlapped on the key frame, to show the spatial distribution of features. The second row shows the distribution of the features in spatio-temporal coordinates.

basic idea is to map unordered feature sets extracted from each video to DPCMs and to compute a weighted similarity summation of the corresponding sub-matrices of DPCMs. Our kernel inherits the merits of the original pyramid matching kernel: it is robust to clutter or outlier features; it is a Mercer kernel; and it has linear computational complexity.

In summary, the main contributions include the following:

- We adapt the covariance descriptor to the 3D spatio-temporal space in order to describe the S-T cuboids. By fusing various kinds of features extracted at the pixel level, the descriptor encodes rich information for describing local properties of human actions.
- We introduce the log-Euclidean Riemannian metric to calculate the similarity between covariance descriptors. By applying the matrix logarithmic transformation to covariance features, we obtain matrices in a Euclidean space. They can be clustered by the *k*-means algorithm to generate the vocabulary of BOVW model.
- We propose a *spatio-temporal Directional Pyramid Co-occurrence Matrix* (DPCM) to represent the video sequence by modeling directed local feature pairs. The DPCM uses feature ensembles instead of individual features, and the feature pairs in the ensembles are accumulated in the co-occurrence matrix. The DPCM goes one step beyond the co-occurrence matrix since it explicitly captures the inner spatio-temporal configuration of the ensemble by considering the relative positional relationships of pairs of features.
- We present a *Directional Pyramid Co-occurrence Matching Kernel* (DPCMK) to measure the similarity of two DPCMs. The DPCMK effectively combines each level in the pyramid structure and can tolerate the intra-class variations of actions. As a Mercer kernel, DPCMK can be readily combined with the SVM classifier.

We evaluate our method on six publicly available datasets including the KTH dataset [5], Weizmann dataset [23], UCF sports dataset [8], UCF CIL action dataset [54], the Feature Films dataset [8], and the facial expression dataset [6]. On all of the six datasets, the proposed method outperforms the traditional BOVW method by a large margin, and is superior to several state-of-the-art methods.

The remainder of the paper is organized as follows. Section II reviews related work on action recognition. Then, Section III details the covariance descriptor of the cuboids and the proposed DPCM. After that, Section IV describes action recognition based on DPCMK. Experimental results are reported in Section V. Finally, Section VI concludes the paper.

II. RELATED WORK

Action recognition [1-3] has attracted a large amount of research attention in the computer vision and image processing community. Some early work uses holistic features extracted from each frame of the video, such as the shape of silhouettes and 3-D joint angles, for describing the action units. Then, a video sequence is mapped into a feature sequence, to which recognition approaches based on the following methods have been applied: probabilistic graphical models such as hidden Markov models [9-12] and dynamic Bayesian networks [13-15].

Recently, the success of local interest points in object recognition has inspired studies using local spatio-temporal features for motion analysis and action recognition. Various statistical approaches, especially the histogram-based analysis, are employed on the local feature set to obtain action representations. Action recognition is then carried out using different classification and clustering approaches, such as the Support Vector Machine (SVM) [5], the Nearest Neighbor Classifier (NNC) [6, 16], and latent topic models [17].

In the next two subsections, we provide a literature review on human action recognition, focusing on local spatio-temporal descriptors and the representation of video sequences.

A. Local Interest Region Descriptors

Extracting local features includes two relatively independent steps: detecting cuboids and describing the cuboids [18]. At each detected interest point, a cuboid is extracted which contains the spatio-temporally windowed pixel values. After obtaining cuboids, one can calculate low-level features, such as gradient and optical flow, at each pixel in the cuboid. The task of the descriptor is to create a compact and discriminative middle-level feature from these low-level features. According to how the middle-level features are constructed, 3D spatio-temporal descriptors are classified into the following categories.

The simplest method is to concatenate all the pixel features in the cuboid [17, 24]. The resulting feature vector usually has a high dimension which is then reduced. An example is given by the PCA-SIFT descriptor proposed by Dollár *et al.* [6], and also called the *cuboid descriptor* in [22]. This type of descriptor is sensitive to small perturbations in the pixel values within the cuboid.

Another type of descriptor measures the distribution of low level statistics inside a cuboid, for example the histogram of gradient values (HOG) and the histogram of optical flows (HOF). There are two main versions of the HOG. One is to allocate the gradient orientations to several bins and count the number of gradients falling into each bin [25]. In the other version, the count of the number of gradients in each bin is weighted by the gradient magnitudes [26]. These descriptors are robust to small perturbations, but their discriminative power is reduced because all spatial information is ignored.

One way of capturing the spatial information is to divide each cuboid into several sub-cuboids and then create histograms for each sub-cuboid [19, 27]. These histograms are later concatenated to form a vector. Typical descriptors of this type include the spatial SIFT [21] for images, the 3D SIFT [20] for videos, and the HOG3D descriptor [28].

The feature vectors of all points in the cuboid comprise a set that can be represented by its statistical properties such as the mean vector, and the covariance matrix. These statistics capture discriminative information for spatio-temporal cuboids. Motivated by this fact, we introduce the covariance descriptor to represent the cuboid, which is fundamentally different from the three kinds of methods mentioned above. The covariance descriptor has been successfully applied to 2D images [7].

The idea of the above mentioned descriptors is to compute invariant feature from the spatio-temporal neighborhood regions of interest point. It is possible to design spatio-temporal descriptors directly from the interest points. For example, Schuldt *et al.* [5] combine a set of image derivatives computed up to a given order to obtain the descriptors in the form of the so-called spatio-temporal jets.

B. Geometrical Modeling for Action Representation

There are many algorithms for combining the geometrical information with BOVW. Fergus *et al.* [29] propose a model, translation and scale invariant pLSA (TSI-pLSA), which extends pLSA (as applied to visual words) to include spatial information of the interest points. They introduce into the classical pLSA model a second latent variable which represents the position of the centroid of the object within the image. Wong *et al.* [30] extend TSI-pLSA from 2D image analysis to 3D video analysis, and propose a pLSA with an implicit shape model (pLSA-ISM) to infer the location of motion in video sequences. Niebles *et al.* [31] define a constellation model for the geometrical arrangement of local features. In the above cases, many parameters and constraints are introduced leading to an increased computational complexity.

Other approaches for combining geometric information include the "spatial pyramid" in image [32] and the "spatio-temporal pyramid" in video [33]. In [32], an image is partitioned into increasingly fine sub-regions and histograms of local features are computed in each sub-region. The resulting "spatial pyramid" is an extension of the orderless bag-of-features image representation. Likewise, in [33], a video is uniformly divided into spatio-temporal grids and the histogram is computed in each grid. However, in action videos, the interest points are usually sparsely distributed in a small number of local regions. In [34] the positions of interest points are clustered in the spatio-temporal space. At each cluster center the histogram of local features is computed.

The co-occurrence matrix [35-36] and the proximity distributions [37] of visual words are proposed as global image descriptors for capturing the geometric information useful for object recognition. In [37], the histogram intersection kernel is used as the proximity distribution kernel to measure the similarities of images. In [35] the stationary distribution derived from the normalized co-occurrence matrix forms the so-called Markov stationary features (MSF), and then the χ^2 distance of the MSF is adopted for the kernel of an SVM classifier. In [36], the co-occurrence matrices are calculated for four selected directions yielding the so-called Directed Markov Stationary Features (DMSF).

In [35-37], it is demonstrated that both the co-occurrence matrix and the proximity matrix are able to improve recognition performance in images. However, the above co-occurrence matrices largely ignore the inner positional relationship between each concurrent feature pair.

There are some approaches that build a video representation directly from the positions of interest points. In [38], the trajectories of interest points are extracted based on the pair-wise SIFT matching over consecutive frames and then three types of features are obtained from the trajectory. In [39], Bregonzio *et al.* propose a different approach in which clouds of interest points are accumulated for different spatio-temporal windows and several features are extracted from the point clouds. These features include shapes, speeds, and the relationship between the clouds and the object areas. Computing these features involves some non-trivial steps such as reliable object detection and segmentation.

In [51, 52], Junejo *et al.* explore the temporal self-similarities within an action sequence and propose a Self-Similarity Matrix (SSM) as the action descriptor. SSMs are approximately invariant under view changes of an action and their diagonals indicate periodicity of the motion. In [53], Sun *et al.* proposed a Joint Self-Similarity Volume (Joint SSV) based on SSM. They construct a Self-Similarity Matrix (SSM) for each frame of a video and then construct Joint SSMs from SSMs of this video. Shen *et al.* [54, 55] decompose each pose into a set of point-triplets to achieve robustness to noise and self-occlusions. They define a matching score based on the pose transition, rather than on the poses themselves. Pose transition captures the temporal information in human motion, while most fundamental matrix based methods only enforce pose-to-pose correspondences. Yan *et al.* [56] build a 4D action feature model by using multi-view action sequences. By elegantly encoding shape and motion of actors observed from

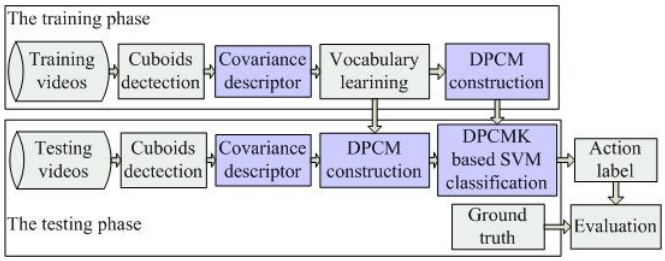


Fig. 2. Flowchart of the proposed action recognition framework.

multiple views, the feature is demonstrated its salient properties to assist in action recognition.

III. VIDEO REPRESENTATION BASED ON DIRECTIONAL PYRAMID CO-OCCURRENCE MODELING

Using local spatio-temporal features, we construct an effective video representation for action recognition. Fig. 2 shows the flowchart of the proposed action recognition framework. Compared with previous studies, our framework is novel mainly in two aspects: (1) the construction of a modified covariance descriptor under the log-Euclidean Riemannian metric to represent the detected 3D cuboids; and (2) the modeling of the video sequence by constructing a directional pyramid co-occurrence matrix for the local covariance features. The resulting representation of a video sequence includes not only local features but also the geometrical-temporal distribution of these local features over 3D space-time.

A. Covariance descriptor of the 3D cuboid

We detect the spatio-temporal interest points for each video using the method proposed by Dollár *et al.* [6]. Subsequently, a cuboid is extracted around each detected interest point. Let f_i be a d-dimensional feature vector associated with the i^{th} pixel in the cuboid. The components of f_i could include features, such as intensity, gradient, and optical flow. The feature vectors of all pixels in the cuboid form a feature vector set $F = \{f_1, f_2, \ldots, f_N\}$, where N is the number of pixels. The covariance matrix associated with the cuboid is defined as:

$$C = \frac{1}{N-1}\sum_{i=1}^{N}(f_i - u)(f_i - u)^T \tag{1}$$

where u is the mean of vectors in F and C is of size $n \times n$. In practice, we use a combination of position, gradient and optical flow to represent the pixel in the cuboid, namely,

$$f_i = \left(x, y, t, L_x, L_y, L_t, |L_x|, |L_y|, |L_t|, v_x, v_y, |v_x|, |v_y|\right) \tag{2}$$

where $|\bullet|$ is the absolute value, (x, y, t) is the positional vector of the pixel in the cuboid, (L_x, L_y, L_t) is the gradient, and (v_x, v_y) is the optical flow vector. The spatial gradients (L_x, L_y) and the temporal gradient L_t are calculated by:

$$\begin{aligned} L_x &= L(x+1, y, t) - L(x-1, y, t) \\ L_y &= L(x, y+1, t) - L(x, y-1, t) \\ L_t &= L(x, y, t+1) - L(x, y, t-1) \end{aligned} \tag{3}$$

where $L(x, y, t)$ is the intensity of the pixel at position (x, y, t). The optical flow is obtained using an implementation of the classical optical flow method by Horn and Schunck [42].

Covariance matrices do not lie in the Euclidean space [7] and cannot be compared directly by the Euclidean metric. Several approaches [7, 43, 44] using covariance descriptors for image regions employ the distance measure proposed in [45]:

$$\rho(C_1, C_2) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(C_1, C_2)} \tag{4}$$

where $\{\lambda_i(C_1, C_2)\}_{i=1,\cdots,n}$ are the generalized eigenvalues of C_1 and C_2, computed from $\lambda_i C_1 x_i = C_2 x_i$ and $x_i \neq 0$ are the generalized eigenvectors. The computational complexity to solve a generalized eigenvalue problem using the Arnoldi iteration is $O(dn^2+d^2n)$ [46], where d is the number of dominant eigenvalues and the matrices are of size $n \times n$. Usually, hundreds of cuboids are obtained from each video and hundreds of thousands are obtained from a video dataset. It is apparent from Eq. (4) that each distance measurement for every pair of covariance matrices requires the solution of a generalized eigenvalue problem. Hence, the computational complexity of comparing all covariance matrices in a video dataset is very high. Besides, the traditional covariance feature in the form of matrix does not lie in a Euclidean space. Therefore, it is nontrivial to cluster this type of feature matrices into centriods that are used as the visual words in the BOVW framework.

The covariance matrix is a symmetric nonnegative definite matrix and in our case it is usually symmetric positive definite (SPD). Arsigny *et al.* [3, 10] propose a novel log-Euclidean Riemannian metric for calculating the statistics of SPD matrices. Under this metric, distances and Riemannian means take a closed form. Therefore, we introduce the log-Euclidean Riemannian metric into the similarity measure of covariance matrices. Specifically, given an $n \times n$ covariance matrix C, the singular value decomposition (SVD) of C is denoted as $U \sum U^T$, where U is an orthogonal matrix, and $\sum = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$ is the diagonal matrix of the eigenvalues of C. The matrix logarithm $\log(C)$ is defined as:

$$\begin{aligned} \log(C) &= \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k}(C - I_n)^k \\ &= U \cdot diag\left(\log(\lambda_1), \log(\lambda_2), \cdots, \log(\lambda_n)\right) \cdot U^T \end{aligned} \tag{5}$$

where I_n is an $n \times n$ identity matrix. Under the log-Euclidean Riemannian metric, the distance between two covariance matrices A and B is $\|\log(A) - \log(B)\|_F$, where $\|\bullet\|_F$ is the Frobenius norm.

Fig. 3 shows the computation of the proposed covariance descriptor. After the pixel-level feature extraction, we use two steps to obtain the covariance descriptor of a cuboid. Let C be the covariance matrix defined by Eq. (1). In the first step, we compute the matrix logarithm $\log(C)$ using Eq. (5). In the second step, we reduce $\log(C)$ to an $n(n+1)/2$ dimensional vector by using its upper triangular matrix as follows:

$$\begin{aligned} &vec(\log(C)) = vec(Z) \\ &= [z_{1,1}\ \sqrt{2}z_{1,2}\ \sqrt{2}z_{1,3} \cdots \sqrt{2}z_{1,n}\ z_{2,2}\ \sqrt{2}z_{2,3} \cdots \sqrt{2}z_{2,n} \cdots z_{n,n}] \end{aligned} \tag{6}$$

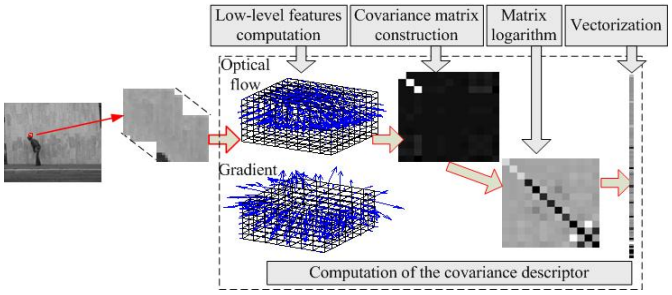


Fig. 3. Computation of the covariance descriptor for the cuboid.

where $Z = \log(C)$ and $z_{i,j}$ is an element of matrix Z. No information is lost because the matrix Z is symmetric. In this paper, the covariance matrix C is of size 13×13, namely n=13. Hence, the right-hand side of Eq. (6) is a 91-dimensional vector. Furthermore, by this vectorization, the following relation holds:

$$\| \log(A) - \log(B) \|_F = \| vec(\log(A)) - vec(\log(B)) \|_2 \qquad (7)$$

Therefore, we can obtain the dissimilarity by computing the Euclidean distance of two corresponding vectors instead of two logarithm matrices.

The proposed covariance descriptor has the following advantages. (1) It combines multiple low-level features in a concise way. (2) It can be computed and compared quickly under the log-Euclidean Riemannian metric. The log-Euclidean mean can be computed approximately 20 times faster than the affine-invariant Riemannian mean [40, 47]. Details of these two metrics can be found in [41, 48]. (3) It is robust to noise. Individual feature vectors corrupted by noise are largely filtered out with an average filter during covariance computation. (4) The covariance matrix does not include any information about the ordering or the number of the feature vectors used to calculate it. This brings a certain degree of invariance to scale, rotation and mis-alignment. In [7] it is proved that large rotations and illumination changes also have little effect on the covariance matrix.

B. Vocabulary Learning

In the BOVW framework, the local covariance features are quantized into a vocabulary. The vocabulary is constructed by clustering a large set of local features from the training videos using the *k*-means clustering method. As described in subsection A, our covariance feature is a 91-dimensional vector and distances between vectors are measured by the Euclidean metric. Therefore, our covariance feature can be input into the *k*-means clustering method. In comparison, the original covariance descriptor and its metric as defined in [7, 43-45] cannot be directly used for *k*-means clustering.

Let $\{v_1, \ldots, v_K\}$ be a vocabulary of K visual words. After building the vocabulary, each covariance feature is mapped to the nearest word in the vocabulary.

C. Spatio-temporal Co-occurrence Matrix for Geometric Context

In visual recognition tasks, constraints obtained from pairs of features have been imposed to improve the standard bag-of-words algorithms. The co-occurrence matrix is an effective non-parameteric model for pair-wise feature distribution. We propose two methods to construct spatio-temporal co-occurrence matrices in the 3D video space. One is based on the ranking according to distances, while the other utilizes directly the values of the distances.

Each video $\boldsymbol{V}$ yields data in the form $\{(\boldsymbol{x}_i, \alpha_i)\}_{1 \le i \le M}$, where $\boldsymbol{x}_i = (x, y, t)$ is the spatio-temporal position vector of the i^{th} local covariance feature (namely, the center of the i^{th} cuboid), α_i is the index of the corresponding visual word, and M is the total number of local features in the video. The distance between two cuboids $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ is measured by the Euclidean distance $\| \boldsymbol{x}_i - \boldsymbol{x}_j \|$ between their centers. The spatio-temporal co-occurrence matrix is defined as $P = (p_{ij}) \in R^{K \times K}$, with each element defined by

$$P(i, j) = p_{ij} = \#\left\{ (\alpha_l, \alpha_m) \,|\, \alpha_l = i,\ \alpha_m = j,\ \|x_l - x_m\| \le d,\ 1 \le l, m \le M \right\} \qquad (8)$$

where α_l and α_m are indices associated with a pair of local features separated by a distance not larger than d, and $\#\{\cdot\}$ means the number of feature pairs satisfying all the conditions listed in the brackets in Eq(8). In fact, d is the radius of the neighborhood of each local feature.

An alternative way of defining a co-occurrence matrix is to use the ranks of neighboring features. We define such a spatio-temporal co-occurrence matrix as $Q=(q_{ij}) \in R^{K \times K}$. The element q_{ij} is the number of the local features with the visual word belonging to the type v_i that are among the r-nearest neighbors of a local feature with a visual word v_j., i.e.,

$$Q(i, j) = q_{ij} = \#\left\{ (\alpha_l, \alpha_m) \,|\, \alpha_l = i, \alpha_m = j, d_{NN}(x_l, x_m) \le r, 1 \le l, m \le M \right\} \qquad (9)$$

where $d_{NN}(\boldsymbol{x}_l, \boldsymbol{x}_m) \le r$ indicates that $\boldsymbol{x}_m$ is among the r^{th} nearest neighbors of $\boldsymbol{x}_l$. Both kinds of spatio-temporal co-occurrence matrices are of size $K \times K$, where K is the size of vocabulary.

The two spatio-temporal co-occurrence matrices differ only in the definition of the neighbor relationship for local features. The second one captures rank information about the relative positions of local features. The rank information is more reliable and robust than the distance information. The distances are affected by changes in scale, or viewpoint. However, the sorting method in the second matrix Q can overcome these problems.

D. The directional Pyramid context modeling

As described above, the spatio-temporal co-occurrence matrices effectively capture the geometric context by modeling undirected local feature pairs of the video sequence. However, the relative positional relationship of the feature pair is not considered. In this section, we present the so-called *Directional Pyramid Co-occurrence Matrix* (DPCM) for modeling the statistics of directed feature pairs. We calculate the directional relationships of feature pairs using a directional vector (θ_s, θ_t), respectively corresponding to spatial direction θ_s and temporal direction θ_t, defined by

$$\theta_s = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \qquad \theta_t = \begin{cases} 1 & t_2 - t_1 \ge 0 \\ -1 & t_2 - t_1 < 0 \end{cases} \qquad (10)$$

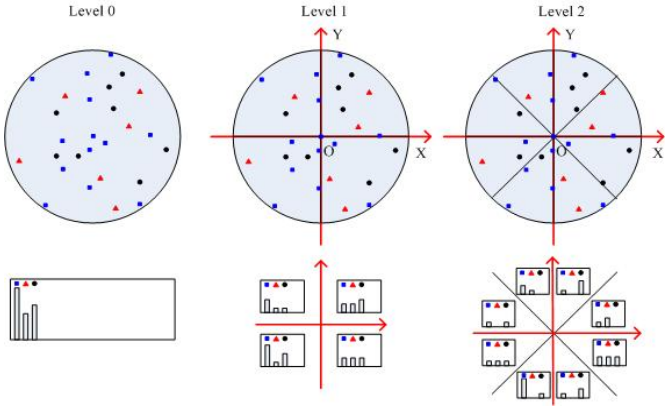


Fig. 4. A toy example of a directional pyramid structure around each local feature in the spatial domain. The first row is the directional pyramid structure of the neighborhood of the central feature only in the spatial domain. Each point denotes a local feature and the three different symbols denote three types of visual words associated with the local features. The second row shows the histograms of visual words obtained for different directions at different levels.

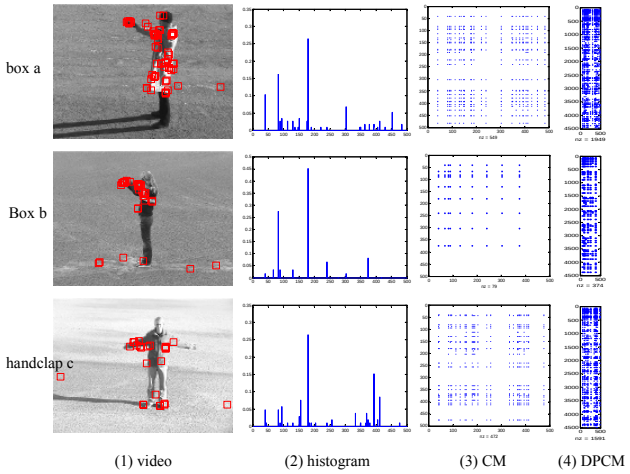
where (x_1, y_1, t_1) is the position vector of the first local feature and (x_2, y_2, t_2) is that of a local feature in the neighborhood of the first local feature. In order to reduce computational complexity and resist local disturbance, we quantify the directional vectors using increasingly finer angle scales and construct a directional pyramid structure. Specifically, level 0 utilizes the undirected local feature pairs. At the finer level l, we define 2^{l+1} bins for the directional parameter θ_s in the spatial domain, and two bins for the directional parameter θ_t in the temporal domain. The directional pyramid structure around each local feature in the spatial domain is as illustrated in Fig. 4. We take the first co-occurrence matrix for example. The central point is the local feature of interest, the circle of radius d defines its neighborhood, and all the local features in the circle are neighboring features of the central feature. Afterwards, the neighborhood of the central feature is partitioned into 2^{l+1} sub-regions in the spatial domain as shown in Fig. 4. The two bins in the temporal domain are combined with the bins in the spatial domain to construct the directional pyramid at the level l ($1 \le l \le L$). In this way, a directional pyramid as shown in Fig. 4 is constructed for each local feature of the video sequence.

After building the directional pyramid structure, we calculate a *spatio-temporal co-occurrence matrix* (STCM) for each direction at each level. The STCM of level 0 is the STCM of undirected cuboid pairs. Without loss of generality, we take the first kind of STCM as an example and introduce the construction of the STCM in other levels. It is the same as in the case of the second kind of STCM. At level 1, eight directional STCM are defined as:

$$P^1_{s,t}(i,j) = \#\{(\alpha_l, \alpha_m) \mid \alpha_l = i,\ \alpha_m = j,\ \|x_l - x_m\| \le d,\ (s-1)\frac{\pi}{2} \le \theta_s < s\frac{\pi}{2},\ \theta_t = t\} \quad (11)$$

$$\theta_s = \arctan\left(\frac{y_l - y_m}{x_l - x_m}\right) \qquad \theta_t = \begin{cases} 1 & t_l - t_m \ge 0 \\ -1 & t_l - t_m < 0 \end{cases}$$

where (x_l, y_l, t_l) and (x_m, y_m, t_m) are the position vectors of the local features α_l and α_m, and s and t respectively denote the bin index of quantified spatial and temporal directions. The



(a)

	histogram	CM	DPCM
Similarity (a , b)	0.5194	0.4987	0.3931
Similarity (a , c)	0.5262	0.3505	0.1972
S (a , b)-S (a , c)	-0.0068	0.1482	0.1959

(b)

Fig. 5. (a) The detected interest points and three kinds of representation of three videos in the KTH dataset. The second column shows the histogram representation of the video. The third and fourth columns are the spatio-temporal co-occurrence matrix (CM) and the 2-level Directional Pyramid Co-occurrence Matrix (DPCM) of the video. (b) The similarities of three videos under different representations.

directional bin index s ranges in {1, 2, 3, 4}, and t ranges in {-1, 1}. Similarly, for level 2, we can calculate 16 directional STPM. In this way, the directional pyramid grows till the finest level L is reached. The number of quantified directions at each level is twice the number at the level below. Moreover, from the definition of the directional STPM in the directional pyramid, we get the following equation:

$$P^0 = \sum_{t=1}^{2}\sum_{s=1}^{4} P^1_{s,t} = \sum_{t=1}^{2}\sum_{s=1}^{8} P^2_{s,t} = \cdots = \sum_{t=1}^{2}\sum_{s=1}^{2^{L+1}} P^L_{s,t} \quad (12)$$

where P^0 denotes the STCM at level 0, and $P^i_{s,t}$ denotes the STCM of the s^{th} spatial direction and the t^{th} temporal direction at level i. In summary, this technique works by partitioning each neighborhood into increasingly finer sub-regions and computing the co-occurrence matrices of feature pairs falling inside each sub-region. The resulting "directional pyramid" is a simple and efficient extension of an undirected co-occurrence matrix video representation. In Fig. 5 (a), we show the histogram, the co-occurrence matrix and the DPCM of three videos on the KTH dataset. Each row is a video and its various representations. The Fig. 5 (b) lists the similarities of these three videos under different video representation, which is analyzed in the next section in detail.

IV. ACTION RECOGNITION BASED ON PYRAMID MATCHING KERNEL OF GEOMETRIC CONTEXT MODEL

With the directional pyramid, each video can be represented as a multi-level directional STCM, namely DPCM. To effectively measure the similarity of two video sequences, we present a *directional pyramid co-occurrence matching kernel* (DPCMK), which serves as a kernel used in the SVM.

A. Directional Pyramid Matching Kernel for DPCM

Each video is represented by multi-resolution matrices as described above in Section III. We create a *Directional Pyramid Co-occurrence Matching Kernel* (DPCMK) to measure the similarity of two video.

Denote a video as $Y = [Y^0, \ldots, Y^l, \ldots, Y^L]$, where Y^l is a list of the co-occurrence matrices at level l. At level l, we concatenate the obtained co-occurrence matrices at all directional ranges of level l to obtain $Y^l = [Y_1^l, Y_2^l, \cdots, Y_{2^{l+2}}^l]$, where Y_d^i is the co-occurrence matrix for the directional index d at level l. The directional index d corresponds to a specific direction (s, t) and in fact the meaning of Y_d^i is equal to that of the symbol "$P_{s,t}^i$" described in the Section III. That is, we build a hierarchical structure for each video and represent the video as a concatenated matrix.

Let $X = [X^0, \ldots, X^L]$ and $Y = [Y^0, \ldots, Y^L]$ be the DPCM representations of two videos. The DPCMK computes a weighted matrix intersection in the hierarchical structure of DPCM. First, we define a "matrix intersection" function S as the similarity for two basic matrices A and B, which measures the "overlap" between two matrices' elements:

$$\mathrm{S}(A, B) = \sum_{i,j} \min\big(A(i, j),\ B(i, j)\big) \tag{13}$$

At each level l, the similarity of X^l and Y^l is defined as the sum of 2^{l+1}*2 matrix intersections of corresponding directional co-occurrence matrices:

$$\begin{aligned} \mathrm{I}(X^l, Y^l) &= \min(X^l,\ Y^l) = \textstyle\sum_{d=1}^{2^{l+2}} \min(X_d^l,\ Y_d^l) \\ &= \textstyle\sum_{i=1}^{K}\sum_{j=1}^{K} \min(x_1^l(i, j), y_1^l(i, j)) + \cdots \\ &\quad + \textstyle\sum_{i=1}^{K}\sum_{j=1}^{K} \min(x_{2^{l+2}}^l(i, j) - y_{2^{l+2}}^l(i, j)) \\ &= \textstyle\sum_{d=1}^{2^{l+2}} \sum_{i=1}^{K}\sum_{j=1}^{K} \min(x_d^l(i, j), y_d^l(i, j)) \end{aligned} \tag{14}$$

where the submatrix X_d^l of X^l is the co-occurrence matrix of the video for directional index d at level l, and $x_d^l(i, j)$ is the element in the i^{th} row and j^{th} column of X_d^l. From Eq. (14), it is seen that the similarity of X^l and Y^l is namely the sum of the minimal value at each corresponding element. In other words, $\mathrm{I}(X^l, Y^l)$ is the number of the matched pairs in the level l. As in PMK [49], the number of the newly matched pairs N^l induced at level l is the difference between successive levels' matrix intersections:

$$N^l = \mathrm{I}(X^l, Y^l) - \mathrm{I}(X^{l+1}, Y^{l+1}) \tag{15}$$

Because level L is the finest level, we compute the number of matches N^l from level L to level 0. The resulting kernel K_Δ is obtained by the weighted sum over the number of matches N^l occurred at each level, and the weight with level l is set to 2^{l-L}:

$$\begin{aligned} \mathrm{K}_\Delta(X, Y) &= \textstyle\sum_{l=0}^{L} \frac{1}{2^{L-l}} N^l \\ &= \textstyle\sum_{l=0}^{L} \frac{1}{2^{L-l}} (I(X^l, Y^l) - I(X^{l+1}, Y^{l+1})) \\ &= \textstyle\sum_{l=0}^{L-1} \frac{1}{2^{l+1}} I(X^{L-l}, Y^{L-l}) + \frac{1}{2^L} I(X^0, Y^0) \end{aligned} \tag{16}$$

where $X^{L+1} = Y^{L+1} = 0$.

Moreover, the number of local feature pairs is unstable in each video, because it relies heavily on video duration and resolution. Hence, we normalize the kernel K_Δ as follows:

$$\mathrm{K}_\Delta'(X, Y) = \frac{\mathrm{K}_\Delta(X, Y)}{\min(\sum_{i,j} X^0(i, j), \sum_{i,j} Y^0(i, j))} \tag{17}$$

where $X^0(i, j)$ is the element of the STCM in level 0, and $\mathrm{K}_\Delta'(X, Y)$ is used as the final DPCMK.

The DPCMK effectively combines similarities over different levels in the hierarchical structure. The newly matched pairs at coarser levels, which are not matched at the finer levels, are also considered in the DPCMK. This can occur in some cases in action recognition, for example when the same action is performed by different persons, or the same action is performed by the same person at different times. Even if these intra-class actions are not matched at fine levels, they can still be treated as matches at a coarser level. Therefore, according to the directional pyramid structure and DPCMK, our approach can overcome the intra-class variations of actions.

In Fig.5 (b), we show the similarities of videos under the histogram based on the traditional BOVW, STCM and DPCM. We respectively use the histogram intersection, matrix intersection and the proposed DPCMK to compute the similarity. The first two videos belong to the "boxing" class and the third video is the "handclapping" class. Moreover, the second video '*b*' has the highest similarity with the first video '*a*' in all the videos belonging to the action class 'boxing' on KTH dataset, and in all the "handclapping" videos, the video '*c*' is the most similar one with the video of '*a*' under the histogram representation. From the table in Fig.5 (b), the similarity between '*a*' and '*b*' is smaller than that between '*a*' and '*c*', which is not consistent with actual request. Obviously, high performance requires that the similarity between two videos with the same action class should be larger than that between one video and another video with different class. However, STCM and DPCM perform well because they meet this actual request. It is likely that STCM and DPCM improve the discriminative power of the video representation by combining the geometric information.

B. SVM classification

We adopt the SVM classifier [50] and use the DPCMK as its kernel function for human action recognition. The DPCMK is directly combined with the SVM classifier by taking advantage of the fact that DPCMK is a Mercer kernel, i.e., a positive semi-definite kernel. This can be derived from the fact that min(·,·) operation is a Mercer kernel and DPCMK is the summation of limited number of minimum operations.

V. EXPERIMENTS

We test our method on six datasets including KTH dataset [5], Weizmann dataset [23], UCF sports dataset [8], UCF CIL action dataset [54], the Feature Films dataset [8], and the facial expression dataset [6].

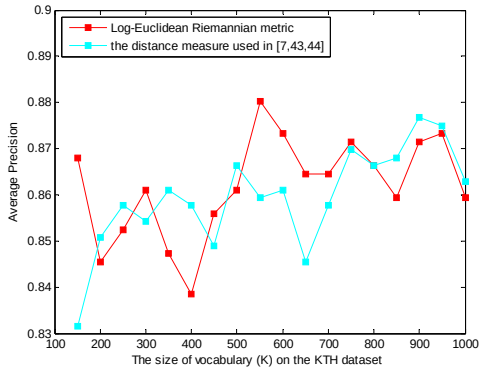


Fig. 6. Comparison of two metrics for covariance descriptors vs. vocabulary size K on the KTH dataset.

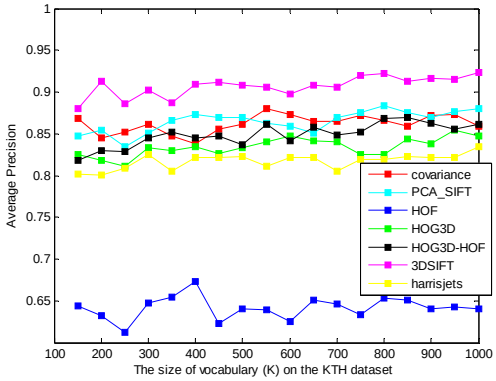


Fig. 7. Recognition accuracy obtained by the seven descriptors vs. vocabulary size K on the KTH dataset.

The KTH dataset contains 599 sequences and six types of actions (walking, jogging, running, boxing, hand waving and hand clapping) performed by 25 subjects in four different scenarios. We conduct three groups of comparison experiments on KTH dataset to investigate the following five aspects: i) the efficiency of using the log-Euclidean Riemannian metric for covariance descriptors with respect to distance measures taken from the literature [7, 43, 44]; ii) performance comparison with several state-of-the-art cuboid descriptors, such as PCA-SIFT [7, 14], histogram of oriented gradients (HOG3D) [3], HOF[22], HOG3D–HOF; iii) performance gain of the DPCM and CM based methods with respect to the traditional BOVW approach (based on the histogram of video words); iv) the influence of vocabulary sizes; v) performance dependency on the neighborhood size in DPCM and CM based methods.

A. Distance measure experiments of Covariance descriptor

According to the discussion in section III, each cuboid is represented by a 91-D vector and the distance between these vectors is measured by the Euclidean distance. We employ the k-means clustering method to construct the vocabulary.

To validate the efficiency of the log-Euclidean Riemannian metric for covariance descriptors, we compare it with the distance measure used in [7, 43, 44] on the KTH dataset. Under the distance measure of [7, 43, 44], the inputs of the k-means clustering method are in the form of the covariance matrices. Therefore, we modify the traditional k-means clustering method and compute distances of the matrices by Eq. (4) instead of the Euclidean distance. In each iteration of the k-means clustering, the generalized eigenvalues of every matrix pair need to be computed, which costs much computation time. Obviously, the construction of the vocabulary based on our descriptor is easier than that based on [7, 43, 44].

The other experimental configurations are all the same. In order to reduce computation time, we represent each video as a histogram of visual words and employ the SVM classifier. We use the histogram intersection kernel for SVM classification, namely, the 'minimum' kernel:

$$\mathrm{K}(h_1, h_2) = \sum_{i=1}^{k} \min\left(h_1(i),\ h_2(i) \right) \qquad (18)$$

where h_1 and h_2 are respectively the histogram representations of two video sequences.

Fig. 6 shows the recognition accuracy curves of these two metrics for the covariance descriptors vs. the vocabulary size K on the KTH dataset. For K={150, 200, …, 1000}, the average recognition accuracy of our covariance descriptor is 86.19% which is 0.24% higher than the latter. Although the recognition performances of the covariance descriptors under these two metrics are similar, our covariance descriptor has low computational cost.

B. Descriptors comparison experiments on the KTH dataset

We compared our descriptors with six other popular descriptors: PCA-SIFT [7, 14], histogram of 3D oriented gradients (HOG3D) [3], HOF[22], HOG3D -HOF, 3D SIFT [20] and Laptev's spatio-temporal jets [5]. Specifically, PCA-SIFT descriptor applies Principal Components Analysis (PCA) to the normalized gradient vector which is formed by flattening the horizontal and vertical gradients of all the points in the cuboid. HOG3D uses the histogram of normalized gradients and HOF uses the histogram of optical flows where the gradients and optical flows are obtained at all the points in the cuboid. HOG3D -HOF is the combination of HOG3D and HOF, namely a concatenation of the normalized gradient histogram and the normalized optical flow histogram. 3D SIFT is an extension of 2D SIFT for images. Laptev's spatio-temporal jets are 34-D vectors $l = (L_x, L_y, L_t, L_{xx}, \cdots, L_{tttt})$ using derivatives, where L is the convolution of the original image with an anisotropic Gaussian kernel with independent spatial variance and temporal variance. Except for descriptors, other experimental configurations are all the same. We use the histogram of visual words to represent each video and employ the histogram intersection as the kernel of the SVM classifier.

Fig. 7 shows the recognition accuracy curves of the seven descriptors vs. the vocabulary size K on the KTH dataset. For K ={150, 200,…,1000}, the 3D SIFT descriptor achieves the best results and the HOF descriptor achieves the worst results. The average accuracy of our covariance descriptor is 86.19% which is 22.02% higher than HOF descriptor and 4.51% lower than the 3D SIFT descriptor. The recognition results of the other five descriptors are similar to each other, and the average accuracy of our covariance descriptor is respectively 2.78%, 1.28%, 4.49% higher than HOG3D descriptor, HOG3D–HOF

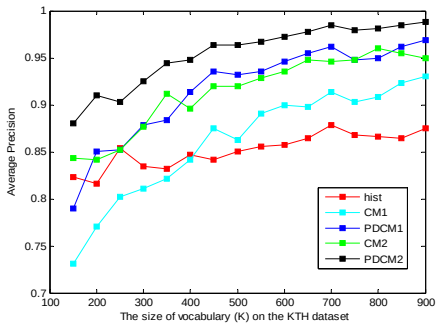


Fig. 9. Recognition accuracy obtained by the five approaches vs. vocabulary size on the KTH dataset.

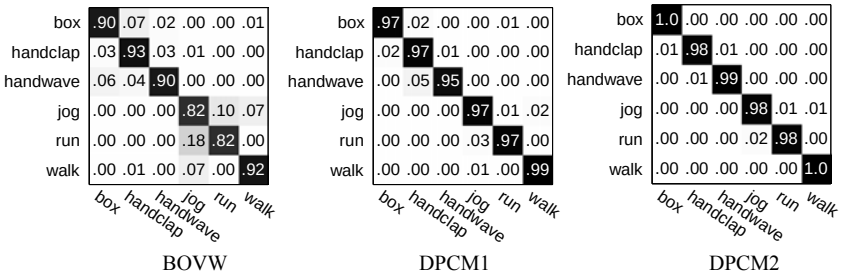


Fig. 10. The confusion matrices of the ordinary BOVW approach and the proposed two kinds of DPCM based approaches on the KTH dataset.

descriptor, Laptev's spatio-temporal jets and 0.31% lower than PCA-SIFT descriptor.

Compared with other descriptors, 3D SIFT descriptor has higher code complexity and computational time, which can be major disadvantages in application to large scale video data. Therefore, although 3D SIFT descriptor achieves higher performance, it is not often employed for action recognition while HOF, HOG3D–HOF, PCA-SIFT descriptors are popular choices. Our covariance descriptor, with low computational complexity, performs better than or comparable to the HOF, HOG3D–HOF and PCA-SIFT descriptors. Therefore, our covariance descriptor can serve as an alternative to action recognition compared with other descriptors.

C. Geometrical modeling comparison experiments on the KTH dataset

It is also shown in Fig.7 that the best of the seven descriptors under the histogram representation of video sequence just achieves a recognition accuracy of 92.36% and still needs further improvement. In this subsection, we conduct experiments using the geometrical modeling based video representation, and evaluate whether the geometrical information improves the action recognition accuracy.

We use the KTH dataset to evaluate the proposed video representation based on DPCM. The three-level pyramid structure is used to model the directed interest point pairs. First, we assess the performances with respect to different size of nearest neighbors in two kinds of DPCM and CM under a vocabulary size K=500, as illustrated in Fig. 8. The absolute values of positional distance based DPCM and CM are denoted as "DPCM1" and "CM1" in Fig.8 (a), while the rank of positional distance based DPCM and CM are denoted as "DPCM2" and "CM2" in Fig.8 (b). The graphs in Fig.8 (a) show the accuracy for DPCM1 and CM1 as functions of the distance d. The graphs in Fig.8 (b) show the accuracy for DPCM2 and CM2 as functions of r. The recognition accuracies of DPCM based method and that of CM based method respectively range from 92.36% to 94.27% and from 86.81% to 88.19% when d >0.1 in Fig.8 (a), and they respectively range from 93.58% to 97.40% and from 88.54% to 94.97% in Fig.8 (b). The dependency of the recognition accuracy on the vocabulary size is not very significant. Thus, we set d to 0.4 and r to 60 in the remaining experiments.

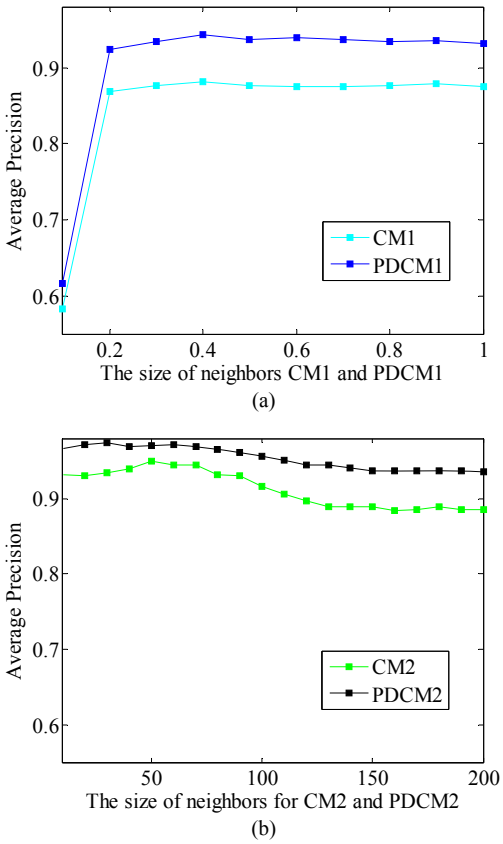


Fig. 8. Recognition accuracy with respect to nearest neighbor sizes in two kinds of DPCM (CM) under vocabulary size K=500 on the KTH dataset.

Furthermore, we compare our DPCM based approach with two other approaches: the traditional BOVW approach and the approach based on co-occurrence matrix. Specifically, the traditional BOVW approach employs a histogram of visual words to represent each video, and it uses the histogram intersection to measure the similarity of histograms for an SVM classifier. This is the method used in subsection V (A) (see Eq. (18)). For the approach based on the co-occurrence matrix, we adopt the same experimental configurations with our DPCM based approach except that a co-occurrence matrix is used instead of the directional pyramid co-occurrence matrix. Fig. 9 displays the accuracy curves of the five approaches vs. the vocabulary size K. It is shown that the second kind of DPCM approach, namely DPCM based on the

TABLE I
COMPARISON OF STATE-OF-THE-ART METHODS ON THE KTH DATASET.

Methods	validation	Rate
Schuldt *et al.* [5]	original	71.72
Schuldt *et al.* [5]	l-o-o	81.70
Dollár *et al.*[6]	l-o-o	81.17
Niebles *et al.* [31]	l-o-o	81.50
Bregonzio *et al.*[39]	l-o-o	93.17
Kim *et al.* [30]	l-o-o	95.33
Our "DPCM1"	l-o-o	**96.88**
Our "DPCM2"	l-o-o	**98.78**

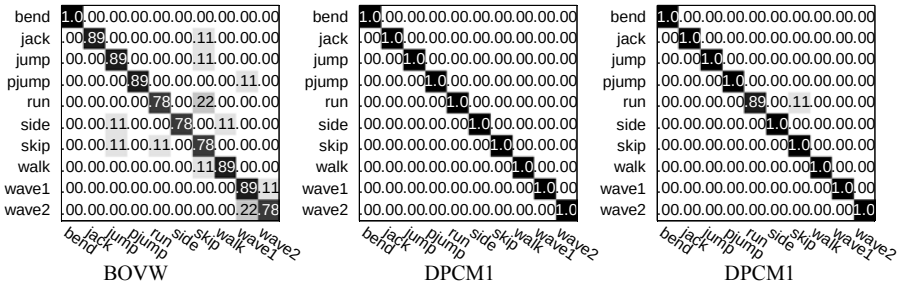


Fig. 13. The confusion matrices of the ordinary BOVW approach and the proposed two kinds of DPCM based approaches on the Weizmann dataset.

ranking of positional distance, has the highest accuracy in all cases. For K={150, 200, 250,…, 900}, the performances of five methods are in the order "DPCM2" > "DPCM1" > "CM2" > "CM1" > "hist" in most cases, where "hist" denotes the traditional BOVW approach. "DPCM2" achieves 95.44% average recognition accuracy, which is respectively 3.80%, 3.98%, 9.31%, and 10.25% higher than the averages for "DPCM1", "CM2", "CM1", and "hist". These experiments validate our claims that the geometrical modeling based approaches improve the recognition accuracy of the BOVW approach by considering the geometrical relationship of local features, and that the DPCM is better than the CM for using the information in directed local features.

Besides, "DPCM2" achieves the best recognition accuracy (i.e. 98.78%), "DPCM1" achieves 96.88%, while the BOVW approach reaches only 88.06%. Fig. 10 shows the confusion matrices of the BOVW approach and our DPCM based approaches on the KTH dataset. Each row of the confusion matrix corresponds to the ground truth class, and each column corresponds to the assigned class. The confusion matrix of DPCM2 approach shows that the "hand" related actions and the "foot" related actions are a little confused within each of these two big action classes, but the two big action classes are always well separated from each other. It is likely that this separation is achieved because the DPCM effectively captures the geometric information.

Table I compares the performances of our method with other recently developed methods. Schuldt *et al.* [5] use the original validation procedure, and divide all sequences with respect to the subjects into a training set (8 persons), a validation set (8 persons) and a test set (9 persons). Other methods in Table I all employ the leave-one-out cross validation (l-o-o). Moreover, we test the method proposed by Schuldt *et al.* [5] under the leave-one-out cross validation for impartial comparison. Schuldt *et al.* [5] and Dollár *et al.* [6] employ local spatio-temporal features and the BOVW model with SVM classification schemes for recognition, but they use different spatio-temporal features. These two methods achieve equivalent accuracy under the leave-one-out cross validation. Our approach achieves the best results by using the DPCM to encode both the appearance and geometric information.

The above reported results for a set of experiments on the KTH dataset show that: i) the proposed approaches based on DPCM outperform the BOVW approach largely; ii) the DPCM based approaches outperform approaches based on the

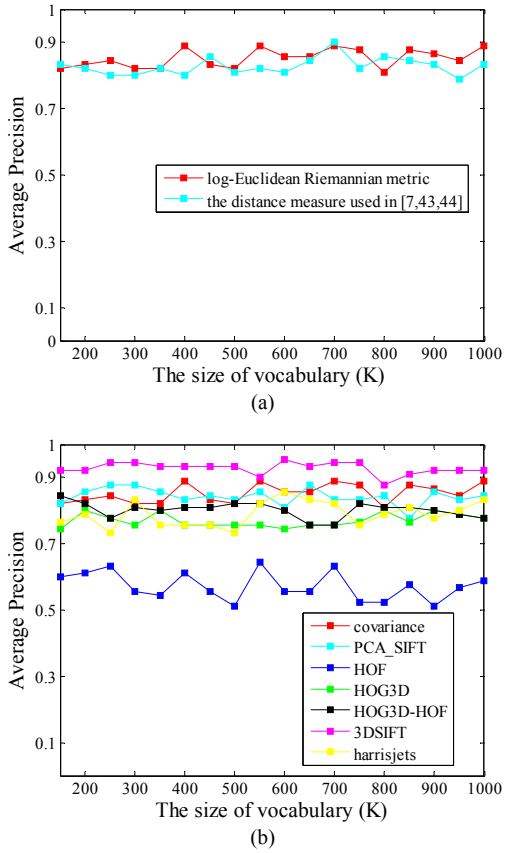


Fig. 11. Cuboid descriptor related comparison experiments on the Weizmann dataset. The figure (a) shows the recognition accuracy curves obtained by two metrics for covariance descriptors vs. vocabulary size K. The figure (b) shows the recognition accuracy curves obtained by the seven descriptors vs. vocabulary size K.

CM; iii) the ranking of positional distance based DPCM approach outperforms DPCM approach based on the absolute values of positional distance; iv) performance varies with the size of the vocabulary, but the approach based on DPCM achieves the best results; v) the DPCM and CM approaches are not very sensitive to the size of neighborhood.

D. Experimental results on the Weizmann dataset

The Weizmann human action dataset contains 93 samples and 10 different actions including Walking, Running, Jumping, Galloping sideways, Bending, One-hand waving, Two-hands waving, Jumping in place, Jumping Jack and Skipping, all performed by 9 subjects. In each run, 8 of the subjects' videos are used as the training set and the remaining one subject's videos form the test set. The results are averaged over 9 runs.

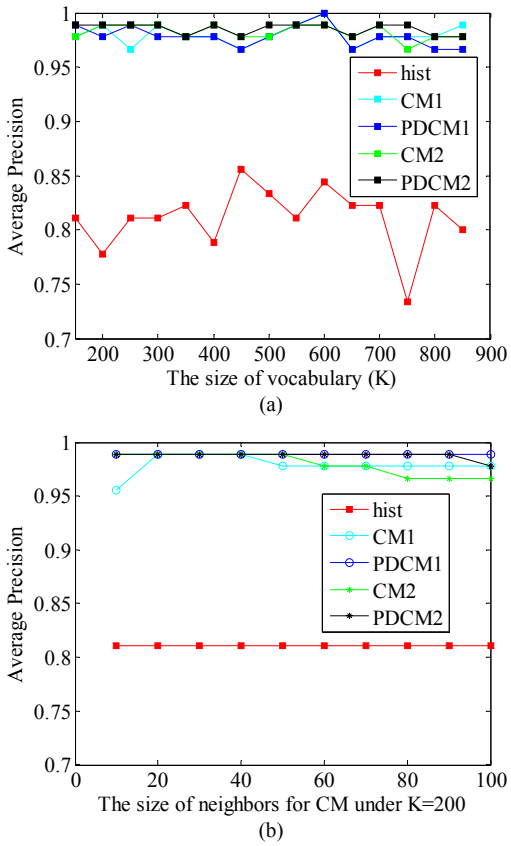


Fig. 12. Cuboid descriptor related comparison experiments on the Weizmann dataset. (a) The recognition accuracy curves obtained by the five approaches vs. vocabulary size K, with d=30 and r=0.3. (b) The recognition accuracy curves with respect to different nearest neighbor sizes in two kinds of DPCM (CM) under vocabulary size K=200.

We perform experiments similar to those on the KTH dataset. These experiments are divided into two groups: cuboid descriptors related experiments and DPCM based geometrical modeling experiments. Fig. 11 shows cuboid descriptors related experimental results on the Weizmann dataset, including recognition accuracy curves of the covariance descriptors based on two metrics and the seven descriptors vs. the vocabulary size K={150,200,…,1000}. In Fig. 11(a), the red line shows the recognition results obtained by our covariance descriptor under the log-Euclidean Riemann metric. The red line is in general higher than the blue line representing the covariance descriptor under metric used in [7, 43, 44]. Moreover, we compare the computational complexity of these two metrics by testing the cost time of computing the distances of all the detected cuboid covariance features under these two metrics. That is because that the distance measure is iteratively required in the clustering method for vocabulary construction. For all the videos in the Weizmann dataset, 8112 cuboids are detected and 8112 covariance features are obtained. We measure the time taken to compute the 8112×8112 distance matrices of all covariance features. As illustrated in Table II, the computation of the distance matrix under log-Euclidean Riemann metric takes only a few seconds, while the time required for metric [7] is more than 10 times longer.

(a) The UCF sport dataset

(b) The UCF CIL action dataset

(c) Feature Films dataset

(d) The facial Expression Database

Fig. 14. Representative frames from videos in the four datasets. Each frame is from one type of class. For (a), (b) and (d), each frame is from one type of class. For (c), the left three frames are from the *kissing* actions, and the right three frames show the *hitting or slapping* actions.

As shown in Fig. 11 (b), we compare our covariance descriptor with six popular descriptors on the Weizmann dataset. For K = {150, 200,…,1000}, the average recognition accuracy of our covariance descriptor is 85.25% which is higher than the other descriptors except the 3D SIFT. The PCA-SIFT descriptor and the HOG3D–HOF descriptor also have good average recognition accuracy, that is, 84.26% and 80.25%. The average recognition accuracies and computation times for seven descriptors are shown in Table III. The 3D SIFT descriptor has a high computational complexity, and its computational time is three times more than that of our covariance descriptor. The main cost of our covariance descriptor arises from the computation of pixel optical flows. Besides, the dimensions of the 3D SIFT descriptor and our covariance descriptor are 640 and 91 respectively. A higher dimension needs more storage space and computational time for subsequent processes such as vocabulary construction, co-occurrence matrix computation and so on. Therefore, for the 3D SIFT descriptor, it costs more time not only in feature calculation but also in using it.

Fig. 12 shows DPCM based experimental results on the Weizmann dataset, including two experiments: the results of the five methods shown as a function of the vocabulary size K and the results of the two kinds of DPCM (CM) methods shown as a function of the size of nearest neighbors d or r. In the two experiments, it can be seen that i) the CM and PDCM based methods, all exceed the BOVW method significantly and achieve good results with about 98% recognition accuracy; the BOVW method only obtains 81.11% average recognition accuracy; ii) the DPCM based approaches are slightly better than the CM based approaches and more stable with respect to changes in the number of nearest neighbors. Therefore, these experiments demonstrate the effectiveness of the proposed

TABLE IV. COMPARISON OF DIFFERENT STATE-OF-THE-ART METHODS ON THE UCF SPORTS DATASET.

Methods	Accuracy (%)
Rodriguez *et al.* [8]	69.2
Wang *et al.* [22]	85.6
Sun *et al.* [42]	86.9
Kovashka *et al.* [58]	87.27
Our approach "DPCM1"	86.67
Our approach "DPCM2"	**87.33**

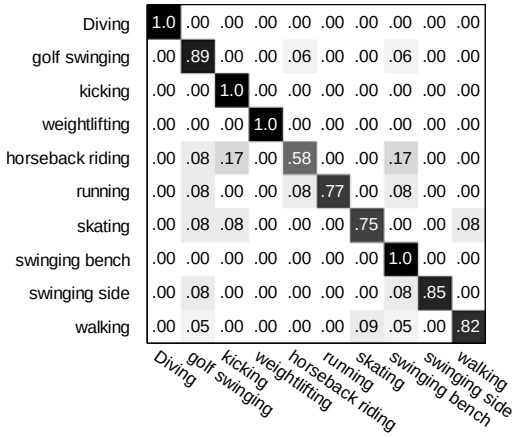


Fig. 15. The confusion matrix of the proposed approach DPCM2 on the UCF sports dataset.

DPCM and CM on the Weizmann dataset.

Further, "DPCM1" achieves the best recognition accuracy (e.g. 100%), "DPCM2" achieves 98.81%, while the BOVW approach only achieves 85.56%. Fig. 13 shows the confusion matrices of the BOVW approach and our DPCM based approaches on Weizmann dataset.

E. Experimental results on UCF sports dataset

This dataset [8] consists of 150 action videos including 10 sport actions, diving, golf swinging, kicking, weightlifting, horseback riding, running, skating, swinging bench, swinging side angle, and walking. It collects a natural pool of actions featured in a wide range of scenes and viewpoints, and in unconstrained environments, as illustrated in Fig. 14 (a).

The UCF sports database is tested in a leave-one-out manner, with each example chosen as a test video in turn, following [8, 58]. Table IV shows the results obtained using several state-of-the-art methods on this dataset. The overall average accuracy for the UCF dataset using our approach "DPCM2" is 87.33%, which demonstrates the effectiveness of our proposed approaches on the realistic and complicated dataset. Fig. 15 shows the confusion matrix of our DPCM2 based approaches on UCF sports dataset. It achieves 100% recognition accuracy for four action classes.

F. Experimental results and analysis on the UCF CIL action dataset

The UCF CIL dataset is collected by Shen *et al.* [54, 55, 57] from the Internet, consisting of 56 sequences of 8 classes of actions. Each action is performed by different subjects, and the videos are taken by different unknown cameras from various viewpoints, as illustrated in Fig. 14 (b).

We employ the leave-one-out manner to test our approach. At each run, one video is used as a test and the remaining videos are used as a training set. The results are reported as the average of all runs. Table V lists the results obtained by our methods (DPCM1 and DPCM2), the BOVW methodbased on our covariance descriptor, the original results of [54], and the point triplet methodproposed in [55]. Our methods achieve 100% accuracy, which is significantly higher than the accuracy of 89.66% obtained by the BOVW method. The results on the UCF CIL action dataset demonstrate the effectiveness of our proposed approach on the multi-view dataset.

TABLE VI
THE RECOGNITION ACCURACIES AND COMPUTATION TIMES FOR TWO METRICS ON THE WEIZMANN DATASET.

Class	Rodriguez *et al.* [8]	Yeffet *et al.* [59]	BOVW	Ours
Kissing	66.4%	77.3%	85.4%	**91.67%**
Slapping	67.2%	84.2%	89.4%	**92.94%**
Average	66.8%	80.75%	87.4%	**92.27%**

TABLE V. COMPARISON OF DIFFERENT STATE-OF-THE-ART METHODS ON THE UCF CIL ACTION DATASET.

Methods	Accuracy
Shen *et al.* [54]	95.83%
Shen *et al.* [55]	**100%**
BOVW	89.66%
DPCM1	**100%**
DPCM2	**100%**

G. Experimental results and analysis on Feature Films

Rodriguez *et al.* [8] collected a dataset of actions performed in a range of film genres consisting of classic old movies, comedies, a scientific movie, a fantasy movie and romantic films. This dataset provides 92 samples of action classes "kissing" and 112 samples of "hitting/slapping." As illustrated in Fig. 14 (c), the extracted samples cover a wide range of backgrounds and view points.

The test for this dataset proceeds in a leave-one-out fashion. Given the significant intra-class variability present in the movie scenes, the recognition task is challenging. Table VI shows several state-of-the-art methods on this dataset. In both categories, our method shows a higher performance than previously reported. In [8], Rodriguez *et al.* also use the BOVW framework together with the PCA-SIFT descriptor. From Table VI, the BOVW method with the proposed covariance descriptor achieves better performance, demonstrating the effectiveness of our descriptor.

H. Experimental results and analysis on the Facial Expression Database

Although our main goal is to recognize human actions, our framework can also be adapted to other application domains that involve spatio-temporal matching. We use our algorithm to classify facial action video sequences on the facial

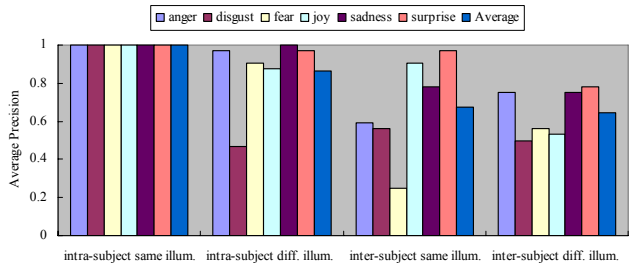


Fig. 16. Average recognition rates of each action class in the facial expression dataset under four different setting cases.

expression database [6]. The face data are performed by two individuals. There are six different classes of emotion expressions and two lighting setups. The expressions are anger, disgust, fear, joy, sadness and surprise. Certain expressions are quite distinct, such as sadness and joy, sadness and surprise. Certain expressions are quite distinct, such as sadness and joy, while others are fairly similar, such as fear and surprise. Under each lighting setup, each individual repeats each of the six expressions eight times. The individual always starts with a neutral expression, expresses an emotion, and returns to neutral, all in about 2 seconds. One representative frame from each action category is shown in Fig.14 (d).

Following [6], in each experiment, we train on a single subject under one of the two lighting setups and test on four cases: (1) the same subject under the same illumination, which is evaluated in a leave-one-out fashion, (2) the same subject under different illumination, (3) a different subject under the same illumination, and (4) a different subject under different illumination. Using these four cases, we investigate how identity and lighting affect the algorithm's performance. In each case, we repeat the experiments four times, where at each time one subject under one lighting setup is used as the training set. The reported results in Fig. 16 are the averages over each set of four experiments, obtained by our proposed DPCM algorithm. The parameter settings are as follows: the vocabulary size is 250, the number of cuboids detected in each video is 30, and the scale in interest point detection is set to 2. Our algorithm generates comparable results to the best results reported in [6].

I. Discussion

The experiments on the KTH dataset and the Weizmann dataset show that: i) our covariance descriptor under the log-Euclidean Riemannian metric is a useful cuboid descriptor for video action recognition, with high computational efficiency; ii) it is beneficial to include geometrical information about the relative positions of cuboids to improve the recognition performance; iii) the proposed DPCM and its DPCMK significantly improve the recognition precision. We achieve the highest classification accuracies among reported results on the KTH and Weizmann datasets, namely, 98.78% and 100%. The experimental results on the UCF sports dataset, the Feature Films dataset, the UCF CIL action dataset and the facial expression dataset show that the proposed method adapts well to realistic datasets with complicated backgrounds, to the multi-view actions and to other application domains such as facial expression recognition.

VI. CONCLUSION

In this paper, we have developed a new framework to recognize human actions from video sequences. In the framework, the covariance matrix of the low-level features from the cuboid is used to represent the local spatio-temporal property of a video sequence under the log-Euclidean Riemannian metric. Lying in the Euclidean space, our covariance features can be clustered by the k-means method to form the vocabulary. We have further proposed a Directional Pyramid Co-occurrence Matrix (DPCM) to represent a video sequence, which effectively captures simultaneously the local appearance information and the geometrical-temporal context information. The discriminative power of the proposed DPCM for video representation has been demonstrated on several benchmark video datasets, in comparison with several state-of-the-arts algorithms. In particular, it greatly improves the results obtained by geometrically unconstrained BOVW approaches, as well as those by Spatio-temporal Co-occurrence Matrix (CM).

REFERENCES

[1] T. B. Moeslund, A. Hilton, V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Understanding*, vol. 104, no. 2, pp. 90-126, 2006.

[2] J. K. Aggarwal, S. Park, "Human motion: modeling and recognition of actions and interactions," in *Proc. 2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pp. 640-647, 2004.

[3] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Comput.*, vol. 28, no. 6, pp. 976-990, 2010.

[4] F. Perronnin, "Universal and Adapted Vocabularies for Generic Visual Categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1243-1256, 2008.

[5] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: A Local SVM Approach," in *Proc. Int. Conf. Pattern Recognit.*, pp. 32-36, 2004.

[6] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition Via Sparse spatio-temporal Features," in *Proc. 2nd Joint IEEE Int. Workshop Visual Surveill. Performance Eval. Tracking and Surveill.*, pp. 65-72, 2005.

[7] O. Tuzel, F. Porikli, and P. Meer, "Region Covariance: A Fast Descriptor for Detection and Classification," in *Proc. Ninth European Conf. Comput. Vis.*, vol. 2, pp. 589-600, 2006.

[8] Mikel D. Rodriguez, Javed Ahmed, Mubarak Shah, "Action MACH: a spatiotemporal maximum average correlation height filter for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, pp. 1-8, June 2008.

[9] F.I. Bashir, A.A. Khokhar, D. Schonfeld, "Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1912-1919, 2007.

[10] N.P. Cuntoor, B. Yegnanarayana, R. Chellappa, "Activity Modeling Using Event Probability Sequences," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 594-607, April 2008.

[11] J.C. Nascimento, M. Figueiredo, J.S. Marques, "Trajectory Classification Using Switched Dynamical Hidden Markov Models," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1338-1348, 2010.

[12] Q. Shi, L. Cheng, L. Wang and A. Smola, "Human Action Segmentation and Recognition Using Discriminative Semi-Markov Models," *Int. J. Comput. Vision*, vol. 93, no. 1, pp. 22-32, 2011.

[13] A. Madabhushi, J.K. Aggarwal, "A Bayesian Approach to Human Activity Recognition," In *Proc. Second IEEE Workshop on Visual Surveillance (VS'99)* , pp. 25-32,1999.

[14] R. Vezzani, M. Piccardi, R. Cucchiara, "An efficient Bayesian framework for on-line action recognition," In *Proc. Inf. Conf. Image Process.*, pp. 3553-3556, 2009.

[15] A. J. Sarkar, Y. Lee, S. Lee, "A Smoothed Naive Bayes-Based Classifier for Activity Recognition," *Iete Technical Review*, vol.27, no, 2, pp. 107-119, 2010.

[16] M. J. Lucena, J. M. Fuertes and N. P. Blanca, "Human Motion Characterization Using Spatio-temporal Features," In *Proc. the 3rd Iberian Conf. Pattern Recognition Image Analysis*, pp. 72-79, 2007.

[17] J. Niebles, H.Wang, and L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial Temporal Words," *Int. J. Comput. Vision*, pp. 299-318, 2008.

[18] K. Mikolajczyk, and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[19] K. Yan, R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 506-513, 2004.

[20] P. Scovanner, S. Ali, M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," In *Proc. ACM Multimedia*, pp. 357-360, 2007.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, pp. 91-110, 2004.

[22] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, "Evaluation of local spatio-temporal features for action recognition," in *Proc. of British Machine Vis. Conf.*, pp. 127-137, 2009.

[23] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, pp.1395–1402, Oct. 2005.

[24] S. Wong, R. Cipolla, "Extracting spatiotemporal interest points using global information," in *Proc. IEEE Int. Conf. Comput. Vis*., pp. 1-8, 2007.

[25] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1-8, 2008.

[26] A. Bosch and A. Zisserman, "Pyramid histogram of oriented gradients (phog)," (http://www.robots.ox.ac.uk/vgg/research/caltech/phog.html)

[27] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. of European Conf. Comput. Vis.*, pp. 428-441, 2006.

[28] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3Dgradients," in *Proc. of British Machine Vis. Conf.*, pp. 995-1004, 2008.

[29] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1816-1823, 2005.

[30] S. Wong, T. Kim and R. Cipolla, "Learning Motion Categories using both Semantic and Structural Information," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1-6, 2007.

[31] J. C. Niebles, and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1-8, 2007.

[32] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2169-2178, 2006.

[33] J. Choi, W. J. Jeon, and S. C. lee, "Spatio-Temporal Pyramid Matching for Sports Videos," in *Proc. ACM Int. Conf. Multimedia Information Retrieval (MIR)*, pp. 291-297, 2008.

[34] C. Yuan, X. Li, W. Hu, H. Wang, "Human action recognition using pyramid vocabulary tree," in *Proc. Ninth Asia Conf. Comput. Vis.*, Xi'an, pp. 527-537, 2009.

[35] J. Li, W. Wu, T. Wang, Y. Zhang, "One step beyond histograms Image representation using Markov stationary features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp.1-8, 2008.

[36] B. Ni, S. Yan, and A. Kassim, "Directed Markov Stationary Features for visual classification," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 825-828, 2009.

[37] H. Ling, S. Soatto, "Proximity Distribution Kernels for Geometric Context in Category Recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1-8, 2007.

[38] J. Sun, X. Wu, S. Yan, L. Cheong, T. Chua, and J. Li, "Hierarchical Spatio-Temporal Context Modeling for Action Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2004-2011, 2009.

[39] M. Bregonzio, S. Gong and T. Xiang, "Recognising Action as Clouds of Space-Time Interest Points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1948-1955, 2009.

[40] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian Detection via Classification on Riemannian Manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713-1727, October, 2008.

[41] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng, "Visual Tracking Via Incremental Log-Euclidean Riemannian Subspace Learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1-8, 2008.

[42] B.K.P Horn, and B.G. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp.185-203, 1981.

[43] P. C. Cargill, C. U. Rius, D. Mery, A. Soto, "Performance Evaluation of the Covariance Descriptor for Target Detection," in *Proc. Int. Conf. Chilean Computer Science Society (SCCC)*, pp.133-141, 2009.

[44] Y. Cai, V. Takala and M. Pietikainen, "Matching Groups of People By Covariance Descriptor," in *Proc. Int. Conf. Pattern Recognit.*, pp.2744-2747, 2010.

[45] W. Forstner, B. Moonen, "A metric for covariance matrices," *Technical report*, Dept. of Geodesy and Geoinformatics, Stuttgart University, 1999.

[46] R. B. Lehoucq, D. C. Sorensen, and C. Yang, "ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods," *SIAM*, 1998.
http://www.ec-securehost.com/SIAM/SE06.html.

[47] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian Framework for Tensor Computing," *Int. J. Comput. Vision*, vol. 66, no. 1, pp.41-66, 2006.

[48] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices," *SIAM J. Matrix Analysis and Applications*, vol. 29, no. 1, pp. 328-347, 2006.

[49] K. Grauman and T. Darrell, "Pyramid match kernels: Discriminative classification with sets of image features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol.2, pp. 1458-1465, 2005.

[50] C. Chang and C. Lin. LIBSVM: a library for SVMs, 2001.

[51] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez, "Cross-View Action Recognition from Temporal Self-Similarities," in Proc. European Conf. Comput. Vis., pp. 1-14, 2008.

[52] I. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-Independent Action Recognition from Temporal Self-Similarities," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 1, pp. 172-185, 2011.

[53] C. Sun, I. Junejo, and H. Foroosh, "Action Recognition using Rank-1 Approximation of Joint Self-Similarity Volume," in Proc. IEEE Int. Conf. Comput. Vis., pp.1007–1012, 2011.

[54] Y. Shen, and H. Foroosh, "View Invariant Action Recognition Using Fundamental Ratios," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp.1-6, 2008.

[55] Y. Shen, and H. Foroosh, "View Invariant Action Recognition from Point Triplets," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 10, pp. 1898-1905, 2009.

[56] P. Yan, S. M. Khan, and M. Shah, "Learning 4D Action Feature Models for Arbitrary View Action Recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp.1-7, 2008.

[57] Y. Shen and H. Foroosh, "View Invariant Recognition of Body Pose from Space-Time Templates," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp.1-6, 2008.

[58] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2010.

[59] L. Yeffet, and L. Wolf, "Local Trinary Patterns for Human Action Recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 492-497, 2009.