Noname manuscript No.
(will be inserted by the editor)

Fusing $\mathcal{R}$ Features and Local Features with Context-aware Kernels for Action Recognition

Chunfeng Yuan · Baoxin Wu · Xi Li · Weiming Hu · Stephen Maybank · Fangshi Wang

Received: date / Accepted: date

Abstract The performance of action recognition in video sequences depends significantly on the representation of actions and the similarity measurement between the representations. In this paper, we combine two kinds of features extracted from the spatio-temporal interest points with context-aware kernels for action recognition. For the action representation, local cuboid features extracted around interest points are very popular using a Bag of Visual Words (BOVW) model. Such representations, however, ignore potentially valuable information about the global spatio-temporal distribution of interest points. We propose a new global feature to capture the detailed geometrical distribution of interest points. It is calculated by using the 3D $\mathcal{R}$ transform which is defined as an extended 3D discrete Radon transform, followed by the application of a two-directional two-dimensional principal component analysis. For the similarity measurement, we model a video set as an optimized probabilistic hypergraph and propose a context-aware kernel to measure high order relationships among videos. The context-aware kernel is more robust to the noise and outliers in the data than the traditional context-free kernel which just considers the pairwise relationships between videos. The hyperedges of the hypergraph are constructed based on a learnt Mahalanobis distance metric. Any disturbing information from other classes is excluded from each hyperedge. Finally, a multiple kernel learning algorithm is designed by integrating the l_2 norm regularization into a linear SVM classifier to fuse the $\mathcal{R}$ feature and the BOVW representation for action recognition. Experimental results on several datasets demonstrate the effectiveness of the proposed approach for action recognition.

C. Yuan, B. Wu and W. Hu
National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
E-mail: cfyuan, bxwu, wmhu@nlpr.ia.ac.cn

X. Li
College of Computer Science and Technology, Zhejiang University, Hangzhou, China
E-mail: xilizju@zju.edu.cn

S. Maybank
Department of Computer Science and Information Systems, Birkbeck College, London, UK
E-mail: sjmaybank@dcs.bbk.ac.uk

F. Wang
School of Software Engineering, Beijing Jiaotong University, Beijing, China
E-mail: fshwang@bjtu.edu.cn

Keywords Action recognition · spatio-temporal interest points · 3D $\mathcal{R}$ transform · hypergraph · context-aware kernel

1 Introduction

Recognition of human actions [29] [8] [60] in videos has many potential applications, such as smart surveillance, human-computer interface, video indexing and browsing, automatic analysis of sports events, and virtual reality. However, it is a challenging task because of geometric variations between intra-class objects or actions, as well as changes in scale, rotation, viewpoint, illumination, and occlusion. Many modern action recognition approaches are based on spatio-temporal interest points via the Bag of Visual Words (BOVW) model [17] [42] [12]. The local interest point features are more robust to noise, occlusion, and geometric variations than global (or large-scale) features. However, most BOVW based representations utilize the local appearance or motion information (e.g. HOG, HOF [16]) of spatio-temporal volumes (e.g. cuboids around interest points), without considering the locations of interest points. In other words, they rely mainly on the discriminative power of individual local cuboid descriptors, whilst ignoring potentially valuable information about the global spatio-temporal distribution of interest points.

In this paper, we propose a novel method to extract a global feature from the locations of interest points for a video sequence. We focus on the geometrical distribution of interest points in 3D space and characterize them from the perspective of geometry. The 2D $\mathcal{R}$ transform [38], as an improved representation of the 2D Radon transform, has been shown to provide effective feature representations of human shapes and silhouettes in images [46]. The 3D discrete Radon transform [35] [7] has been successfully applied to object classification in 3D models. However, the 3D $\mathcal{R}$ transform is little utilized. We deduce the form and properties of the 3D $\mathcal{R}$ transform, based on the 3D discrete Radon transform, and apply the 3D $\mathcal{R}$ transform to the representation of spatio-temporal interest points for the task of action recognition [53]. Afterwards, we apply $(2D)^2$PCA [57] to the $\mathcal{R}$ transform, to reduce the dimension of the obtained feature. The obtained $\mathcal{R}$ feature has several unique advantages: (1) 3D $\mathcal{R}$ transform captures the specific global distribution of spatio-temporal interest points; (2) It is invariant to geometry transformations and robust to noise, both of which are required for the effective action representation; (3) It is easy to compute and avoids the non-trivial task of foreground object segmentation and the problems involved in the BOVW method such as selecting the optimal spatio-temporal descriptor, clustering for constructing a codebook, and selecting codebook size.

The $\mathcal{R}$ feature captures the global geometrical distribution information, while the BOVW based representation utilizes the discriminative power of individual local features. They naturally complement each other. Therefore, the most important issue in the subsequent steps is how to combine these two features to enhance the classification performance. In general, most existing classification methods only utilize the pairwise similarities between videos. The pairwise similarities based kernel is a context-free kernel which is only based on the individual videos themselves. If the representation of a video is corrupted, the pairwise similarities associated with it may change significantly. In order to alleviate this problem, we introduce a new type of kernel, referred to as a “context-aware” kernel, which measures the neighborhood coherence of videos and models the high order relationships among videos. Specifically, we build an optimized probability hypergraph to capture the context information. For hyperedge construction, we adopt a distance metric learning method in which a centroid and its k nearest neighbors belong to the same label. The vertices in each resulting

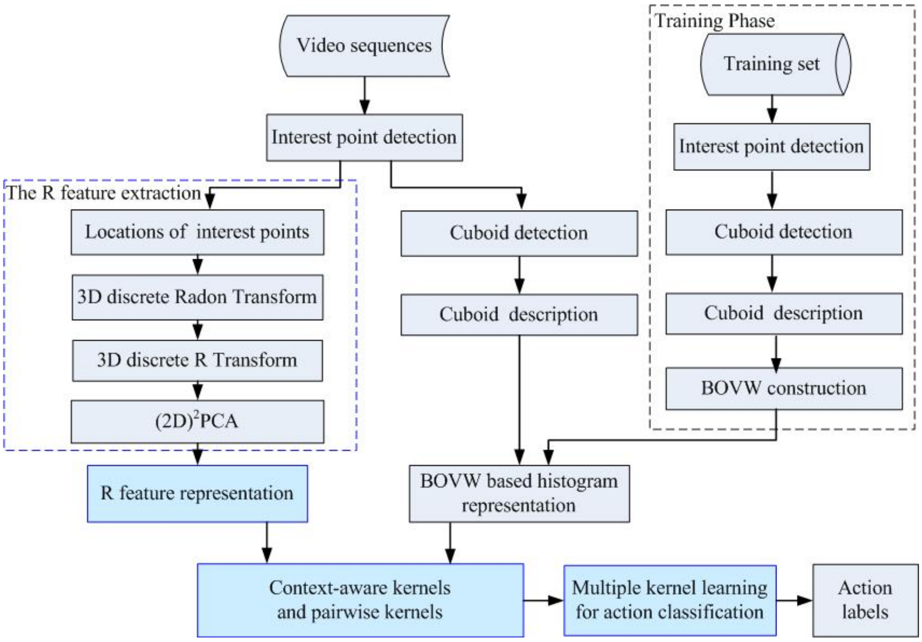


Fig. 1: The flowchart of the proposed approach for action recognition.

hyperedge have the same action class. Hence, the context of a video captured by the optimized probability hypergraph is not mixed with the videos from other classes and can assist in improving the classification performance. The combination of the proposed context-aware kernel and the pairwise kernel produces a more accurate method of measuring the similarities between videos. Finally, we introduce a multiple kernel learning (MKL) algorithm to automatically determine the optimal combination of pairwise kernels and context-aware k-ernels both based on two features. The MKL algorithm integrates the l_2 norm regularization into a linear SVM classifier for action recognition. Since the MKL objective function is non-smooth, we propose a new efficient optimization algorithm to minimize it. To sum up, the proposed approach fuses different features, and also captures the underlying contextual information from videos. Fig. 1 shows the flowchart of the proposed approach for action recognition.

The contributions of this paper are summarized below.

- A $\mathcal{R}$ feature based on the 3D $\mathcal{R}$ transform is proposed to capture the specific global spatio-temporal distribution of interest points. It is complementary to the conventional BOVW action representation method. The 3D $\mathcal{R}$ transform has some invariance to geometrical transformations and robustness to noise.
- A context-aware kernel is developed via an optimized probability hypergraph to measure the high order relationships among videos and enhance the classification performance. Moreover, the context-aware kernel is less sensitive to noise than the traditional pairwise similarity kernel.
- We introduce the max-margin metric learning algorithm to hyperedge construction and generate an optimized probability hypergraph, where the vertices in each hyperedge

have identical labels. In this way, hyperedges stay "pure" without disturbance from other class videos. The resulting context-aware kernel is more suitable for classification.

- The multiple kernel learning algorithm is employed for action recognition. It can efficiently incorporate the pairwise kernels and context-aware kernels of multiple features to improve the recognition performance.

The remainder of the paper is organized as follows. Section 2 gives a review of the related work. Section 3 introduces the proposed $\mathcal{R}$ feature. Section 4 describes the proposed optimized probability hypergraph based-context-aware kernel and the multiple kernel learning algorithm for action classification. Section 5 reports experimental results on five human action datasets. Section 6 concludes the paper.

2 Related Work

Action recognition includes two essential steps: action representation and classification based on the representations. To address this task, many papers propose an effective feature or combine multiple features to improve the representation ability. Other papers harness discriminative classification methods to accurately distinguish videos with different action classes. In the following three sections we provide a brief review on three aspects of this research that are most related to our work.

2.1 Geometrical Construction of Local Interest Points for Action Representation

Several algorithms have been proposed to integrate geometrical information into the BOVW model. A common way is to use multi-scale pyramids [18] or spatio-temporal grids [16] [6] to produce a coarse description of the feature layout. These algorithms uniformly divide the 3D space into spatio-temporal grids and then compute the histogram of local features in each grid. Ni *et al.* [25] propose an adaptive motion feature pooling scheme that utilizes human poses as side information.

Co-occurrence patterns encoding partial geometrical information are more discriminative than individual features, and have been applied in action recognition. Savarese *et al.* [31] propose spatial-temporal correlograms to model the temporal co-occurrence patterns of spatial-temporal features for action recognition. Yuan *et al.* [54][55] discover meaningful spatially co-occurrence patterns of visual words using frequent itemset mining (FIM). Such visual collocation patterns have better representation power and are less ambiguous than individual visual features. Moreover, they are statistically more meaningful and effective than the frequent itemsets. These patterns belong to one type of co-occurrence pattern, namely the conjunction form (AND). In [56][47], two types of co-occurrence patterns, the conjunction (AND) and disjunction (OR) of binary features, are discussed. In [56], a two-stage frequent pattern mining method is proposed to discover the optimal co-occurrence patterns with minimum empirical error from a noisy training dataset. In [47], a novel branch-and-bound search-based mining method is proposed to mine both AND/OR co-occurrence patterns at arbitrary orders directly in one stage, and it is order of magnitude faster than previous two-stage mining algorithm.

There are some approaches that build a video representation directly from the positions of interest points. Sun *et al.* [36] extract trajectories of interest points based on pairwise SIFT matching over consecutive frames. Three types of features are obtained from the trajectories. Wang *et al.* [41] introduce a video representation based on dense trajectories (tracklets). These tracklets describe most of short-term motion contained in a video. Gaidon *et al.* [9] further propose a spectral divisive clustering algorithm to decompose the motion content of a video into a hierarchy of data-driven parts by using hierarchical clustering on the set of tracklets. These dense trajectories based methods achieve exciting performance on many complex datasets. Trajectories are extracted depending on dense sampling which guarantees a good coverage of foreground motion as well as of the surrounding context but with a large increase in the computation complexity.

Recently, Bregonzio *et al.* [3] 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 clouds and object areas. However, computing these features involves some non-trivial steps such as reliable object detection and segmentation. Subsequently, these features are quantified by the BOVW model, and then each video is represented as a histogram. The $\mathcal{R}$ feature proposed below is also extracted from the cloud of interest points but the extraction method is completely different. It utilizes the location information of interest points and does not require object detection. It is a global feature without needing to map into a visual word based on the BOVW model like the cloud feature in [3].

2.2 Multiple Feature Combination for Action Representation

Usually, different features emphasize different aspects of actions and are suitable for different datasets. It is natural to fuse them to improve the performance. A large number of feature fusion approaches have recently appeared in the literature for action recognition.

In [22], two types of features, a quantized vocabulary of local cuboid features and a quantized vocabulary of spin-images, are used. A fusion framework is developed, based on the concept of Fiedler embedding. Different entities are embedded into a common Euclidian space, thus enabling the use of simple Euclidian distances for discovering relationships between features. Wang *et al.* [43] combine two types of features for better action representation, namely the quantized vocabulary of cuboid features and the quantized vocabulary of silhouette projection (SP) histograms. The feature fusion is realized in a simple way by directly concatenating the cuboid features and SP features together. Finally, they apply the GP classification model for action recognition. In [37], four features are extracted, including two kinds of local descriptors, namely 2D and 3D SIFT descriptors, both based on 2D interest points, and two kinds of holistic features, namely Zernike moments from single frame and a motion energy image. The basic idea of feature fusion is to concatenate all the feature vectors produced by different approaches to form a larger feature vector as the input to a classifier such as Support Vector Machine (SVM). Mikolajczyk and Uemura [24] use local descriptors and optical flow information to form a vocabulary forest of local motion-appearance features to recognize actions. Wang and Yuan [44] use two features for the action representation, namely histograms of oriented gradients (HOG) and motion boundary histograms (MBH) extracted from dense appearance trajectories. They perform spectral clustering in different feature types separately and capture co-occurrence patterns across multiple feature types. Under a novel transductive learning formulation, the spectral clustering results in different feature types and the formed co-occurrence patterns influence each other to improve the clustering or classification performance for visual recognition.

It is observed that incorporating multiple features is desirable to enhance action recognition. Moreover, all the above mentioned approaches extract cuboid descriptors as local features and derivatives from single frames as holistic features. The derivatives have no direct relation with local cuboid features. In this paper, we combine the local cuboid feature with a new kind of holistic feature which summarises information from the whole video sequence and is based on the spatio-temporal interest points.

2.3 Classification Methods for Action Recognition

During the recognition phase, many methods are adopted including probabilistic graphical models (such as hidden Markov models [34]), dynamic Bayesian networks [15], boosting [47], the Support Vector Machine (SVM), the Nearest Neighbor Classifier (NNC), and latent topic models [26] etc.. Among them, the SVM method is most popular by virtue of its effectiveness for recognition and its simplicity in use. The success of SVM depends on the design of appropriate similarity kernels. In general, most existing SVM methods exert pairwise similarity kernels. This type of kernel is actually based on a graph, in which vertices are videos and edge weights indicate the similarities between two videos. Hence only the pairwise relationship between two samples is considered and any higher order relationships are ignored. Hypergraph learning [52] addresses this problem, because a set of vertices is connected by a hyperedge. When a dataset is represented by a hypergraph, hyperedges can be viewed as the contexts of samples, and the samples with their contexts are used to generate the similarity kernel. This kernel differs from the pairwise similarity kernel and is called a context-aware kernel.

Hypergraph learning has achieved promising performance in many applications. For example, Liang *et al.* [21] introduce a content-sensitive hypergraph to represent the image and then utilize an incremental hypergraph partitioning to generate candidate regions for the final salient object detection. Zhang *et al.* [58] construct a feature correlation hypergraph to model high order relations among multimodal features for object recognition in images. Li *et al.* [20] construct three types of hypergraphs: the pairwise hypergraph, the k nearest neighbor (kNN) hypergraph, and the high order over-clustering hypergraph. They further design a discriminative hypergraph partitioning criterion for face recognition and handwritten digit recogniton. Huang *et al.* [10] propose a probabilistic hypergraph ranking for image retrieval, and Yu *et al.* [52] propose an adaptive hypergraph learning for image classification. Hong *et al.* [11] propose a novel classification method based on structured SVM and hypergraph regularization (Hyper-SSVM) for action recognition in motion capture data. A trade-off parameter is used to balance the structured relevance and the regularization item. Our method also leverages the hypergraph for action recognition, but the usage of the hypergraph is totally different from that in [11] . Hong *et al.* [11] introduce a hypergraph regularization item into the optimization objective of a structured SVM framework, where the correlations of similar samples in the training process are taken into consideration by the hypergraph regularization. However, in our method we model a video set as an optimized probability hypergraph and propose a context-aware kernel based on the hypergraph to directly measure high order relationships among videos.

Existing hypergraphs are usually constructed based on the k nearest neighbor method. Each image is taken as a “centroid” vertex and a hyperedge is formed by a centroid and its k nearest neighbors [20]. Huang *et al.* [10] assign each vertex to a hyperedge in a probabilistic way and propose a probabilistic hypergraph. Instead of generating a hyperedge for each vertex, Yu *et al.* [52] generate a group of hyperedges by varying the neighborhood size k in a specified range. These hypergraphs model the high order grouping information by using a hyperedge to link multiple samples. However, the labels of the samples are ignored in hypergraph construction and hence the resulting hyperedge may include samples from multiple classes. This mixing of the information from different classes is likely to reduce the classification performance. Our paper improves the hypergraph construction algorithm by considering label information and forming 'pure' hyperedges, such that action recognition performance is improved.

3 $\mathcal{R}$ Features and Local Features Based on Spatio-Temporal Interest Points

Spatio-temporal interest point methods have been extensively studied in the field of video representation. Among all the methods, the histogram representation of the local cuboid features around interest points is widely used for its simplicity and efficiency. However, the histogram representation only utilizes the local appearance/motion information of interest points and ignores their location information. Therefore, we propose a global $\mathcal{R}$ feature based on the interest point location information to complement the local motion feature for joint video representation. In the following, we first briefly introduce the histogram representation based on the BOVW model. Then we describe the proposed $\mathcal{R}$ feature and its properties.

3.1 Histogram Representation Based on Spatio-temporal Interest Points

We first perform the spatio-temporal interest point detection for a given video using the Harris3D detector [17]. Afterwards, we employ the HOG/HOF feature [42] to describe the cuboid extracted around each interest point. So, a video V is denoted as $\{(\hat{\mathbf{x}}_i, \boldsymbol{\alpha}_i)\}$, $1 \le i \le N$, where $\hat{\mathbf{x}}_i$ is the spatio-temporal position vector of the i^{th} detected interest point, $\boldsymbol{\alpha}_i$ is the HOG/HOF feature, and N is the total number of interest points detected in the video.

For the BOVW based representation, local HOG/HOF features $\{\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_m\}$ from a training set are quantized to form a codebook (i.e. BOVW) by using the k-means clustering method. Each HOG/HOF feature is mapped into a visual word in the BOVW. Then, each video is represented as a histogram with regard to all visual words, formulated as:

$$H = (n_1, n_2, \cdots, n_d), \tag{1}$$

where n_i denotes the occurrence frequency of the i^{th} visual word in this video, and d is the number of visual words.

3.2 The $\mathcal{R}$ Feature Based on Spatio-Temporal Interest Points

The proposed $\mathcal{R}$ feature is based on the 3D $\mathcal{R}$ transform of the video sequence. The 3D $\mathcal{R}$ transform is an extension of the 3D Radon transform. Thus, we first perform the 3D Radon transform on the video sequence with detected interest points. The minimal spatio-temporal window containing all the interest points extracted from a video is regarded as a 3D model. Let $\mathbf{M}$ represent this 3D model and $f(\mathbf{x})$ be the binary function defined on the 3D model, where $\mathbf{x} = (x, y, t)$ denotes the position of a point in $\mathbf{M}$. The binary function $f(\mathbf{x})$ is defined as:

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is an interest point, namely } \mathbf{x} \in \{\hat{\mathbf{x}}_i, 1 \le i \le N\} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Namely, the 3D model $\mathbf{M}$ of a video sequence is a discrete 3D array in spatio-temporal space and the function $f(\mathbf{x})$ is the index variable of $\mathbf{M}$.

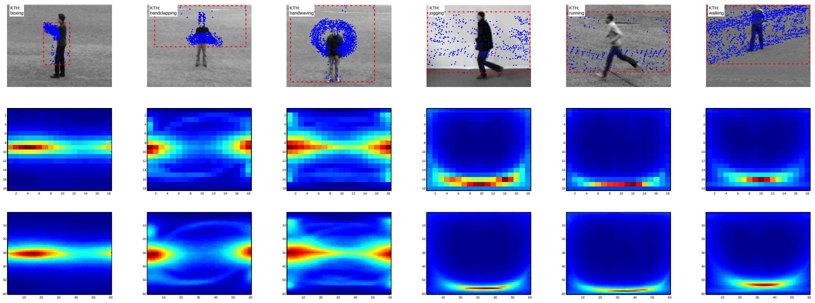


Fig. 2: The 3D $\mathcal{R}$ transform of six videos belonging to different action classes in the KTH dataset. From the first row to last row: interest points in blue, $\mathcal{R}$ transform by sampling (θ, ϕ) with an internal 10, and $\mathcal{R}$ transform by sampling (θ, ϕ) with an internal 2.

The 3D discrete Radon transform is defined by summing the interpolated samples of a discrete 3D array lying on planes which satisfy certain constraints [35]. Let $\{\mathbf{x}_j\}_{j=1}^{J}$ be all the 3D points in the array $\mathbf{M}$. The 3D discrete Radon transform of the $f(\mathbf{x})$ is defined by [7]:

$$T_f(\boldsymbol{\eta}, \rho) = \sum_{j=1}^{J} f(\mathbf{x}_j)\delta(\mathbf{x}_j^T\boldsymbol{\eta} - \rho) = \sum_{i=1}^{N} \delta(\hat{\mathbf{x}}_i^T\boldsymbol{\eta} - \rho) \;, \tag{3}$$

where J is the total number of points in $\mathbf{M}$, $\boldsymbol{\eta}$ is a unit vector in 3D space, ρ is a real number, and $\delta(\cdot)$ is the Dirac delta function. The unit vector $\boldsymbol{\eta}$ using spherical coordinates is written as: $\boldsymbol{\eta} = [\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta]$, where θ is the polar angle measured from a fixed zenith direction and ϕ is the azimuthal angle in the xy-plane from the x-axis. Thus, equation (3) is rewritten in a spherical coordinate system as:

$$T_f(\rho, \theta, \phi) = \sum_{j=1}^{J} f(x_j, y_j, t_j) \cdot \delta(x_j \cos\phi \sin\theta + y_j \sin\phi \sin\theta + t_j \cos\theta - \rho) \;. \tag{4}$$

where ρ is the distance from a plane to the origin and this plane is normal to the direction $\boldsymbol{\eta}$.

The 3D Radon transform can be easily calculated, but it is not invariant to translation, scaling and rotation [38]. To overcome this problem, we define the $\mathcal{R}$ Transform of the 3D Radon transform, inspired by the $\mathcal{R}$ Transform of the 2D Radon transform proposed in [38]. The 2D $\mathcal{R}$ transform is defined as the integral of the square of the 2D Radon transform over the parameter ρ. Therefore, we define the 3D $\mathcal{R}$ transform as follows:

$$\mathcal{R}_f(\theta, \phi) = \int_{-\infty}^{\infty} T_f^2(\rho, \theta, \phi) d\rho \;. \tag{5}$$

Observed from equations (4) and (5), each interest point is first projected into all planes with parameters (ρ, θ, ϕ) and then the $\mathcal{R}$ transform is obtained by the integral of the square of projections over ρ. Therefore, the 3D $\mathcal{R}$ transform efficiently describes the geometrical distribution of interest points. Afterwards, we perform scale normalization to the $\mathcal{R}$ transform by the following equation:

$$\mathcal{R}'_f(\theta, \phi) = \frac{\mathcal{R}_f(\theta, \phi)}{\max_{\theta,\phi}\{\mathcal{R}_f(\theta, \phi)\}} \;. \tag{6}$$

The minimum value of the $\mathcal{R}$ transform $\mathcal{R}_f(\theta, \phi)$ is non-negative. By the normalization defined in (6), the values of the scaled version $\mathcal{R}'_f(\theta, \phi)$ are in the range of 0 to 1. The scale normalization makes $\mathcal{R}_f(\theta, \phi)$ achieve the robustness to scaling variation. In our application to human action recognition, several videos may differ widely in the numbers of the detected interest points, because the length of the video and the action intensity in videos are both different. According to (3), the different numbers of interest points cause scaling variation of $\mathcal{R}_f(\theta, \phi)$. Therefore, we employ the normalization defined in (6) to reduce the influence caused by the different numbers of interest points.

The $\mathcal{R}$ transform $\mathcal{R}_f(\theta, \phi)$ represents the distribution of interest points. By sampling two parameters θ and ϕ, $\mathcal{R}_f(\theta, \phi)$ yields a 2D matrix. Fig. 2 shows the 3D $\mathcal{R}$ transform of six videos belonging to different action classes in the KTH dataset. In the first row, all the interest points detected in a video are superposed on a single frame. It can be seen that the geometrical distribution of interest points varies according to different action classes and is very helpful for improving the action recognition accuracy. The second and third rows respectively exhibit the 3D $\mathcal{R}$ transform obtained by sampling (θ, ϕ) in the range of 1 to 180 degrees, firstly in intervals of 10 degrees and secondly in intervals of 2 degrees. The more samples of θ and ϕ, the more detailed the characterization of the interest points' distribution, but the larger the size of the matrix.

In order to reduce the dimension and improve the robustness of the $\mathcal{R}$ feature, we apply the 2-Directional 2-Dimensional PCA, i.e. $(2D)^2$PCA, to the matrix obtained from the $\mathcal{R}$ transform. The $(2D)^2$PCA, introduced in [57], simultaneously calculates 2DPCA in the row and column directions and obtains higher recognition accuracy than PCA and two-dimensional PCA (2DPCA) [50]. Let $R \in \mathbb{R}^{m \times n}$ be the matrix obtained from the $\mathcal{R}$ transform for a video sequence, let $\{R_k, k = 1, 2, \cdots, M\}$ be the corresponding matrices of all the M video sequences in a video set, and let $\overline{R} = (1/M) \sum_{k=1}^{M} R_k$ be the mean matrix of the set. Then the covariance matrix S of the video set and its transposed matrix S' are evaluated by:

$$
\begin{aligned}
S &= \frac{1}{M} \sum_{k=1}^{M} (R_k - \overline{R})^T (R_k - \overline{R}) \ , \\
S' &= \frac{1}{M} \sum_{k=1}^{M} (R_k - \overline{R})(R_k - \overline{R})^T \ .
\end{aligned}
\tag{7}
$$

It has been proven that the optimal projection matrix P is constructed from the orthonormal eigenvectors of S corresponding to the p largest eigenvalues. The optimal projection matrix Q is obtained by computing the eigenvectors of S' corresponding to the q largest eigenvalues. After projecting each matrix $\{R_k, k = 1, 2, \cdots, M\}$ onto P and Q, we obtain the low-dimensional matrix $G_k \in \mathbb{R}^{q \times p}$ corresponding to R_k as follows.

$$
G_k = Q^T R_k P \ . \tag{8}
$$

Finally, we use the resulting low-dimensional matrix G_k as the final feature.

The $\mathcal{R}$ feature is different from other features derived from local interest points in that it represents the whole video by encoding the detailed geometrical distribution of the interest points in space-time. It avoids the non-trivial problems of selecting the optimal spatio-temporal descriptor, clustering to obtain a codebook, and selecting the codebook size that are faced by previous interest point based methods. Moreover, the computational cost is not high and it is even much lower than that of the interest point based video-level histogram feature. Specifically, the implementation includes embedding all interest points into a series of 3D planes to obtain the Radon transform, and then summing the square of the Radon transform over the parameter ρ. There are $m \times n$ planes, where m and n are the sampling numbers of parameters θ and ϕ respectively. Let l denote the variation range of parameter ρ, and l is less than or equal to N, the number of interest points in a video sequence. In experiments, m and n are set to 18, and N is about 1000. The computational complexity of the R feature is $O(Nmnl)$, which in this case is about $O(10^8)$ arithmetical operations.

3.3 The Properties of the $\mathcal{R}$ Feature

In this subsection, we derive the following properties of the proposed new $\mathcal{R}$ Transform.

For a scale factor α, we have

$$\frac{1}{\alpha^2}\int_{-\infty}^{\infty} T_f^2(\alpha\rho,\theta,\phi)d\rho = \frac{1}{\alpha^3}\int_{-\infty}^{\infty} T_f^2(\nu,\theta,\phi)d\nu = \frac{1}{\alpha^3}\mathcal{R}_f(\theta,\phi)\ . \tag{9}$$

For a spatio-temporal translation by (x_0, y_0, t_0), we have

$$\int_{-\infty}^{\infty} T_f^2(\rho - x_0\cos\phi\sin\theta - y_0\sin\phi\sin\theta - t_0\cos\theta,\theta,\phi)d\rho$$
$$= \int_{-\infty}^{\infty} T_f^2(\nu,\theta,\phi)d\nu = \mathcal{R}_f(\theta,\phi). \tag{10}$$

For the rotation with angles (θ_0, ϕ_0), we have

$$\int_{-\infty}^{\infty} T_f^2(\rho,\theta+\theta_0,\phi+\phi_0)d\rho = \mathcal{R}_f(\theta+\theta_0,\phi+\phi_0)\ . \tag{11}$$

From (9), the amplitude of the $\mathcal{R}$ transform changes from $1/\alpha^2$ to $1/\alpha^3$ for a scale factor α. From (11), the R transform is changed from $\mathcal{R}_f(\theta,\phi)$ to $\mathcal{R}_f(\theta+\theta_0,\phi+\phi_0)$ for the rotation with angles (θ_0,ϕ_0). This means that the phase of $\mathcal{R}_f$ changes by (θ_0,ϕ_0), and the amplitude of $\mathcal{R}_f$ does not change. Therefore, from equations (9)-(11) we can see: first, $\mathcal{R}$ transform is invariant to translation; second, scaling leads to amplitude scaling; and third, rotation results in phase shift. These properties make the $\mathcal{R}$ transform useful for representing the distribution of interest points for action recognition.

As the above analyzed, the obtained $\mathcal{R}$ feature is invariant to some geometric transformations. Moreover, it is tolerant to noise in interest points because the $\mathcal{R}$ transform is robust to small perturbations of parts of the shape[46]. Besides, the $(2D)^2$PCA has a certain ability to reduce noise. Fig. 3 shows some examples to demonstrate the robustness of our $\mathcal{R}$ feature. In the first row, two videos about "boxing" are performed in different environments and by two persons in different clothing, but their $\mathcal{R}$ features are similar. It demonstrates the robustness of the $\mathcal{R}$ feature. In the second row, the first video "handclapping 1" is shot with a fixed camera and the second video "handclapping 2" is shot with camera zooming in and out. The scales of video "handclapping 2" change continually. However, the $\mathcal{R}$ features of these two videos are almost same except for a small difference in amplitude scaling. In the last row, the person jogs horizontally from the left to the right in the first video "jogging 1", and the person jogs in a line from the bottom right to the up left in the second video "jogging 2". The distributions of interest points undergo rotation and a change in scale. Their $\mathcal{R}$ features differ by a small change in amplitude and by a small shift in the vertical direction. In all, although the videos for a given class are very different, their $\mathcal{R}$ features can be easily discriminated from those of other classes. It can be seen that our $\mathcal{R}$ feature is robust to geometric transformations and appearance variations.

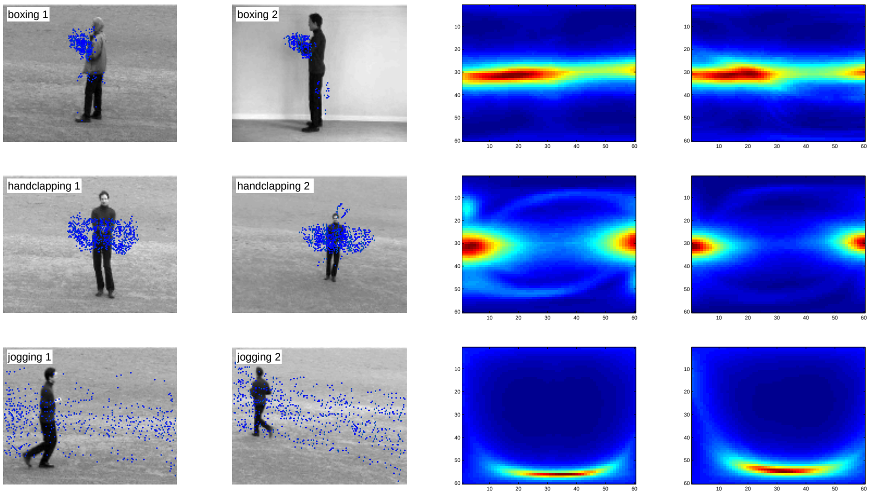


Fig. 3: The robustness of the $\mathcal{R}$ features for six videos belonging to three action classes in the KTH dataset. The $\mathcal{R}$ features in the third column respectively correspond to the videos in the first column, and The $\mathcal{R}$ features in the fourth column respectively correspond to the videos in the second column.

4 Optimized Probability Hypergraph based Context-aware Kernels for Action Recognition

Given the two representations of each video, our aim is to obtain the labels of test videos from the labeled training videos. The crucial step in this process is to compute the similarities between videos, and build kernel matrices based on the similarities. In most approaches, the kernel matrix is computed based on the pairwise comparisons of videos, but such kernels are sensitive to noise, outliers, and so on. In this section, we propose two context-aware kernels to model the high-order relationship among the videos.

We first propose a k-nearest neighbor (kNN) based probability hypergraph for context-aware kernel construction. This probability hypergraph is built in an unsupervised way, in which videos from different action classes may be put into the same hyperedge. As a result, the kernel obtained from this hypergraph may involve some disturbance and not classify video reliably. This may tend to increase the probability of classification errors. In order to solve this problem, we propose an optimized probability hypergraph based on distance metric learning. Finally, the pairwise kernels and context-aware kernels, both computed from all features, are combined together by multiple kernel learning for action classification.

4.1 The k-nearest Neighbor Based Probabilistic Hypergraph for Context-aware Kernel Construction

Assume that $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ is a finite set of videos, and A is an $n \times n$ affinity matrix over $\mathcal{V}$ where $A(i, j) \in [0, 1]$ measures the similarity between videos v_i and v_j. We define A using a Gaussian kernel over the Euclidean distances of videos in the feature space, i.e.,

$A(i, j) = exp(-\|v_i - v_j\|^2/\sigma^2)$, where σ is the average distance among all vertices. In most action recognition methods, the affinity matrix A is directly used as the kernel and input into the SVM classifier. However, A considers only the pairwise relationships between two videos, and it ignores the higher order relationships. If the representation of a video is inaccurate and corrupted, the pairwise similarities may change significantly. Modeling the high order relationships among videos alleviates this problem and improves the classification performance.

In order to explore the high order relationships among videos, we employ a hypergraph construction mechanism. We regard each video in the dataset as a vertex and construct a probabilistic hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{w})$ with the vertex set $\mathcal{V}$, the hyperedge set $\mathcal{E}$, and the hyperedge weight vector $\mathbf{w}$. The hyperedge set $\mathcal{E}$ is a family of subsets of $\mathcal{V}$ such that $\bigcup_{e \in \mathcal{E}} = \mathcal{V}$. Each hyperedge e_i is assigned a positive weight $\mathbf{w}(e_i)$, which is computed as follows:

$$\mathbf{w}(e_i) = \sum_{v_j \in e_i} A(i, j) . \tag{12}$$

In our method, a hyperedge connects a centroid vertex and its k nearest neighbors, and a hyperedge is generated for each vertex. There are n hyperedges, and each hyperedge connects $k + 1$ vertices. Therefore, the hypergraph $\mathcal{G}$ can be represented by a $| \mathcal{V} | \times | \mathcal{E} |$ incidence matrix H:

$$h(v_i, e_j) = \begin{cases} A(j, i) & \text{if } v_i \in e_j , \\ 0 & \text{otherwise} . \end{cases} \tag{13}$$

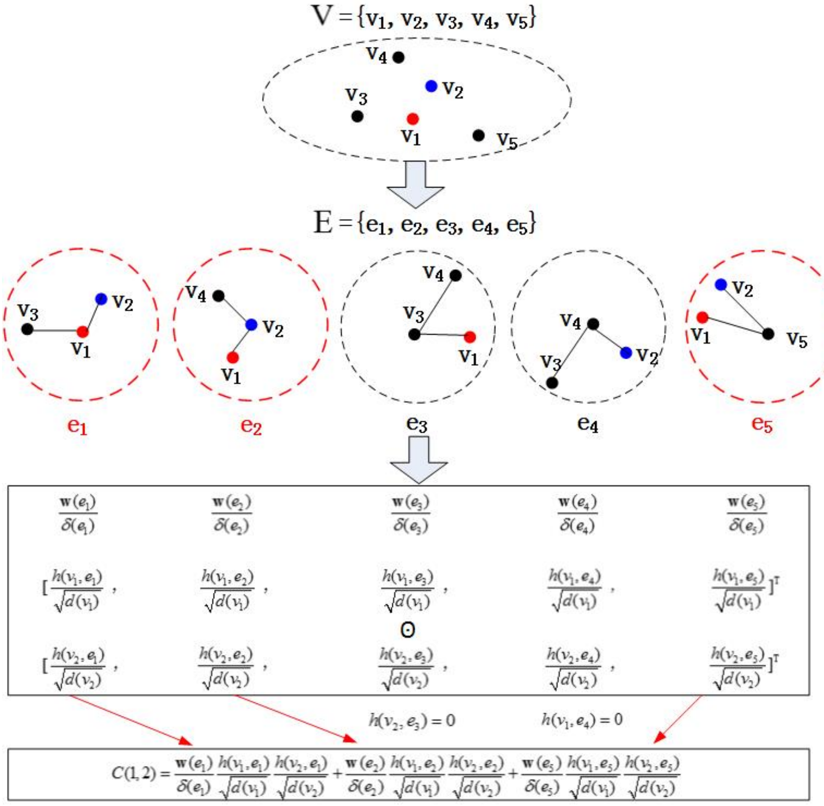
The incidence matrix describes the vertex-to-hyperedge relationship and determines the hypergraph structure. According to the formulation above, a vertex v_i is “softly” assigned to e_j based on the similarity $A(j, i)$ between vertices v_i and v_j, where v_j is the centroid vertex of e_j. In this way, $\mathcal{G}$ is a probabilistic hypergraph [10], which describes not only the higher order grouping information but also the local relationship between vertices within each hyperedge. Based on h, the degree of a vertex $v \in \mathcal{V}$ is defined by $d(v) = \sum_{e \in \mathcal{E}} \mathbf{w}(e) h(v, e)$. The degree of a hyperedge $e \in \mathcal{E}$ is defined as $\delta(e) = \sum_{v \in \mathcal{V}} h(v, e)$. We use D_v, D_e, and W to denote the diagonal matrices containing the vertex degrees, the hyperedge degrees, and the hyperedge weights respectively.

Based on these definitions, the similarity $C(i, j)$ between videos v_i and v_j in a kNN based probabilistic hypergraph is defined as:

$$C(i, j) = \sum_{t=1}^{n} \frac{\mathbf{w}(e_t)}{\delta(e_t)} \frac{h(v_i, e_t)}{\sqrt{d(v_i)}} \frac{h(v_j, e_t)}{\sqrt{d(v_j)}} . \tag{14}$$

The first term $\frac{\mathbf{w}(e_t)}{\delta(e_t)}$ measures the average weight of a hyperedge. The second term $\frac{h(v_i, e_t)}{\sqrt{d(v_i)}} = \frac{h(v_i, e_t)}{\sqrt{\sum_{e \in \mathcal{E}} \mathbf{w}(e) h(v_i, e)}}$ characterizes the normalized vertex-to-hyperedge membership between v_i and e_t. Thus, the hypergraph similarity $C(i, j)$ can be interpreted as the inner product of two vertex-to-hyperedge vectors $[\frac{h(v_i, e_1)}{\sqrt{d(v_i)}}, \cdots, \frac{h(v_i, e_n)}{\sqrt{d(v_i)}}]^T$ and $[\frac{h(v_j, e_1)}{\sqrt{d(v_j)}}, \cdots, \frac{h(v_j, e_n)}{\sqrt{d(v_j)}}]^T$. The corresponding matrix form of equation (14) is: $C = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$. Our hypergraph similarity matrix C is closely related to the hypergraph Laplacian matrix [59]. In [59], the hypergraph Laplacian matrix is defined as $\Delta = I - \Theta$, where I denotes the identity matrix and Θ is exactly the same as our hypergraph similarity matrix C.

Fig. 4 illustrats a simplified example of the context-aware kernel construction based on the probability hypergraph. Therefore, for a given video dataset we obtain a context-aware hypergraph similarity matrix C, which captures higher order relationships between videos. The hyperedge constructed for each video is regarded as its context.



$$\left[\frac{h(v_1,e_1)}{\sqrt{d(v_1)}}, \frac{h(v_1,e_2)}{\sqrt{d(v_1)}}, \frac{h(v_1,e_3)}{\sqrt{d(v_1)}}, \frac{h(v_1,e_4)}{\sqrt{d(v_1)}}, \frac{h(v_1,e_5)}{\sqrt{d(v_1)}}\right]^T$$

$$C(1,2) = \frac{\mathbf{w}(e_1)}{\delta(e_1)}\frac{h(v_1,e_1)}{\sqrt{d(v_1)}}\frac{h(v_2,e_1)}{\sqrt{d(v_2)}} + \frac{\mathbf{w}(e_2)}{\delta(e_2)}\frac{h(v_1,e_2)}{\sqrt{d(v_1)}}\frac{h(v_2,e_2)}{\sqrt{d(v_2)}} + \frac{\mathbf{w}(e_5)}{\delta(e_5)}\frac{h(v_1,e_5)}{\sqrt{d(v_1)}}\frac{h(v_2,e_5)}{\sqrt{d(v_2)}}$$

Fig. 4: A simplified illustration for the context-aware kernel construction based on the probability hypergraph. Assume that the vertex set $\mathcal{V} = \{v_1, v_2, \cdots, v_5\}$. For each vertex, we construct a hyperedge containing the connections between this vertex and its 2 nearest neighbors.
The five dashed circles represent five hyperedges. Finally, as in (14), the similarity between two vertices is computed as the weighted inner product of the two corresponding vertex-to-hyperedge vectors.

4.2 Optimized Probability Hypergraph for Context-aware Kernel Construction

The crucial issue in defining a hypergraph is the hyperedge construction which determines the hypergraph structure. In the above subsection, each hyperedge is constructed by the kNN method, which is very popular in hypergraph studies [10] [52] [20]. However, since the kNN method uses simple Euclidean distances to measure the sample dissimilarities in an unsupervised way, it is often happens that some of the samples in the k nearest neighbors have different labels from the centroid sample. In this way, the information from different classes is mixed into the context of the centroid sample. This is likely to reduce the classification performance. In order to solve this problem, we train a Mahalanobis distance metric, which aims at making the k nearest neighbors belong to the same class as their centroid samples and generating “pure” hyperedges. Based on the k nearest neighbors obtained by the Mahalanobis distance metric, we construct an optimized probability hypergraph for the context-aware kernel construction. The Mahalanobis distance metric is learnt with a large margin nearest neighbor (LMNN) algorithm [49].

Assume a training video set consisting of n labeled videos $\{(\mathbf{v}_i, y_i)\}_{i=1}^n$, where $\mathbf{v}_i \in \mathbb{R}^d$ is the video action representation and $y_i \in \{1, 2, \cdots, c\}$ is the action label. Here, c is the number of classes. The Mahalanobis distance between two videos $\mathbf{v}_i$ and $\mathbf{v}_j$ is defined as

$$d_M(\mathbf{v}_i, \mathbf{v}_j) = \|L(\mathbf{v}_i - \mathbf{v}_j)\| = \sqrt{(\mathbf{v}_i - \mathbf{v}_j)^T M(\mathbf{v}_i - \mathbf{v}_j)}\,, \tag{15}$$

where $L : \mathbb{R}^d \to \mathbb{R}^d$ is a linear transformation to be learned, and the matrix $M = L^T L$ is a symmetric positive definite matrix parameterizing the Mahalanobis distance metric induced by the linear transformation L. When M is equal to the identity matrix, d_M is the Euclidean distance in the affinity matrix A defined in the above subsection. We aim at learning a linear transformation L such that under the metric (15), the k nearest neighbors of a vertex belong to the same class as the vertex, while samples from different classes are separated by a large margin. In order to learn this linear transformation, for each video we identify its k nearest neighbors within the videos with the same class as its k *target neighbors*, and its other neighboring videos with different class labels as its *impostors* based on the Euclidean metric. Specifically, for an input $\mathbf{v}_i$ and its target neighbors $\mathbf{v}_j$, its impostor $\mathbf{v}_k$ is defined as:

$$d_M(\mathbf{v}_i, \mathbf{v}_k)^2 - d_M(\mathbf{v}_i, \mathbf{v}_j)^2 = \|L(\mathbf{v}_i - \mathbf{v}_k)\|^2 - \|L(\mathbf{v}_i - \mathbf{v}_j)\|^2 \le 1. \tag{16}$$

In a word, the impostor $\mathbf{v}_k$ of an input is any differently labeled sample that invades any of its target neighbors plus unit margin. We define a set of triples $S = \{(i, j, k) : y_j = y_i, y_k \neq y_i\}$, where $\mathbf{x}_i$ is an input, $\mathbf{x}_j$ is an target neighbor of $\mathbf{x}_i$, and $\mathbf{x}_k$ is an impostor of $\mathbf{x}_i$. The LMNN algorithm learns the transform matrix L of a Mahalanobis metric which keeps each input $\mathbf{x}_i$ closer to its *target neighbors* than *impostors* by a large margin. By introducing nonnegative slack variables $\{\xi_{ijk}\}$ for all triplets $(i, j, k) \in S$, the optimization problem can be formulated as a semidefinite program (SDP) as follows:

$$\begin{aligned} & \min_M \sum_{(i,j)\in S} d_M(\mathbf{x}_i, \mathbf{x}_j)^2 + c \sum_{(i,j,k)\in S} \xi_{ijk} \\ \text{subject to:}\quad & (i, j, k) \in S \\ & (1)\ d_M(\mathbf{x}_i, \mathbf{x}_k)^2 - d_M(\mathbf{x}_i, \mathbf{x}_j)^2 \ge 1 - \xi_{ijk} \\ & (2)\ \xi_{ijk} \ge 0 \\ & (3)\ M \ge 0\,. \end{aligned} \tag{17}$$

This optimization function consists of two competing terms. The first term penalizes large distances between each input and its target neighbors in order to make k nearest neighbors with the same label be closer. The second term penalizes the impostors in order to keep impostors maintain a large margin of distance and do not threaten to “invade” their corresponding target neighbors. Moreover, this function is optimized locally and does not require that all similarly labeled samples be tightly clustered. It can be solved by standard online SDP solvers. Since impostors are sparse and few constraints are active in our case, we employ a speedup solver [49] based on sub-gradient descent in both the matrices L and M. In [49], it is shown that this speedup solver works much faster than generic solvers.

By applying the learned linear transformation L to the video dataset, we obtain a new video feature set $\{(L\mathbf{v}_i, y_i)\}_{i=1}^n$ where the k nearest neighbors of each point $L\mathbf{v}_i$ have the same action class y_i. Subsequently, we use the k nearest neighbors, computed based on the new feature set, to construct a new probabilistic hypergraph. The subsequent construction of this hypergraph is similar to that in the above subsection except that the new feature set is used. The computational complexity of this step is $O(n^2)$, where n is the number of video sequences. Finally, we obtain a new context-aware similarity matrix C_o based on this optimized probabilistic hypergraph by the analytical formula defined in (14). This formula involves matrix inversion and multiplication, where the matrix inversion is only required for diagonal matrices and the involved matrix H is very sparse. Since this context-aware similarity matrix is directly obtained by an analytical formula without iteration, its computational complexity is not high.

In the experiments, the linear transformation L is learnt using the training set. Since our aim is to solve the offline classification problem, we use both the training and test videos to build the hypergraph and the context-aware kernels. But during this processing, we just utilize their features without the action label information. Besides, we can deal with unseen test samples in an online way. At first, we use the training set to construct one hypergraph (denoted as $\mathcal{G}_{old}$). Next, when each unseen test sample arrives we build a hyperedge for this test sample using its k-nearest neighbors from the training set and then add this hyperedge into the hypergraph $\mathcal{G}_{old}$ to form a new hypergraph, $\mathcal{G}_{new}$. In $\mathcal{G}_{new}$, the hyperedges involving vertices in the training set are not updated in order to save computational time. In this paper, the video representation $\mathbf{v}_i$ is the histogram feature or the $\mathcal{R}$ feature as described in Section 3. We compute a context-aware similarity matrix C_o for each kind of feature and thus obtain two context-aware similarity matrices in total.

4.3 Multiple Kernel Learning for Action Classification

For each video sequence, we extract two features, the histogram feature based on the BOVW model and the $\mathcal{R}$ feature. Utilizing each video feature, we compute a pair-wise similarity kernel and a context-aware kernel based on the optimized probability hypergraph. Thus four kernels are obtained in total. To automatically determine the optimal combination of kernels, we introduce the Multiple Kernel Learning (MKL) algorithm [39] [13] to assign optimal weights to different kernels.

As in subsection 4.2, a set of training samples $\mathbb{S} = \{(\mathbf{v}_i, y_i)\}_{i=1}^{n}$ is given, where $\mathbf{v}_i \in \{H_i, G_i\}$ includes the histogram feature H_i computed by (1) and the $\mathcal{R}$ feature G_i computed by (8) of an input action video, and y_i is an action class label associated with $\mathbf{v}_i$. According to the definitions in subsections 4.1 and 4.2, we obtain a set of $n \times n$ base kernel matrices $\{K_1, K_2, K_3, K_4\}$ with the forms defined as follows

$$\begin{aligned}
K_1(i, j) &= A(H_i, H_j) \,, \\
K_2(i, j) &= A(G_i, G_j) \,, \\
K_3(i, j) &= C_o(H_i, H_j) \,, \\
K_4(i, j) &= C_o(G_i, G_j) \,, \qquad (18)
\end{aligned}$$

where A denotes the pairwise affinity kernel and C_o denotes the context-aware kernel based on the optimized probability hypergraph. The optimal kernel matrix K is defined as the linear combination of the base kernel matrices

$$K = \sum_{l=1}^{4} \lambda_l K_l \,, \qquad (19)$$

```
Algorithm 1: Minimax optimization of Multiple Kernel Learning
Input: a set of training samples $\mathbb{S} = \{(\mathbf{v}_i, y_i)\}_{i=1}^{n}$, and base kernel matrices $\{K_1, K_2, K_3, K_4\}$.
Initialization: Set $t \leftarrow 0$ and initialize $\Lambda^0$ randomly.
repeat
    1: Compute $K^t$ by equation (19).
    2: Solve the dual problem with a fixed kernel $K^t$ and regularization $r(\Lambda^t)$
       using an SVM solver and obtain the optimal solution $\alpha^*$.
    3: Compute $\nabla D(\Lambda^t)$ through equation (27).
    4: Use the projected gradient descent to update the base kernel weight $\Lambda^t$
       through equation (28).
    5: Update $t \leftarrow t + 1$;
until converged
output: $\alpha^*$ and $\Lambda^* = \Lambda^t$.
```

where $\Lambda = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]^T$ is a weight vector. The objective of the MKL is to learn a predictor function with the form

$$f(A) = \omega^T \phi(\mathcal{V}) + b, \quad \omega \in R^{d(\mathcal{H})}, \quad b \in R\,, \tag{20}$$

where $\phi : \mathcal{V} \to \mathcal{H}$ is a feature mapping from the original input video space $\mathcal{V}$ to a Hilbert space $\mathcal{H}$ and the feature mapping is implicitly represented by $K(\mathbf{v}_i, \mathbf{v}_j) = \phi(\mathbf{v}_i)^T \phi(\mathbf{v}_j)$. We include a regularization of base kernel weights in an SVM framework. Since the two types of features are complementary, and the pairwise kernel and high-order context-aware kernel are also complementary, the four base kernels to be combined are complementary. So there should be a smoothness constraint on the weights of the base kernels to keep their diversity. Therefore, we apply $l_2 - norm$ regularization, in the form $r(\Lambda) = \sum_{i=1}^{4} \lambda_i^2$. The primal optimization problem is

$$\min_{\omega,b,\xi,\lambda} \frac{1}{2}\omega^T \omega + C_1 \sum_{i=1}^{n} \xi_i + C_2 r(\Lambda)\,, \tag{21}$$

$$\text{subject to} \quad y_i(\omega^T \phi(\mathbf{v}_i) + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0, \;\; 1 \leq i \leq n, \tag{22}$$

$$\lambda_l > 0, \;\; 1 \leq l \leq 4\,, \tag{23}$$

where ω corresponds to the support vectors, and C_1, C_2 are two constants controlling the importance of hinge loss and regularization on kernel weights respectively. The whole optimization problem is similar to that of the standard C-SVM, except an additional regularization $r(\Lambda)$ in the objective function (21) and the additional constraints $\lambda_l > 0$ in (23). Let $\mathbf{1}$ denote a vector with all its entries equal to 1, and let Y be a diagonal matrix with action class labels on its diagonal. The corresponding dual problem is described as

$$\min_{\Lambda} D(\Lambda), \tag{24}$$

$$\text{where } D(\Lambda) = \max_{\alpha} \mathbf{1}^T \alpha - \frac{1}{2}\alpha^T Y K Y \alpha + C_2 r(\Lambda), \tag{25}$$

$$\text{subject to} \quad \mathbf{1}^T Y \alpha = 0, \;\; 0 \leq \alpha \leq C_1, \;\; \Lambda \geq 0, \tag{26}$$

where α is a Lagrange multiplier, and K is the kernel matrix defined by equation (19). We adopt the minimax optimization strategy of [5] to solve this dual problem.

The minimax algorithm proceeds in two stages to calculate α and Λ iteratively. In the first stage, Λ is fixed so that $r(\Lambda)$ and K are constants, and D is the standard SVM dual with kernel matrix K. Therefore, we can use any SVM solver to maximize D and obtain the optimal solution α^*. In the second stage, α^* is fixed and $D(\Lambda)$ is estimated by the projected gradient descent method. As proved in Lemma 2 of [5], D can be differential with respect to Λ as if α^* did not depend on Λ. The derivatives of $D(\Lambda)$ are given by

$$\frac{\partial D}{\partial \lambda_l} = 2C_2\lambda_l - \frac{1}{2}\alpha^{*T} Y K_l Y \alpha *, \quad 1 \le l \le 4. \tag{27}$$

The weights are updated and projected to a feasible set by the following equation

$$\lambda_l^{t+1} = \max(0, \lambda_l^t - s^t \frac{\partial D}{\partial \lambda_l}), \tag{28}$$

where s^t is the step size determined by the Armijo rule [1] to ensure convergence. The projection defined in equation (28) always ensures that $\Lambda \ge 0$. The two stages are repeated until convergence.

The whole minimax optimization algorithm is shown in Algorithm 1. The output of Algorithm 1 is the optimal parameters α^* and Λ^*. Given a new unlabeled video sequence $\mathbf{v} = \{H, G\}$, we obtain a video graph kernel $K^*(\mathbf{v}, \mathbf{v}_i)$ based on the learned parameter Λ^* by equations (18) and (19). Finally, the action class of this test video is determined by the sign of $\sum_{i=1}^{n} \alpha_i^* y_i K^*(\mathbf{v}, \mathbf{v}_i) + b^*$.

This MKL algorithm works fairly fast and its computational complexity is acceptable. At the first step, we employ a fast SVM solver [33] whose run time required to obtain a solution of accuracy ϵ is $\widetilde{O}(1/\epsilon)$. The second step is the projected gradient descent which is very easy to perform. Moreover, recent work focuses on developing fast solvers to train SVM and MKL for large-scale datasets, and applying these work to our method will make it faster. For example, Kloft *et al.* [13] develop two fast interleaved optimization strategies for MKL.

5 Experiments

We tested our approach on five human action datasets: the KTH dataset [32], the robustness dataset [2], the UCF sports dataset [30], the UCF films dataset [30], and the Hollywood2 dataset [23]. Representative frames from the four datasets are shown in Fig. 5. We emphasize that, unlike many other systems, our approach requires no preprocessing steps such as object segmentation and tracking.

5.1 Parameter Evaluation for the Proposed $\mathcal{R}$ Feature

There are two parameters θ and ϕ in the 3D $\mathcal{R}$ transform used to compute the proposed $\mathcal{R}$ feature. We evaluated the effect of the sampling intervals of these two parameters on the $\mathcal{R}$ feature on the KTH dataset. Moreover, we tested to see if the performance is improved by applying $(2D)^2$PCA to refine the feature obtained from the $\mathcal{R}$ transform. We performed leave-one-person-out cross-validation to make the performance evaluation on the KTH database.

The ranges of θ and ϕ are both from 0 to 180. Fig. 6 demonstrates the performance with respect to θ and ϕ with seven different sampling intervals, comprising 30, 25, 20, 15, 10,

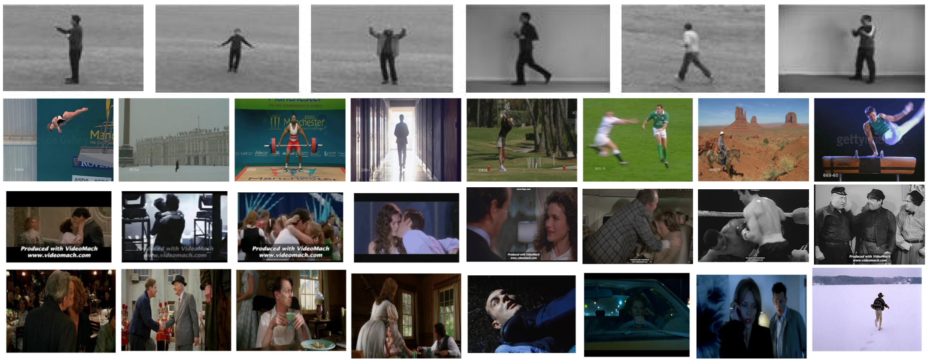


Fig. 5: Sample frames from video sequences of the KTH dataset (the top row), the UCF sports dataset (the second row), the UCF feature films dataset (the third row), and the Hollywood2 dataset (the bottom row).

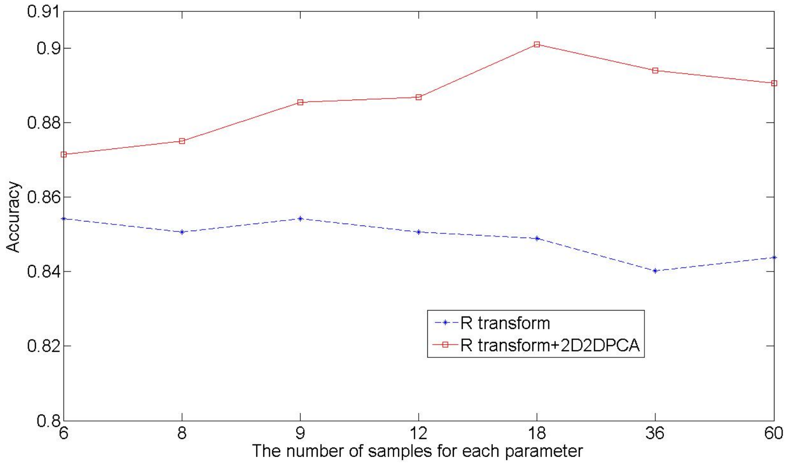


Fig. 6: Recognition accuracy obtained by the $\mathcal{R}$ transform feature and the $(2D)^2$PCA feature with respect to seven different samplings of the two parameters θ and ϕ on the KTH dataset.

5, and 3. The blue curve is the obtained recognition accuracy using the $\mathcal{R}$ transform feature without $(2D)^2$PCA, and the red curve is the recognition accuracy using the $(2D)^2$PCA to refine the $\mathcal{R}$ transform feature. From Fig. 6, the following points are observed.

(1) The sampling frequency has little influence on the final result. The best accuracy of 91.67% was obtained when the sampling intervals of θ and ϕ were both set to 10.
(2) The features obtained by $(2D)^2$PCA yield a higher recognition accuracy in most cases than the $\mathcal{R}$ transform features on their own. The former achieves 90.31% average recognition accuracy, and the latter achieves 84.9%.

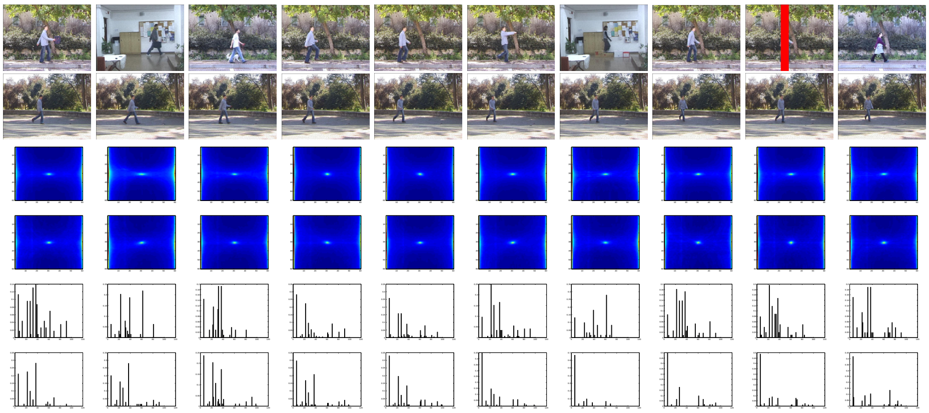


Fig. 7: Representative frames, $\mathcal{R}$ features and histogram features of 20 videos from the robustness dataset. The top two rows from left to right and from top to bottom are the representative frames of videos for walk when swinging a bag, walk when carrying a briefcase, walk with a dog, walk with knees up, limping, moonwalk, walk with occluded feet, normal walk, walk occluded by a ”pole”, walk in a skirt, and walk in ten different viewpoints (varying between 0 and 81 relative to the image plane with steps of 9) respectively. The remaining rows show the corresponding $\mathcal{R}$ features and histogram features.

These points demonstrate that $\mathcal{R}$ transform feature is an effective descriptor and the $(2D)^2$PCA further improves the discrimination of the $\mathcal{R}$ transform feature. In all other experiments on the five datasets, we set the sampling intervals of θ and ϕ to 10 and employed $(2D)^2$PCA on the $\mathcal{R}$ transform to obtain the final $\mathcal{R}$ feature in the form of an 18×18 matrix.

5.2 Robustness Evaluation for the Proposed $\mathcal{R}$ Feature

We evaluated the robustness of the proposed $\mathcal{R}$ feature with respect to challenging factors such as changes in viewpoint, changes in clothes and motion styles, etc. We performed experiments on the robustness dataset [2]. The robustness dataset includes two video sequence sets, the deformed set and the multi-view set. The deformed set includes ten test video sequences of people walking in various difficult scenarios in front of different non-uniform backgrounds. The multi-view set includes ten sequences respectively showing the “walk” action captured from a different viewpoint (varying between 0 and 81 relative to the image plane with steps of 9). The representative frames of all videos in robustness dataset are shown in the top two rows of Fig. 7.

We used a cross-dataset approach in order to acquire classification results on the robustness dataset. Specifically, we used the robustness dataset only for testing, while training was performed on the KTH dataset, following [27]. The BOVW model based histogram feature and the $\mathcal{R}$ feature of all videos in robustness dataset are shown in Fig. 7. It can be seen that although videos in this dataset have large diversity, their $\mathcal{R}$ features are similar, but the histogram features differ widely. The intra-class variations lead to large variations in local motion and appearance, but the whole geometric distribution of interest points does not change very much. Therefore, in this case the $\mathcal{R}$ feature is more efficient due to capturing the whole geometric information.

The classification results of these two methods are listed in Table 1. Since the test samples have large intra-class variance and many of them have small inter-class difference from some samples with the "jog" or "run" classes in the training set, it is difficult to classify all test samples correctly. For the histogram feature based method, 5 out of 20 videos in the robustness dataset are wrongly classified. The algorithm in [27] correctly classified 9 sequences on the deformed set and 6 sequences on the multi-view set, with all the confusions being between the walking and jogging classes. Wang *et al.* [45] only tested on the deformed set and correctly classified 8 sequences as listed in Table 1. However, the proposed R feature based method is better for dealing with cases in which there are large intra-class variance and small inter-class difference, and achieved a correct classification of all the videos. This shows that the $\mathcal{R}$ feature has relatively low sensitivity to considerable changes in scale, viewpoint and clothes, high irregularity in walking forms, etc.

In order to further evaluate the proposed R feature, we performed the experiments where some "jog" videos from the KTH dataset are used as the test set together with the robustness dataset. Hence, there are two action classes, "walk" and "jog", in the test set. Specifically, we chose the robustness dataset and all four "jog" videos performed by one person in the KTH dataset as the test set, and used the remaining videos in the KTH dataset as the training set in each run. The final results were the averages from 25 runs, since there are in total 25 subjects in the KTH dataset. We tested two methods, the histogram feature based method and the $\mathcal{R}$ feature based method. The histogram feature based method achieved accuracies of 75% and 80.21% for the "walk" class and the "jog" class respectively. The R feature based method achieved accuracies of 100% and 83.33% for the "walk" class and the "jog" class respectively. The R feature based method outperforms the histogram feature based method for both action classes. Moreover, when the test set included the "jog" videos, the R feature based method still achieved a correct classification of all the "walk" videos in the robustness dataset.

5.3 Parameter Evaluation for the Context-aware Kernel

We have proposed two kinds of context-aware kernels, one based on the *k*NN probabilistic hypergraph and the other one based on an optimized probabilistic hypergraph. Both of these two kernels have only one parameter k, which is the number of neighbors for hyperedge construction. We evaluated the parameter k on the KTH dataset. In the BOVW model, the size of codebook was set to 500. We tested four methods as follows.

(1) The BOVW based histogram feature was used. The final kernel for SVM classification was the combination of the pairwise kernel and the context-aware kernel constructed from the *k*NN probabilistic hypergraph. This method is referred to as 'BOVW+Context'.

(2) The BOVW based histogram feature, was still used, but the final kernel for SVM classification was the combination of the pairwise kernel and the context-aware kernel constructed from the proposed optimized probabilistic hypergraph. The optimized probabilistic hypergraph is obtained by a large margin nearest neighbor (LMNN) algorithm. This method is referred to as 'BOVW+O Context' which differs from 'BOVW+Context' just in adding the distance metric learning based on LMNN.

(3) The experimental configuration of this method was same as that of the first method except that the proposed $\mathcal{R}$ feature replaced the BOVW based histogram feature. This method is referred to as 'R+Context'.

Table 1: Recognition results for Wang and Suter[45] method, the BOVW method, and the proposed $\mathcal{R}$ feature method on the robustness dataset.

Test Sequence	Wang and Suter[45]	BOVW	R feature
walk with swinging a bag	walk	walk	walk
walk with carrying a briefcase	walk	walk	walk
walk with a dog	run	walk	walk
walk with knees up	walk	walk	walk
walk with limping	walk	jog	walk
moonwalk	jump	jog	walk
walk with occluded feet	walk	walk	walk
normal walk	walk	walk	walk
walk occluded by a ”pole”	walk	walk	walk
walk in a skirt	walk	walk	walk

Test Sequence	BOVW	R feature
Walk in 0^0	walk	walk
Walk in 9^0	walk	walk
Walk in 18^0	walk	walk
Walk in 27^0	walk	walk
Walk in 36^0	walk	walk
Walk in 45^0	walk	walk
Walk in 54^0	walk	walk
Walk in 63^0	jog	walk
Walk in 72^0	jog	walk
Walk in 81^0	jog	walk

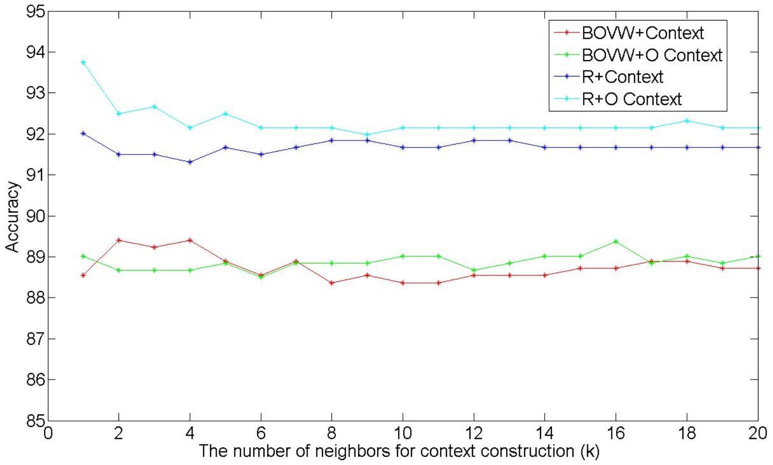


Fig. 8: Recognition accuracies of four methods with respect to *k*, the number of neighbors for context construction, on the KTH dataset.

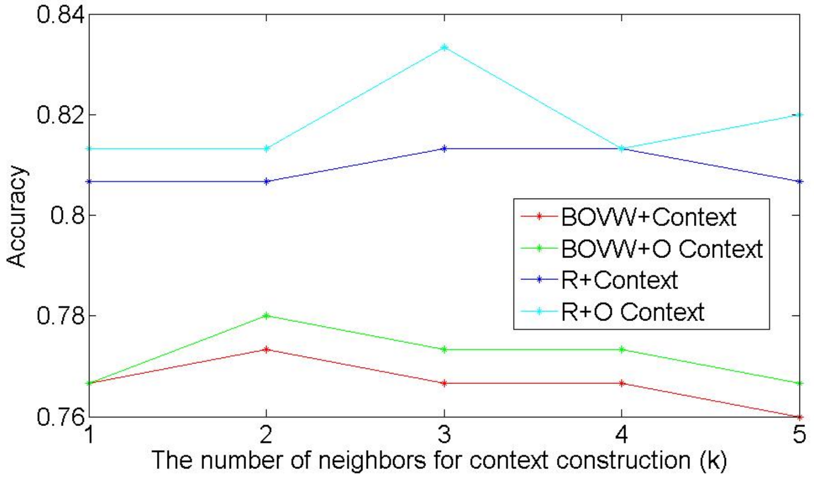


Fig. 9: Recognition accuracies of four methods with respect to k, the number of neighbors for context construction, on the UCF sports dataset.

(4) The proposed $\mathcal{R}$ feature was used together with the combination of kernels in method (2). This method is referred to as 'R+O Context'.

Fig. 8 shows the experimental results of the above four methods, with values of parameter k in the range 1 to 20. From Fig. 8, it can be seen that the accuracy does not vary much with k. But, when k is less than or equal to 5, the accuracies are a little higher than most of those when k is larger than 5. This is probably because that there are large intra-class variances for the videos with same action class. On the KTH dataset, each person performs each class of actions four times. Hence, the four videos with the same action performed by one person are usually very similar but a number of videos with the same action performed by different persons differ greatly.

Besides, from Fig. 8 it can be seen that the optimized probabilistic hypergraph based context-aware kernel approaches ('BOVW+O Context', 'R+O Context') outperform the kNN probabilistic hypergraph based context-aware kernel approaches ('BOVW+Context', 'R+Context') in most cases. However, the improvement of 'BOVW+O Context' compared with 'BOVW+Context' is not very large for the following reasons. On the KTH dataset, for each action class there are about 100 samples. By the leave-one-person-out cross-validation, there are still about 96 samples for each action class in the training set. So for 'BOVW+Context', the k nearest neighbors in the construction of the probabilistic hypergraph have the same action class as the centroid vertex with a large probability, when k ranges from 1 to 20. In this situation, the optimized probabilistic hypergraph learnt by LMNN in 'BOVW+O Context' is similar to the probabilistic hypergraph in 'BOVW+Context'. Therefore the improvement is only marginal.

Moreover, we performed the same experiments as shown in Fig. 8 on the UCF sports dataset and the UCF feature films dataset. The results are shown in Fig. 9 and Fig. 10. On the UCF sports dataset, there are in total 6 videos with respect to the "lifting" action class. So we change k from 1 to 5. In Fig. 9, we achieve better results when k is equal to 3. In Fig. 10, we achieve better results when k is equal to 5 and 13. Therefore, in other experiments we set k

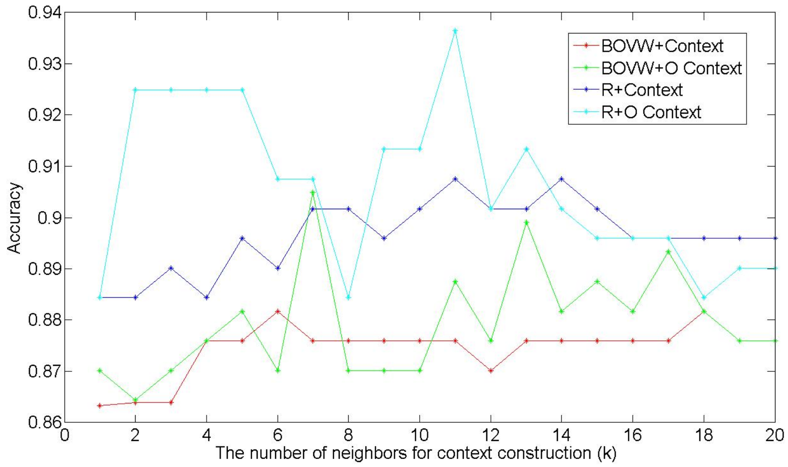


Fig. 10: Recognition accuracies of four methods with respect to k, the number of neighbors for context construction, on the feature films dataset.

to 3 in order to achieve relatively good results and at the same time to reduce computational cost.

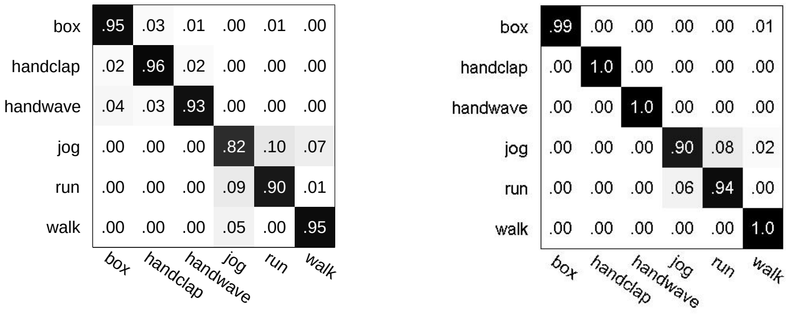
5.4 Experiments on the KTH Database

The KTH video database contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. There are in total 599 sequences in the dataset. We performed leave-one-person-out cross-validation to make the performance evaluation. In each run, 24 subjects' videos were used as the training set and the remaining one person's videos as the test set. The final results were the averages from 25 runs.

To evaluate the proposed $\mathcal{R}$ feature and context-aware kernel methods, we performed three groups of comparison experiments. In the first group of experiments, we tested two single feature based methods using the traditional pairwise kernel without the context-aware kernel, namely the BOVW based histogram feature based method and the $\mathcal{R}$ feature based method, referred to as 'BOVW' and 'R' respectively. In the second group, we compared two kinds of context-aware kernels, one based on the kNN probabilistic hypergraph and the other one based on an optimized probabilistic hypergraph. There are four experiments, 'BOVW+Context', 'BOVW+O Context', 'R+Context', and 'R+O Context'. In other words, these experiments all used one out of the two features and one out of the two context-aware kernels. In the third group, we compared our fusion approaches with three other feature fusion approaches: the feature-level fusion approach [43][37], the similarity kernel-level fusion approach, and Yuan *et al.*'s method [53]. All of these fusion approaches combined the proposed two features. Specifically, the feature-level fusion approach concatenated the two normalized feature vectors to form a larger feature vector as input to the SVM classifier.

Table 2: Comparison of three groups of methods on the KTH dataset.(%)

KTH	box	hand clap	hand wave	jog	run	walk	Average
BOVW	95.83	90.63	95.83	75.00	80.21	92.71	88.37
R	94.79	95.83	92.71	82.29	89.58	94.79	91.67
BOVW+Context	95.83	92.71	95.83	75.00	81.25	94.79	89.23
BOVW+O Context	96.88	92.71	95.83	77.08	79.17	94.79	89.41
R+Context	96.88	94.79	92.71	85.42	88.54	98.96	92.88
R+O Context	100	91.67	93.75	89.58	88.54	98.96	93.75
feature-fusion	98.96	97.92	95.83	86.46	90.63	95.83	94.27
kernel-fusion	97.92	98.96	95.83	87.50	88.54	98.96	94.61
Yuan *et al.*[53] BOVW+R	97.92	95.83	94.79	85.33	89.58	97.92	93.23
Yuan *et al.*[53] R+BOVW	100	98.96	96.88	87.50	89.58	100	95.49
ours	98.96	100	100	89.58	93.75	100	**97.05**



	box	handclap	handwave	jog	run	walk
box	.95	.03	.01	.00	.01	.00
handclap	.02	.96	.02	.00	.00	.00
handwave	.04	.03	.93	.00	.00	.00
jog	.00	.00	.00	.82	.10	.07
run	.00	.00	.00	.09	.90	.01
walk	.00	.00	.00	.05	.00	.95

	box	handclap	handwave	jog	run	walk
box	.99	.00	.00	.00	.00	.01
handclap	.00	1.0	.00	.00	.00	.00
handwave	.00	.00	1.0	.00	.00	.00
jog	.00	.00	.00	.90	.08	.02
run	.00	.00	.00	.06	.94	.00
walk	.00	.00	.00	.00	.00	1.0

Fig. 11: The confusion matrices for the proposed methods on the KTH dataset. The left one is based on the proposed $\mathcal{R}$ feature. The right one is based on the two features and our MKL approach.

For the similarity kernel-level fusion approach, we separately computed the similarity matrix for each feature, and then utilized the weighted sum of the two obtained similarity matrices as the final kernel of SVM, which is formulated as follows:

$$K_2(\mathbf{v}_i, \mathbf{v}_j) = \alpha A(H_i, H_j) + (1 - \alpha) A(G_i, G_j) \ , \tag{29}$$

where α is the weight coefficient and is decided by the cross-validation method. Yuan *et al.* [53] use one feature for context selection and a second feature for context-aware kernel computation. Since two features are used for these two steps, there are two methods, ‘BOVW+R’ and ‘R+BOVW’ with the former for context calculation and the latter for kernel calculation. Our fusion method combined four kernels via multiple kernel learning for action classification described in Subsection 4.3.

Table 2 lists the three groups of experimental results on the KTH dataset. Fig. 11 shows the confusion matrices of the $\mathcal{R}$ feature based method and our fusion method on the KTH dataset. Table 2 shows the following points.

(1) The $\mathcal{R}$ feature based method achieves 91.67% accuracy, which is 3.3% higher than the BOVW model based method. This proves the efficiency of the proposed $\mathcal{R}$ feature.

(2) All the context-aware kernel based methods (‘BOVW+Context’, ‘BOVW+O Context’, ‘R+Context’, ‘R+O Context’) outperform the corresponding traditional pairwise kernel based methods (‘BOVW’, ‘R’). Both of the context-aware kernels improve the recognition performance.
(3) The optimized probabilistic hypergraph based context-aware kernel methods (‘BOVW+O Context’, ‘R+O Context’) outperform the kNN probabilistic hypergraph based context-aware kernel methods (‘BOVW+Context’, ‘R+Context’). It demonstrates that the optimized probabilistic hypergraph further enhances the performance by utilizing the distance metric learning based on LMNN. This is because that the distance metric learning makes hyperedges in the optimized probabilistic hypergraph be pure and then prevents interference of other action classes.
(4) The methods for fusing two features all achieve higher accuracies than the single feature based methods. The $\mathcal{R}$ feature and the BOVW model based histogram feature are complementary and their combination improves action recognition.
(5) Our fusion method obtains higher accuracy than the other two common fusion methods, which demonstrates the effectiveness of our proposed fusion strategy.

5.5 Experiments on the UCF Sports Dataset

The UCF sports dataset consists of 150 action videos including 10 sport actions, diving, golf swinging, kicking, weightlifting, horseback riding, running, skating, swinging Bench, swinging from side angle, and walking. It contains a natural pool of actions featured in a wide range of scenes and viewpoints, and in unconstrained environments.

The UCF sports database was tested in a leave-one-out manner, cycling through the test videos one at a time, following [30] [14] [51]. In the BOVW model, the size of the codebook was 800. On the UCF sports database, we performed three groups of experiments similar to those on the KTH dataset. The results are shown in Fig. 12. Similar results to those for the KTH dataset were obtained. This demonstrates the effectiveness of our proposed $\mathcal{R}$ feature and fusion method on a realistic and complicated dataset. The overall average accuracy for the UCF dataset using our method is 90.67%.

Fig. 13 shows the confusion matrices of the $\mathcal{R}$ feature based method and our fusion method on the UCF sports dataset. In addition, Table 3 presents a comparison of our results with state-of-the-art results on the KTH and UCF datasets, which indicate that our method outperforms the listed methods. With our method, the overall average accuracies are 90.67% for the UCF dataset and 97.05% for the KTH dataset. The results on the UCF dataset demonstrate clearly the effectiveness of the proposed method on a realistic dataset. In particular, among these state-of-the-art methods Yuan *et al.*’s method [53] differs from ours only in the feature fusion technique. They use the same features as ours, but use one feature for context selection and the other feature for context-aware kernel computation. In our method, two context-aware kernels of two features are combined together by multiple kernel learning. By the new learning techniques, our method achieves the improvements of 1.56% and 3.34% on the KTH and UCF sport datasets respectively compared with the best results obtained in [53].

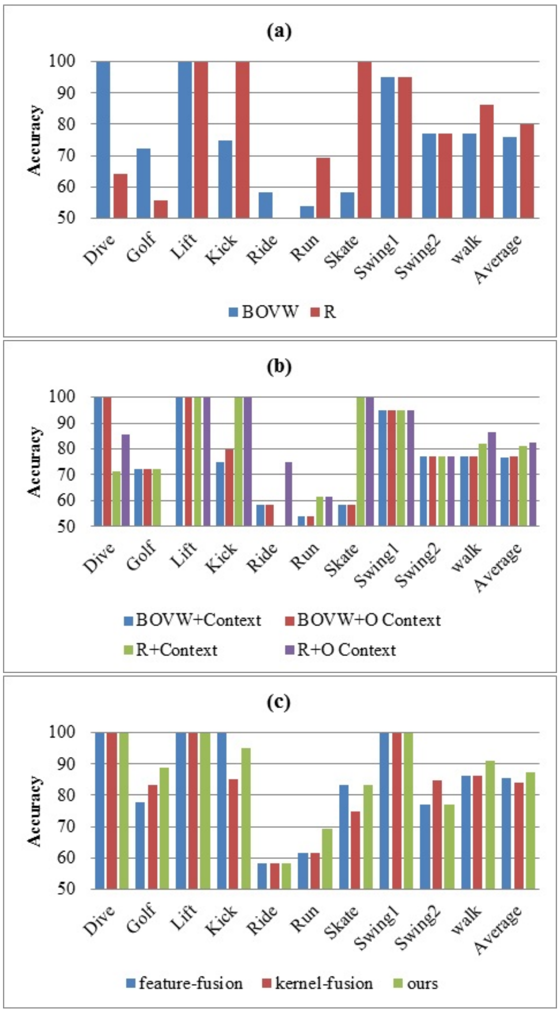


Fig. 12: Comparison of three groups of methods on the UCF sports dataset. (a) is the recognition accuracy comparison of the single feature based methods. (b) is the recognition accuracy comparison of the four context-aware kernel methods. (c) is the recognition accuracy comparison of three fusion methods.

5.6 Experiments on the UCF Feature Films Dataset

Rodriguez *et al.* [30] 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 class "kissing" and 112 samples of "hitting/slapping." The extracted samples cover a wide range of backgrounds and view points. The test for this dataset proceeded in a leave-one-out fashion. Table 4 shows the results obtained by several state-of-the-art methods on this dataset. In both categories, our fusion

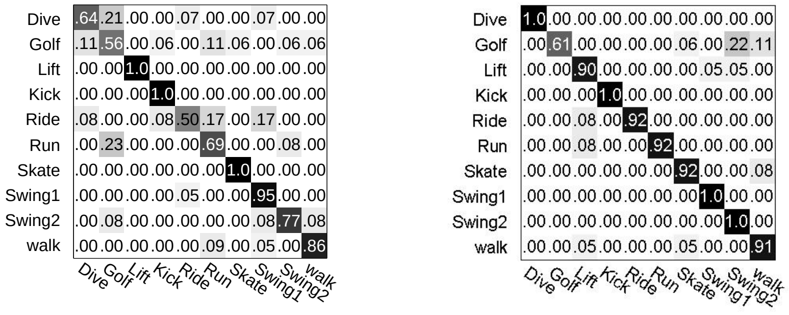


Fig. 13: The confusion matrices for the proposed methods on the UCF sports dataset. The left one is based on the propose $\mathcal{R}$ feature. The right one is based on the two features and our MKL method.

Table 3: Comparison of our method with state-of-the-art methods on the KTH and UCF sport datasets. (%)

	KTH	UCF sport
Our method	**97.05**	**90.67**
Bregonzio *et al.* [3]	93.17	-
Bregonzio *et al.* [4]	-	86.9
Sun *et al.* [37]	94.0	-
Yeffet and Wolf [51]	90.1	79.2
Wang *et al.* [42]	92.1	85.6
Kovashka *et al.* [14]	94.53	87.27
Le *et al.* [19]	93.9	86.5
Wang *et al.* [40]	94.2	88.2
Yuan *et al.* [53]	95.49	87.33

method shows a higher performance. Bregonzio *et al.* [4] achieve an accuracy of 96.75% which is a little lower than our method on the UCF feature films dataset. But Bregonzio *et al.* [4] achieve an accuracy of 86.90% which is much lower than our 90.67% on the UCF Sport dataset.

5.7 Experiments on the Hollywood2 Dataset

The Hollywood2 dataset [23] has 12 classes collected from 69 Hollywood movies. We used the clean training dataset. The performance was evaluated as suggested in [23], i.e., by computing the average precision (AP) for each of the action classes and reporting the mean AP over all classes (mAP). Finally, the evaluation results for Hollywood2 dataset are presented in Table 5. Each of the four methods in the top row was based on only one kind of feature. Excepting the proposed $\mathcal{R}$ feature, the other three methods were based on 3D SIFT, HOF, and HOG with the BOVW model respectively. The proposed $\mathcal{R}$ feature achieves 42.8% which demonstrates its effectiveness on a realistic dataset. The four methods in the bottom row all combined multiple features. Our method employed the 3D SIFT feature and the proposed $\mathcal{R}$

Table 4: Comparison of our method with state-of-the-art methods on the UCF feature films dataset. (%)

	Kiss	Slap	Average
Rodriguez *et al.*[30]	66.4	67.2	66.8
Yeffet and Wolf [51]	77.3	84.2	80.75
Oshin *et al.* [28]	82.4	77.2	79.8
Wu *et al.* [48]	97.6	94.4	96.0
Bregonzio *et al.* [4]	97	**96.5**	96.75
BOVW	85.4	89.4	87.4
R	95.8	83.1	90.7
Ours	**98.96**	95.29	**97.24**

Table 5: Comparison of our method with state-of-the-art methods on Hollywood2 dataset. (%)

	3D SIFT	R	HOG[42]	HOF[42]
mAP	49.68	42.84	39.4	45.8
	Ours	Wang[42]	Le[19]	Wang[40]
mAP	57.39	47.7	53.3	**58.3**

feature. Wang *et al.* [40] utilized four features (trajectory, HOG, HOF, and MBH) and employed a dense sampling strategy. The result is 54.6% using the KTL tracking. Our method achieves comparable results to Wang *et al.* [40]. The proposed $\mathcal{R}$ feature does not achieve the best result among the one feature based method, but the combination with the SIFT feature highly improves the performance.

6 Conclusions

In this paper we have presented a new action recognition framework based on spatio-temporal interest points. First, we have proposed a new holistic descriptor, the 3D $\mathcal{R}$ transform on spatio-temporal interest points, to capture the detailed global geometrical distribution. Second, we have proposed a new context-aware kernel to measure the similarity of video representations. The context-aware kernel models high-order relationships among video in order to overcome the disadvantage of the traditional pairwise context-free kernel, which is sensitive to noise and outliers in the data. Moreover, an optimized hypergraph has been proposed for context-aware kernel construction. It makes the context have the same action class as the centroid and prevents disturbance from other classes. Experimental results on several datasets have demonstrated the effectiveness of our proposed $\mathcal{R}$ feature and context-aware kernel method.

Our method has the following limitations: The proposed $\mathcal{R}$ transform is not appropriate for modeling dense points or random points, since the geometrical distribution of the dense points or random points is not discriminative for action recognition; Compared with the traditional methods which just combine the pair-wise kernels with SVM, the additional context-aware kernel computing and multiple kernel learning in our method improve the performance but at the same time increase the computational complexity.

7 Acknowledgements

This work is partly supported by the 973 basic research program of China (Grant No. 2014CB349303), the Natural Science Foundation of China (Grant No. 61472421, 61472420, 61303086, 61202327), the Project Supported by CAS Center for Excellence in Brain Science and Intelligence Technology, and the Project Supported by Guangdong Natural Science Foundation (Grant No. S2012020011081).

References

1. Armijo, L. (1966). Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1), 1-3.
2. Blank, M., Gorelick, L., Shechtman, E., Irani, M., & Basri, R. (2007). Actions as space-time shapes. *PAMI*, 29(12), 2247-2253.
3. Bregonzio, M., Gong, S., & Xiang, T. (2009). Recognising action as clouds of space-time interest points. In *CVPR* (pp. 1948-1955) .
4. Bregonzio, M., Li, J., Gong, S., & Xiang, T. (2011). Discriminative topics modelling for action feature selection and recognition. In *BMVC* (pp. 1-11).
5. Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine learning*, 46(1-3), 131-159.
6. Choi, J., Jeon, W. J., & Lee, S. C. (2008). Spatio-temporal pyramid matching for sports videos. In *ACM MIR* (pp. 291-297).
7. Daras, P., Zarpalas, D., Tzovaras, D., & Strintzis, M. G. (2004). Shape matching using the 3D Radon tranform. In *3DPVT* (pp. 953-960).
8. Ellis, C., Masood, S., Tappen, M., LaViola, J., & Sukthankar, R. (2013). Exploring the trade-off between accuracy and observational latency in action recognition. *International Journal of Computer Vision*, 101(3), 420-436.
9. Gaidon, A., Harchaoui, Z., & Schmid, C. (2014). Activity representation with motion hierarchies. *International Journal of Computer Vision*, 107(3), 219-238.
10. Huang, Y., Liu, Q., Zhang, S., & Metaxas, D. (2010). Image retrieval via probabilistic hypergraph ranking. In *CVPR* (pp. 3376-3383).
11. Hong, C., Yu, J., & Chen, X. (2014). Structured action classification with hypergraph regularization. In *IEEE International Conference on Systems, Man and Cybernetics (SMC)* (pp. 2853-2858).
12. Ikizler-Cinbis, N., & Sclaroff, S. (2010). Object, scene and actions: combining multiple features for human action recognition. In *ECCV* pp. 494-507.
13. Kloft, M., Brefeld, U., Sonnenburg, S., & Zien, A. (2011). Lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 12, 953-997.
14. Kovashka, A., & Grauman, K. (2010). Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*(pp. 2046-2053).
15. Kulkarni, K., Evangelidis, G., Cech, J., & Horaud, R. (2014). Continuous action recognition based on sequence alignment. *International Journal of Computer Vision*.
16. Laptev, I., Marszalek, M., Schmid, C., & Rozenfeld, B. (2008). Learning realistic human actions from movies. In *CVPR* (pp. 1-8).
17. Laptev, I. (2005). On space-time interest points. *IJCV*, 64(2), 107-123.
18. Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR* (pp. 2169-2178).
19. Le, Q., Zou, W., Yeung, S., & Ng, A. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR* (pp. 3361-3368).
20. Li, X, Hu, W., Shen, C., Dick, A., & Zhang, Z. (2013). Context-aware hypergraph construction for robust spectral clustering. *IEEE Trans. on Knowledge and Data Engineering*, 1-10.
21. Lianga, Z., Chi, Z., Fu, H., & Fenga, D. (2012). Salient object detection using content-sensitive hypergraph representation and partitioning. *Pattern Recognition*, 45, 3886-3901.
22. Liu, J., Ali, S., & Shah, M. (2008). Recognizing human actions using multiple features. In *CVPR* (pp. 1-8).
23. Marzalek, M., Laptev, I., & Schmid, C. (2009). Actions in context. In *CVPR* (pp. 2929-2936).
24. Mikolajczyk, K., & Uemura, H. (2008). Action recognition with motion-appearance vocabulary forest. In *CVPR* (pp. 1-8).

25. Ni, B., Moulin, P., & Yan, S. (2014). Pose adaptive motion feature pooling for human action analysis. *International Journal of Computer Vision*, 1-20.
26. Niebles, J., Wang, H., & Fei-Fei L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 793, 299-318.
27. Oikonomopoulos, A., Patras, I., & Pantic, M. (2011). Spatiotemporal localization and categorization of human actions in unsegmented image sequences. *IEEE Trans. on Image Process*, 20(4), 1126-1140.
28. Oshin, O., Gilbert, A., Bowden, R. (2011). Capturing the relative distribution of features for action recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops (FG 2011)* (pp.111-116).
29. Poppe, R. (2010). A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6), 976-990.
30. Rodriguez, M. D., Ahmed, J., & Shah, M. (2008). Action MACH: a spatiotemporal maximum average correlation height filter for action recognition. In *CVPR* (pp. 1-8).
31. Savarese, S., Pozo, A., Niebles, J., & Fei-Fei, L. (2008). Spatial-temporal correlatons for unsupervised action classification. In *IEEE Workshop on Motion and Video Computing* (pp. 1-8).
32. Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: A local SVM approach. In *ICPR* (pp. 32-36).
33. Shalev-Shwartz, S., Singer, Y., Srebro, N., & Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1), 3-30.
34. Shi, Q., Cheng, L., Wang, L., & Smola, A. (2011). Human action segmentation and recognition using discriminative semi-Markov models. *International Journal of Computer Vision*, 93(1), 22-32.
35. Shkolnisky, Y., & Averbuch, A. (2003). 3D Fourier based discrete Radon transform. *Applied and Computational Harmonic Analysis*, 15(1), 33-69.
36. Sun, J., Wu, X., Yan, S., Cheong, L., Chua, T., & Li, J. (2009). Hierarchical spatio-temporal context modeling for action recognition. In *CVPR* (pp. 2004-2011).
37. Sun, X., Chen, M., & Hauptmann, A. (2009). Action recognition via local descriptors and holistic features. In *CVPR* (pp. 58-65).
38. Tabbone, S., Wendling, L., & Salmon, J. (2006). A new shape descriptor defined on the Radon transform. In *CVIU* (pp. 42-51).
39. Varma, M., & Bodla, R. (2009). More generality in efficient multiple kernal learning. In *ICML* (pp. 1065-1072).
40. Wang, H., Kläser, A., Laptev, I., Schmid, C., & Liu, C. (2011). Action recognition by dense trajectories. In *CVPR* (pp. 3169-3176).
41. Wang, H., Kläser, A., Schmid, C., & Liu, C. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 1031, 60-79.
42. Wang, H., Ullah, M. M., Kläser, A., Laptev, I., & Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *BMVC*.
43. Wang, L., Zhou, H., Low, S.C., & Leckie, C. (2009). Action Recognition via Multi-Feature Fusion and Gaussian Process Classification. In *WACV* (pp. 1-6).
44. Wang, H., & Yuan, J. (2015). Collaborative multi-feature fusion for transductive spectral learning. *IEEE Transactions on Cybernetics*, 45(3), 465-475.
45. Wang, L., & Suter, D. (2007). Learning and matching of dynamic shape manifolds for human action recognition. *IEEE Transactions on Image Processing*, 16(6), 1646-1661.
46. Wang, Y., Huang, K., & Tan, T. (2007). Human activity recognition based on $\mathcal{R}$ transform. In *CVPR* (pp. 1-8).
47. Weng, C., & Yuan, J. (2015). Efficient mining of optimal AND/OR patterns for visual recognition. *IEEE Transactions on Multimedia*, 17(5), 626-635.
48. Wu, B., Yuan, C., & Hu, W. (2014). Human action recognition based on context-dependent graph kernels. In *CVPR* (pp. 2609C2616).
49. Weinberger, K., & Saul, L. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 207-244.
50. Yang, J., Zhang, D., Frangi, A.F., & Yang, J. (2004). Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *PAMI*, 26(1), 131-137.
51. Yeffet, L., & Wolf, L. (2009). Local trinary patterns for human action recognition. In *ICCV* (pp. 492-497).
52. Yu, J., Tao, D., & Wang, M. (2012). Adaptive hypergraph learning and its application in image classification. *IEEE Transactions on Image Process*, 21(7), 3262-3272.
53. Yuan, C., Li, X., Hu, W., Lin, H., Maybank, S., & Wang, H. (2013). 3D R transform on spatio-temporal interest points for action recognition. In *CVPR* (pp. 724-730).
54. Yuan, J., Wu, Y., & Yang, M. (2007). Discovery of collocation patterns: From visual words to visual phrases. In *CVPR* (pp. 1-8).

55. Yuan, J., & Wu, Y. (2012). Mining visual collocation patterns via self-supervised subspace learning. *IEEE Transactions on Systems, Man, Cybernetics B, Cybernetics*, 42(2), 334-346.
56. Yuan, J., Yang, M., & Wu, Y. (2011). Mining discriminative co-occurrence patterns for visual recognition, In *CVPR* (pp. 2777-2784).
57. Zhang, D., & Zhou, Z. (2005). $(2D)^2$PCA: 2-Directional 2-Dimensional PCA for efficient face representation and recognition. *Neurocomputing*, 69(1-3), 224-231.
58. Zhang, L., Gao, Y., Hong, C., Feng, Y., Zhu, J., & Cai, D. (2014). Feature correlation hypergraph: Exploiting high-order pPotentials for multimodal recognition. *IEEE Transactions on Cybernetics*, 44(8), 1408-1419.
59. Zhou, D., Huang, J., & Schölkopf, B. (2006). Learning with hypergraphs: Clustering, classification, and embedding. In *NIPS* (pp. 1601-1608).
60. Zhu, F., & Shao, L. (2014). Weakly-supervised cross-domain dictionary learning for visual recognition. *International Journal of Computer Vision*, 109(1-2), 42-59.