Kernelized Multiview Projection for Robust Action Recognition

Ling Shao[1] · Li Liu[1] · Mengyang Yu[1]

Received: 14 October 2014 / Accepted: 23 September 2015 / Published online: 5 October 2015

Abstract Conventional action recognition algorithms adopt a single type of feature or a simple concatenation of multiple features. In this paper, we propose to better fuse and embed different feature representations for action recognition using a novel spectral coding algorithm called Kernelized Multiview Projection (KMP). Computing the kernel matrices from different features/views via time-sequential distance learning, KMP can encode different features with different weights to achieve a low-dimensional and semantically meaningful subspace where the distribution of each view is sufficiently smooth and discriminative. More crucially, KMP is linear for the reproducing kernel Hilbert space, which allows it to be competent for various practical applications. We demonstrate KMP's performance for action recognition on five popular action datasets and the results are consistently superior to state-of-the-art techniques.

Keywords Human action recognition · Sequential distance learning · Multiple view fusion · Dimensionality reduction · Spectral coding

Communicated by Martial Hebert.

✉ Ling Shao
ling.shao@ieee.org

Li Liu
li2.liu@northumbria.ac.uk

Mengyang Yu
m.y.yu@ieee.org

[1] Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne NE1 8ST, UK

1 Introduction

Human action recognition has been widely applied to human-computer interaction, human behavior analysis, video surveillance, robotics and so on. Traditional action recognition techniques are mainly based on single feature representations, either global (Shao et al. 2014) or local (Laptev et al. 2008). For local feature extraction, an unsupervised detection technique, such as: cuboid detector (Dollár et al. 2005), is first applied to locate the spatio-temporal interest points around which the salient features, e.g., histogram of 3D oriented gradients (3DHOG) (Klaser and Marszalek 2008), 3D scale invariant feature transforms (3DSIFT) (Scovanner et al. 2007), or histogram of optical flow (HOF) (Laptev et al. 2008), are extracted. Then, the bag-of-visual-words scheme is employed to embed these local features into a whole histogram representation. On the one hand, local feature based methods tend to be more robust and effective in challenging scenarios, while this kind of representation is often not precise and informative because of the quantization error during codebook construction and the loss of structural relationships among local features. On the other hand, global representations (Bobick and Davis 2001; Ji et al. 2013; Taylor et al. 2010) describe the action clip as a whole. Thus, it would be more informative to capture the discriminative features along both spatial and temporal dimensions. Unfortunately, global methods are sensitive to shift, scaling, occlusion, and cluttering, which commonly exist in action sequences.

Notwithstanding the remarkable results achieved by both local and global methods in some cases, most of them are still based on single feature representations. Since variations in lighting conditions, intra-class differences, complex backgrounds and viewpoint and scale changes all lead to obstacles for robust feature extraction and action classification, single feature representations cannot handle the realistic tasks to a satisfactory extent. In some situations, the direct concatenation of different features such as (Wang et al. 2013) can improve the performance over single features. However, the concatenation will make the representation quite lengthy and the relationship between different features is not exploited.

In practice, a typical action clip can be represented by different views/features, e.g., gradient, shape, color, texture and motion. Generally speaking, these views from different feature spaces always maintain their particular statistical characteristics. Accordingly, it is desirable to incorporate these heterogeneous feature descriptors into one compact representation, leading to the multiview learning approaches (Long et al. 2008; Xia et al. 2010; Xu et al. 2014, 2015). These techniques have been designed for multiview data classification (Zien and Ong 2007), clustering (Bickel and Scheffer 2004) and feature selection (Zhao and Liu 2008). For such multiview learning tasks, the feature representations are usually very high-dimensional for each view. However, little effort has been paid to learning low-dimensional and compact representations for multiview computer vision tasks. Thus, how to obtain a comprehensively low-dimensional embedding to discover the discriminative information from all views is a worthy research topic, since the effectiveness and efficiency of the methods drop exponentially as the dimensionality increases, which is commonly referred to as the curse of dimensionality.

In this paper, we propose to encode different feature representations for action recognition using a novel multiview subspace learning method called Kernelized Multiview Projection (KMP). Our preliminary study shows KMP can produce outstanding results for image classification (Yu et al. 2015). For action recognition, the spatio-temporal nature of a video sequence has to be considered and represented in a meaningful manner. Particularly, each action clip is first described by several individual views using frame-based representations, which contain the whole human body with the complete information of spatial structure and share the advantages with the global representation methods. Therefore, the adopted representation can be regarded as a semi-holistic representation of human actions. It inherits the advantages of global features in the spatial dimension and meanwhile has the superiority of local features in the temporal axis. To further preserve the sequential information of actions (Zhang and Tao 2012), for each view, the dynamic time warping (DTW) (Berndt and Clifford 1994) technique is applied to form radial basis function (RBF) sequential kernels. Having obtained kernel values for each view in the reproducing kernel Hilbert space (RKHS), KMP is able to fuse the features from different views, which have different dimensions, by exploring the complementary property of different views and finally finds a unique low-dimensional subspace where the distribution of each view is sufficiently smooth and discriminative. Different from multiple kernel learning methods (Gönen and Alpaydin 2011) which include linear and nonlinear approaches to learn the fused kernel matrix based on the maximum margin criterion, KMP also investigate the similarity and local information of features from each view.

The rest of this paper is organized as follows. In Sect. 2, we give a brief review of the related work. The details of our method are described in Sect. 3. Section 4 reports the experimental results. Finally, we conclude this paper in Sect. 5.

2 Related Work

A simple multiview embedding framework is to concatenate the feature vectors from different views together as a new representation and utilize an existing dimensionality reduction method directly on the concatenated vector to obtain the final mulitiview representation. Nonetheless, this kind of concatenation is not physically meaningful because each view has a specific characteristic. And, the relationship between different views is ignored and the complementary nature of intrinsic data structure of different views is not sufficiently explored.

One feasible solution is proposed in (Long et al. 2008), namely, distributed spectral embedding (DSE). For DSE, a spectral embedding scheme is first performed on each view, respectively, producing the individual low-dimensional representations. After that, a common compact embedding is finally learned to guarantee that it would be similar with all single-view's representations as much as possible. Although the spectral structure of each view can be effectively considered for learning a multiview embedding via DSE, the complementarity between different views is still neglected.

To effectively and efficiently learn the complementary nature of different views, multiview spectral embedding (MSE) is introduced in (Xia et al. 2010). The main advantage of MSE is that it can simultaneously learn a low-dimensional embedding over all views rather than separate learning as in DSE. Additionally, MSE shows better effectiveness in fusing different views in the learning phase.

However, both DSE and MSE are based on nonlinear embedding, which leads to a serious computational complexity problem. In particular, when we apply them to classification or retrieval tasks, the methods have to be retrained for learning the low-dimensional embedding when new test data are used. Besides, this kind of mechanism causes an uncertain training phase, since the low-dimensional representations of training data are always changing after retraining the model for a new test sample. Due to their nonlinearity nature, this will cause heavily computational costs and even become impractical for realistic and large-scale scenarios.

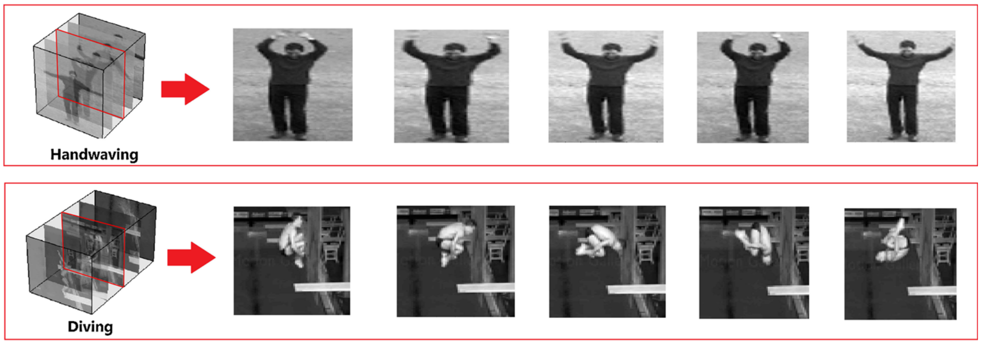


Fig. 1 Illustration of selected middle frames from actions "Handwaving" and "Diving"

Therefore, in this paper, we propose a robust linear projection embedding method for RKHS, namely, KMP. It is noteworthy that, different from non-linear approaches, once the learning phase of KMP is finished and the projection is learned, it will be fixed and can be directly used to embed the new test samples without any re-training (Fig. 1).

3 Methodology

Our recognition system is composed of the following main stages: (1) Pose description: For each video sequence, a set of visual features is extracted from each frame to represent the pose appearing in it. (2) Sequential distance kernel learning: Each feature view is computed into a kernel matrix via our proposed Gaussian-sequential learning. (3) Kernelized Multiview Projection: KMP is able to successfully explore the complementary property of different views and finally finds a discriminative low-dimensional subspace to fuse all views into a single feature vector. (4) Action recognition: the SVM with the RBF kernel is finally applied to categorize actions into different classes. The flowchart of the proposed method is illustrated in Fig. 2. We will detail the above stages in the following sections.

3.1 Notations

We are given N training video sequences $\{v_1, \ldots, v_N\}$ and M different descriptors are used for multiview feature extraction. For the i-th view and p-th video sequence, X_p^i represents the matrix composed of the feature column vector of i-th view in time-sequential order. Since the dimensions of various descriptors are different, kernel matrices $K_1, \ldots, K_M \in \mathbb{R}^{N\times N}$ are constructed in Sect. 3.3 for the fusion of different views. Our task is to output an optimal projection matrix $P \in \mathbb{R}^{N\times d}$ and weights $\{\alpha_1, \ldots, \alpha_M\}$ $(\sum_{i=1}^{M} \alpha_i = 1)$ for kernel matrices such that the fused feature matrix $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T = KP = (\sum_{i=1}^{M} \alpha_i K_i)P$ can represent original data comprehensively.

3.2 Incremental Naive Bayes Denoising

In a video sequence, however, not all of the poses are informative and discriminative for action recognition. Some poses may carry neither complete nor accurate information and would even contain common patterns shared by various action types. Since these poses in a video sequence cannot represent the action well and would cause confusion during the classification phase, a weakly supervised method, termed incremental Naive Bayes filter (INBF), has been carried out to filter the noisy representation and keep the relatively representative and discriminative poses, i.e., the key poses.

For each action category, ten action sequences are randomly selected. We choose a small set of discriminative poses for a certain action type from each action sequence as the INBF initial positive samples (labeled as $y = 1$), and the remaining frames are adopted as the negative ones ($y = 0$). As illustrated in Fig. 1, the five frames in the middle of an action sequence are selected as discriminative poses. We repetitively apply the above procedure to each action type. INBF is then regarded as an unsupervised online learning strategy.

For the i-th feature view, the representation of each pose (frame) s is $\mathbf{x}^i(s) = (x_1^i(s), \ldots, x_D^i(s)) \in \mathbb{R}^D$. Since all the features we extracted are based on statistical histograms, we assume all elements in x^i are independently distributed and model them with a naive Bayes classifier:

$$
\begin{aligned}
P(\mathbf{x}^i) &= \log \frac{\Pi_{m=1}^{D} \Pr(x_m^i|y=1)\Pr(y=1)}{\Pi_{m=1}^{D} \Pr(x_m^i|y=0)\Pr(y=0)} \\
&= \sum_{m=1}^{D} \log \frac{\Pr(x_m^i|y=1)}{\Pr(x_m^i|y=0)}.
\end{aligned}
\tag{1}
$$

Note that we make the assumption of a uniform prior, i.e., $\Pr(y = 1) = \Pr(y = 0)$, and $y \in \{0, 1\}$ is a binary variable which represents the positive and negative sample labels, respectively.

Furthermore, in either statistics or physics, real-world data distribution empirically follows the same form, i.e.,

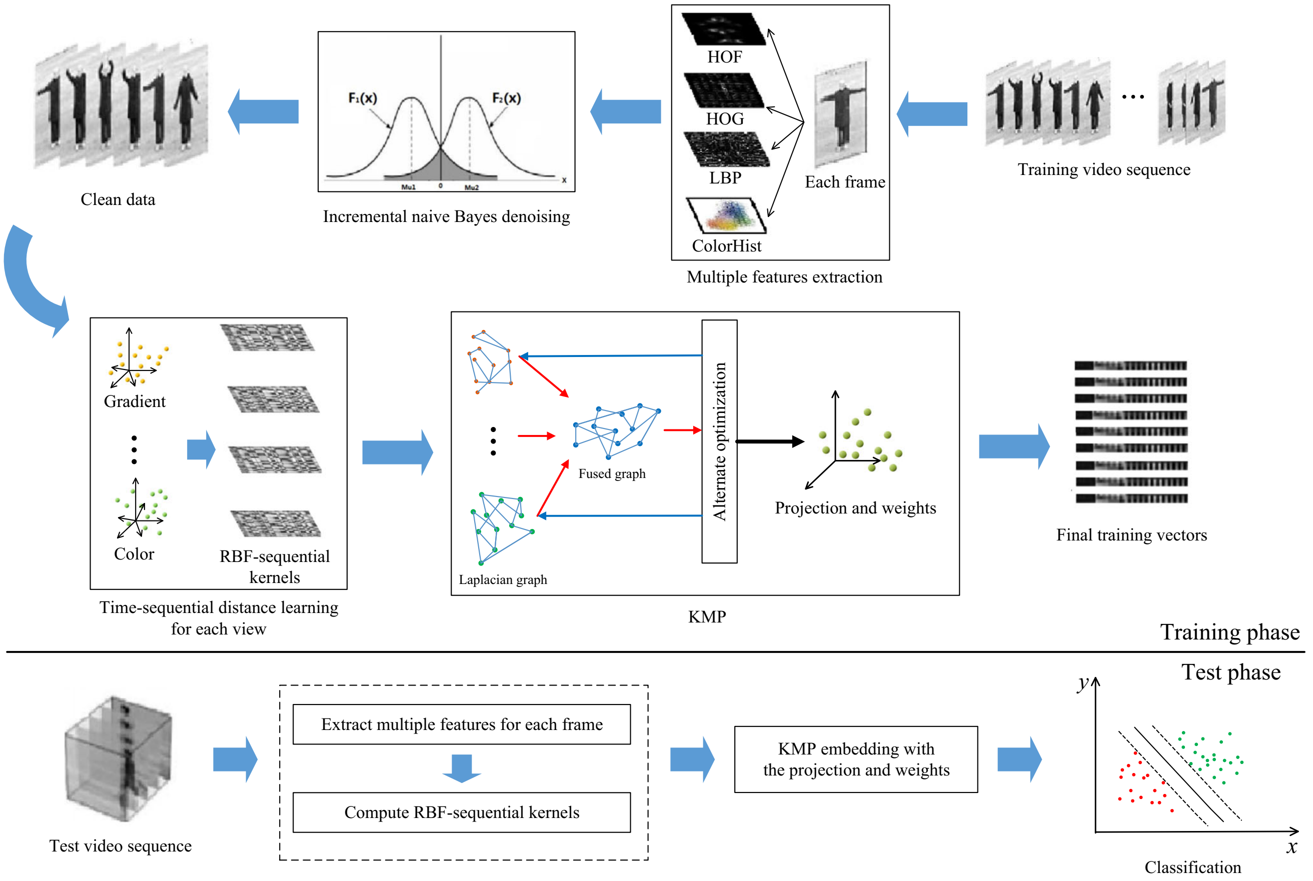


Fig. 2 Working flow of the proposed method. Multiple features are extracted from training video data for each frame. Based on the data after incremental naive Bayes denoising, the dynamic time warping is performed to construct the kernel matrices for each view. Then a projection matrix and weights for kernel matrices are derived by an EM-like alternate optimization procedure

Gaussian distribution. Thus, the conditional distributions $x_m^i|y = 1$ and $x_m^i|y = 0$ in the classifier $P(\mathbf{x}^i)$ are assumed to be Gaussian distributed with the four-tuple $(\mu_{y=1}^m, \mu_{y=0}^m, \sigma_{y=1}^m, \sigma_{y=0}^m)$, which satisfy

$$x_m^i|y = 1 \sim N\left(\mu_{y=1}^m, \sigma_{y=1}^m\right)$$

and

$$x_m^i|y = 0 \sim N\left(\mu_{y=0}^m, \sigma_{y=0}^m\right).$$

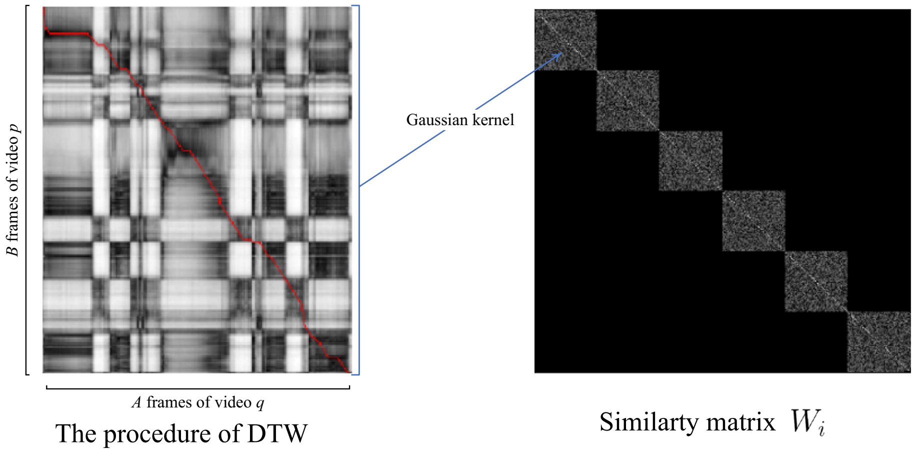
Up to now, for a certain feature view, we can initialize a group of naive Bayes models for each action type, and the training sequence is successively employed through all the models. The Gaussian parameters in INBF can be then incrementally updated as follows:

$$\mu_{y=1}^m \leftarrow \lambda\mu_{y=1}^m + (1-\lambda)\mu_{y=1},$$
$$\sigma_{y=1}^m \leftarrow \sqrt{\lambda\left(\sigma_{y=1}^m\right)^2 + (1-\lambda)(\sigma_{y=1})^2 + \lambda(1-\lambda)\left(\mu_{y=1}^m - \mu_{y=1}\right)^2}, \tag{2}$$

where λ > 0 denotes the learning rate of INBF, $\mu_{y=1} = \frac{1}{S}\sum_{s|y(s)=1} x_m^i(s)$, $\sigma_{y=1} = \sqrt{\frac{1}{S}\sum_{s|y(s)=1}(x_m^i(s) - \mu_{y=1})^2}$ and $S = |\{s|y(s) = 1\}|$. And $\mu_{y=0}^m$ and $\sigma_{y=0}^m$ have similar update rules. The above solutions are easily obtained by maximum likelihood estimation. In this way, we can use INBF to keep the representative frames for the later learning phase and discard irrelevant frames to decrease the influence of noise.

3.3 RBF Sequential Kernel Construction

For the i-th view, since we extract features from the frames of video sequences, each video sequence can be described by a set of features with a sequential order (along the temporal axis). The similarity between video v_p and video v_q under view i: $k_i(v_p, v_q)$ can be measured via DTW (Berndt and Clifford 1994). Therefore, the kernel function can be defined as: $k_i(v_p, v_q) = \exp(-\frac{DTW(X_p^i, X_q^i)^2}{2\sigma^2})$, where $DTW(X_p^i, X_q^i)$ indicates the sequential distance computed via DTW and σ is a standard deviation in the RBF kernel. In this way, we can easily obtain the kernel matrices for different views using the above equation.



The procedure of DTW

Similarty matrix W_i

Fig. 3 Illustration of the similarity matrix construction

3.4 Kernelized Multiview Projection

Based on the above kernel construction, we can obtain kernel matrices $K_1, \dots, K_M \in \mathbb{R}^{N \times N}$ with the same size for M views with different dimensions. Furthermore, we use the label of training video sequences to supervise the calculation of the similarity matrix W_i for the i-th view. Then each component of W_i is computed as follows:

$$(W_i)_{pq} = \begin{cases} \exp\left(-\frac{DTW(X_p^i, X_q^i)^2}{2\sigma^2}\right), & C(p) = C(q) \\ 0, & otherwise \end{cases}, \qquad (3)$$

where $C(p)$ is the label function which indicates the label of video v_p and $p, q = 1, \dots, N$. In fact, the similarity matrix W_i is a block matrix consisting of some submatrices of kernel matrix K_i as illustrated in Fig. 3. Then we have the diagonal matrix D_i in which $(D_i)_{pp} = \sum_q (W_i)_{pq}$ and the Laplacian matrix $L_i = D_i - W_i$ for each view i.

Due to the complementary nature of different descriptors, we assign different weights for different views. The goal of KMP is to find the basis of a subspace in which the lower-dimensional representation can preserve the intrinsic structure of original data. Therefore, we impose a set of nonnegative weights $\alpha = (\alpha_1, \dots, \alpha_M)$ on the similarity matrices $W_1, \dots, W_M$ and we have the fused similarity matrix $W = \sum_{i=1}^M \alpha_i W_i$ and the fused Laplacian matrix $L = \sum_{i=1}^M \alpha_i L_i$.

For the kernel matrix, since we use the same method (DTW) to compute kernel values and similarities, we can also define the fused kernel matrix $K = \sum_{i=1}^M \alpha_i K_i$. In fact, suppose ϕ_i is the substantial feature map for kernel K_i, i.e., $K_i = \phi_i(X^i)^T \phi_i(X^i)$, then the fused kernel value is computed by the feature vector concatenated by the mapped vectors via $\phi_1, \dots, \phi_M$, since we have

$$\begin{aligned} K &= \sum_{i=1}^M \alpha_i K_i = \sum_{i=1}^M \alpha_i \phi_i(X^i)^T \phi_i(X^i) \\ &= \begin{bmatrix} \sqrt{\alpha_1}\phi_1(X^1) \\ \vdots \\ \sqrt{\alpha_M}\phi_M(X^M) \end{bmatrix}^T \begin{bmatrix} \sqrt{\alpha_1}\phi_1(X^1) \\ \vdots \\ \sqrt{\alpha_M}\phi_M(X^M) \end{bmatrix} \\ &= \phi(X)^T \phi(X), \end{aligned}$$

where $\phi(\cdot) = [\sqrt{\alpha_1}\phi_1(\cdot)^T, \cdots, \sqrt{\alpha_M}\phi_M(\cdot)^T]^T$ is the fused feature map and $X = (X^1, \dots, X^M)$ is the M-tuple consisting of features from all the views.

To preserve the fused locality information, we need to find the optimal projection for the following optimization problem:

$$\arg\min_{\mathbf{v}} \sum_{ij} \|\mathbf{v}^T \psi_i - \mathbf{v}^T \psi_j\|^2 (W)_{ij}, \qquad (4)$$

where ψ_i is the fused mapped feature, i.e., $[\psi_1, \dots, \psi_N] = \phi(X)$. Through simple algebra derivation, the above optimization problem can be transformed to the following form:

$$\arg\min_{\mathbf{v}} \mathrm{Tr}(\mathbf{v}^T \phi(X) L \phi(X)^T \mathbf{v}). \qquad (5)$$

With the constraint $\mathrm{Tr}(\mathbf{v}^T \phi(X) D \phi(X)^T \mathbf{v}) = 1$, minimizing the objective function in Eq. (5) is to solve the following generalized eigenvalue problem:

$$\phi(X) L \phi(X)^T \mathbf{v} = \lambda \phi(X) D \phi(X)^T \mathbf{v}. \qquad (6)$$

Note that each solution of problem (6) is a linear combination of $\psi_1, \dots, \psi_N$, and there exits N-tuple $\mathbf{p} = (p_1, \dots, p_N) \in \mathbb{R}^N$ such that $\mathbf{v} = \sum_{i=1}^N p_i \psi_i = \phi(X)\mathbf{p}$. For matrix V consisting of all the solutions, there exists a matrix P such that $V = \phi(X)P$. Therefore, with the additional constraint

$\mathrm{Tr}(P^T\phi(X)D\phi(X)^TP) = 1$, we can formulate the new objective function as follows:

$$\begin{aligned}&\arg\min_{P,\alpha} \mathrm{Tr}(P^TKLKP)\\ &\text{s.t. } \mathrm{Tr}(P^TKDKP)=1,\ \sum_{i=1}^{M}\alpha_i=1,\ \alpha_i\geq 0,\end{aligned} \tag{7}$$

or in the form without the trace constraint:

$$\arg\min_{P,\alpha} \frac{\mathrm{Tr}(P^TKLKP)}{\mathrm{Tr}(P^TKDKP)},\ \text{s.t. } \sum_{i=1}^{M}\alpha_i=1,\ \alpha_i\geq 0. \tag{8}$$

3.5 Alternate Optimization via Relaxation

In this section, we employ a procedure of alternate optimization (Bezdek and Hathaway 2002; Tao et al. 2007) to derive the solution of the optimization problem. To the best of our knowledge, it is difficult to find its optimal solution directly, especially for the weights in (8). To optimize α, we derive a relaxed objective function from the original problem. The output of the relaxed function can ensure that the value of the objective function in (8) is in a small neighborhood of the true minimum.

For a fixed α, finding the optimal projection P is simply reduced to solve the generalized eigenvalue problem

$$KLK\mathbf{p}=\lambda KDK\mathbf{p}, \tag{9}$$

and set $P=[\mathbf{p}_1,\ldots,\mathbf{p}_d]$ corresponds to the smallest d eigenvalues based on the Ky-Fan theorem (Bhatia 1997).

Next, we fix the projection P to update α individually. Without loss of generality, we first consider the condition that $M=2$, i.e., there are only two views. Then the optimization problem (8) is reduced to

$$\arg\min_{P,\alpha} \frac{\mathrm{Tr}(P^TKLKP)}{\mathrm{Tr}(P^TKDKP)},\ \alpha_1+\alpha_2=1,\ \alpha_1,\alpha_2\geq 0. \tag{10}$$

For simplicity, we denote $L_{ijk}=\mathrm{Tr}(P^TK_iL_kK_jP)$ and $D_{ijk}=\mathrm{Tr}(P^TK_iD_kK_jP)$, $i,j,k\in\{1,2\}$. Then we can simply find that $L_{ijk}=L_{jik}$ and $D_{ijk}=D_{jik}$.

With the Cauchy-Schwarz inequality (Hardy et al. 1952), the relaxation for the objective function in (10) is shown in Eq. (11),

$$\begin{aligned}\frac{\mathrm{Tr}(P^TKLKP)}{\mathrm{Tr}(P^TKDKP)} &= \frac{\mathrm{Tr}\Big(P^T(\alpha_1K_1+\alpha_2K_2)(\alpha_1L_1+\alpha_2L_2)(\alpha_1K_1+\alpha_2K_2)P\Big)}{\mathrm{Tr}\Big(P^T(\alpha_1K_1+\alpha_2K_2)(\alpha_1L_1+\alpha_2L_2)(\alpha_1K_1+\alpha_2K_2)P\Big)}\\
&= \frac{\alpha_1^3L_{111}+2\alpha_1^2\alpha_2L_{121}+\alpha_1\alpha_2^2L_{221}+\alpha_1^2\alpha_2L_{112}+2\alpha_1\alpha_2^2L_{122}+\alpha_2^3L_{222}}{\alpha_1^3D_{111}+2\alpha_1^2\alpha_2D_{121}+\alpha_1\alpha_2^2D_{221}+\alpha_1^2\alpha_2D_{112}+2\alpha_1\alpha_2^2D_{122}+\alpha_2^3D_{222}}\\
&\leq \frac{1}{\alpha_1^3L_{111}+2\alpha_1^2\alpha_2L_{121}+\alpha_1\alpha_2^2L_{221}+\alpha_1^2\alpha_2L_{112}+2\alpha_1\alpha_2^2L_{122}+\alpha_2^3L_{222}}\\
&\quad\times\left(\frac{\left(\alpha_1^3L_{111}\right)^2}{\alpha_1^3D_{111}}+\frac{\left(2\alpha_1^2\alpha_2L_{121}\right)^2}{2\alpha_1^2\alpha_2D_{121}}+\frac{\left(\alpha_1\alpha_2^2L_{221}\right)^2}{\alpha_1\alpha_2^2D_{221}}+\frac{\left(\alpha_1^2\alpha_2L_{112}\right)^2}{\alpha_1^2\alpha_2D_{112}}+\frac{\left(2\alpha_1\alpha_2^2L_{122}\right)^2}{2\alpha_1\alpha_2^2D_{122}}+\frac{\left(\alpha_2^3L_{222}\right)^2}{\alpha_2^3D_{222}}\right)\\
&= \frac{1}{\alpha_1^3L_{111}+2\alpha_1^2\alpha_2L_{121}+\alpha_1\alpha_2^2L_{221}+\alpha_1^2\alpha_2L_{112}+2\alpha_1\alpha_2^2L_{122}+\alpha_2^3L_{222}}\\
&\quad\times\left(\alpha_1^3L_{111}\frac{L_{111}}{D_{111}}+2\alpha_1^2\alpha_2L_{121}\frac{L_{121}}{D_{121}}+\alpha_1\alpha_2^2L_{221}\frac{L_{221}}{D_{221}}+\alpha_1^2\alpha_2L_{112}\frac{L_{112}}{D_{112}}+2\alpha_1\alpha_2^2L_{122}\frac{L_{122}}{D_{122}}+\alpha_2^3L_{222}\frac{L_{222}}{D_{222}}\right)\\
&= \sum_{i,j,k\in\{1,2\}} w_{ijk}(\alpha_1,\alpha_2)\frac{L_{ijk}}{D_{ijk}},\end{aligned} \tag{11}$$

where w_{ijk} is the coefficient of $\frac{L_{ijk}}{D_{ijk}}$ and $\sum_{i,j,k\in\{1,2\}} w_{ijk}=1$. In this way, the objective function in (10) is relaxed to a weighted sum of $\frac{L_{ijk}}{D_{ijk}}$. Thus, minimizing the weighted sum of the right-hand-side in (11) can lower the objective function value in (10). Note that

$$\alpha_1^2\alpha_1=\frac{1}{2}\alpha_1\cdot\alpha_1\cdot 2\alpha_2\leq\frac{1}{2}\left(\frac{\alpha_1+\alpha_1+2\alpha_2}{3}\right)^3=\frac{4}{27},$$

and then the weights without containing α_1^3 and α_2^3 are always smaller than a constant. Therefore, we only ensure that a part of the terms in the weighted sum is minimized, i.e., to solve the following optimization problem:

$$\mathop{\arg\min}_{\alpha_1,\alpha_2} w_{111}\frac{L_{111}}{D_{111}} + w_{222}\frac{L_{222}}{D_{222}}. \tag{12}$$

Since w_{111} and w_{222} are the functions of (α_1, α_2), we first find the optimal weights without parameters. To avoid trivial solution, we assign an exponent $r > 1$ on each weight. The relaxed optimization will be

$$\mathop{\arg\min}_{\beta_1,\beta_2} \beta_1^r\frac{L_{111}}{D_{111}} + \beta_2^r\frac{L_{222}}{D_{222}}, \text{ s.t. } \beta_1 + \beta_2 = 1, \beta_1, \beta_2 \geq 0. \tag{13}$$

For (13), we have the Lagrangian function with the Lagrangian multiplier η:

$$L(\beta_1, \beta_2, \eta) = \beta_1^r\frac{L_{111}}{D_{111}} + \beta_2^r\frac{L_{222}}{D_{222}} - \eta(\beta_1 + \beta_2 - 1). \tag{14}$$

We only need to set the derivatives of L with respect to β_1, β_2 and η to zeros as follows:

$$\frac{\partial L}{\partial \beta_1} = r\beta_1^{r-1}\frac{L_{111}}{D_{111}} - \eta = 0, \tag{15}$$

$$\frac{\partial L}{\partial \beta_2} = r\beta_2^{r-1}\frac{L_{222}}{D_{222}} - \eta = 0, \tag{16}$$

$$\frac{\partial L}{\partial \eta} = \beta_1 + \beta_2 - 1 = 0. \tag{17}$$

Then β_1 and β_2 can be calculated by

$$\begin{aligned} \beta_1 &= \frac{(L_{222}D_{111})^{\frac{1}{r-1}}}{(L_{222}D_{111})^{\frac{1}{r-1}} + (L_{111}D_{222})^{\frac{1}{r-1}}}, \\ \beta_2 &= \frac{(L_{111}D_{222})^{\frac{1}{r-1}}}{(L_{222}D_{111})^{\frac{1}{r-1}} + (L_{111}D_{222})^{\frac{1}{r-1}}}. \end{aligned} \tag{18}$$

Having acquired β_1 and β_2, we can obtain α_1 and α_2 by the corresponding relationship between the coefficients of the functions in (12) and (13):

$$\frac{\alpha_1^3 L_{111}}{\alpha_2^3 L_{222}} = \frac{w_{111}}{w_{222}} = \frac{\beta_1^r}{\beta_2^r}. \tag{19}$$

With the constraint $\alpha_1 + \alpha_2 = 1$, we can easily find that

$$\begin{aligned} \alpha_1 &= \frac{\left(\beta_1^r L_{222}\right)^{\frac{1}{3}}}{\left(\beta_1^r L_{222}\right)^{\frac{1}{3}} + \left(\beta_2^r L_{111}\right)^{\frac{1}{3}}}, \\ \alpha_2 &= \frac{\left(\beta_2^r L_{111}\right)^{\frac{1}{3}}}{\left(\beta_1^r L_{222}\right)^{\frac{1}{3}} + \left(\beta_2^r L_{111}\right)^{\frac{1}{3}}}. \end{aligned} \tag{20}$$

Hence, for the general M-view situation, we also have the corresponding relaxed problems:

$$\mathop{\arg\min}_{\sum_{i=1}^M \alpha_i = 1} \sum_{i,j,k\in\{1,\ldots,M\}} w_{ijk}(\alpha_1, \ldots, \alpha_M)\frac{L_{ijk}}{D_{ijk}} \tag{21}$$

and

$$\mathop{\arg\min}_{\beta_1,\ldots,\beta_M} \sum_{i=1}^M \beta_i^r \frac{L_{iii}}{D_{iii}}, \text{ s.t. } \sum_{i=1}^M \beta_i = 1, \ \beta_i \geq 0. \tag{22}$$

The coefficients $(\beta_1, \ldots, \beta_M)$ and $(\alpha_1, \ldots, \alpha_M)$ can be obtained in similar forms:

$$\beta_i = \frac{(D_{iii}/L_{iii})^{\frac{1}{r-1}}}{\sum_{j=1}^M (D_{jjj}/L_{jjj})^{\frac{1}{r-1}}}, \ i = 1, \ldots, M \tag{23}$$

and

$$\alpha_i = \frac{\left(\beta_i^r/L_{iii}\right)^{\frac{1}{3}}}{\sum_{j=1}^M \left(\beta_j^r/L_{jjj}\right)^{\frac{1}{3}}}, \ i = 1, \ldots, M. \tag{24}$$

Although the weight α obtained in the above procedure is not the global minimum, the objective function is ensured in a range of small values. We let F_1 and F_2 be the objective functions in (8) and (21), respectively, and let

$$F_3 = \sum_{i=j=k} w_{ijk}\frac{L_{ijk}}{D_{ijk}} = \sum_{i=1}^M w_{iii}\frac{L_{iii}}{D_{iii}}. \tag{25}$$

We can find that $F_1 \leq F_2$ and if there exists $\alpha_i = 1$ for some i, then $F_1 = F_2 = F_3$. During the alternate procedure, for optimizing P, F_1 is minimized, and for optimizing α, F_3 is minimized. Denote $m_1 = \max(F_1 - F_3)$ and $(P_1, \alpha_1) = \arg\max(F_1 - F_3)$, then we have

$$\begin{aligned} \min F_3 + m_1 &\leq F_3(P_1, \alpha_1) + (F_1 - F_3)(P_1, \alpha_1) \\ &= F_1(P_1, \alpha_1) \leq \max F_1, \end{aligned}$$

and we can define the following nonnegative continuous function:

$$F_4(P, \alpha) = \max\Big(F_1(P, \alpha), \min_\alpha\big(F_3(P, \alpha) + m_1\big)\Big). \tag{26}$$

Note that $\min_\alpha\big(F_3(P, \alpha) + m_1\big)$ is independent of α, thus for any P, there exists α_0, such that $F_1(P, \alpha_0) = \min_\alpha\big(F_3(P, \alpha)+m_1\big)$. If we impose the above alternate optimization on F_4, F_4 is nonincreasing and therefore converges. Though α dose not converge to a certain point, the range of F_1 is reduced to a small district, i.e., smaller than $\min_\alpha F_3$ plus a constant. It is also worthwhile to note that F_3 is actually the weighted sum of the objective functions for preserving each view's locality information. However, the optimization for F_3 still learns information from each view separately, i.e., the locality similarity is not fused. We summarize the KMP in Algorithm 1.

Algorithm 1 Kernelized Multiview Projection

```
Require: The training video sequences $\{v_1, \ldots, v_N\}$ and parameter $r > 1$.
Ensure: The projection matrix $P \in \mathbb{R}^{N \times d}$ and the weights $\alpha = (\alpha_1, \ldots, \alpha_M) \in \mathbb{R}^M$ for kernel matrices.
Extract multiple features from each training video and obtain clean data matrices $X_p^i$, $p = 1, \ldots, N$, $i = 1, \ldots, M$ via the INBF in time-sequential order.
Compute the kernel matrices $K_1, \ldots, K_M \in \mathbb{R}^{N \times N}$ and the Laplacian matrices $L_1, \ldots, L_M \in \mathbb{R}^{N \times N}$ via DTW for $M$ views.
Initialize $\alpha \leftarrow (\frac{1}{M}, \cdots, \frac{1}{M})$;
repeat
    Compute the fused kernel matrix $K = \sum_{i=1}^{M} \alpha_i K_i$ and the fused Laplacian matrix $L = \sum_{i=1}^{M} \alpha_i L_i$;
    Compute $P$ by solving the generalized eigenvalue problem (9);
    Compute coefficients $\beta = (\beta_1, \cdots, \beta_M)$ by Eq. (23);
    Transform $\beta$ to $\alpha$ by Eq. (24);
until $F_4$ defined in Eq. (26) converges.
```

During the testing phase, having acquired the data from each view $X_{test}^1, \cdots, X_{test}^M$ of a test video sequence v_{test}, we first compute the kernel values to form the representation of v_{test} in RKHS of each view:

$$\mathbf{k}_{test}^i = (k_i(v_1, v_{test}), \cdots, k_i(v_N, v_{test})), \; i = 1, \ldots, M,$$

where $k_i(\cdot, \cdot)$ is the kernel function defined in Sect. 3.3. Using the weights $(\alpha_1, \ldots, \alpha_M)$ optimized by Algorithm 1, we have the fused representation of v_{test}: $\mathbf{k}_{test} = \sum_{i=1}^{M} \alpha_i \mathbf{k}_{test}^i$. Then the final fused representation of v_{test} in the reduced space is $\mathbf{y}_{test} = \mathbf{k}_{test} P$.

4 Experiments and Results

In this section, we evaluate our KMP systematically on five action datasets: KTH (Schuldt et al. 2004), UCF YouTube (Liu et al. 2009), UCF Sports (Rodriguez et al. 2008), Hollywood2 (Marszalek et al. 2009) and HMDB51 (Kuehne et al. 2011) respectively. Some representative frames of these datasets are illustrated in Fig. 4. In the rest of this section, we will first introduce the details of the used datasets and their corresponding experimental settings. After that, the compared results will be presented and discussed.

4.1 Datasets

The **KTH** dataset is the benchmark dataset commonly used for action recognition with 599 video clips. Particularly, it contains six different action classes (i.e., boxing, handclapping, handwaving, jogging, running and walking), which are performed by 25 subjects under 4 different scenarios. Following the pre-processing step mentioned in (Yao et al. 2010), the coarse 3D bounding boxes are extracted from all the raw action sequences and further normalized into an equal size of 100 × 100 of each frame. In our experiments, we adopt two usually used settings to compare the final results. The first one is the original experimental setting of the authors, i.e., divide the data into a test set with nine subjects: 2, 3, 5, 6, 7, 8, 9, 10, 22 and the rest form the training set. We finally report the average accuracy over all classes as the performance measure. The other setting is the common leave-one-person-out cross-validation.

The **UCF YouTube** dataset contains 1168 video clips with 11 action categories: *basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog*. We also extract the bounding boxes according to the original paper (Liu et al. 2009). Each frame of the sequences is further normalized into the size of 100 × 100. This dataset is relatively challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, and illumination conditions. Following the original setup in (Liu et al. 2009), a leave-one-out scheme is adopted. The average accuracy over all classes is reported as the final performance.

The **UCF Sports** dataset has 10 classes of human actions with 150 collected broadcast videos. This collection represents a natural pool of actions featured in a wide range of scenes and viewpoints with a large intra-class variability. For this dataset, we use the provided bounding boxes and resize each video frame to a normalized size of 100 × 100. In our experiments, we use a fivefold cross- validation setup mentioned in (Rodriguez et al. 2008), adopting 4/5th of the total number of sequences in each category for training and the rest for testing. The final recognition rate is averaged over the fivefolds.

The **Hollywood2** dataset is a collection of 1707 action samples comprising 12 types of action from 69 different Hollywood movies. For this dataset, we deliberately use the full-sized sequences without any bounding boxes. In our experiments, we use the proposed KMP on a training set of 823 sequences and a test set with 884 sequences following the original setting.

The **HMDB51** dataset contains 6849 realistic action sequences collected from a variety of movies and online videos. Specifically, it has 51 action classes and each has at least 101 positive samples. In our experiments, coarse bounding boxes have been extracted from all the sequences through masks released with the dataset and initialized into the size of 100 × 120 for each frame. We adopt the official setting

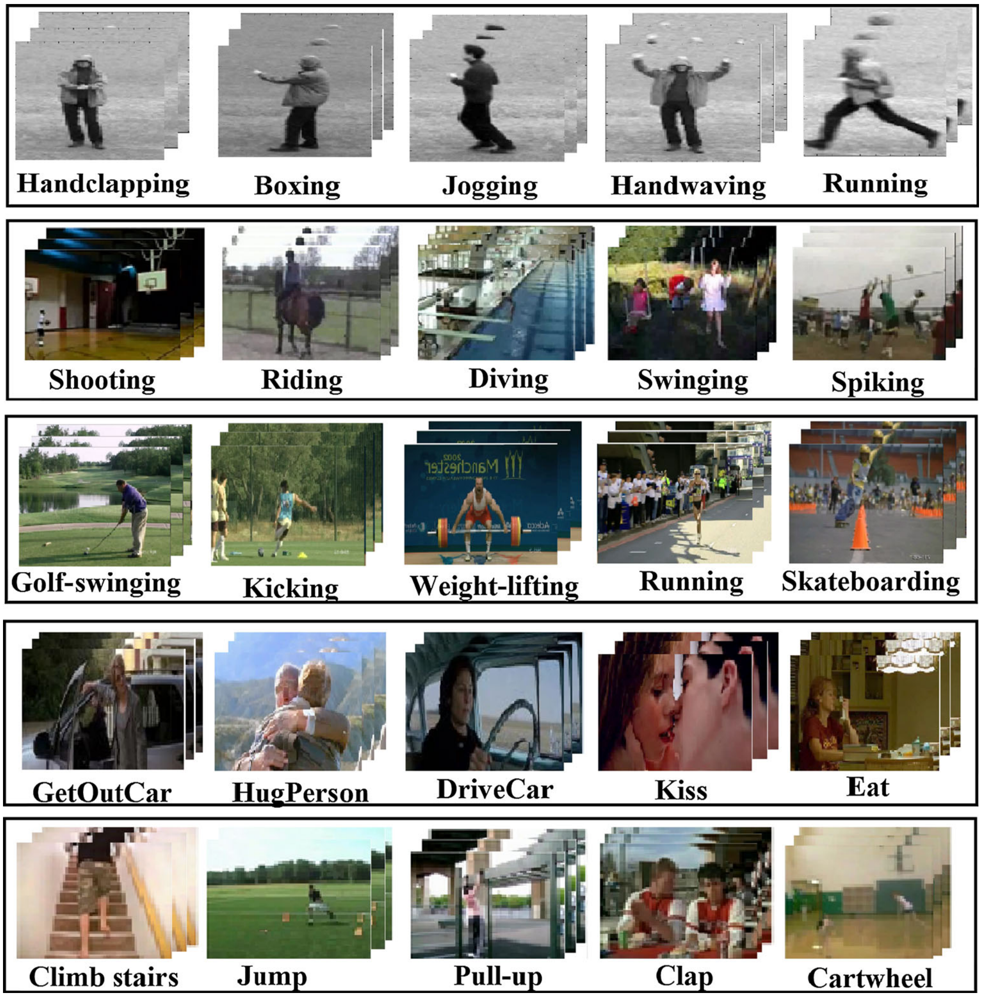


Fig. 4 Some example frames of five datasets: KTH, UCF YouTube, UCF Sports, Hollywood2 and HMDB51 (ordered from the top to the bottom)

of (Kuehne et al. 2011) with three train/test splits. Each split has 70 training and 30 testing clips for each class.

4.2 Multiview Pose Feature Extraction

With the increasing complexity of recognition scenarios, using a single type of feature representation is difficult to satisfy the required accuracies in vision tasks, especially for some realistic applications.

Given a frame containing one pose, we would like to first describe it with multiview informative features. The descriptors are expected to capture the gradient, motion, texture and color information, which are the main cues of a pose. We, therefore, employ the HOF (Laptev et al. 2008), the histogram of oriented gradients (HOG) (Dalal and Triggs 2005), the local binary pattern (LBP) (Ahonen et al. 2004) and color histogram (ColorHist), respectively, for pose representation.

HOF: A fast and effective algorithm to capture the action movement based on the Lucas-Kanade optical flow. Specifically, we calculate HOF between any adjacent frames and each motion region is divided into sub-regions with a 5×5 grid. For each sub-region, a 12-bin histogram is computed to accumulate the motion orientation within 360 degrees. Thus, the length of the final vector of HOF is $5 \times 5 \times 12 = 300$.

HOG: A powerful gradient descriptor. In particular, a 9-bin histogram over [0,180] degrees is computed to accumulate the gradient orientation over a 5×5 cell. The length of the vector is $5 \times 5 \times 9 = 225$.

LBP: LBP features tolerate against illumination changes and are computationally efficient. The operator labels the pixels of an image by thresholding a 3×3 neighborhood of each pixel with the center value and considering the results as a binary number and a 256-bin histogram of the LBP labels computed over a region is used as a texture descriptor.

Note that, all the above three features are extracted on the gray-scale frames.

ColorHist: For each channel of RGB, a 64-bin histogram is used. Thus the final ColorHist has $3 \times 64 = 192$ dimensions.

Table 1 Dimensions of four features for action recognition

Feature representation	Dimension
Histogram of oriented gradients (HOG)	225
Histograms of optical flow (HOF)	300
Local binary pattern (LBP)	256
Color histogram (ColorHist)	192
Total dimension	973

In this way, each pose from a video frame is represented by four different feature views which can describe the thorough information of this frame/pose.

4.3 Compared Methods and Settings

For action recognition, a video sequence can be usually described using differentfeature representations, i.e., multiview representation, in high dimensional feature spaces. In this paper, we adopt four different feature representations (i.e., HOG, HOF, LBP, ColorHist) to describe a video sequence. Table 1 illustrates the original dimensions of these features. We systematically compare our proposed KMP with two related multi-kernel fusion methods. In particular, KMP denotes that the RBF sequential kernels are combined by the proposed method:

$$K = \sum_{i=1}^{M} \alpha_i K_i,$$

where the weight α_i is obtained via alternate optimization. AM indicates that the kernels are combined by arithmetic mean:

$$K_{AM} = \frac{1}{M} \sum_{i=1}^{M} K_i,$$

and GM denotes the combination of kernels through geometric mean:

$$K_{GM} = \left(\prod_{i=1}^{M} K_i \right)^{\frac{1}{M}}.$$

Besides, we also include the best performance of the single-view-based spectral projection (BSP), the average performance of the single-view-based spectral projection (ASP) and concatenation of multiview embeddings in our compared experiments. All of AM, GM , BSP, ASP and multiview embedding concatenation are based on the locality preserving projections (LPP) (He and Niyogi 2004) technique. In addition, two non-linear embedding methods DSE and MSE are adopted in our comparison, as well. In DSE and MSE, the Laplacian embedding (LE) (Belkin and Niyogi 2001) is adopted.

Table 2 Runtime(seconds) of the training and test phases with d = 80 on different datasets

Datasets	Training time (s)	Test time (s)
KTH	460.15	1.89
UCF YouTube	1533.0	4.12
UCF Sports	170.9	1.01
Hollywood2	1220.5	4.03
HMDB51	3250.8	12.95

All of the above methods are evaluated on seven different lengths of codes (20, 30, 40, 50, 60, 70, 80). Under the same experimental setting, all the parameters used in the compared methods have been strictly chosen according to their original papers. For KMP/MSE, the optimal balance parameter r for each dataset is selected from one of {2, 3, 4, 5, 6, 7, 8, 9, 10 } with the step of 1, which yields the best performance by ninefold cross-validation on the training data. The best σ in kernel construction is also selected by the cross-validation on the training data. All experiments are performed using Matlab 2013a on a server configured with a 12-core processor and 128 G of RAM running the Linux OS (Table 2).

4.4 Results

In Table 3, we first illustrate the performance of the single-view representation on all five datasets. In detail, we compute the RBF sequential kernel and weight matrix for a certain single view and input them to our KMP system. Since only a single view is used in KMP, it can be regarded as the procedure of kernelized LPP. From the comparison, we can easily observe that the HOG and HOF features consistently outperform the LBP descriptor in low dimensional feature space. The lowest accuracy is always obtained by ColorHist. Furthermore, we also include the long representation, which is concatenated by all the four low-dimensional feature representations, and the proposed KMP for multiview fusion based reduction into this comparison. It is shown that the concatenated representation can reach better performance than any of the single views, but is significantly lower than our KMP. Specifically, the best accuracies achieved by KMP are 97.5, 87.6, 95.8, 64.3 and 49.8 % on KTH, UCF YouTube, UCF Sport, Hollywood2 and HMDB51, respectively. Additionally, the results of the multiple kernel learning based on SVM (MKL-SVM) (Gönen and Alpaydin 2011) are listed in Table 3 using the same four feature descriptors. The training time and the test time of KMP are listed in Table 2. The runtime of the training phase includes the multiview feature extraction, the INBF process, the construction of kernel matrices via DTW and the optimization of KMP.

Table 3 Performance comparison (%) between the proposed KMP and single feature representations

Accuracy	Dataset				
	KTH	UCF YouTube	UCF sports	Hollywood2	HMDB51
HOG	92.3 (50)	82.6 (70)	91.5 (50)	52.9 (70)	42.3 (50)
HOF	91.6 (70)	81.9 (70)	90.7 (50)	56.7 (70)	39.7 (50)
LBP	80.2 (50)	70.5 (40)	74.6 (30)	32.1 (30)	22.4 (30)
ColorHist	42.7 (20)	31.1 (30)	37.2 (30)	19.4 (20)	18.1 (30)
Concatenation	93.8 (190)	85.4 (210)	93.1 (160)	60.5 (190)	46.0 (160)
MKL-SVM	91.4	82.5	94.3	58.9	47.5
KMP	**97.5** (60)	**87.6**(80)	**95.8** (50)	**64.3** (80)	**49.8** (70)

The numbers in parentheses indicate the dimensions of the representations. For MKL-SVM, DTW is also used to construct the kernel matrix (as illustrated in Fig. 3) for each view and then MKL-SVM is applied to final classification

Bold values indicate highest performance

In Tables 4, 5 and 6, six different multiview embedding schemes are compared with the proposed KMP on the KTH, UCF YouTube and UCF Sports respectively. From the whole tendency, the proposed KMP always leads to the best performance for action recognition. Meanwhile, arithmetic mean (AM) and geometric mean (GM) achieve higher recognition accuracies than the best performance of the single-view-BSP and the ASP. DSE produces worse performance than MSE and sometimes even obtains lower results than AM, but generates better performance than others, since a more meaningful multiview combination scheme is adopted in DSE. Beyond these, it is obviously observed that, with different target dimensions, the final results change a lot. Although both KMP and MSE consider the similarity matrix of each view, KMP maps data into the RKHS which is more suitable for linearly inseparable data in realistic situations. Usually, the best results via KMP appear from d = 50 to d = 80. For instance, the highest accuracy on the KTH dataset is on the dimension of 60 and the best performance on the UCF Sports and UCF YouTube happens when d = 50 and d = 80, respectively (Fig. 5).

Table 4 Performance comparison (%) on the KTH dataset with different feature fusion methods

Dimension	Method						
	Arithmetic mean (AM)	Geometric mean (GM)	BSP	ASP	DSE	MSE	KMP
d = 20	86.8	84.5	85.6	72.4	85.9	86.0	**88.9**
d = 30	88.7	83.6	88.4	74.4	88.0	87.7	**91.4**
d = 40	91.6	86.2	91.0	71.3	89.6	91.7	**93.7**
d = 50	93.0	90.4	92.3	73.6	92.5	93.9	**95.0**
d = 60	93.3	90.7	91.5	75.3	93.8	94.2	**97.5**
d = 70	93.6	92.0	91.8	74.8	93.8	93.5	**96.2**
d = 80	92.5	91.1	92.1	75.0	93.3	93.7	**96.8**

Bold values indicate highest performance

Table 5 Performance comparison (%) on the UCF YouTube dataset with different feature fusion methods

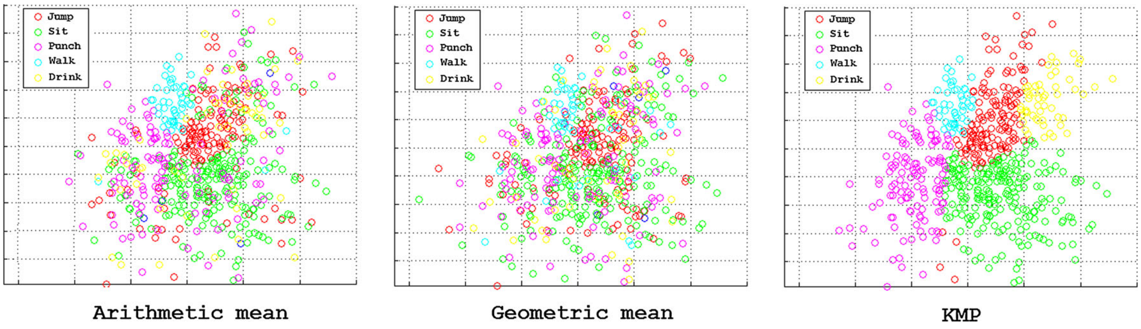
Dimension	Method						
	Arithmetic mean (AM)	Geometric mean (GM)	BSP	ASP	DSE	MSE	KMP
d = 20	72.9	71.8	71.5	58.2	72.1	73.6	**76.0**
d = 30	75.0	74.2	72.8	59.4	74.0	75.2	**78.6**
d = 40	79.5	77.4	77.7	62.5	78.2	80.8	**82.0**
d = 50	82.3	80.8	80.3	61.8	81.3	82.5	**84.2**
d = 60	82.1	81.3	80.9	64.2	81.7	82.5	**85.6**
d = 70	82.9	82.2	82.6	66.0	83.0	83.3	**85.0**
d = 80	84.2	83.0	82.3	66.3	83.5	84.5	**87.6**

Bold values indicate highest performance

Table 6 Performance comparison (%) on the UCF Sports dataset with different feature fusion methods

Dimension	Method						
	Arithmetic mean (AM)	Geometric mean (GM)	BSP	ASP	DSE	MSE	KMP
d = 20	82.8	82.0	81.3	65.2	83.2	86.2	**88.5**
d = 30	87.3	86.5	87.0	68.3	87.5	88.0	**91.6**
d = 40	93.0	92.4	89.6	71.0	93.2	93.0	**94.7**
d = 50	93.0	92.9	91.5	73.4	93.8	**95.8**	**95.8**
d = 60	93.8	92.7	90.8	73.0	94.0	94.5	**95.5**
d = 70	93.2	93.0	91.2	71.7	93.6	**95.1**	94.8
d = 80	92.3	91.6	90.2	72.8	90.7	92.6	**94.3**

Bold values indicate highest performance



Arithmetic mean Geometric mean KMP

Fig. 5 Illustration of low-dimensional distributions of three different multi-kernel fusion schemes (illustrated with data of five actions from the HMDB51 dataset)

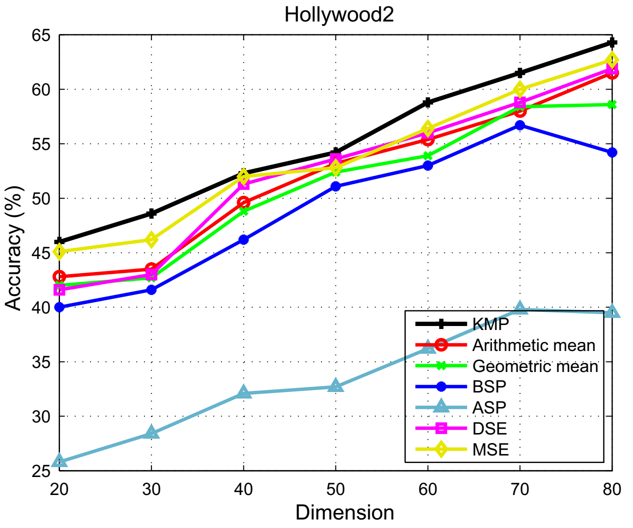


Fig. 6 Performance comparison (%) on the Hollywood2 dataset with different feature fusion methods

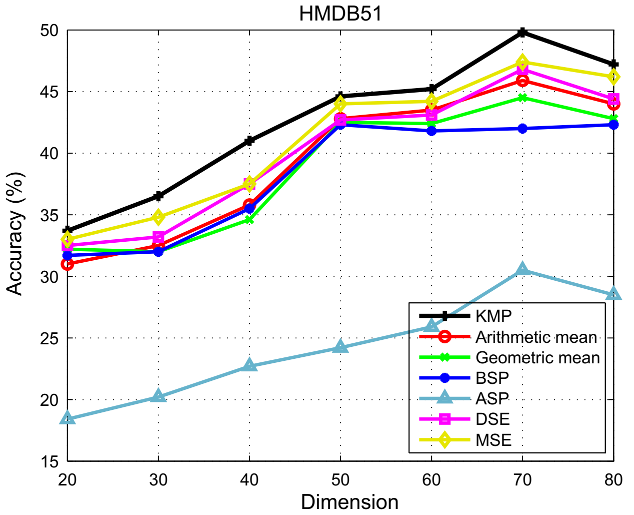


Fig. 7 Performance comparison (%) on the HMDB51 dataset with different feature fusion methods

Similar behaviors can also be seen on the Hollywood2 and HMDB51 datasets. From Fig. 6, we can observe that with the increase of the dimension, all the curves of compared methods on the Hollywood2 dataset are climbing up except for ASP and BSP, both of which have a decrease when the dimension exceeds 70. However, on the HMDB51 dataset, the results in comparison always climb up then go down when the length of dimension increases (see Fig. 7). Besides, from these figures, we can also discover that all the curves have the same tendency of change. All of the above compared methods including MKL-SVM are trained on the same multiview features after INBF.

Furthermore, Table 7 illustrates the performance variation of KMP with respect to the balance parameter r; the

Table 7 Performance (%) of KMP with different r values on the KTH dataset

Dimension	Parameter value								
	r = 2	r = 3	r = 4	r = 5	r = 6	r = 7	r = 8	r = 9	r = 10
d = 20	87.0	87.0	87.5	87.8	**88.9**	88.7	88.0	88.0	87.4
d = 30	89.4	90.1	90.5	91.0	91.3	**91.4**	**91.4**	90.7	89.3
d = 40	87.2	89.0	89.4	91.2	92.0	93.5	93.5	**93.7**	93.2

Bold values indicate highest performance

Table 8 The effectiveness (%) for INBF with $d = 80$ on different datasets

Datasets	KMP with INBF	KMP without INBF
KTH	96.8	95.2
UCF YouTube	87.6	84.8
UCF Sport	94.3	91.5
Hollywood2	64.3	62.2
HMDB51	47.1	45.8

dimensionality of the low-dimensional embedding d is fixed at 20,30 and 40 respectively on the KTH dataset. By adopting the ninefold cross-validation scheme on the training data, it is demonstrated that the higher dimension prefers a larger r in our KMP. Moreover, Fig. 5 shows the low-dimensional (2-dimensional) embeddings obtained by AM, GM and KMP on the HMDB51 dataset. Our proposed KMP can well separate different categories, since it takes the semantically meaningful data structure of different views into consideration for embedding. The effectiveness of the INBF procedure in the training phase is demonstrated in Table 8.

At last, we also compare our results with the state-of-the-art approaches published in major vision conferences and journals in Table 9. In a sense, this kind of comparison is not fair enough, since different features with different methods are applied in different publications. Thus, we only treat this as a general evaluation of recent results. For the four datasets: KTH, UCF YouTube, UCF Sports and Hollywood2, our KMP approach either outperforms state-of-the-art methods or achieves the competitive results compared with published results. For the HMDB51 dataset, the proposed KMP has not shown better results than that reported in Wang and Schmid (2013) and Simonyan and Zisserman (2014) due to the powerful features they introduced, but doubles the result shown in the original paper that introduced this dataset (Kuehne et al. 2011). As a dimensionality reduction method, the proposed KMP can also adopt trajectory-based features or deep-learned features as different views for multiview learning. Considering that our action representation is semi-holistic and does not require an interest points detection phase, the results achieved by KMP are outstanding.

Table 9 Performance comparison (%) of KMP with state-of-the-art methods in the literature

KTH		**UCF YouTube**		**UCF Sports**		**Hollywood2**		**HMDB51**	
Liu et al. (2015)	93.5	Brendel and Todorovic (2010)	77.8	AFMKL Wu et al. (2011)	91.3	Wang et al. (2011)	58.3	Kuehne et al. (2011)	22.8
Schindler and Van Gool (2008)	92.7	Le et al. (2011)	75.8	GMKL Wu et al. (2011)	85.2	Taylor et al. (2010)	46.6	Sapienza et al. (2012)	31.53
Wang et al. (2009)	92.1	Bhattacharya et al. (2011)	76.5	Wang et al. (2011)	88.2	Liu et al. (2009)	53.2	Liu et al. (2013)	36.5
Laptev et al. (2008)	91.8	Sapienza et al. (2012)	80.4	Le et al. (2011)	86.5	Gilbert et al. (2011)	50.9	Jiang et al. (2012)	40.7
Jhuang et al. (2007)	91.7	Wang et al. (2011)	84.2	Kovashka and Grauman (2010)	87.3	Le et al. (2011)	53.3	Wang et al. (2013)	46.6
Klaser and Marszalek (2008)	91.4	Kihl et al. (2015)	87.6	O'Hara and Draper (2012)	91.3	Jiang et al. (2012)	59.5	Wang and Schmid (2013)	**57.2**
Wang et al. (2013)	94.2			Wang et al. (2013)	88.0	Wang et al. (2013)	58.2	Simonyan and Zisserman (2014)	55.4
Kihl et al. (2015)	94.7			Vrigkas et al. (2014)	95.1	Wang and Schmid (2013)	64.3		
				Sun et al. (2014)	86.6	Kihl et al. (2015)	60.2		
Our method	**97.5**	Our method	**87.6**	Our method	**95.8**	Our method	**64.3**	Our method	49.8

Bold values indicate highest performance

5 Conclusion

In this paper, we have presented an effective subspace learning framework based on KMP for action recognition. KMP can encode a variety of features in different ways, to achieve a semantically meaningful embedding. Specifically, KMP is able to successfully explore the complementary property of different views and finally finds a unique low-dimensional subspace where the distribution of each view is sufficiently smooth and discriminative. KMP can be regarded as a fused dimensionality reduction method for multiview data.

We have systematically evaluated our approach on five human action datasets: KTH, UCF YouTube, UCF Sports, Hollywood2 and HMDB51, and the corresponding results show that the proposed approach achieves better or competitive results with state-of-the-art methods. For future work, we plan to combine the current KMP approach with semi-supervised learning for other computer vision tasks.

References

Ahonen, T., Hadid, A., & Pietikäinen, M. (2004). Face recognition with local binary patterns. In *European Conference on Computer Vision*, Prague.

Belkin, M. & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*. New York: MIT Press.

Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. *KDD Workshop*, *10*, 359–370.

Bezdek, J. C. & Hathaway, R. J. (2002). Some notes on alternating optimization. In *AFSS International Conference on Fuzzy Systems*, Calcutta.

Bhatia, R. (1997). *Matrix analysis*. New York: Springer.

Bhattacharya, S., Sukthankar, R., Jin, R., & Shah, M. (2011). A probabilistic representation for efficient large scale visual recognition tasks. In *IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs.

Bickel, S. & Scheffer, T. (2004). Multi-view clustering. In *International Conference on Data Mining*, Brighton.

Bobick, A. F., & Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(3), 257–267.

Brendel, W., & Todorovic, S. (2010). Activities as time series of human postures. In *European Conference on Computer Vision*, Heraklion.

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego.

Dollár, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In: *2nd joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance* (pp. 65–72).

Gilbert, A., Illingworth, J., & Bowden, R. (2011). Action recognition using mined hierarchical compound features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 883–897.

Gönen, M., & Alpaydin, E. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, *12*, 2211–2268.

Hardy, G. H., Littlewood, J. E., & Pólya, G. (1952). *Inequalities*. Cambridge: Cambridge University Press.

He, X. & Niyogi, P. (2004). Locality preserving projections. In *Advances in Neural Information Processing Systems*, New York.

Jhuang, H., Serre, T., Wolf, L., & Poggio, T. (2007). A biologically inspired system for action recognition. In *International Conference on Computer Vision*. New York: IEEE Press.

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3d Convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231.

Jiang, Y. G., Dai, Q., Xue, X., Liu, W., & Ngo, C. W. (2012). Trajectory-based modeling of human actions with motion reference points. In *European Conference on Computer Vision*, Florence.

Kihl, O., Picard, D., & Gosselin, P. H. (2015). A unified framework for local visual descriptors evaluation. *Pattern Recognition*, *48*(4), 1174–1184.

Klaser, A. & Marszalek, M. (2008). A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, Leeds.

Kovashka, A. & Grauman, K. (2010). Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, Singapore.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). Hmdb: A large video database for human motion recognition. In: IEEE International Conference on Computer Vision.

Laptev, I., Marszalek, M., Schmid, C., & Rozenfeld, B. (2008). Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

Le, Q. V., Zou, W. Y., Yeung, S. Y., & Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

Liu, J., Luo, J., & Shah, M. (2009). Recognizing realistic actions from videos in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, Leeds.

Liu, L., Shao, L., Li, X., & Lu, K. (2015). Learning spatio-temporal representations for action recognition: A genetic programming approach. *IEEE Transactions on Cybernetics*. doi:10.1109/TCYB.2015.2399172.

Liu, L., Shao, L., & Rockett, P. (2013). Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition. *Pattern Recognition*, *46*(7), 1810–1818.

Long, B., Philip, S. Y., & Zhang, Z. M. (2008). A general model for multiple view unsupervised learning. In *International Conference on Data Mining*, New York.

Marszalek, M., Laptev, I., & Schmid, C. (2009). Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

O'Hara, S. & Draper, B. A. (2012). Scalable action recognition with a subspace forest. In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

Rodriguez, M. D., Ahmed, J., & Shah, M. (2008). Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

Sapienza, M., Cuzzolin, F., & Torr, P. H. (2012). Learning discriminative space-time actions from weakly labelled videos. In *British Machine Vision Conference*, San Diego.

Schindler, K. & Van Gool, L. (2008). Action snippets: How many frames does human action recognition require? In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: A local SVM approach. *International Conference on Pattern Recognition*, *3*, 32–36.

Scovanner, P., Ali, S., & Shah, M. (2007). A 3-dimensional SIFT descriptor and its application to action recognition. In *International Conference on Multimedia*, Bulgaria.

Shao, L., Zhen, X., Tao, D., & Li, X. (2014). Spatio-temporal laplacian pyramid coding for action recognition. *IEEE Transactions on Cybernetics*, *44*(6), 817–827.

Simonyan, K. & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems* (pp. 568–576).

Sun, L., Jia, K., Chan, T. H., Fang, Y., Wang, G., & Yan, S. (2014). Dl-sfa: Deeply-learned slow feature analysis for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2625–2632).

Tao, D., Li, X., Wu, X., & Maybank, S. J. (2007). General tensor discriminant analysis and gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(10), 1700–1715.

Taylor, G., Fergus, R., LeCun, Y., & Bregler, C. (2010). Convolutional learning of spatio-temporal features. In *European Conference on Computer Vision*.

Vrigkas, M., Karavasilis, V., Nikou, C., & Kakadiaris, I. A. (2014). Matching mixtures of curves for human action recognition. *Computer Vision and Image Understanding*, *119*, 27–40.

Wang, H., Klaser, A., Schmid, C., & Liu, C.L. (2011). Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

Wang, H., Kläser, A., Schmid, C., & Liu, C. L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, *103*(1), 60–79.

Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, San Diego.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., & Schmid, C., et al. (2009). Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, New York.

Wu, X., Xu, D., Duan, L., & Luo, J. (2011). Action recognition using context and appearance distribution features. In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

Xia, T., Tao, D., Mei, T., & Zhang, Y. (2010). Multiview spectral embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *40*(6), 1438–1446.

Xu, C., Tao, D., & Xu, C. (2014). Large-margin multi-viewinformation bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(8), 1559–1572.

Xu, C., Tao, D., & Xu, C. (2015). Multi-view intact space learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. doi:10.1109/TPAMI.2015.2417578.

Yao, A., Gall, J., & Van Gool, L. (2010). A hough transform-based voting framework for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

Yu, M., Liu, L., & Shao, L. (2015). Kernelized multiview projection. arXiv:1508.00430.

Zhang, Z., & Tao, D. (2012). Slow feature analysis for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(3), 436–450.

Zhao, Z. & Liu, H. (2008). Multi-source feature selection via geometry-dependent covariance analysis. In *Third Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery* (pp. 36–47).

Zien, A. & Ong, C.S. (2007). Multiclass multiple kernel learning. In *International Conference on Machine Learning*, New York.