Cross-View Action Recognition via a Continuous Virtual Path

Zhong Zhang, Chunheng Wang, Baihua Xiao, Wen Zhou, Shuang Liu and Cunzhao Shi
State Key Laboratory of Management and Control for Complex Systems, CASIA
{zhong.zhang,chunheng.wang,baihua.xiao,wen.zhou,shuang.liu,cunzhao.shi}@ia.ac.cn

Abstract

In this paper, we propose a novel method for cross-view action recognition via a continuous virtual path which connects the source view and the target view. Each point on this virtual path is a virtual view which is obtained by a linear transformation of the action descriptor. All the virtual views are concatenated into an infinite-dimensional feature to characterize continuous changes from the source to the target view. However, these infinite-dimensional features cannot be used directly. Thus, we propose a virtual view kernel to compute the value of similarity between two infinite-dimensional features, which can be readily used to construct any kernelized classifiers. In addition, there are a lot of unlabeled samples from the target view, which can be utilized to improve the performance of classifiers. Thus, we present a constraint strategy to explore the information contained in the unlabeled samples. The rationality behind the constraint is that any action video belongs to only one class. Our method is verified on the IXMAS dataset, and the experimental results demonstrate that our method achieves better performance than the state-of-the-art methods.

1. Introduction

Recognizing human actions from videos play a key role in computer vision and pattern recognition due to its wide and significant applications. The importance is strongly driven by the need for human computer interaction, video surveillance and multimedia retrieval. Recently, in the field of action representation, several strategies have been proposed by researchers to make action representation more discriminative, such as space-time pattern templates [28], 2D shape matching [16, 19, 27], optical flow patterns [5], trajectory-based representation [22], and spatio-temporal interest points [4, 18]. Especially, methods based on spatio-temporal interest points together with bag-of-words model have shown promising performance. Since these approaches do not rely on preprocessing techniques, e.g. background modeling or body-part tracking, they are relatively robust to noise, background changing and illumination variation.

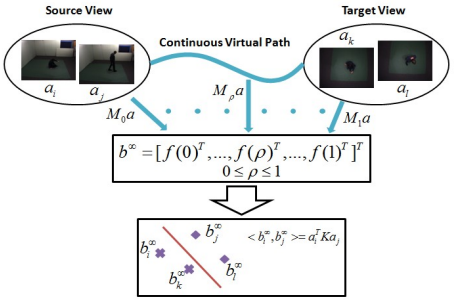

Figure 1. View knowledge transfer via a continuous virtual path. Action feature **a** is projected to form the virtual view f_ρ ($0 \le \rho \le 1$) on the continuous virtual path by transformation matrix M_ρ, and then all the virtual views are concatenated to form an infinite-dimensional feature $\mathbf{b}^\infty$. Inner product between them defines our virtual view kernel which can be computed in a close-form. The virtual view kernel can be readily used to construct any kernelized classifiers.

Furthermore, some other methods [13, 29, 30] derive from this model, which exploits the spatial and temporal contexts as another type of information for describing interest points. These approaches are effective for recognizing actions observed from similar viewpoints, but their performance tends to degrade sharply when the viewpoint changes significantly. This is because the same action looks very different when observed from different views. Hence, action models learned using labeled samples in one view are less discriminative for recognizing actions in a different view. The intuitional approach of training a separate classifier for each view may be impractical owing to lack of labeled samples.

In this paper, we present a novel kernel-based approach for cross-view action recognition via a continuous virtual path. Imagine there is a virtual path connecting the source view and the target view, and each point on the virtual path refers to a virtual view. An action feature is transformed to a virtual view on the virtual path by a particular class of linear projections. Then, all the virtual views are integrated into an infinite-dimensional feature. Since the infinite-dimensional feature contains all the virtual views from the source to the target view, it is robust to view changes. As

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.347

these infinite-dimensional features cannot be used directly, we propose a virtual view kernel to measure the similarity between two infinite-dimensional features. Concretely, it is defined by inner produce between two infinite-dimensional features. The main idea is illustrated in Fig. 1. In addition, we solve the virtual view kernel under an information theoretic framework that allows maximizing discrimination among action classes. Our approach can deal with three working modes as [15]. The first one is *correspondence mode*. Like the approaches in [7, 15, 17], an unlabeled action sample observed simultaneously in both views yields a corresponding pair, so that these pairs can be used in the training stage. Besides, our approach can handle the situation usually considered by the domain adaptation, transfer learning, and covariate shift [1, 2, 8, 9]. There are two settings for domain adaptation. One is semi-supervised domain adaptation, where the target view contains a small amount of labeled samples without corresponding pairs. We refer to this working mode as *partially labeled mode*. The other is unsupervised domain adaptation where the target view is completely unlabeled, which is referred as *unlabeled mode*.

It can be seen that there are several unlabeled samples from the target view in the above three modes, and it is usually insufficient to construct a good classifier only using labeled samples or corresponding pairs. Hence, how to effectively utilize unlabeled samples from the target view is key to cross-view action recognition. In this paper, the information contained in these unlabeled samples is explored by a constraint strategy, which is based on the rationality that any sample belongs to only one class. The experimental results demonstrate that our approach outperforms the state-of-the-art approaches in all the three modes on the IXMAS dataset.

1.1. Related Work

A lot of approaches have been proposed to address the problem of cross-view action recognition. Some of these approaches rely on geometric constraints [28], body joints detection and tracking [20, 21], and 3D models [14, 26]. For example, Rao *et al.* [21] presented a view-invariant representation of human action to capture the dramatic changes in the speed and direction of the trajectory using spatio-temporal curvature of 2D trajectory. However, this kind of approaches require some challenging techniques, such as body joints detection and tracking, or alignment between views. Junejo *et al.* [11, 12] introduced a temporal self-similarity matrix, which was view stable.

Recently, transfer learning approaches are employed to address cross-view action recognition. Farhadi *et al.* [6] generated split-based features in the source view using Maximum Margin Clustering and then transferred the split values to the corresponding frames in the target view. Liu *et al.* [17] learned a cross-view bag of "bilingual words" using corresponding pairs. Then, the action videos are represented by "bilingual words" in both views. Li *et al*'s work [15], which also explored the idea of using virtual view to overcome the problem of view changes, is close to ours. However, there are two significant differences. One difference is that their work only samples several virtual views, while our kernel-based method utilizes all the virtual views on the virtual path. This can keep all the visual information on the virtual path and eliminate the requirement to tune the parameter needed in [15]. The other is that their work only uses labeled samples or corresponding pairs to train the model, while our method makes full use of unlabeled samples from target view as well.

2. Approach

In this section, we start by reviewing the method which obtains multiple virtual views by sampling virtual path. Second, we present our virtual view kernel. Third, we formulate the problem under an information theoretic framework so as to maximize discrimination among action classes. Fourth, we present a constraint on unlabeled samples, then the optimization procedure is given. Finally, we introduce the implementation details and extensions.

2.1. Multiple Virtual Views by Sampling Virtual Path

Imagine that there is a virtual path $V(\rho), \rho \in [0, 1]$ connecting the source view V_S and the target view V_T, where $V(0) = V_S$ and $V(1) = V_T$ [15]. A particular class of linear projections is adopted to transform the action features on the virtual path $V(\rho)$. Let $f(\rho) = M_\rho \mathbf{a}$ be a virtual view on the virtual path $V(\rho)$, where M_ρ is a transformation matrix and $\mathbf{a} \in \mathbb{R}^{D\times 1}$ is an action feature vector. In the special case, $f_S = f(0) = M_S^T \mathbf{a}$ and $f_T = f(1) = M_T^T \mathbf{a}$ are the source virtual view and target virtual view respectively, corresponding to the two endpoints of the virtual path. Here M_S and M_T are both $D \times d$ transformation matrices satisfying $M_S^T M_S = I$ and $M_T^T M_T = I$, i.e. they both have orthogonal columns of unit-length.

The task is to compute the virtual view $f(\rho)$ on the virtual path, i.e. M_ρ, when the source and target transformation matrixes M_S and M_T have been given. The columns of M_S and M_T are of the unit length and therefore lie on a hyper-sphere. It can be seen that each column of M_ρ is a point on a segment line where the corresponding columns of M_S and M_T are the two endpoints. Concretely, the transformation matrix $M_\rho = [m_{\rho,1}, m_{\rho,2}, ..., m_{\rho,d}]$ is computed from $M_S = [m_{S,1}, m_{S,2}, ..., m_{S,d}]$ and $M_T = [m_{T,1}, m_{T,2}, ..., m_{T,d}]$ as [15, 23]:

$$m_{\rho,i} = \frac{(1-\rho)m_{S,i} + \rho m_{T,i}}{[\rho^2 + (1-\rho)^2 + 2\rho(1-\rho)m_{S,i}^T m_{T,i}]^{1/2}}. \quad (1)$$

We can see that the above equation is actually a geodesic between $m_{S,i}$ and $m_{T,i}$. The radial projection to unit norm of the straight line joining the two points is the geodesic between the two points [23].

Multiple virtual views are sampled on the virtual path $V(\rho)$ at L intervals $\rho_1, \rho_2, ..., \rho_L$ ($0 < \rho_1 < \rho_2 < ... < \rho_L < 1$) [15]. After that the transformation matrix M_{ρ_i} of the corresponding virtual view f_{ρ_i} can be obtained from Eq. (1). The final representation of an action video is simply obtained by concatenating the transformation features $M_{\rho_i}^T \mathbf{a}$ into a single long feature vector:

$$\hat{\mathbf{a}} = [(M_S^T \mathbf{a})^T, (M_{\rho_1}^T \mathbf{a})^T, ..., (M_{\rho_L}^T \mathbf{a})^T, (M_T^T \mathbf{a})^T]^T. \quad (2)$$

The final feature vector implicitly incorporates multiple virtual view transformations, which change from the source view to the target view. Hence, a classifier trained by these feature vectors can be robust to view changes.

2.2. Virtual View Kernel

The above method, however, only samples several virtual views on the virtual path, it neglects the information provided by the other virtual views. Furthermore, this method has to adopt cross-validation to determine the parameter L (the number of virtual views sampling on the virtual path). Intuitively, if we utilize all the virtual views on the virtual path, the above drawbacks can be naturally overcome. Nevertheless, when the original feature projects to all the virtual views, it changes to an infinite-dimensional feature. Thus we cannot use this representation directly. In this work, we propose a kernel-based method to measure the similarity between two infinite-dimensional features. The proposed virtual view kernel is expected to be the measurement of similarity that is robust to the viewpoint changes. In other word, although actions belong to the same class are observed from different views, the values of similarity computed by virtual view kernel are high enough to classify. Concretely, we employ the inner products to construct a linear kernel, which can be readily used to construct any kernelized classifiers. We next show that our kernel-based method don't need to compute and store all the virtual views.

Given two original D-dimensional feature vectors $\mathbf{a}_i$ and $\mathbf{a}_j$, we compute their virtual views $f(\rho)$ for a continuous ρ from 0 to 1, and then concatenate all the virtual views into infinite-dimensional feature vectors $\mathbf{b}_i^\infty$ and $\mathbf{b}_j^\infty$. The proposed virtual view kernel is defined as the inner product between them:

$$< \mathbf{b}_i^\infty, \mathbf{b}_j^\infty >= \int_0^1 (M_\rho^T \mathbf{a}_i)^T (M_\rho^T \mathbf{a}_j) d\rho = \mathbf{a}_i^T K \mathbf{a}_j, \quad (3)$$

where

$$K = \int_0^1 M_\rho M_\rho^T d\rho = \sum_{n=1}^{d} \int_0^1 m_{\rho,n} m_{\rho,n}^T d\rho. \quad (4)$$

The Eq. (3) is the "kernel trick", where a kernel function induces inner products between infinite-dimensional features. $K \in \mathbb{R}^{D \times D}$ is defined as the virtual view kernel, which can measure the similarity between two feature vectors. The more similar are the two feature vectors, the larger the inner product value is, otherwise the smaller the value is.

The matrix K can be computed by substituting Eq. (1) into Eq. (4):

$$\begin{aligned} K = &\sum_{n=1}^{d} A_n \cdot m_{S,n} m_{S,n}^T + B_n \cdot m_{T,n} m_{T,n}^T \\ &+ C_n \cdot (m_{S,n} m_{T,n}^T + m_{T,n} m_{S,n}^T), \end{aligned} \quad (5)$$

where

$$\begin{aligned} A_n &= \frac{1}{2(1-t_n)} + \frac{t_n^2}{t_n - 1} \cdot \frac{\theta_n}{(1-t_n^2)^{1/2}}; \\ B_n &= \frac{1}{2(1-t_n)} - t_n \cdot \frac{\theta_n}{(1-t_n^2)^{1/2}}; \\ C_n &= \frac{1}{2(1-t_n)} (\frac{\theta_n}{(1-t_n^2)^{1/2}} - 1). \end{aligned} \quad (6)$$

Here $t_n = \cos\theta_n = m_{S,n}^T m_{T,n}$, and $m_{S,n}, m_{T,n} \in R^{D \times 1}$ are the n-th columns of M_S and M_T. From Eq. (5), we can see that our virtual view kernel has a closed-form solution.

Since our method utilizes all the virtual views on the virtual path, it not only takes full advantage of the visual information provided by the continuous virtual path, but also saves the cost of tuning the parameter L (the number of virtual views sampling on the virtual path).

2.3. Maximizing Discrimination

In this subsection, we discuss the problem of choosing discriminative values for M_S and M_T, because our virtual view kernel K are totally confirmed by transformation matrices M_S and M_T. For convenience, we first discuss a two-class problem, and the multi-class problem can be handled as a set of two-class problems using one versus all approach.

In the unlabeled mode, all the labeled training samples are from the source view. In the partially labeled mode, only a part of samples from the target view are labeled as training data. In the above two cases, we desire to maximize discrimination between the two classes using all the available labeled samples. Concretely, the values of similarity between different class samples are forced to be different from the values between the same class samples. To this end, with the help of mutual information, the problem can be formulated by:

$$\max_{M_S, M_T} I(V; c). \quad (7)$$

where $V = \{\mathbf{a}_i^T K \mathbf{a}_j\}$, $\mathbf{a}_i$ and $\mathbf{a}_j$ are the labeled feature vectors, and I is the mutual information which measures

the degree of dependence between two random variables. When $\mathbf{a}_i$ and $\mathbf{a}_j$ are from the same class (positive pair), c is set to 1; otherwise c is set to 0 for negative pair. Eq. (7) can be rewritten on the basis of the differential entropy H:

$$\begin{aligned} I(V;c) &= H(V) - H(V|c) \\ &= H(V) - P(c=1)H(V_P) - P(c=0)H(V_N). \end{aligned} \tag{8}$$

where V_P and V_N are the set of similarity values computed from positive and negative pairs respectively. To solve the Eq. (8), we approximate the differential entropy H using V, i.e. a set of $\mathbf{a}_i^T K \mathbf{a}_j$. In the assumption of V drawn from a 1-dimensional Gaussian distribution, we obtain $H(V) = \frac{1}{2}[1 + \ln(2\pi\sigma)]$, in which the variance σ can be estimated from V. In addition, we assume that the prior probabilities for the two classes are equal, so Eq. (8) can be approximated by:

$$I(V;c) \approx \ln\sigma - \frac{1}{2}\ln\sigma_P - \frac{1}{2}\ln\sigma_N, \tag{9}$$

where σ, σ_P and σ_N are variances computed from V, V_P and V_N respectively.

In the case of correspondence mode, we also solve the problem under the information theoretic framework. Let $\{\mathbf{a}_{Sl}, \mathbf{a}_{Tl}\}_{l=1}^{n_c}$ denote the corresponding sample pairs. These pairs are unlabeled, but belong to the same class observed simultaneously in the source view and target view. Since $\mathbf{a}_S$ and $\mathbf{a}_T$ describe the same action class, we expect that they have high value of similarity, i.e. $\max \mathbf{a}_S^T K \mathbf{a}_T$. Let δ denote $\mathbf{a}_S^T K \mathbf{a}_T$. For computational convenience, we change the equation to its equivalent form:

$$\min_{M_S, M_T} g(\delta) = 1 - \frac{1}{1 + e^{-|\delta|}}, \tag{10}$$

where $g(\delta) \in (0, 0.5]$. We add a penalty term $H(\{g(\delta), -g(\delta)\})$ to Eq. (7):

$$\max_{M_S, M_T} I(V;c) - \alpha H(\{g(\delta), -g(\delta)\}), \tag{11}$$

where α is a parameter which controls the importance of corresponding pairs. If we assume $\delta = \mathbf{a}_S^T K \mathbf{a}_T$ are drawn from a 1-dimensional Gaussian distribution, $\{g(\delta), -g(\delta)\}$ could be a Gaussian distribution with zero mean. We want to minimize $H(\{g(\delta), -g(\delta)\})$ for two purposes. First, we expect the distribution of $\{g(\delta), -g(\delta)\}$ to be compact. Second, we hope it is close to the origin so as to maximize $\delta = \mathbf{a}_S^T K \mathbf{a}_T$. As before, we also approximate the mutual information and differential entropy in terms of variances. The objective in Eq. (11) is reformulated by:

$$\ln\sigma - \frac{1}{2}\ln\sigma_P - \frac{1}{2}\ln\sigma_N - \alpha\ln\sigma_\delta, \tag{12}$$

where σ_δ is the correlation coefficient for $\{g(\delta), -g(\delta)\}$, not the variance. A minimization of $\ln\sigma_\delta$ will yield $g(\delta)$ concentrating around 0, by which we make the value of similarity between the pair $(\mathbf{a}_S, \mathbf{a}_T)$ greater.

2.4. Constraint on Unlabeled Samples

The above three modes only utilize the labeled samples or corresponding pairs, yet these samples may be insufficient to construct a good classifier. Thus, how to effectively leverage unlabeled samples from the target view is crucial to cross-view action recognition. In this work, we impose a constraint on the unlabeled samples from the target view. The constraint is based on the fact that any sample belongs to only one class. For the two-class problem, the constraint is equivalent to maximize the absolute value of following formula:

$$\gamma = \mathbf{a}_P^T K \mathbf{a}_u - \mathbf{a}_N^T K \mathbf{a}_u, \tag{13}$$

where $\mathbf{a}_u$, $\mathbf{a}_P$ and $\mathbf{a}_N$ are the unlabeled feature vectors from the target view, positive feature vectors and negative feature vectors respectively. Note that $\mathbf{a}_P$ and $\mathbf{a}_N$ could be either from the source view or target view because virtual view kernel K is robust to view changes. For computational convenience, we also change this equation according to Eq. (10), i.e. $g(\gamma)$. This constraint can be further written within the information theoretic framework:

$$\min_{M_S, M_T} H(\{g(\gamma), -g(\gamma)\}). \tag{14}$$

As previous, since we expect Eq. (14) to be not only compactly distributed but also close to the origin, the distribution of $\gamma = \mathbf{a}_P^T K \mathbf{a}_u - \mathbf{a}_N^T K \mathbf{a}_u$ is assumed to be subject to Gaussian distribution. With this constraint, the objective Eq. (11) is rewritten as:

$$\max_{M_S, M_T} I(V;c) - \alpha H(\{g(\delta), -g(\delta)\}) - \beta H(\{g(\gamma), -g(\gamma)\}), \tag{15}$$

where β is a parameter which controls the importance of unlabeled samples. When $\alpha = 0$, it is the unlabeled mode or partially labeled mode with the constraint:

$$\max_{M_S, M_T} I(V;c) - \beta H(\{g(\gamma), -g(\gamma)\}). \tag{16}$$

The Eq. (15) is approximated by:

$$\ln\sigma - \frac{1}{2}\ln\sigma_P - \frac{1}{2}\ln\sigma_N - \alpha\ln\sigma_\delta - \beta\ln\sigma_\gamma, \tag{17}$$

where σ_γ is the correlation coefficient for $\{g(\gamma), -g(\gamma)\}$. When $\alpha = 0$, this equation could be used to approximate Eq. (16).

2.5. Optimization Algorithm

In this section, we present the optimization algorithm to solve the objective Eq. (8), (11), (15) and (16). We denote the objective functions as $W(M_S, M_T)$ in the following discussion. We adopt a greedy axis-rotating approach

that iteratively searches for transformations (M_S, M_T) that maximize W [15]. Specifically, we seek matrices $R_S(t)$ and $R_T(t) \in \mathbf{SO}(D)$, so that the estimate at step t is $M_S(t) = R_S(t)M_S(t-1)$ and $M_T(t) = R_T(t)M_T(t-1)$, where $\mathbf{SO}(D)$ is the D-dimensional special orthogonal group. Here $\mathbf{SO}(D)$ corresponds to a set of rotation operations in $\mathbb{R}^D$, so the resulting $M_S(t), M_T(t)$ will be orthonormal matrixes as well. Essentially, we find a pair of $R_S(t)$ and $R_T(t)$ to provide a steep ascent in W. According to the Lie algebra, the optimal rotation direction for M_S and M_T can be found by:

$$\begin{aligned} R_{S,n} &= exp(n\mu \sum_{i,j} c_{i,j}(E_{i,j} - E_{j,i})), \\ R_{T,n} &= exp(n\mu \sum_{k,l} c_{k,l}(E_{k,l} - E_{l,k})), \end{aligned} \tag{18}$$

where $2 \leq i, k \leq D$, $i+1 \leq j \leq D$ and $k+1 \leq l \leq D$. Here μ is step length, n is the step number for searching optimal rotation direction, and $E_{i,j}$ is a matrix whose (i,j)-th element is one and all others are zero. In addition, $c_{i,j}$ and $c_{k,l}$ can be obtained by:

$$\begin{aligned} c_{i,j} &= \Delta W_{S,i,j} / (\sum_{i,j} \Delta W_{S,i,j}^2 + \sum_{k,l} \Delta W_{T,k,l}^2)^{1/2}; \\ c_{k,l} &= \Delta W_{T,k,l} / (\sum_{i,j} \Delta W_{S,i,j}^2 + \sum_{k,l} \Delta W_{T,k,l}^2)^{1/2}. \end{aligned} \tag{19}$$

The $\Delta W_{S,i,j}$ can be approximated by:

$$\begin{aligned} \Delta W_{S,i,j} = &\{W(R_{S,i,j}M_S(t-1), M_T(t-1)) - \\ &W(M_S(t-1), M_T(t-1))\}/\epsilon, \end{aligned} \tag{20}$$

Analogously, $\Delta W_{T,k,l}$ is approximated by:

$$\begin{aligned} \Delta W_{T,k,l} = &\{W(M_S(t-1), R_{T,k,l}M_T(t-1)) - \\ &W(M_S(t-1), M_T(t-1))\}/\epsilon, \end{aligned} \tag{21}$$

where ϵ is a small positive number.

$$\begin{aligned} R_{S,i,j} &= exp(\epsilon(E_{i,j} - E_{j,i})); \\ R_{T,k,l} &= exp(\epsilon(E_{k,l} - E_{l,k})). \end{aligned} \tag{22}$$

The iterative algorithm terminates when $M_S(t) = M_S(t-1)$, and $M_T(t) = M_T(t-1)$. The above process is illustrated in Algorithm 1. The mathematical principle behind this algorithm and the details can be found in [10, 15].

Algorithm 1: Greedy Axis Rotation

Input: $M_S(0)$, $M_T(0)$, $\epsilon > 0$, $\delta > 0$
Output: $M_S(t)$, $M_T(t)$
Initialize the initialization of $M_S(0)$, $M_T(0)$ is described in Section 2.6;

while 1 **do**
  If $M_S(t) = M_S(t-1)$, and $M_T(t) = M_T(t-1)$ break;
  1. For all the i, j, k and l, calculate:
    1) $R_{S,i,j}$ and $R_{T,k,l}$ according to Eq. (22)
    2) $\Delta W_{S,i,j}$ and $\Delta W_{T,k,l}$ according to Eq. (20), (21)
    3) $c_{i,j}$ and $c_{k,l}$ according to Eq. (19)
  2. The optimal rotation direction $R_{S,n}$ and $R_{T,n}$ can be computed by Eq (18), where $n^* = \arg\max_n W(R_{S,n}M_S(t-1), R_{T,n}M_T(t-1))$
  3. $R_S(t) = R_{S,n^*}$ and $R_T(t) = R_{T,n^*}$
  4. $M_S(t) = R_S(t)M_S(t-1)$ and $M_T(t) = R_T(t)M_T(t-1)$
end

2.6. Implementation Details and Extensions

Before training our model, we should determine the working mode and extract the corresponding single-view action feature vector from each training action video. Once the virtual view kernel K is trained, we compute the values of similarity between any two training samples. In our experiments, we use SVM [3] as classifier.

It is important to choose good initializations for M_S and M_T. We employ the basis of principal subspaces of the source and target samples as the initializations of M_S and M_T respectively. For a Q-class action recognition problem, we learn Q binary one-against-all models as described above. The final classification is determined by selecting the model whose SVM outputs the maximum response. If we have G source views, we simply sum the response values from the G classifiers, and then make a binary decision with the threshold at 0. For a Q-class, G source views problem, we select the class which achieves the maximum sum of response values.

3. Experimental Results

3.1. Dataset and Low-level Feature Extraction

We test our approach on the IXMAS multi-view action dataset [26], which contains eleven daily-life actions, such as check watch, punch, and turn around. Each action is performed three times by twelve actors and observed from five different views including four side views and one top view.

For fair comparison, we extract the same low-level action descriptors as [17, 15]. Concretely, we first extract the local feature, i.e. the spatio-temporal interest points proposed in [4]. To detect the interest points, a 2D Gaussian filter and then a 1D-Gabor filter are applied to an action video, and the interest points are detected at the local maximum response. We extract up to 200 cuboids from each action video. Each cuboid is represented by a 100-dimensional descriptor learned by PCA. These descriptors are further quantized to 1000 codewords by k-means clustering and each action video is represented by a histogram

Table 1. Cross-view recognition accuracy on the IXMAS dataset in correspondence mode. Each row is a source view and each column a target view. The four accuracy numbers in a tuple are the average recognition accuracy of [17], [15], VVK, and VVKC respectively.

%	C0	C1	C2	C3	C4
C0		(79.9, 81.8, 84.5, **86.3**)	(76.8, 88.1, 90.6, **93.1**)	(76.8, 87.5, 90.6, **91.5**)	(74.8, 81.4, 83.8, **85.4**)
C1	(81.2, 87.5, 88.3, **90.5**)		(75.8, 82.0, 85.3, **87.8**)	(78.0, **92.3**, 90.2, 91.3)	(70.4, 74.2, 79.7, **83.4**)
C2	(79.6, 85.3, 88.1, **90.4**)	(76.6, 82.6, 83.1, **84.4**)		(79.8, 82.6, 86.1, **87.1**)	(72.8, 76.5, 78.4, **81.6**)
C3	(73.0, 82.1, 83.5, **86.3**)	(74.1, 81.5, 83.8, **85.2**)	(74.4, 80.2, 84.0, **85.3**)		(71.2, 70.0, 74.7, **77.2**)
C4	(82.0, 78.8, 84.2, **85.9**)	(68.3, 73.8, 74.6, **76.2**)	(74.0, 77.7, 82.0, **84.5**)	(71.1, 78.7, 80.3, **83.1**)	
Ave.	(79.0, 83.4, 86.0, **88.3**)	(74.7, 79.9, 81.5, **83.0**)	(75.2, 82.0, 85.5, **87.7**)	(76.4, 85.3, 86.8, **88.3**)	(71.2, 75.5, 79.2, **81.9**)

Table 2. Cross-view recognition accuracy on the IXMAS dataset in partially labeled mode. Each row is a source view and each column a target view. The three accuracy numbers in a tuple are the average recognition accuracy of [15], VVK, and VVKC respectively.

%	C0	C1	C2	C3	C4
C0		(63.6, 68.2, **71.5**)	(60.6, 65.9, **68.9**)	(61.2, 65.4, **67.3**)	(52.6, 60.4, **64.2**)
C1	(61.0, 67.2, **70.5**)		(62.1, 66.3, **69.8**)	(65.1, 71.7, **74.2**)	(54.2, 60.8, **62.3**)
C2	(63.2, 66.0, **67.8**)	(62.4, 68.1, **71.8**)		(71.7, 75.5, **79.2**)	(58.2, 65.1, **66.5**)
C3	(64.2, 67.5, **68.7**)	(71.0, 77.4, **80.0**)	(64.3, 67.7, **70.4**)		(56.6, 60.3, **63.8**)
C4	(50.0, 53.5, **55.4**)	(59.7, 64.7, **67.3**)	(60.7, 68.3, **72.6**)	(61.1, 65.6, **68.0**)	
Ave.	(59.6, 63.6, **65.6**)	(64.2, 69.6, **72.7**)	(61.9, 67.1, **70.4**)	(64.8, 69.6, **72.2**)	(55.4, 61.7, **64.2**)

using bag-of-words model [25]. To complement the local feature, we then extract global shape-flow feature [24]. Specifically, three channels features are extracted from each frame: horizontal optical flow, vertical optical flow, and silhouette. Then PCA is again used to reduce the dimensionality. Descriptors from neighboring frames are concatenated with the current frame descriptor to incorporate temporal information. The histogram vector is built over 500 quantized codewords. Finally, for each action video, we concatenate the local and global features to form a 1500-dimensional feature vector.

3.2. Pairwise Cross-view Recognition

In this section, we verify our algorithm on all possible pairwise view combinations (twenty in total for five views) in all three modes.

Correspondence mode: For an accurate comparison to [17] and [15], we follow the same leave-one-action-class-out strategy for choosing the orphan action which means that each time we only consider one action class for testing in the target view. The final results are reported according to average accuracy for all action classes in each view. Note that the orphan action class is not used to train the virtual view kernel and establishes corresponding pairs. The corresponding pairs are randomly chosen from the non-orphan training samples and these pairs account for 30% of the non-orphan samples. We set the transformed virtual view dimension d to 20. Meanwhile, we set α to 4 in Eq. (11) and set $\alpha = 4$ and $\beta = 3$ in Eq. (15).

The recognition accuracy is shown in Table 1 for all possible source-target view combinations. We compare our algorithms, i.e. virtual view kernel (VVK) and virtual view kernel with constraint on unlabeled samples (VVKC), with [17] and [15]. Note that we omit the accuracy of [6] and [7], since they report lower results than [17] and [15] in most view combinations. Some interesting observations can be made from Table 1. First, our algorithms (VVK and VVKC) outperform all five possible target views with varying source views on average recognition accuracies, which can be seen in the last row of Table 1. Second, our VVKC achieves better results than [17] in all the view combinations and obtains better results than [15] except only one view combination. Third, our VVK is superior to [15] with all view combinations but the combination of source view C1 and target view C3, due to sampling all the virtual views on the virtual path. Finally, since our algorithm takes full advantage of the unlabeled samples from the target view, the average accuracy of VVKC is about 2% better than VVK.

Partially labeled and unlabeled modes: For partially labeled mode, we set β to 3, and set d to 20. We compare our approach with multiple virtual views (MVV) proposed in [15], and the results are shown in Table 2. The labeled samples from the target view take up 30% of all the target view samples as [15]. From Table 2, it is can been that our approaches (VVK and VVKC) outperform MVV [15] in all the view combinations. Once again, we prove the effectiveness of our algorithm on partially labeled mode.

We then study the recognition accuracy as the proportion of labeled samples from the target view which increases from 0% to 30% in steps of 10%. The average recognition accuracies are shown in Fig. 2, from which we can see that our VVK and VVKC achieve better results in all situations.

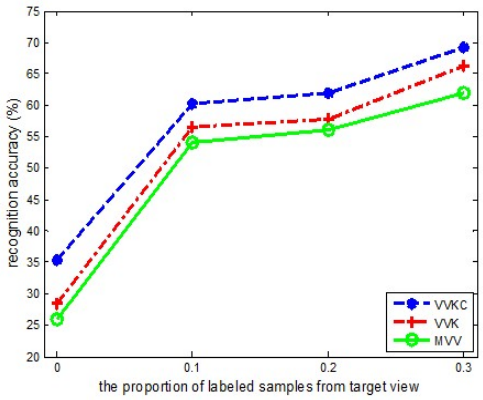

Figure 2. Cross-view action recognition accuracies on the IXMAS dataset compared with baseline from [15] when a varying proportion of samples are labeled in the target view.

Table 3. Recognition accuracy of non-discriminative virtual views (NDVV), VVK, and VVKC on the IXMAS dataset.

%	Correspondence	Partially labeled	Unlabeled
NDVV	76.6	52.6	24.5
VVK	83.8	66.3	28.4
VVKC	**85.8**	**69.2**	**35.3**

It is worth noting that when proportion of labeled samples from the target view is 0%, it degenerates into unlabeled mode. In unlabeled mode, VVKC gains about 7% over VVK, while about 9% over MVV. Therefore, the increase in accuracy demonstrates the advantage of the constraint on unlabeled samples.

3.3. Effect of Maximizing Discrimination

The virtual view kernel K (M_S and M_T) is learned under an information theoretic framework so as to maximize discrimination. To test the effect of maximizing discrimination, we let M_S and M_T be the bases of the principal subspaces of the source and target samples respectively, and directly compute K as Eq. (5) without discriminatively learning. The modification is similar to the method of Gong *et al.* [8]. Table 3 shows that the recognition ability of non-discriminatively algorithm reduces significantly for all the three working modes. This indicates that maximizing discrimination plays an important role on cross-view action recognition.

3.4. Multiple Source Views

Our algorithm can also handle multiple source views problem. We choose a target view and use all other four views as sources. We train the classifiers on the four source-target pairs and fuse these classifiers as the method proposed in Section 2.6. The average accuracies in correspondence and partially labeled modes are presented in Table 4 and Table 5 respectively. Our algorithms (VVK and VVKC) obtain better results than the baselines in both correspondence mode and partially labeled mode. In addition, comparing Table 4 with Table 1 and Table 5 with Table 2, we can see that the fusion strategy of multiple source views performs better than single source view.

Table 4. Cross-view action recognition accuracy (%) with multiple source views in correspondence mode.

Target view	C0	C1	C2	C3	C4
Liu *et al.* [17]	86.2	81.1	80.1	83.6	82.8
Li *et al.* [15]	85.1	82.1	82.2	85.7	77.6
VVK	86.5	83.3	85.7	88.6	82.4
VVKC	**89.2**	**85.6**	**88.0**	**90.7**	**83.6**

Table 5. Cross-view action recognition accuracy (%) with multiple source views in partially labeled mode.

Target view	C0	C1	C2	C3	C4
Li *et al.* [15]	62.0	65.5	64.5	69.5	57.9
VVK	64.5	71.6	69.2	72.8	63.1
VVKC	**66.4**	**73.5**	**71.0**	**75.4**	**66.4**

Table 6. Cross-view action recognition accuracy (%) under different α and β in correspondence mode.

β \ α	2	4	6	8
1	84.3	84.7	83.0	81.2
2	84.6	84.8	83.2	82.9
3	85.2	**85.8**	84.4	83.5
4	84.1	84.3	83.6	81.9

Table 7. Cross-view action recognition accuracy (%) under different β. The two accuracy numbers in a tuple are the average recognition accuracy of partially labeled and unlabeled modes.

β	1	2	3	4
	(65.7, 32.6)	(68.0, 34.5)	(**69.2**, **35.3**)	(67.6, 33.8)

3.5. Influence of Parameters Variances

We further evaluate the performance of the proposed VVKC with respect to α and β in all the three modes which control the importance of corresponding pairs and unlabeled samples. For correspondence mode shown in Table 6, we can see that when $\alpha = 4$ and $\beta = 3$, the result is the best. When $\beta = 3$, the results are the best for both partial labeled and unlabeled modes as shown in Table 7. The conclusion can be generalized to VVK as well, i.e. $\alpha = 4$ for correspondence mode.

4. Conclusion

We propose a kernel-based method for cross-view action recognition. The method constructs a continuous virtual path between the source view and the target view. The proposed virtual view kernel utilizes all the virtual views on the virtual path to learn new feature representations that are robust to change in views. Furthermore, we impose a constraint on unlabeled samples from the target view for further performance improvement. The experimental results demonstrate that our method achieves better results than the state-of-the-art methods in cross-view action recognition.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 60933010, No. 61172103 and No. 61271429 and National High-tech R&D Program of China (863 Program) under Grant No. 2012AA041312.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. *In Proc. of NIPS*, page 137, 2007.

[2] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. *In Proc. of NIPS*, 2010.

[3] C. Chang and C. Lin. Libsvm: a library for support vector machines. *ACM Trans. on TIST*, 2(3):27, 2001.

[4] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. *In Proc. of ICCV workshop: VS-PETS*, 2005.

[5] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *In Proc. of ICCV*, pages 726–733, 2003.

[6] A. Farhadi and M. Tabrizi. Learning to recognize activities from the wrong view point. *In Proc. of ECCV*, pages 154–166, 2008.

[7] A. Farhadi, M. Tabrizi, I. Endres, and D. Forsyth. A latent model of discriminative aspect. *In Proc. of ICCV*, pages 948–955, 2009.

[8] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. *In Proc. of CVPR*, pages 2066–2073, 2012.

[9] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. *In Proc. of ICCV*, pages 999–1006, 2011.

[10] B. Hall. Lie algebras, and reprensentations: An elementary introduction. *Springer*, 2003.

[11] I. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. *In Proc. of ECCV*, pages 293–306, 2008.

[12] I. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *PAMI*, 33(1):172–185, 2011.

[13] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. *In Proc. of CVPR*, pages 2046–2053, 2010.

[14] R. Li, T. Tian, and S. Sclaroff. Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. *In Proc. of ICCV*, pages 1–8, 2007.

[15] R. Li and T. Zickler. Discriminative virtual views for cross-view action recognition. *In Proc. of CVPR*, pages 2855–2862, 2012.

[16] Z. Lin, Z. Jiang, and L. Davis. Recognizing actions by shape-motion prototype trees. *In Proc. of ICCV*, pages 444–451, 2009.

[17] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. *In Proc. of CVPR*, pages 3209–3216, 2011.

[18] J. Liu, Y. Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. *In Proc. of CVPR*, pages 461–468, 2009.

[19] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. *In Proc. of CVPR*, pages 1–8, 2007.

[20] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *IJCV*, 66(1):83–101, 2006.

[21] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *IJCV*, 50(2):203–226, 2002.

[22] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. *In Proc. of ECCV*, pages 577–590, 2010.

[23] A. Srivastava, I. Jermyn, and S. Joshi. Riemannian analysis of probability density functions with applications in vision. *In Proc. of CVPR*, pages 1–8, 2007.

[24] D. Tran and A. Sorokin. Human activity recognition with metric learning. *In Proc. of ECCV*, pages 548–561, 2008.

[25] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. *In Proc. of BMVC*, 2009.

[26] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. *In Proc. of ICCV*, pages 1–7, 2007.

[27] S. Xiang, F. Nie, Y. Song, and C. Zhang. Contour graph based human tracking and action sequence recognition. *Pattern Recognition*, 41(12):3653–3664, 2008.

[28] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. *In Proc. of CVPR*, pages 984–989, 2005.

[29] Z. Zhang, C. Wang, B. Xiao, W. Zhou, and S. Liu. Action recognition using context-constrained linear coding. *Signal Processing Letters, IEEE*, 19(7):439–442, 2012.

[30] Z. Zhang, C. Wang, B. Xiao, W. Zhou, and S. Liu. Contextual fisher kernels for human action recognition. *In Proc. of ICPR*, pages 437–440, 2012.