Discriminative Virtual Views for Cross-View Action Recognition

Ruonan Li and Todd Zickler
Harvard School of Engineering and Applied Science
{ruonanli,zickler}@seas.harvard.edu

Abstract

We propose an approach for cross-view action recognition by way of 'virtual views' that connect the action descriptors extracted from one (source) view to those extracted from another (target) view. Each virtual view is associated with a linear transformation of the action descriptor, and the sequence of transformations arising from the sequence of virtual views aims at bridging the source and target views while preserving discrimination among action categories. Our approach is capable of operating without access to labeled action samples in the target view and without access to corresponding action instances in the two views, and it also naturally incorporate and exploit corresponding instances or partial labeling in the target view when they are available. The proposed approach achieves improved or competitive performance relative to existing methods when instance correspondences or target labels are available, and it goes beyond the capabilities of these methods by providing some level of discrimination even when neither correspondences nor target labels exist.

1. Introduction

We consider the challenge of recognizing human actions across changes in the observer's viewpoint. Opportunities for the use of action analysis in domains such as surveillance, video indexing/retrieval, and human-computer interaction are growing fast [16, 18, 1], but realizing this potential relies on the ability to accurately interpret human activities from a broad range of viewing directions. In a typical action recognition setting, spatio-temporal features are computed from a video to represent the underlying action. These features can be powerful in discriminating between different actions observed from similar viewpoints, but since the same action can appear quite different when observed from different directions, the utility of these features degrades when the viewpoint changes more significantly.

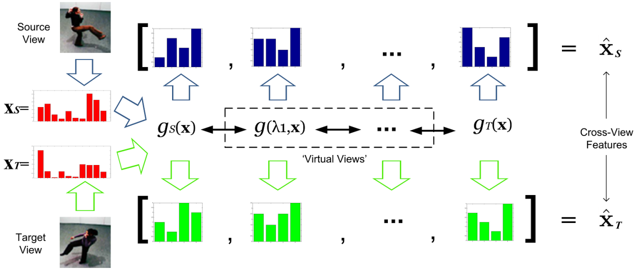

Figure 1. Knowledge transfer using 'virtual views'. Action descriptors $\mathbf{x}$ from different views are augmented into cross-view feature vectors $\hat{\mathbf{x}}$ by applying a finite sequence of linear transformations $g(\lambda_i, \mathbf{x})$ to each descriptor $\mathbf{x}$. We introduce a flexible, semi-supervised framework for learning the transform-sequences in a way that can exploit various forms of partial labeling for the two camera angles.

The brutal-force approach of training independent classifiers for each action in each view does not scale well due the requirement of excessive labeled training data, so a possible line of attack is to search for view-invariant features, representations, or models that can be used for all viewpoints. One approach is to infer three-dimensional scene structure so that the derived action descriptors can be adapted from one view to another through geometric reasoning [29, 26, 15, 10, 4], while another is to search for spatio-temporal features of a video sequence that are insensitive to changes in view angle [17, 21, 23, 22, 3, 28]. Recent view-invariant approaches include [27] and [13]. The former learns a classifier on examples taken from various views, and the latter introduces a temporal self-similarity matrix and demonstrates its view stability empirically.

Another emerging family of approaches address cross-view action recognition by adapting features, representations, or recognition models trained on one or more source views to a target view where the recognition task will be performed [8, 7, 14]. This boils down to drawing some form of statistical connections between view-dependent features extracted from different viewing directions. This is attractive because it reduces reliance on accurately inferring explicit camera geometry, extended motion trajectories, and three-dimensional actor models. A notable example of this knowledge transfer approach is the work of Farhadi *et. al.* [8, 7], who rely on simultaneous multi-view observations of the same action instance to explicitly identify maps between one view's features and those of another, thereby allowing a classifier learned in one view to be adapted by suitably reorganizing its weights. Another example is the work of Liu *et. al.* [14] who rely on the same style of input to learn a cross-view bag of 'bilingual words' representation in which each bilingual word represents the co-occurrence of one visual word in one view with another visual word in another view.

We propose a different approach to view knowledge transfer that significantly relaxes the requirements on the training data. Instead of requiring access to simultaneous multi-view observations of the same action instance, our approach can leverage a variety of weak supervisions, including cases in which action categories are labeled in only one camera angle and there are no links or labels at all in another. As depicted visually in Figure 1, the conceptual idea is to construct 'virtual views' between action descriptors from one viewpoint and those from another. We imagine that an action descriptor transforms continuously between one viewpoint and another, and we compute 'virtual views' as a sequence of transformed descriptors obtained by making a finite number of stops along the way. The intermediate views are virtual because they exist only in an abstract feature space and are not identified with any physical change in camera position. Taken together, the sequence of transformed descriptors represents an augmented feature that embeds the statistical transition between two views, and by developing a discriminative method for learning the sequence of transform operators, we ensure that these augmented 'cross-view' features can be used to meaningfully compare actions descriptors from different viewpoints.

Our key technical contribution is an information-theoretic framework that allows learning discriminative 'virtual view' transformations using a wide variety of partial labelings. Like the approaches in [8, 14], it can exploit the case in which an unlabeled action instance (execution) observed simultaneously in both views yields an matched pair, so that a few of such (unlabeled) pairs are available. We refer to this working mode as the *correspondence mode*. At the same time, our approach can also operate under the conditions usually considered by the transfer learning or domain adaptation paradigm [9, 6, 19, 2, 11], where the samples in the target domain are usually partially labeled while matched instances with the source view may not exist. We refer to this working mode as the *partially labeled mode*. In addition to the these two working modes, our approach can operate in a third mode where the target view is completely unlabeled and no target instances are matched to the source instances. We refer to it as the *unlabeled mode*. Experiments show that our approach provides improved or competitive performance as existing methods when operating in the first two modes, and that it provides some discrimination in the third.

2. Discriminative Virtual Views

Consider source view V_S and target view V_T, and imagine that they are connected by some virtual path $V(\lambda)$, $0 \leq \lambda \leq 1$, with $V(0) = V_S$ and $V(1) = V_T$. Recall that this virtual path does not correspond to physical changes in camera position, but instead is associated with transformations of action descriptors. For the transformations of action descriptors along the virtual path $V(\lambda)$, we will use a particular class of linear projections. To this end, it is convenient to express the transformation associated with the source view as $g_S(\mathbf{x}) = A_S^T \mathbf{x}$ and that associated with the target view as $g_T(\mathbf{x}) = A_T^T \mathbf{x}$, where $\mathbf{x}$ is a D-dimensional raw action descriptor (*e.g.*, histogram on a vocabulary of visual words) computed from either the source view (in the former case) or the target view (in the latter case). Here A_S, A_T are both $D \times d$ matrices satisfying $A_S^T A_S = I$ and $A_T^T A_T = I$, *i.e.*, they both have orthogonal columns of unit-length, and induce a linear dimensionality deduction.

We represent the view change along the virtual path $V(\lambda)$ implicitly as alterations of the feature extractors g_S and g_T (and thus the matrices A_S and A_T). For this purpose, we define $g(\lambda, \mathbf{x}) = A_\lambda^T \mathbf{x}$ for $0 < \lambda < 1$, where A_λ is also a $D \times d$ transformation matrix, $g(0, \mathbf{x}) = g_S(\mathbf{x})$, and $g(1, \mathbf{x}) = g_T(\mathbf{x})$. Sampling the virtual path $V(\lambda)$ at a finite number of intervals $\lambda_1, \lambda_2, \cdots, \lambda_L$ $(0 < \lambda_1 < \lambda_2 < \cdots < \lambda_L < 1)$ yields a sequence of 'virtual views' $V(\lambda_1), V(\lambda_2), \cdots, V(\lambda_L)$, and the consecutive incremental 'jumps' from $V(0) = V_S$ to $V(\lambda_1)$, $V(\lambda_2)$, *etc.*, through to $V(1) = V_T$ are intended to establish a smooth bridge between the visual information existing in the two views. Since we have associated a view V with a transform g uniquely identified by a matrix A , the sequence of virtual-view transforms $g(\lambda_1, \mathbf{x}) = A_{\lambda_1}^T \mathbf{x}, g(\lambda_2, \mathbf{x}) = A_{\lambda_2}^T \mathbf{x}, \cdots, g(\lambda_L, \mathbf{x}) = A_{\lambda_L}^T \mathbf{x}$ can provide a sequence of 'virtual' features that characterize the smooth changes of the features from the source to the target. Refer again to Figure 1.

The major questions to be answered are how to choose effective transformations g_S, g_T (*i.e.*, A_S, A_T) and how to alter the transformations to define the virtual path $g(\lambda, \mathbf{x})$ (*i.e.*, A_λ). In 2.1, we show that for a given pair of transformations A_S, A_T, there exists a particular 'shortest' path connecting the two, allowing the virtual views to be obtained analytically. Then, in 2.2 we formulate the problem of identifying the optimal pair (A_S, A_T) under our three distinct working modes, so that in each case the augmented cross-view features are discriminative among action categories. Finally, we provide the algorithm to solve this problem and determine the optimal A_S and A_T in 2.3.

2.1. Obtaining a Virtual Path

For the moment, let us assume that the source and target view transformations A_S and A_T have been given, and our task is to compute the transforms A_λ that connect them along a virtual path. To this end, we aim to determine a path of $D \times d$ matrices from A_S to A_T. There are various ways to establish such connections between the two matrices, among which one possibility is to look into the space of all $D \times d$ matrices and make use of its geometry [24]. However, manipulation in this space is computationally inconvenient, so we pursue an alternative approach. By construction the columns of A_S and A_T are of unit length and therefore lie on a hyper-sphere. Thus, a natural definition for a continuous path between the ith column of A_S and the ith column of A_T is the segment of the great circle that connects them. We define a closed-form path between the matrices as wholes by separately identifying the D geodesics between their D corresponding columns, and then traveling simultaneously along these geodesics from the columns of A_S at rates that guarantee simultaneous arrival at columns of A_T. Specifically, to get the transforms $A_\lambda, \lambda = \lambda_1, \lambda_2, \cdots, \lambda_L$ along the virtual path that connects $A_S = [\mathbf{a}_{S,1}, \mathbf{a}_{S,2}, \cdots, \mathbf{a}_{S,D}]$ to $A_T = [\mathbf{a}_{T,1}, \mathbf{a}_{T,2}, \cdots, \mathbf{a}_{T,D}]$, we compute

$$\mathbf{a}_{\lambda,i} = \frac{(1-\lambda)\mathbf{a}_{S,i} + \lambda \mathbf{a}_{T,i}}{\lambda^2 + (1-\lambda)^2 + 2\lambda(1-\lambda)\mathbf{a}_{S,i}^T \mathbf{a}_{T,i}}, \tag{1}$$

and then obtain $A_\lambda = [\mathbf{a}_{\lambda,1}, \mathbf{a}_{\lambda,2}, \cdots, \mathbf{a}_{\lambda,D}]$.

Note that columns of an A_λ constructed in this way are not necessarily orthogonal, but remain unit-norm. The preservation of unit-length guarantees that the transformed feature $A_\lambda^T \mathbf{x}$ is at the same scale as $A_S^T \mathbf{x}$ and $A_T^T \mathbf{x}$. To create our augmented cross-view feature, we simply concatenate the transformed features into a single long feature vector:

$$\hat{\mathbf{x}} = [(A_S^T \mathbf{x})^T, (A_{\lambda_1}^T \mathbf{x})^T, \cdots, (A_{\lambda_L}^T \mathbf{x})^T, (A_T^T \mathbf{x})^T]^T. \tag{2}$$

This new feature implicitly incorporates the smooth change from one view to the other, and therefore bridges the two views and serves as a new, unified feature vector.

2.2. Maximizing Discrimination

Since our virtual view transforms are completely determined by matrices A_S and A_T, we now turn to the question about how to choose good values for A_S and A_T. Let us consider a two-class problem (multi-class problems can be treated as a set of two-class problems using one versus all approach) with positive training examples $\{(\mathbf{x}_{P,i}, 1)\}_{i=1}^{n_P}$ and negative training examples $\{(\mathbf{x}_{N,j}, -1)\}_{j=1}^{n_N}$. In the *unlabeled mode*, all these labeled samples come from the source view. For the *partially labeled mode*, only a minority of the above training samples come from the target view. In either case, we would like to maximize our ability to discriminate between the two classes in all available labeled samples. To this end, we seek transformations A_S and A_T that maximize the mutual information between cross-view feature $\hat{\mathbf{x}}$ and the class label $c \in \{1, -1\}$:

$$\max_{A_S, A_T} I(\hat{\mathbf{x}}; c). \tag{3}$$

Note that

$$\begin{aligned} I(\hat{\mathbf{x}}; c) &= H(\hat{\mathbf{x}}) - H(\hat{\mathbf{x}}|c) \\ &= H(\hat{\mathbf{x}}) - P(c=1)H(\hat{\mathbf{x}}_P) - P(c=-1)H(\hat{\mathbf{x}}_N), \end{aligned} \tag{4}$$

so (3) can be written in terms of the differential entropy $H(\hat{\mathbf{x}})$.

To solve (3), we approximate differential entropy $H(\hat{\mathbf{x}})$ using a finite set of samples. Assuming that the samples of cross-view feature $\hat{\mathbf{x}}$ are drawn from a Gaussian distribution, we may write $H(\hat{\mathbf{x}}) = \frac{1}{2}\ln((2\pi e)^{d(L+2)} \det \Sigma)$, in which the covariance matrix Σ can be estimated from samples $\hat{\mathbf{x}}$. Further assuming equal prior probabilities for the two classes, we approximate the objective in (3) by

$$I(\hat{\mathbf{x}}; c) \doteq \ln\det\Sigma_{all} - \frac{1}{2}\ln\det\Sigma_P - \frac{1}{2}\ln\det\Sigma_N, \tag{5}$$

where $\Sigma_{all}, \Sigma_P, \Sigma_N$ are covariance matrices computed from all labeled samples, the positive samples, and the negative samples respectively.

We may take a similar approach to choose the optimal transformation pair A_S and A_T in the *correspondence mode*. Specifically, labeled samples can be written as $\{(\mathbf{x}_{P,i}^{(S)}, 1)\}_{i=1}^{n_P}$ and $\{(\mathbf{x}_{N,j}^{(S)}, -1)\}_{j=1}^{n_N}$ since in this mode the labels are not shared across the two views. The instances in correspondence, meanwhile, can be expressed as $\{(\mathbf{x}_k^{(S)}, \mathbf{x}_k^{(T)})\}_{k=1}^{n_C}$ where the unlabeled pair $(\mathbf{x}^{(S)}, \mathbf{x}^{(T)})$ describes the same instance (execution) of the an unlabeled action in two views. We expand all $\mathbf{x}^{(S)}$ and $\mathbf{x}^{(T)}$ to get $\hat{\mathbf{x}}^{(S)}$ and $\hat{\mathbf{x}}^{(T)}$, and define $\Delta\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(S)} - \hat{\mathbf{x}}^{(T)}$ for each pair $(\mathbf{x}^{(S)}, \mathbf{x}^{(T)})$ corresponding to the same instance. Since the pair $(\hat{\mathbf{x}}^{(S)}, \hat{\mathbf{x}}^{(T)})$ describes the same instance of an action, we expect $\Delta\hat{\mathbf{x}}$ to be close to zero. In addition to maximizing the mutual information between $\hat{\mathbf{x}}$ and the class label $c \in \{1, -1\}$, we add penalty $H(\Delta\hat{\mathbf{x}})$ to solve

$$\max_{A_S, A_T} I(\hat{\mathbf{x}}; c) - \gamma H(\Delta\hat{\mathbf{x}}). \tag{6}$$

As previously, we approximate the mutual information in terms of covariance matrices and assume the cross-view feature $\Delta\hat{\mathbf{x}}$ to be Gaussian distributed with zero mean, since we expect it to be not only compactly distributed but also close to the origin. The objective in (6) is therefore approximated by

$$\begin{aligned} I(\hat{\mathbf{x}}; c) - \gamma H(\Delta\hat{\mathbf{x}}) \doteq & \ln\det\Sigma_{all} - \frac{1}{2}\ln\det\Sigma_P \\ & - \frac{1}{2}\ln\det\Sigma_N - \gamma\ln\det\Sigma_\Delta, \end{aligned} \tag{7}$$

where Σ_Δ is the *correlation* matrix, not the covariance matrix, for all $\Delta\hat{\mathbf{x}}$'s. A minimization of $\det \Sigma_\Delta$ will yield $\Delta\hat{\mathbf{x}}$'s concentrating around $\mathbf{0}$, by which we enforce the correspondence between the pair $(\hat{\mathbf{x}}^{(S)}, \hat{\mathbf{x}}^{(T)})$. A practical issue that may arise is a rank deficiency in any of the covariance/correlation matrices. In this case we first determine the minimum rank among all involved matrices (say, r), and use the product of the top r large eigenvalues of each matrix to approximate its determinant.

In fact, our learning algorithm for maximizing (7) can not only exploit the semi-supervisions considered in the three working modes, but also accommodate any mixture of those modes: We simply need to encode the information regarding available labels and corresponding instances respectively into the covariance/correlation matrices Σ_{all}, Σ_P, Σ_N, and Σ_Δ.

2.3. Obtaining the Optimal Virtual Views

We now go on to present the algorithm with which we optimize the two objectives (5) and (7) above. For simplicity, we denote the objectives in both (5) and (7) as $J(A_S, A_T)$ in the following discussion.

We employ a greedy algorithm that iteratively searches for transformations (A_S, A_T) that maximize J. To use a gradient based approach, we need to evaluate $\frac{\partial J(A_S,A_T)}{\partial A_S}$ and $\frac{\partial J(A_S,A_T)}{\partial A_T}$ subject to $A_S^T A_S = I$ and $A_T^T A_T = I$, which is difficult. Instead, we consider an axis-rotating approach. Let $A_S(t-1)$ to be the estimate for A_S and $A_T(t-1)$ to be the estimate for A_T at iteration $t-1$. We seek matrices $R_S(t), R_T(t) \in \mathbf{SO}(D)$, *i.e.*, the D-dimensional special orthogonal group, so that the estimate at step t is $A_S(t) = R_S(t)A_S(t-1)$ and $A_T(t) = R_T(t)A_T(t-1)$. In essence, we seek a pair of $R_S(t), R_T(t)$ to provide a steep ascent in J. Note that $\mathbf{SO}(D)$ corresponds to the set of rotation operations in $\mathbb{R}^D$, thus the resulting $A_S(t), A_T(t)$ will be orthonormal matrices as well. We summarize the algorithm by which we obtain the optimal $R_S(t), R_T(t)$ and consequently $A_S(t), A_T(t)$ from $A_S(t-1), A_T(t-1)$ in Algorithm 1. The mathematical principle behind this algorithm involves approximate gradient computation on $\mathbf{SO}(D)$ and is briefly introduced in the Appendix. More details on $\mathbf{SO}(D)$ can be found, for example, in [12].

Algorithm 1 Greedy Axis Rotation.

1. Input: $A_S(t-1)$, $A_T(t-1)$, $\epsilon > 0$, $\delta > 0$, $N > 0$;
2. For $2 \le i \le D, i+1 \le j \le D$, compute $R_{S,i,j} = \exp(\epsilon(E_{i,j} - E_{j,i}))$, and $\Delta J_{S,i,j} = J(R_{S,i,j}A_S(t-1), A_T(t-1)) - J(A_S(t-1), A_T(t-1))/\epsilon$, where $E_{i,j}$ is a matrix whose (i,j)th element is one and all others are zero;
3. For $1 \le k \le D, k+1 \le l \le D$, compute $R_{T,k,l} = \exp(\epsilon(E_{k,l} - E_{l,k}))$, and $\Delta J_{T,k,l} = J(A_S(t-1), R_{T,k,l}A_T(t-1)) - J(A_S(t-1), A_T(t-1))/\epsilon$;
4. Compute $c_{i,j} = \frac{\Delta J_{S,i,j}}{(\sum_{i',j'} \Delta J^2_{S,i',j'} + \sum_{k',l'} \Delta J^2_{T,k',l'})^{\frac{1}{2}}}$, and $c_{k,l} = \frac{\Delta J_{T,k,l}}{(\sum_{i',j'} \Delta J^2_{S,i',j'} + \sum_{k',l'} \Delta J^2_{T,k',l'})^{\frac{1}{2}}}$;
5. Let $R_{S,n} = \exp(n\delta \sum_{i,j} c_{i,j}(E_{i,j} - E_{j,i}))$ and $R_{T,n} = \exp(n\delta \sum_{k,l} c_{i,j}(E_{k,l} - E_{l,k}))$, find $R_S(t), R_T(t)$ by
$$n^* = \arg \max_{0 \le n \le N} J(R_{S,n}A_S(t-1), R_{T,n}A_T(t-1)), \tag{8}$$
and then $R_S(t) = R_{S,n^*}$, $R_T(t) = R_{T,n^*}$;
6. Output: $A_S(t) = R_S(t)A_S(t-1)$, and $A_T(t) = R_T(t)A_T(t-1)$. ■

In practice, we initialize $A_S(0)$ and $A_T(0)$ as described in the next section, and iterate Algorithm 1 until $A_S(t) = A_S(t-1)$ and $A_T(t) = A_T(t-1)$.

3. Implementation Details and Extensions

The first step in training our model is to determine the working mode and extract the corresponding single-view action descriptors from each training video. In all cases we use an equally-spaced sequence for the path parameter λ, *i.e.*, $\lambda_i = \frac{i}{L+1}$. Once the optimal transformations are computed, we compute cross-view features $\hat{x}$ for all training samples and use the subset of labeled samples to train a cross-view action classifier. There are many possible choices for the classifier, and in the experiments we use the Multiple Kernel Learning SVM (MKL-SVM) [20].

For any testing observation $\mathbf{x}$ from target view, we compute its cross-view feature $\hat{\mathbf{x}}$ using all transformations obtained from training stage, and then evaluate MKL-SVM at this cross-view feature.

Initialization. Good choices for initializing A_S and A_T can expedite the training procedure. For the source view, we find it effective to use an orthonormal basis that spans the d dimensional subspace determined by the Fisher discriminant for the labeled samples in that view. For the target view, we simply use the basis of the Fisher discriminant subspace if labeled samples are available, or that of the principal subspace if not.

Multiple Action Classes. For an M-class action recognition problem, we learn M binary one-against-all models as described above. The final classification is determined by selecting the model whose MKL-SVM yields the maximum response.

Multiple Source Views. In many applications we may have w source views with $w > 1$. In this case, given a test instance from the target view, we simply aggregate the response values from the w MKL-SVM classifiers on their respective cross-view features $\hat{\mathbf{x}}$, and then make a binary decision with the threshold at 0. For a M-class problem, we select the class which achieves the maximum aggregated response value.

4. Experimental Evaluation

Following [8, 7, 14], we evaluate our approach on the IXMAS multi-view action dataset [26] which contains eleven categories including actions like `walk`, `kick`, and `throw`. Each action is performed three times by twelve actors taken from five different views including four side views and one top view. To enable appropriate comparison, we use the same low-level action descriptors used in [14]. Specifically, the action is represented by a concatenation of a spatio-temporal interest-point-based descriptor [5] and a shape-flow descriptor [25]. The two types of descriptors serve as complementary local and global characterizations of the motion. For the local interest point based descriptor, a 2-D Gaussian filter and then a 1D-Gabor filter are applied to the video, and the interest points are detected at the local maximum response. The parameters for the two filters are $\sigma = 2$ and $\tau = 1.5$ respectively, and at most 200 maxima are extracted from each video. Then, the spatio-temporal volumes around the maxima are extracted, and gradient-based descriptors are computed and reduced to 100 dimensions via PCA. These descriptors are further quantized to visual words by k-means clustering. Eventually, each action is represented by a histogram over 1000 visual-words. For the global shape-flow feature, a three channels descriptor is computed from each frame: horizontal optical flow, vertical optical flow, and silhouette. Each of these channels has the same dimension as the input frame, and PCA is again employed to reduce the dimensionality. Descriptors from neighboring frames are concatenated with the current frame descriptor to incorporate temporal information. Finally, the histogram vector is built over 500 quantized visual words. See [5, 25] for more details.

4.1. Pairwise Cross-View Recognition

We first look into all possible pairwise view combinations (twenty in total for five views) to evaluate the proposed approach. We begin with the *correspondence mode* and compare with existing approaches. We then show results on *partially labeled* and *unlabeled modes*.

Correspondence mode: We follow the same data separation scheme as in [14] (inherited from [8, 7]) for fair comparison. This is a *leave-one-action-class-out* scheme, where we consider one action class (called an 'orphan action') in the target view, and exclude all videos of that class when learning the quantized visual words and establishing correspondence. The instances in correspondence are randomly selected from the non-orphan training actions, and approximately 30% of the non-orphan samples serve as such pairs. We adopt the six-fold cross-validation scheme of [14] to build the discriminative virtual views as well as train the classifier on the augmented cross-view feature vectors. We sample ten virtual views from the virtual path, and set the transformed virtual view dimension to $d = 20$. The MKL-SVM, meanwhile, consists of nine Gaussian kernels with bandwidths in powers of 10 ranging from 10^{-4} to 10^{4}. The final performance is reported on an average over all actions classes.

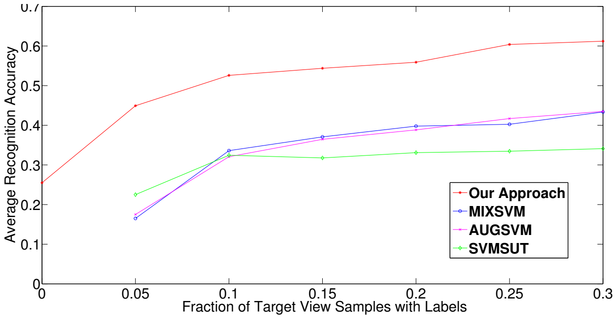

Figure 2. Cross-view action recognition accuracy on the IXMAS dataset compared with baselines from [2] when a varying fraction of samples are labeled in the target view. Note that our approach provides some discrimination even when no target labels are available (*i.e.*, at 0%).

The recognition accuracy is shown in Table 1 for all possible source-target view combinations, as compared to baselines [7], and [14]. (We omit the accuracy of [8] since it reports the lowest in all cases). It is seen that our approach is outperformed by [7] on two source-target combinations and by [14] on one combination, while it achieves a uniform improvement over all baselines on the other combinations. In particular, our approach achieves increased accuracy on average for all five possible target views with varying source views, though the increase on camera 4, the top view, is less significant than the others.

Partially labeled and unlabeled modes: As mentioned earlier, the view-transfer problem has much in common with other transfer learning or domain adaptation problems. Therefore, we consider cross-view recognition as in a similar setting as semi-supervised classification, where a small portion of the samples from the target view is labeled, and we compare our performance to that obtained by methods studied in [2]. Again, we employ a six-fold cross-validation strategy, and provide class labels to randomly selected samples from the target view. We again sample ten virtual views from the virtual path, and set the dimension of the transformed feature to $d = 20$. Three types of SVMs used in [2] are employed in our experiment for comparison: SVM-

Table 1. Cross-view action recognition accuracy on the IXMAS dataset when matched instances are available between the source view and the target view (correspondence mode). Each row is a source view and each column a target view. The three accuracy numbers in a triple are the average recognition accuracy of [7], [14], and our approach respectively.

%	c0	c1	c2	c3	c4
c0		(79, 79.9, **81.8**)	(79, 76.8, **88.1**)	(68, 76.8, **87.5**)	(76, 74.8, **81.4**)
c1	(72, 81.2, **87.5**)		(74, 75.8, **82.0**)	(70, 78.0, **92.3**)	(66, 70.4, **74.2**)
c2	(71, 79.6, **85.3**)	(82, 76.6. **82.6**)		(76, 79.8, **82.6**)	(72, 72.8, **76.5**)
c3	(75, 73.0, **82.1**)	(75, 74.1, **81.5**)	(79, 74.4, **80.2**)		(**76**, 71.2, 70.0)
c4	(80, **82.0**, 78.8)	(73, 68.3, **73.8**)	(73, 74.0, **77.7**)	(**79**, 71.1, 78.7)	
Average	(74, 79.0, **83.4**)	(77, 74.7, **79.9**)	(76, 75.2, **82.0**)	(73, 76.4, **85.3**)	(72, 71.2, **75.5**)

Table 2. Cross-view action recognition accuracy on the IXMAS dataset when some labels are available in the target view but there are no matched pairs (partially labeled 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 SVMSUT, AUGSVM, MIXSVM from [2], and our approach respectively.

%	c0	c1	c2	c3	c4
c0		(39.8, 42.8, 36.8, **63.6**)	(42.1, 45.2, 46.8, **60.6**)	(41.6, 47.2, 42.7, **61.2**)	(28.8, 30.5, 36.7, **52.6**)
c1	(35.7, 44.1, 39.4, **61.0**)		(42.0, 43.5, 51.8, **62.1**)	(28.5, 47.1, 45.8, **65.1**)	(25.1, 43.6, 40.2, **54.2**)
c2	(36.1, 53.7, 49.1, **63.2**)	(42.0, 50.5, 49.4. **62.4**)		(43.0, 53.5, 45.0, **71.7**)	(30.4, 39.1, 46.9, **58.2**)
c3	(31.6, 46.3, 39.3, **64.2**)	(30.3, 42.5, 42.5, **71.0**)	(36.0, 48.8, 51.2, **64.3**)		(28.7, 37.5, 38.9, **56.6**)
c4	(24.7, 37.0, 40.3, **50.0**)	(27.0, 35.0, 42.5, **59.7**)	(36.7, 44.4, 40.4, **60.7**)	(31.1, 37.2, 40.7, **61.1**)	
Average	(32.0, 45.3, 42.6, **59.6**)	(34.8, 42.7, 42.8, **64.2**)	(39.2, 45.4, 47.5, **61.9**)	(36.1, 46.2, 43.5, **64.8**)	(28.3, 37.6, 40.7, **55.4**)

SUT, AUGSVM, and MIXSVM. SVMSUT trains a single classifier on all labeled samples from both views and treats each sample as independent. AUGSVM uses a new feature vector which reserves space for both views, and fills an original feature into its corresponding space to obtain the new features. MIXSVM, meanwhile, trains two SVM's on the source and target and then learns an optimal linear combination of the two. Since we use MKL-SVM instead of a single SVM, we use MKL versions of the three baselines as well for comparison. (We refer to them using their original names even though we actually use their MKL version). The kernel types and parameters remain the same as in the previous experiment, and we use the fusion scheme for multiple action class introduced in the previous section.

We vary the fraction of the labeled samples from the target view in increments of 5% up to 30%. The average recognition accuracies for different fractions are shown in Figure 2, from which a substantial improvement is observed for our approach relative to the baselines. Note that the left side of the graph (0%) corresponds to *unlabeled mode* in which no target samples have labels for training. Our approach handles this mode seamlessly, and it outperforms the baselines that have access to labeled target samples (ours is 26% accurate with no target labels while the others is less than 26% accurate even with 5% of target labels). Also note that AUGSVM directly combines source and target samples into a single vector while we augment either the source or target feature by the discriminative virtual features. Therefore, one can view AUGSVM as a limiting case of our framework in which the number of virtual views is set to zero. Our increase in accuracy therefore demonstrates the advantage of using the virtual views.

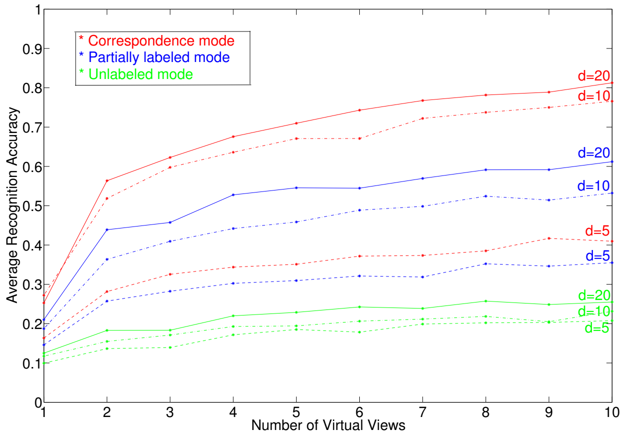

Figure 3. Cross-view action recognition accuracy on the IXMAS dataset for all three working modes (correspondence mode, partially labeled mode, unlabeled mode) with a varying number of virtual views and a varying dimension *d* for each virtual view.

4.2. Effects of Varying Parameters

In the previous experiments we use ten virtual views, each with a 20-dimensional feature. To investigate the effects of changes in these parameters, we vary the number of virtual views from 1 to 10 and set the dimension for each virtual view to be 5, 10, or 20. All three working modes are evaluated, and the overall recognition accuracy is given in Figure 3. A significant jump is observed from one virtual view to two virtual views, after which the accuracy increases only incrementally. Also, the dimension increase from 10 to 20 leads to mild accuracy improvement, espe-

Table 3. Cross-view action recognition accuracy on the IXMAS dataset with [11], non-discriminative virtual views (NDVV), and our approach, under all three working modes.

%	Correspondence	Partially labeled	Unlabeled
[11]	63.3	52.8	19.8
NDVV	72.7	50.9	22.4
Ours	81.2	61.1	26.0

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

Target View	c0	c1	c2	c3	c4
[14]	**86.2**	81.1	80.1	83.6	**82.8**
Our Approach	85.1	**82.1**	**82.2**	**85.7**	77.6

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

Target View	c0	c1	c2	c3	c4
SVMSUT	38.5	43.4	50.3	51.0	35.1
AUGSVM	54.2	50.8	58.1	49.5	46.9
MIXSVM	46.4	44.2	52.3	47.7	44.7
Our Approach	62.0	65.5	64.5	69.5	57.9

cially for *correspondence mode*. These observations imply that one may use a relatively smaller number of virtual views and lower dimensions per view unless a very high accuracy is desired.

4.3. Non-Discriminative Virtual Views

The transformations A_S, A_T and A_λ are learned through mutual information maximization to discriminate action categories. How will the performance be affected if these transformations are not learned discriminatively? To answer this question, we let A_S, A_T be the bases of the principal subspaces of the source and target samples respectively, and directly compute A_λ following 2.1 without the optimizations in 2.2 or 2.3. This modification reduces our approach to bridging the source and target by non-discriminative projections, similar to the method of Gopalan *et. al.* [11]. In Table 3, we compare results of our proposed approach to those of this non-discriminative version, as well as to [11]. The average recognition accuracy of these non-discriminative approach suffers a significant drop for all three working modes, which underscores the benefit of learning the virtual views discriminatively.

4.4. Multiple Source Views

To explore the benefits of having multiple source views, we select a target view and use all other four views as sources. Classifiers trained on the four source-target pairs are fused using the method presented in Section 3. We again compare *correspondence mode* with matched pairs available with the strategy in [14], and compare *partially labeled mode* with the three domain transfer SVMs, for which the fusion of multiple classifiers is the same. The average accuracy is provided in Table 4 and Table 5. Comparing Table 4 with Table 1 we find moderate performance gain by fusing multiple source views, while [14] sees a substantial increase. Overall, we accomplish a comparable accuracy with [14]. By comparing Table 5 to Table 2, it is also interesting to note that for the *partially labeled mode* the performance gain from a single source view is more significant on the baseline SVMs than on our approach, though our fused classifier still reports the best accuracy. This may imply that our view transfer method, which attempts to bridge two views via a smooth path, has more thoroughly exploited the connection between the source and the target, so that additional source views only contribute limited additional discrimination.

5. Conclusion

We propose an approach for cross-view action recognition, in which the source and target views are explicitly connected by a smooth virtual path represented as a sequence of linear transformations of action descriptors. The linear transformations are selected discriminatively based on a measure of mutual information in a training set between the virtual views and class labels. This view-transfer mechanism operates under a variety of weakly supervised scenarios (matched source-target pairs, partial target labels without matched pairs, and no target labels or matches), which have been considered quite separately. In all cases, our performance compares to or improves upon the state of the art.

Appendix: Approximate Gradient Ascent on $\mathbf{SO}(D)$

Consider the generic optimization problem

$$\max_{R\in\mathbf{SO}(D)} J(RA).$$

The steepest ascent direction is the gradient of J with respect to R. The gradient on $\mathbf{SO}(D)$ is defined as a vector $\nabla J \in \mathbf{so}(D)$, where $\mathbf{so}(D)$ is the associated Lie algebra, such that

$$\nabla J = \arg \max_{\xi\in\mathbf{so}(D),\|\xi\|=1} \frac{\partial J(A)}{\partial \xi}.$$

Here $\frac{\partial J(A)}{\partial \xi}$ is the directional derivative of J along ξ. To find the optimal ξ, we express it in terms of a linear combination of the basis axes of $\mathbf{so}(D)$:

$$\xi = \sum_{i,j} c_{i,j}(E_{i,j} - E_{j,i}), 2 \le i \le D, i+1 \le j \le D,$$

where we have employed the fact that $E_{i,j} - E_{j,i}, 2 \le i \le D, i+1 \le j \le D$ is the basis of $\mathbf{so}(D)$. Consequently, the search for a gradient direction becomes

$$\nabla J = \arg\max_{c_{i,j}} \frac{\partial J(A)}{\partial(\sum_{i,j} c_{i,j}(E_{i,j} - E_{j,i}))}, s.t. \sum_{i,j} c_{i,j}^2 = 1.$$

We first approximate the directional derivative along a linear combination of basis axes by the linear combination of directional derivatives along the axes, *i.e.*,

$$\nabla J \doteq \arg\max_{c_{i,j}} \sum_{i,j} c_{i,j} \frac{\partial J(A)}{\partial(E_{i,j} - E_{j,i})}, s.t. \sum_{i,j} c_{i,j}^2 = 1, \tag{9}$$

and then approximate the partial derivative $\frac{\partial J(A)}{\partial(E_{i,j}-E_{j,i})}$ by its finite difference as

$$\frac{\partial J(A)}{\partial(E_{i,j} - E_{j,i})} \doteq \frac{J(\exp(\epsilon(E_{i,j} - E_{j,i}))A) - J(A)}{\epsilon} \triangleq \Delta J_{i,j}$$

in which ϵ is a small positive number. As a result, the optimization (9) has close-form solution

$$c_{i,j} = \frac{\Delta J_{i,j}}{(\sum_{i',j'} \Delta J_{i',j'}^2)^{\frac{1}{2}}}.$$

We hence find an approximate gradient on $\mathbf{SO}(D)$, namely ∇J, and the final step is a line search along ∇J at a step length δ at $J(\exp(n\delta\nabla J)A)$. By jointly considering R_S and R_T we reach the greedy axis rotation algorithm in Algorithm 1.

Acknowledgement This work was supported by the National Science Foundation under awards IIS-0905243 and IIS-0835338.

References

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3), 2011.

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

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[4] T. Darrell, I. Essa, and A. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 18:1236 – 1242, 1996.

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Hierarchical motion history images for recognizing human motion. In *ICCV workshop:VS-PETS*, 2005.

[6] G. Elidan, G. Heitz, and D. Koller. Learing object shape: from drawings to images. In *CVPR*, 2006.

[7] A. Farhadi, M. Kamali, I. Endres, and D. Forsyth. A latent model of discriminative aspect. In *ICCV*, 2009.

[8] A. Farhadi and M. Tabrizi. Learning to recognize activities from the wrong view point. In *ECCV*, 2008.

[9] L. Fei-Fei, P. Perona, and R. Fergus. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 2006.

[10] D. Gavrila and L. Davis. 3d model-based tracking of humans in action: a multi-view approach. In *CVPR*, 1996.

[11] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.

[12] B. C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer, 2003.

[13] I. Junejo, E. Dexter, I. Laptev, and P. Patrick. View-independent action recognition from temporal self-similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 2011.

[14] J. Liu, M. Shah, B. Kuipersy, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011.

[15] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *ICCV*, 2007.

[16] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006.

[17] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *International Journal of Computer Vision.*, 66:83 – 101, 2006.

[18] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[19] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *CVPR*, 2008.

[20] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

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

[22] S. Seitz and C. Dyer. View-invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25:231 – 251, 1997.

[23] T. F. Syeda-Mahmood, M. Vasilescu, and S. Sethi. Recognizing action events from multiple viewpoints. In *IEEE Workshop on Detection and Recognition of Events in Video*, 2001.

[24] T.Arias, A.Edelman, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal of Matrix Analysis and Applications*, 20:303–353, 1998.

[25] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *ECCV*, 2008.

[26] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d examplars. In *ICCV*, 2007.

[27] D. Weinland, M. Ozuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV*, 2010.

[28] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding.*, 104:249 – 257, 2006.

[29] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, 2005.