Cross-View Action Recognition via View Knowledge Transfer

Jingen Liu[†], Mubarak Shah[‡], Benjamin Kuipers[†], Silvio Savarese[†]
[†]Dept. of Electrical Engineering and Computer Science, University of Michigan
{liujg,kuipers,silvio}@umich.edu

[‡]Dept. of Electrical Engineering and Computer Science, University of Central Florida
shah@eecs.ucf.edu

Abstract

In this paper, we present a novel approach to recognizing human actions from different views by view knowledge transfer. An action is originally modelled as a bag of visual-words (BoVW), which is sensitive to view changes. We argue that, as opposed to visual words, there exist some higher level features which can be shared across views and enable the connection of action models for different views. To discover these features, we use a bipartite graph to model two view-dependent vocabularies, then apply bipartite graph partitioning to co-cluster two vocabularies into visual-word clusters called bilingual-words *(i.e., high-level features), which can bridge the semantic gap across view-dependent vocabularies. Consequently, we can transfer a BoVW action model into a bag-of-bilingual-words (BoBW) model, which is more discriminative in the presence of view changes. We tested our approach on the IXMAS data set and obtained very promising results. Moreover, to further fuse view knowledge from multiple views, we apply a Locally Weighted Ensemble scheme to dynamically weight transferred models based on the local distribution structure around each test example. This process can further improve the average recognition rate by about 7%.*

1. Introduction

Recognizing human actions from videos has received considerable attention in computer vision during the past few years. The ever growing interest in characterizing human actions is in part due to the increasing number of real-world applications such as action centric video indexing and retrieval, human-computer interaction, activity monitoring in surveillance scenarios, and so on. However, it remains challenging to recognize actions from different views.

In general, human actions can cause spatiotemporal patterns of appearance or motion that can be in turn used for action recognition in videos. Based on this observation, many visual representations have been developed for recognizing actions. Some leading representations include space-time pattern templates [3, 27], shape features [36, 24, 7, 19], interest point based representations [15, 26, 29, 20], learned geometrical models of the human body parts [12], and motion/optical flow patterns [1, 25, 36]. Although most of features can be quite powerful in recognizing actions from similar views, their performance tend to dramatically decrease as the viewpoint changes. One reason for this is that the same action may look very different when observed from different angles (as shown in Fig. 1) and consequently the action models learned using low-level features become less discriminative. One possible solution is to maintain a separate classifier for each viewpoint. However, this may be impractical as it is difficult to acquire sufficient labeled examples for each view and it becomes infeasible as the number of action categories increases. Instead, we argue that it is more flexible to transfer action knowledge across views by exploring the connections between view-dependent features. In this work, we present a novel approach to discovering these connections for transferring action models across two views. This process is illustrated in Fig. 2.

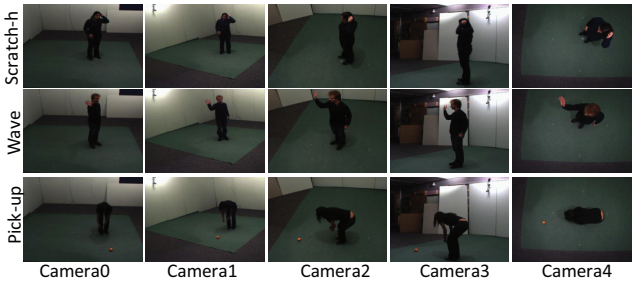

Figure 1. Exemplar frames from IXMAS multi-view data set. Each row shows one action viewed from different angles.

Our approach starts with two sets of **unlabelled** videos $\{v_i^1\}_{i=1}^N$ and $\{v_i^2\}_{i=1}^N$, where each set is taken from a different view. We construct individual visual vocabularies for both views and model an action video as a Bag of Visual Words (BoVW). We consider two action models built using two different vocabularies to be just like two articles written in two different languages. In order to tell whether these

two articles belong to a same category, we must either translate one of them to the other language or translate both of them into an interlingua, as is used in machine translation [14]. Similarly, before comparing two heterogenous action models we need to *transfer* ("translate") them into a common "language", say an action view "interlingua". Hence, generating such a view "interlingua" from two views (vocabularies) becomes critical. On the other hand, we notice that even if two vocabularies are mutually independent, they eventually describe the same set of action concepts (by "concept" we mean action category or sub-category). Therefore, we conjecture that there exist high-level semantic features that can bridge the semantic gap between two vocabularies, as well as between low-level features and action concepts. Our conjecture is consistent with the idea of constructing semantic vocabularies for action recognition [20][26] by modelling human actions in a hierarchical manner. Since high-level features are shared across two vocabularies, we call them *bilingual-words*, which form our action view "interlingua".

In order to abstract *bilingual-words*, we choose to model the relationship between two vocabularies as a weighted bipartite graph, where two disjoint vertex sets correspond to two vocabularies and edges connect visual-words from different vocabularies. Unlike [13] and [17], where a bipartite graph is used to model the *document-to-word* relationship, we employ a bipartite graph to model the *visual-word to visual-word* relationship. We compute the edge weight connecting two visual-words as their semantic similarity, which can be estimated from the training data matrix $\mathcal{M}$ (as shown in Fig. 2). Next, we apply a bipartite graph partitioning technique to establish a many-to-many mapping between the two vocabularies. Many algorithms have been proposed to partition bipartite graph [17][13]. We adapt the spectral graph co-clustering technique [17] to cluster both vocabularies simultaneously because their clusterings induce each other. The resulting visual-word clusters are *bilingual-words*. After clustering, we transfer the actions in different views from BoVW representation to a Bag-of-Bilingual-Words (BoBW) representation, such that new action representations are view invariant.

Afterwards, suppose we have labelled training examples for action classes $\mathcal{Z} = \{z_i\}_i^L$ in view 1 (*source* view,$\mathcal{V}^s$), then with the help of BoBW we can directly apply the classifiers learned for L classes in the *source* view to recognize novel actions in view 2, say *target* view $\mathcal{V}^t$. Note that the **unlabelled** videos used for discovering *bilingual-words* are NOT from action classes $\mathcal{Z}$. This guarantees that actions of classes $\mathcal{Z}$ in the *target* view are unknown to the classification models learned in the *source* view. In other words, we do NOT use actions from $\mathcal{Z}$ to discover *bilingual-words*.

Suppose we have n-1 source views $\{\mathcal{V}_i^s\}_1^{n-1}$, how can we fuse the knowledge transferred from varied *source* views

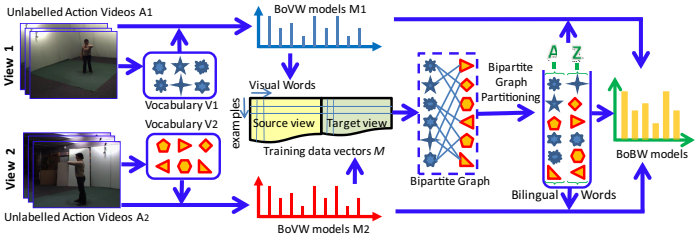

Figure 2. The process of discovering *bilingual-words*. Two vocabularies V_1 and V_2 are created from two views independently. Then action videos from different views are represented by varied BoVW models. By constructing a training data matrix $\mathcal{M}$, whose rows correspond to action examples associated with two corresponding instances from two views and columns denoting visual-words, we employ a bipartite graph to model the relationship between two vocabularies, and conduct graph partitioning to obtain the shared *bilingual-words*. As a result, the heterogenous BoVW models are transferred into BoBW models, which are discriminative under view changes.

to the *target* view? Which mechanism can we use to combine varied predictions provided by different transferred classification models? One solution would be to linearly combine them with fixed weights. However, we know that different test samples may favor predictions from different models M_i, because a good M_i with low test error implies that the training and testing data share similar data distributions. Thus, we employ the Locally Weighted Ensemble (LWE) scheme [18] to dynamically estimate combination weights for every test example by checking the consistency of data structures between the model predictions (near the test example) and the real data distribution (near the test example). Here, we assume that the testing examples are available.

1.1. Related Work

Several geometry-based approaches have been proposed for multi-view action recognition. For example, [3] employed epipolar geometry to perform view-invariant action recognition. The fundamental matrix constraints are applied on trajectories of an action captured from varied viewpoints. Parameswaran *et al.* [34] extracted view-invariant features by checking the planarity of the body joints. Rao *et al.* [4] presented an action representation to capture the dramatic changes of actions using spatiotemporal curvature of 2-D trajectories. These methods require reliable body joints detection and tracking, which are still challenging problems, thereby limiting their applications. Instead of using the geometric measurement of body joints, [6] and [30] performed 3D reconstruction for multi-view action recognition. However, 3D reconstruction requires strict alignment between views and is computationally expensive. Lv *et al.* [7] proposed a graphical model named *Action Net* connecting 2D key poses to represent 3D shapes for action recognition. Another group of approaches [32][5] tried to directly estimate 3D shapes and poses from multi-view inputs for action recognition. Weinland *et al.* [35] proposed to use the

hierarchical classifiers on local 3D HOG descriptors to obtain global classification decision, which handles viewpoint changes by learning a classifier on training examples taken from various views.

Rather than using geometry constraints, Junejo *et al.* [16][22] presented a very simple and interesting action representation called Self-Similarity Matrix, which is constructed by computing the pairwise similarity between any pair of figure-centric frames. However, experiments show that this approach performs poorly on the top views (very different from other views) of the IXMAS data set. Another very recent work [8] proposed to train a discriminative aspect model to handle the view invariance but it requires good parameter initialization.

Moreover, Farhadi *et al.*[2] employed Maximum Margin Clustering to generate split-based features in the source view, then a classifier (predictor) is trained to predict split-based features in the target view. Consequently, the split-based features are transferable across views. This approach is close to our work, but there are several significant differences. First, their work requires feature-to-feature correspondence at the frame-level to train a predictor, while we only need video-to-video correspondence (no cuboid-to-cuboid and word-to-word correspondence is required). Second, in their work the mapping is provided by a trained predictor, while our mapping between visual-words and bilingual-words is straightforward and efficient. No classifier is needed. Third, their approach does not have mutual communication between two views in the process of mapping construction, while our approach simultaneously co-cluster two vocabularies, which allows the vocabularies to exchange information during the construction of mapping.

Our work is also relevant to transfer learning, which has been explored in machine learning to transfer knowledge across different domains or tasks. Please refer to [23] for a review. Moreover, some works [31, 9, 10, 11] have used concepts from transfer learning for addressing object and image classification problems.

1.2. Our Contributions

In summary, we seek to solve the cross-view action recognition problem from a different perspective by *transferring* action models across views. In contrast to the aforementioned approaches for multi-view action recognition, our approach has the following advantages: i) It is more flexible. Unlike earlier view-invariant action recognition, our method does not require geometry constraints, human body joint detection and tracking, 3D reconstruction, and so on. ii) We do not require strict temporal alignment. Unlike [2], which require frame-level alignment, we only need to relate action videos performed by an actor under different views for discovering *bilingual-words*. iii) Our approach can simultaneously co-cluster visual-words of two views into *bilingual-words* by modeling two vocabularies as a bipartite graph. This co-clustering can make full use of the fact that the visual-words clustering of two views induce each other. iv) Our method is unsupervised when discovering *bilingual-words*. It only requires video-level correspondence across two views for the training data set without category labels. vi) To fuse knowledge transferred from varied views, we apply the Locally Weighted Ensemble method to successfully combine the prediction outputs of varied transferred models. We test our approach on the IXMAS multi-view action data set [6] and obtain very promising results.

2. Low-Level Action Representation

The primary feature used in our experiments is the spatiotemporal interest point feature. To detect the interest points, we applied a 2D Gaussian filter to a video, followed by a 1D-Gabor, and the interest points are detected at the local maximum response. This detector is proposed in [29]. The parameters for the two filters are set to $\sigma = 2$ and $\tau = 1.5$ in this paper. Then the ST volumes around the points are extracted and gradient-based descriptors are learned by PCA. These descriptors are further quantized to visual words by k-means clustering. Afterwards, each action is represented by a histogram vector $\mathbf{h}$ of visual-words (i.e., BoVW model).

The 3D interest point feature is able to capture rich local motion features, but not the global shape. So we further extract shape and flow feature, called the shape-flow descriptor, as an auxiliary feature from each frame, as is proposed in [33]. We believe it is complementary to the 3D interest point feature. Specifically, from every frame three channels features are extracted: horizontal optical flow, vertical optical flow, and silhouette. PCA is used to reduce the dimensionality of all these features. In order to capture temporal information, the feature information from neighbor frames are integrated into the current frame descriptor by simply concatenating feature vectors. See [33] for more details. These descriptors are also quantized into visual words, and an action video is represented by a histogram of words.

3. View Knowledge Transferring

Let $D^s_{Tr} = \{(\mathbf{h}^s_d, l_d)\}^n_{d=1}$ be a training dataset, where $\mathbf{h}^s_d$ (i.e., a histogram of visual-words) is a training example from the *source* view and $l_d \in \mathcal{Z}$ is the class label. We learn a classifier M_i for classes $\mathcal{Z}$ from training data D^s_{Tr}. Our goal is to transfer view knowledge across views such that the classifier M_i can be used to recognize novel actions taken in the *target* view. In this task, M_i does not see training examples of classes $\mathcal{Z}$ from the *target* view. To achieve this goal, we discover the $bilingual - words$ from another **unlabelled** training dataset $D^U_T = I_d = (\mathbf{h}^s_d, \mathbf{h}^t_d)^m_{d=1}$, where $(h)^s_d$ and $\mathbf{h}^t_d$ are two corresponding action videos from the *source* and *target* view respectively. Note that the action category of any video in D^U_T is not in $\mathcal{Z}$, which means the

action classes of D_T^U and the classes $\mathcal{Z}$ are disjoint. As such, we can guarantee that *bilingual-words* are not affected by the actions of $\mathcal{Z}$ from the *target* view. The details of our approach are addressed in this section. Here, we consider pairwise views (vocabularies) $\mathcal{V}^s$ and $\mathcal{V}^t$. We first build a bipartite graph for two vocabularies (section 3.1), and then apply bipartite graph partitioning to simultaneously co-cluster the two vocabularies into *bilingual-words* in section 3.2.

3.1. Bipartite Graph Modelling

Let G=(V,E,$\mathbb{W}$) be a graph, where V, E and $\mathbb{W}$ are the vertex set, edge set and weight matrix respectively. G is a bipartite graph if $V = X \bigcup Y$ with $X \bigcap Y = \emptyset$ and each edge connects two vertices in X and Y. Here, X and Y represent two vocabularies $\mathcal{V}^s$ and $\mathcal{V}^t$ for the source and target view respectively. The weight matrix $\mathbb{W} = \begin{pmatrix} 0 & S \\ S^T & 0 \end{pmatrix}$, where S is a $|\mathcal{V}^s| \times |\mathcal{V}^t|$ matrix representing the similarity between any pair of visual-words from two vocabularies.

Matrix S can be constructed from the unlabelled training data $D_T^U = I_d = (\mathbf{h}_d^s, \mathbf{h}_d^t)_{d=1}^m$ as follows. Each example I_d corresponds to two instances $\mathbf{h}_d^s$ and $\mathbf{h}_d^t$ (two histograms over vocabulary $\mathcal{V}^s$ and $\mathcal{V}^t$ respectively).

To better capture the relationship between data d and visual-words x, we further replace each entry of $\mathbf{h}_d^s(x)$ and $\mathbf{h}_d^t(x)$ by the Pointwise Mutual Information (PMI) [20] between d and x, which is a measure of association. If histogram $\mathbf{h}$ is normalized, we approximately estimate PMI of video d and visual-word x as follows,

$$pmi(d,x) \approx log(\frac{\mathbf{h}_{dx}}{\sum_d \mathbf{h}_{dx} \sum_x \mathbf{h}_{dx}}). \quad (1)$$

where we treat histogram entry $\mathbf{h}_{dx}$ as an empirical joint probability $p(d,x)$. As a result, we obtain new action representation $\hat{\mathbf{h}}_d^s$ and $\hat{\mathbf{h}}_d^t$, which captures more semantic correlation between actions (videos) and visual-words. From the updated feature vectors, we then construct a $m \times (|\mathcal{V}^s|+|\mathcal{V}^t|)$ matrix $\mathcal{M}$, whose rows correspond to action examples and columns to visual-words (Column 1 to $|\mathcal{V}^s|$ correspond to visual-words in vocabulary $\mathcal{V}^s$, as shown in Fig 2). As such, each visual word is embedded into the column space of $\mathcal{M}$. Afterwards, an entry $\mathcal{S}(i,j)$ of S can be computed as $exp(d(x_i^s, x_j^t)/2\sigma^2)$, where $d(x_i^s, x_j^t)$ is the Euclidean distance between two visual words in the column space of matrix $\mathcal{M}$.

3.2. Bilingual-Words Discovery

With the constructed bipartite graph G on $\mathcal{V}^s$ and $\mathcal{V}^t$, we seek to discover the *bilingual-words*. This section describes how to discover *bilingual-words* by graph partitioning. We start with graph bi-partitioning.

3.2.1 Graph Bi-Partitioning.

A bi-partition of graph $G(V,E,W)$ ($V = \mathcal{V}^s \bigcup \mathcal{V}^t$) is defined by $\mathbf{\Pi}(V_1, V_2)$, where $V_1 = A^s \bigcup A^t$ and $V_2 = \bar{A}^s \bigcup \bar{A}^t$ ($A^i \bigcup \bar{A}^i = \mathcal{V}^i, i = s,t$). For any two subsets of vertices $S \subset \mathcal{V}^s$ and $T \subset \mathcal{V}^t$, we define $f_{\mathcal{W}}(S,T) = \sum_{i \in S, j \in T} w_{ij}$, which measures the association between sets S and T. To partition graph vertices into clusters, we seek a partition $\mathbf{\Pi}(V_1, V_2)$, such that Eq 2 is minimized.

$$cut(V_1, V_2) = f_{\mathcal{W}}(A^s, \bar{A}^t) + f_{\mathcal{W}}(\bar{A}^s, A^t). \quad (2)$$

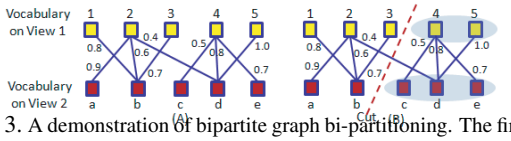

Figure 3. A demonstration of bipartite graph bi-partitioning. The first row (yellow nodes) and second row (red nodes) correspond to two vocabularies, respectively. The numbers attached to edges are the weights. (A) Before the partition. (B) After the partition. The red dotted line gives the optimal cut, resulting in two clusters (1 2 3; a b), (4 5; c d e). As a result, the mappings $\{(123),(ab)\}$ and $\{(45),(cde)\}$ correspond to two bilingual words.

One toy example of a bipartite graph and its bi-partition is shown in Fig. 3. The objective function can be approximately solved by spectral clustering, which starts from constructing the Laplacian matrix L as follows,

$$L(i,j) = \begin{cases} -w_{i,j} & \text{if } e_{ij} \in E \\ \sum_k w(i,k) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

From the perspective of graph embedding, bi-partitioning a graph can be understood as projecting vertices onto two points +1 and -1 in the discrete case. A good projection shall minimize $\sum_{(i,j)\in E} w_{ij}(q_i - q_j)^2$, where q_i is the projection value of vertex i. Intuitively, larger w_{ij} will result in higher possibility of projecting vertices i and j onto a same point. It can be shown that the following equation holds,

$$cut(V_1, V_2) = \frac{1}{4}\mathbf{q}^T L \mathbf{q} = \frac{1}{4} \sum_{(i,j)\in E} w_{ij}(q_i - q_j)^2, \quad (4)$$

where $\mathbf{q}$ is a vector of projected values. Obviously, one trivial solution of the minimizing problem is to project all vertices onto either +1 or -1. However, we look for a objective function that can achieve not only minimized cut value but also a balanced partition. Hence, normalized graph cuts is proposed as [21][17],

$$Ncut(V_1, V_2) = \frac{cut(V_1, V_2)}{f_{\mathcal{W}}(A^s, A^t)} + \frac{cut(V_1, V_2)}{f_{\mathcal{W}}(\bar{A}^s, \bar{A}^t)}. \quad (5)$$

The partition that minimizes $Ncut(V_1, V_2)$ is the optimal partition of graph G. It can be proved that the second smallest eigenvector of the generalized eigenvalue problem $Lz = \lambda Dz$ (where $D(i,i) = \sum_j w_{ij}$) provides a real relaxed solution.

3.2.2 Efficient Solution via SVD Decomposition

For partitioning a bipartite graph, a more computationally efficient solution is proposed in [17] to perform SVD on a variation of $\mathcal{S}$. For a bipartite graph G, we have,

$$L(i,j) = \begin{pmatrix} D_1 & -S \\ -S^T & D_2 \end{pmatrix}, \text{and } D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \quad (6)$$

where $D_1(i,i) = \sum_j w_{ij}$ and $D_2(j,j) = \sum_i w_{ij}$. Let matrix $\hat{S} = D_1^{-1/2} S D_2^{-1/2}$, it can be proved that the second eigenvector of L can be expressed in terms of left and right singular vectors of $\hat{S}$ as follows (refer to [17]),

$$z_2 = \begin{pmatrix} D_1^{-1/2} u_2 \\ D_2^{-1/2} v_2 \end{pmatrix} \quad (7)$$

In fact, the left and right singular vectors u_2 and v_2 provide us the bipartitions of $\mathcal{V}^s$ and $\mathcal{V}^t$ respectively. Then the k-means algorithm can be used to cluster the 1-dimensional data of z_2. In this way $\mathcal{V}^s$ and $\mathcal{V}^t$ are simultaneously clustered. Since two vocabularies are correlated to each other, the clustering on one vocabulary naturally induces the clustering of the other.

3.2.3 K-way Bipartite Graph Partitioning

The two-way clustering can be extended to K-way clustering by either recursively partitioning the sub-set or applying k-means on $l = \lceil logK \rceil$ eigenvectors. The procedure of discovering *bilingual-words* is summarized in table 1.

3.2.4 Action Models Transfer

The bilingual-words $\mathcal{C}_i = (\mathcal{V}_i^s, \mathcal{V}_i^t)_1^K$ basically serve as a many-to-many mapping between two set of visual-words. The clustering can be regarded as a mapping function defined on visual-words: $\mathcal{C}_j = f(x_i)$. As a result, we can transfer an action from a BoVW model to a Bag of Bilingual-words (BoBW) model as $\mathbf{h}(\mathcal{C}_j) = \sum_{x_i:\mathcal{C}_j=f(x_i)} \mathbf{h}(x_i)$. With actions in both views represented by BoBW model, we obtain a transferable classification model M_i on $\mathcal{V}^s$, and directly use it on $\mathcal{V}^s$ for recognition.

4. Classification Models Fusion

Suppose there are n-1 source views $\mathcal{V}^s$ and one target view $\mathcal{V}^t$, one problem is how to combine the transferred knowledge from each source view to recognize novel actions in the target view. Because each source view classifier/model M_i may provide partial knowledge, we employ the Locally Weighted Ensemble (LWE) [18] to effectively fuse all predictions when all test examples are available.

To generalize the problem, let x and y be an action instance and its label, respectively. Given classifiers $\{M_i\}_1^{n-1}$ and testing data set D_{Ts}^t in the target view, the recognition task is to estimate a posterior probability as follows,

$$p(y|x) = \sum_{i=1}^{n-1} p(y|x, M_i, D_{Ts}^t) p(M_i|D_{Ts}^t), \quad (8)$$

where probability $p(y|x, M_i, D_{Ts}^t)$, namely $p(y|x, M_i)$ due to $x \in D_{Ts}^t$, is the output of M_i on x and $p(M_i|D_{Ts}^t)$ is the probability to select M_i given known testing data D_{Ts}^t. As

Objective: Given D_T^U, an unlabelled training action data and two corresponding vocabularies of $\mathcal{V}^s$ and $\mathcal{V}^t$, discover K *bilingual words* $\{\mathcal{C}_i = (\mathcal{V}_i^s, \mathcal{V}_i^t)\}_1^K$.

1. Create the training data matrix $\mathcal{M}$ (with the same structure shown in Fig. 2 (b)) for the unlabelled training data set D_T^U.
2. Construct the correlation $\mathcal{S}$ between two sets of vertices of G, namely $\mathcal{V}^s$ and $\mathcal{V}^t$. Each entry $\mathcal{S}(i,j)$ is computed as $exp(d(x_i^s, x_j^t)/2\sigma^2)$.
3. Compute $D_1(i,i) = \sum_j w_{ij}$, $D_2(j,j) = \sum_i w_{ij}$, and matrix $\hat{S} = D_1^{-1/2} S D_2^{-1/2}$.
4. Apply SVD on $\hat{S}$, and select $l = \lceil log_2 k \rceil$ number of its left and right singular vectors: $U = (u_2, ..., u_{l+1})$ and $V = (v_2, ..., v_{l+1})$.
5. Composite matrix Z = $\begin{pmatrix} D_1^{-1/2} U \\ D_2^{-1/2} V \end{pmatrix}$, whose size is $(|\mathcal{V}^s| + |\mathcal{V}^t|) \times l$.
6. Run k-means clustering on l-dimensional row data of Z to obtain K clusters of visual-words, which are the bilingual-words $\{\mathcal{C}_i = (\mathcal{V}_i^s, \mathcal{V}_i^t)\}_1^K$.

Table 1. The procedure of discovering bilingual-words.

$x \in D_{Ts}^t$, $p(M_i|D_{Ts}^t)$ is actually equal to $p(M_i|x)$, which is locally adapted for each x and represents the effectiveness of model M_i for x in D_{Ts}^t. By defining the weight $w_i^x = p(M_i|x)$, we have $p(y|x) = \sum_{i=1}^{n-1} p(y|x, M_i) w_i^x$. Theoretically, the error of LWE on each test example x is not greater than that of any single model (see [18]).

If $p(y|x)$ is known, we may estimate the weights w_i^x by minimizing the square error between the prediction and ground truth. However we are unable to x is in D_{Ts}^t, we are unable to obtain the real value of $p(y|x)$. Intuitively, however, a model M_i should have higher weights for x if x is covered by the knowledge of M_i. Nevertheless, we still have no idea which region is covered by M_i. Based on the clustering assumption [28] that $p(y|x)$ is not expected to change much in a dense area, which means the decision boundary probably occurs at the area where $p(x)$ is lower, if the clustering boundary for the region where x is located agrees with the decision boundary of M_i, we can assume that the distribution $p(y|x, M_i)$ is similar to the true distribution $p(y|x)$, which means we can assign higher weight to model M_i at x. In other words, if the outputs of M_i at the area surrounding x have higher consistency with the clustering results, M_i can obtain a higher weight at x.

As a result of these observations, we employ a clustering-based approach to estimate the weight associated with M_i for x. We first construct two graphs, $G_T = (V, E_T)$ and $G_M = (V, E_M)$, for clustering and classification respectively, where $V = D_{Ts}^t$. In graph G_T, if two test examples are classified into the same category, there is an edge between them. For graph G_M, the edges exist be-

(%)	Camera 0		Camera 1		Camera 2		Camera 3		Camera 4	
	woTran	wTran	woTran	wTran	woTran	wTran	woTran	wTran	woTran	wTran
Cam0			14.40	75.46	10.69	64.40	10.61	67.68	19.09	65.99
Cam1	16.12	75.72			11.11	64.23	7.41	68.10	9.22	56.02
Cam2	10.27	70.33	11.80	66.25			12.90	71.34	8.08	62.42
Cam3	11.15	73.74	8.59	65.62	9.98	71.30			9.30	58.04
Cam4	8.80	71.34	8.46	66.29	9.22	70.88	10.06	63.55		

Figure 4. Performance comparison of action recognition with and without model transfer. The rows and columns correspond to training and testing view, respectively. *woTran*-columns and *wTran*-columns contain the results of recognition with and without action model transfer. The average accuracies are 10.9% and 67.4% for *woTran* and *wTran* respectively.

tween any pair of test examples that are clustered into the same group. If x's neighbors on both graphs have a large overlap, it implies higher local weight. So we can use the percentage of overlap between examples V_T (neighbors of x in G_T) and V_M (neighbors of x in G_M) to approximate the weight: $w_k^x \propto s(G_M, G_T, x) = \frac{|V_M \bigcap V_T|}{|V_M \bigcup V_T|}$, where we can consider $s(G_M, G_T, x)$ as the degree of consistency.

5. Experimental Results and Discussion

5.1. Data Set and Experimental Setup

We test our approach on the IXMAS multi-view action data set [6] which contains eleven daily-live actions. Each action is performed three times by twelve actors taken from five different views: four side views and one top view (see Fig.1 for some examples).

We extract at most 200 cuboids from each video. Each cuboid is represented by a 100-dimensional descriptor learned using PCA. We subsequently apply k-means to quantize these interest point descriptors into N=1,000 visual words. With these basic view-dependent vocabularies, we conduct experiments on all possible pairwise view combinations (twenty in total for five views) to evaluate the proposed transfer method. For a better comparison to [2], we follow the same *leave-one-action-class-out* strategy, which means that each time we only consider one action class for testing in the target view (this action classes is not used to construct *bilingual words*). The final results are reported in terms of average accuracy for all action classes in each view. The training data D_T^U used for discovering bilingual-words are randomly selected from actions excluding the orphan action. With learnt bilingual-words, six multi-class classifiers are trained on the source view in a 6-fold cross-validation manner and employed to recognize actions from the target view. SVM with histogram intersection kernel is chosen as our classifier.

5.2. Transferring models across pairwise views

In this section, we want to verify the performance of transferring models across pairwise views. Initially, an action video is represented by BoVW model. We first try to recognize novel actions from *target* view by directly using classifiers trained on *source* view without model transfer. The results are shown in Fig. 4 (i.e., *woTran*-columns).

(%)	Camera 0		Camera 1		Camera 2		Camera 3		Camera 4	
	A	B	A	B	A	B	A	B	A	B
Cam 0			68.27	70.16	60.47	68.14	64.33	65.87	30.56	50.80
Cam 1	72.87	75.88			53.36	60.65	60.27	64.53	27.57	50.42
Cam 2	57.23	65.44	31.39	63.55			73.72	70.58	50.84	64.44
Cam3	48.06	62.86	30.68	64.52	71.93	65.87			27.57	54.61
Cam 4	31.35	62.33	17.46	60.52	59.47	68.18	39.56	61.45		

Figure 5. Performance comparison of two transfer strategies: appearance-similarity-based (A columns) and bipartite-graph-based (B columns) strategy. The average accuracies for A and B are 48.8% and 63.5% respectively.

Note that the action models from two views are heterogenous, we are not surprised to see the results are not much better than random guess. On the other hand, we transfer all actions in both views from BoVW models to BoBW models, followed by action classification. The number of discovered *bilingual-words* is 100 for this experiment. The results are shown in Fig. 4 (i.e., *wTran*-columns). The performance is very promising considering that the classifiers are trained on data taken from different views. It also demonstrates that BoBW models are discriminative under view changes. To further demonstrate the performance of cross-view recognition, we applied single view classification, which means we trained and tested the classifiers on the same view. The average accuracies are 82.01%, 80.99%, 78.32%, 82.41% and 75.57% for views 0 to 4 respectively. We see that the performance of cross-view recognition is very close to single view classification for most view combinations.

Additionally, we want to demonstrate the performance of appearance-based transfer. In fact, a visual-word is a m-dimensional vector (m=100 in this paper), which is the center of a group of cuboid descriptors in k-means clustering. Hence, it is very natural to build the mapping between two vocabularies by comparing the distance between visual-words in R^m space. In the second experiment we examine the transfer based on appearance similarity. For computational simplicity, we use the vocabulary generated in the source view to quantize the cuboid descriptors for the target view. So there is one physical vocabulary (from the source view) with one mirror in the target view. To compare with our bipartite-graph-based approach, we treat one vocabulary as two virtual vocabularies and perform the bipartite graph co-clustering to generate 100 *bilingual-words*. The results for both experiments are reported in Fig. 5. Columns A show the results of appearance-similarity-based method, while columns B for bipartite-graph-based method. We can see that for some combinations such as (*camera 1*, *camera 0*) and (*camera 2*, *camera 3*), both methods have very competitive results. We can conjecture that these pairs of views might be very similar, which means they may share similar appearance for some actions. We check the data set and find out some actors were oriented differently (relative to cameras) compared to the majority of actors, which actually makes these pairs of views share similar appearance to some extent. However, the bipartite graph co-clustering per-

	Camera 0				Camera 1				Camera 2				Camera 3				Camera 4			
(%)	Ours	A	B	C	Ours	A	B	C	Ours	A	B	C	Ours	A	B	C	Ours	A	B	C
C0					79.9	72	77.6	79	76.8	61	69.4	79	76.8	62	70.3	68	74.8	30	44.8	76
C1	81.2	69	77.3	72					75.8	64	73.9	74	78.0	68	67.3	70	70.4	41	43.9	66
C2	79.6	62	66.1	71	76.6	67	70.6	82					79.8	67	63.6	76	72.8	43	53.6	72
C3	73.0	63	69.4	75	74.1	72	70.0	75	74.4	68	63.0	79					66.9	44	44.2	76
C4	82.0	51	39.1	80	68.3	55	38.8	73	74.0	51	51.8	73	71.1	53	34.2	79				
Ave.	79.0	61	63.0	74	74.7	67	64.3	77	75.2	61	64.5	76	76.4	63	58.9	73	71.2	40	46.6	72

Figure 6. Cross-view action recognition performance of different approaches on IXMAS dataset. Columns 'Ours', A, B and C correspond to our approach, [2]'s approach, [16]'s approach and [8]'s approach. The overall average accuracies are 75.3%, 58.1%, 59.4% and 74.4% for them respectively (best viewed in PDF file).

form much better for the remaining combinations. This is because *bilingual-words* produce a more meaningful mapping between two (virtual) vocabularies.

As was mentioned in Sec 2, a shape-flow descriptor can capture global shape information and temporal information. It is complementary to the 3D interest point feature. So in this experiment, we also extract a vocabulary (size of 500) of shape-flow descriptor for each view. Then for each view, we have a hybrid vocabulary (size of 1,500) of 3D interest point and shape-flow features. Then we conduct cross view recognition for 20 view combinations. The results are listed in Fig.6. As compared to the results in Fig.4, the performance has been improved about 8% in average accuracy by introducing the shape-flow descriptor. We also list three additional sets of state-of-the-art results on cross-view action recognition reported in [2], [16] and [8] in Fig. 6 for comparison. We are particularly interested in [2], since they also transfer action models across views. We notice our performance is better than that of [2] and [16]. It is interesting to note that our method can perform much better (about 10% − 40%) than the [2] and [16] when *camera 4* is involved in either training or testing. If we look at the data set (see Fig. 1), *Camera 4* was set above the actors, so it captured totally different actions. Hence, we believe the recognition results on *Camera 4* are more important for evaluating a cross-view action recognition approach. Our performance is competitive to that of [8], even though our approach is unsupervised whereas [8] requires supervision.

Compared with [16] and [8], one limitation of our approach is that it implicitly assumes the target view in the test stage is known, whereas the view from which the test video is taken is usually unknown. To cope with this problem, one option is to estimate the view of the test video before recognition, just like Farhadi *et al.* in [2] which achieves high accuracy in view estimation. Alternatively, we can also simply apply our approach by treating every view as the target view, and then the view that gives the highest confidence on action recognition is selected as the target view for the test video.

(%)	Camera 0	Camera 1	Camera 2	Camera 3	Camera 4	Average
LWE	**86.6**	**81.1**	**80.1**	**83.6**	**82.8**	**82.8**
Global	80.6	78.5	78.0	78.3	73.0	77.7
Junejo et al.	74.8	74.5	74.8	70.6	61.2	71.2
Liu et al.	76.7	73.3	72.0	73.0	N/A	73.8
Weinland et al.	86.7	89.9	86.4	87.6	66.4	83.4

Figure 7. Performance comparison of different model fusion methods, as well as state-of-the-art approaches. Row LWE and Global shows the recognition rates of model fusion on each testing views using LWE and Global weighting methods respectively. And the following three rows list performance of approaches proposed by Junejo *et al.* [16], Liu *et al.* [19] and "Weinland *et al.*[35].

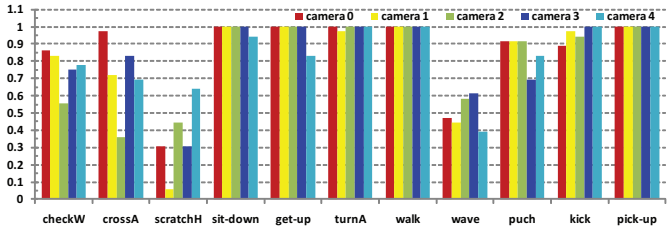

Figure 8. Recognition performance of transferred model fusion on each action category.

5.3. Multiple transferred models fusion

In the above experiments, for each target (testing) view there are four recognition results provided by the classification models trained on the rest of the four views. We employ the LWE approach to combine the results. K-means clustering is used to group the testing data into N clusters. Fig. 7 row *LWE* shows the performance of model fusion with N=5 on the output of our experiments in Fig. 6). By comparing the results in Fig. 6 (the row of "ave") and 7, we see LWE obtains significant performance improvements (about 5% to 10%) on each testing view. For comparison, we also try a global weighting method, which assign 1 to each model for all testing examples. The results are also shown in Fig. 7. Obviously, LWE can perform better than the global weighting method.

Fig. 7 lists some results of the state-of-the-art approaches on the IXMAS dataset as well. Our performance is better than that of [19] and [16], and competitive to [35]. In fact, the other three approaches trained their classifiers using examples from all five views, but our approach train our classifiers on four views but not on the testing view (target view). Specifically, the classification conducted on the target view are accomplished by combining the models transferred from the other four views for recognition. No classifiers are trained on the target view. We believe this capability is very attractive.

Moreover, it is very interesting to look at the recognition rate of each action category in Fig. 8. We can observe that the task of action model transfer is very hard for some ac-

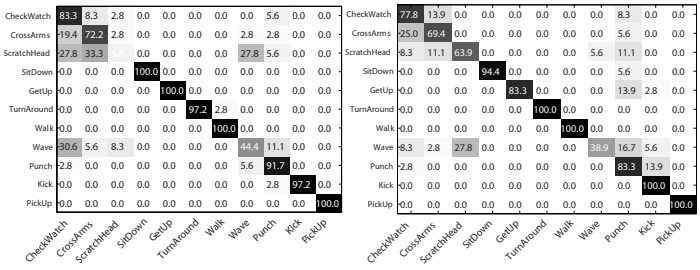

Figure 9. The confusion tables for *camera 1*(Left) and *camera 4*(Right) (Best viewed in PDF file).

tions and some views. For example, "scratch-head" has less than a 5% recognition rate for *camera 1*. One of the reasons might be that the majority of the motion is blocked by human bodies in *camera 1* and thus very slight motion can be observed in this view. This can also explain that the actions only associated with arms (e.g., "check-watch", "cross-arms" etc.) have much lower recognition rate than those actions associate with entire human body ("sit-down","turn-around",etc.). Moreover, it is surprising that some actions such as "walk" and "turn-around" which are supposed to share very similar motions achieve very good performance. We further check the confusion tables generated from the outputs of LWE method (Fig. 9 shows two confusion tables for *camera 1* and *camera 4*, respectively.), and we notice some actions are hard to be distinguished when viewed from a certain viewpoint. This may be due to that for some actions the models are difficult to be transferred from varied views to a certain view. For instance, most "scratch head" actions in the side view *camera 1* are misclassified as "wave" and "cross arms", while there is less confusion in the top view *camera 4*.

6. Conclusion

In this paper, we address the problem of recognizing an unknown action from an unseen (target) view using training data taken from other (source) views. For this purpose, we propose a novel bipartite-graph-based approach to learn *bilingual-words* from two view-dependent vocabularies in an unsupervised manner. By means of the *bilingual-words*, we are able to transfer actions from both views from BoVW models to BoBW models, which are discriminative under view changes. We have extensively tested our approach on the publicly available IXMAS multi-view data set and obtained very promising results. Additionally, in order to fuse the decisions provided by different source views, we employed the Locally Weighted Ensemble method to dynamically combine the predictions from varied source views for each test example. Thus, we can further improve average accuracy by about 7%.

7. Acknowledgements

Thanks for helpful comments from colleagues and the reviewers. This work is supported by the National Science Foundation (Grant CPS-0931474).

References

[1] A. A. Efros, A. C. Berg, G. Mori and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

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

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

[4] C. Rao, A. Yilmaz, and M. Shah. View-invariant reprentation and recognition of actions. In *IJCV*, 2002.

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

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

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

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

[9] L. Fei-Fei, P. Perona, and R. Fergus. One-shot learning of object categories. *PAMI*, 28(4), 2006.

[10] M. Fink. Object classification from a single example utilizing class relevance metrics. In *NIPS*, 2004.

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

[12] G. Mori, X. Ren, A. A. Efros and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, 2004.

[13] H. Zha, X. He, C. Ding, H. Simon and M. Gu. Bipartite graph partitioning and data clustering. In *CIKM*, 2001.

[14] W. Hutchins and H. Somers. *An introduction to machine translation*. Academic Press New York, 1992.

[15] I. Laptev and T. Lindeberg. Space time interest points. In *CVPR*, 2003.

[16] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez. Cross-view action recognition from temporal self-similarities. In *ECCV*, 2008.

[17] I.S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *SIGKDD*, 2001.

[18] J. Gao, W. Fan, J. Jiang, and J. Han. Knowledge transfer via multiple model local structure mapping. In *SIGKDD*, 2008.

[19] J. Liu, S. Ali and M. Shah. Recognizing human actions using multiple features. In *CVPR*, 2008.

[20] J. Liu, Y. Yang and M. Shah. Learning semantic visual vocabuaries using diffusion distance. In *CVPR*, 2009.

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. In *PAMI*, 2000.

[22] I. Junejo, E. Dexter, I. Laptev, and P. Patrick. View-independent action recognition from temporal self-similarities. *PAMI*, 33(1), 2011.

[23] S. Kaskil and J. Peltonen. Learning from relevant tasks only. *Springer, Heidelberg*.

[24] K.M. Cheung,S. Baker and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *CVPR*, 2003.

[25] J. Little and J. E. Boyd. Recognizing people by their gait: The shape of motion. *Journal of Computer Vision Research*.

[26] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

[27] M. Blank, M. Irani and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[28] B. S. O. Chapelle and A. Zien. Semi-supervised learning. *The MIT Press*, 2006.

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

[30] P. Yan, S.M. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *CVPR*, 2008.

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

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

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

[34] V. Parameswaran and R. Chellappa. View invariance for human action recognition. In *IJCV*, 2006.

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

[36] Z. Lin, Z. Jiang and L.S. Davis,. Recognizing actions by shape-motion prototype trees. In *CVPR*, 2009.