Cross-view Action Recognition via Dual-Codebook and Hierarchical Transfer Framework

Chengkun Zhang, Huicheng Zheng, Jianhuang Lai

School of Information Science and Technology, Sun Yat-sen University, 510006
Guangzhou, China
zhangchk@mail2.sysu.edu.cn, {zhenghch, stsljh}@mail.sysu.edu.cn

Abstract. In this paper, we focus on the challenging cross-view action recognition problem. The key to this problem is to find the correspondence between source and target views, which is realized in two stages in this paper. Firstly, we construct a Dual-Codebook for the two views, which is composed of two codebooks corresponding to source and target views, respectively. Each codeword in one codebook has a corresponding codeword in the other codebook, which is different from traditional methods that implement independent codebooks in the two views. We propose an effective co-clustering algorithm based on semi-nonnegative matrix factorization to derive the Dual-Codebook. With the Dual-Codebook, an action can be represented based on Bag-of-Dual-Codes (BoDC) no matter it is in the source view or in the target view. Therefore, the Dual-Codebook establishes a sort of codebook-to-codebook correspondence, which is the foundation for the second stage. In the second stage, we observe that, although the appearance of action samples will change significantly with viewpoints, the temporal relationship between atom actions within an action should be stable across views. Therefore, we further propose a hierarchical transfer framework to obtain the feature-to-feature correspondence at atom-level between source and target views. The framework is based on a temporal structure that can effectively capture the temporal relationship between atom actions within an action. It performs transfer at atom levels of multiple timescales, while most existing methods only perform video-level transfer. We carry out a series of experiments on the IXMAS dataset. The results demonstrate that our method obtained superior performance compared to state-of-the-art approaches.

1 Introduction

Recently, action recognition has gained much attention in computer vision due to its extensive applications in video surveillance [26], human-machine interaction, medical assistance for elders [1, 25], etc. Previous work has proposed some popular features for recognizing actions, such as space-time point features [5, 14, 19], shape features [3, 17, 18, 32], optical-flow features [17, 6]. These features have led to remarkable action recognition performance for typical scenarios where there are only limited viewpoint variations. However, when the view point changes significantly, traditional approaches for action recognition would suffer from serious drop of performance [10, 11].

Several approaches have been proposed to address action recognition across views. One category of approaches rely on 3D reconstruction [27, 30, 21, 9]. Some other approaches directly use 2D image or geometric constraints across different views [31, 22, 23, 8, 16]. Besides, temporal self-similarities have also been exploited for view-invariant feature extraction [13, 12].

Recently, some works seek to transfer action model from the source view to the target view, and have received satisfactory results. Farhadi et al. [7] proposed to use Maximum Margin Clustering to build split-based features for frames. Then, they transfer them among corresponding frames across different views. Liu et al. [20] constructed "bilingual words" by using the co-occurrence of visual words from source and target views. By representing the videos as a bag of bilingual words (BoBW), they can transfer the action model at the video-level across different views. Zheng et al. [34] proposed to build a transferable dictionary pair by forcing the videos of the same action to have the same sparse coefficients across views. These approaches are attractive, for they have little dependence on the 3D model reconstruction of actions, reliable body joints detection and tracking, and the geometric information across different views.

However, existing approaches generally implement codebooks separately trained in the source and target views, which cannot guarantee reliable correspondence between visual words. This will degrade the performance of these approaches on transferring action models from the source view to the target view. In this paper, we propose to construct a Dual-Codebook for the source and target views. Unlike traditional codebook learning approaches, we model the construction of Dual-Codebook as a co-clustering problem and propose an effective algorithm to solve it. Our Dual-Codebook consists of two codebooks, one for each of the two views. Since it is obtained by co-clustering, not isolated clustering in source and target views, each codeword in one codebook has a corresponding codeword in the other codebook. This means that our Dual-Codebook contains basic view-correspondence, i.e., a codebook-to-codebook correspondence across two views. To our knowledge, this has never been explored before.

Furthermore, existing approaches usually transfer action models at the video-level, ignoring the sequential composition of atom actions during the execution of the full action. Such a strategy will not be discriminative enough when multiple actions contain similar atom actions following different occurrence orders, such as sit down and get up. To resolve this problem, we propose a novel hierarchical framework for transferring action models across different views. Specifically, we divide action videos into several segments along the time dimension at each level of this framework. Each segment contains an atom action within a short time interval. Then, we enforce similar sparse representations for each pair of corresponding segments from the source and target views by learning a transferable pairwise dictionary. The inputs of the learning procedure are the Bag-of-

Dual-Codes (BoDC) of these segments. This is different from implementations of existing dictionary learning strategies, which are usually based on separately generated codebooks. The representation generated in this way is more robust to view changes, as demonstrated experimentally.

This paper presents the following contributions.

1. A Dual-Codebook is constructed for source and target views. We propose an effective co-clustering algorithm to learn the Dual-Codebook. The Dual-Codebook achieves the codebook-to-codebook correspondence across different views.

2. We propose a hierarchical transfer framework based on Dual-Codebook. The framework transfers the action model at the atom-level on different timescales and achieves the feature-to-feature correspondence across different views.

3. We evaluate our method on the IXMAS dataset, and demonstrate the superiority of our method compared to state-of-the-art methods.

2 Dual-Codebook Construction

In this section, we firstly model the process of learning Dual-Codebook as a co-clustering problem. Then, we propose an iterative algorithm to solve this problem effectively, which is based on semi-nonnegative matrix factorization.

We consider two kinds of actions: shared actions and orphan actions as in [7]. *Shared actions* are observed in both source and target views, and *orphan actions* are only observed in the source view during training. We only use shared actions to construct Dual-Codebook in the training phase, and use the samples of orphan action in the target view as test samples in the classification phase. This setting means that we do not use the correspondence across pairwise views for the orphan action.

2.1 Problem Formulation

The classical k-means algorithm aims to minimize the representation error of the given set of data points, and can be modeled as follows. Let $Y \in R^{d\times N}$ be the set of N d-dimensional data points. Then, the codebook of K-cluster centroids $C \in R^{d\times K}$ can be obtained by solving the following optimization problem

$$\{C^*, X^*\} = \arg\min_{C,X} \|Y - CX\|_F^2 \tag{1}$$

where $X \in R^{K\times N}$ is the cluster indicators of the N data points. Here, $\|.\|_F$ is the Frobenius norm of a matrix. Since C contains both positive and negative entries, and the entries in X should be nonnegative, if we allow the entries in X to range over $(0, 1)$, the k-means clustering can be seen as semi-nonnegative matrix factorization [4].

Existing methods for cross-view action recognition usually implement codebooks obtained by k-means clustering separately in source and target views.

As a result, these codebooks cannot guarantee correspondence with each other. We propose to construct a Dual-Codebook across two different views, which is composed of two codebooks, one in each view. Each pair of codewords that hold the same column number in these two codebooks is a pair of corresponding codewords across source and target views. The codewords in the Dual-Codebook are generated while maintaining pairwise associations across two views, which is very different from traditional codebook learning approaches.

To establish the association, we reduce the distance between the histograms of each pair of corresponding frames in the source and target views. We argue that, if two codewords from these two views correspond to each other, their frequency in the corresponding action videos should be close. Hence, by reducing the distance between the corresponding histograms from the source and target views, we can obtain corresponding codewords across the two views.

Suppose there are N_s, N_t feature points extracted from the videos of shared actions in source and target views, respectively. Let $Y_s \in R^{d\times N_s}, Y_t \in R^{d\times N_t}$ denote the sets of these feature points in source and target views. The Dual-Codebook $\{C_s, C_t\}$, where $C_s, C_t \in R^{d\times K}$ correspond to source and target views, respectively, can be learned by minimizing the following objective function

$$\begin{gathered} f(C_s, C_t, X_s, X_t) = \alpha \left\|X_s A_s - X_t A_t\right\|_F^2 + \left\|Y_s - C_s X_s\right\|_F^2 + \left\|Y_t - C_t X_t\right\|_F^2 \\ s.t.\ X_s \geq 0_{K\times N_s}, X_t \geq 0_{K\times N_t} \end{gathered} \tag{2}$$

where $X_s \in R^{K\times N_s}$, $X_t \in R^{K\times N_t}$ denote the cluster indicators of the feature points in source and target views, respectively. α is a positive constant. Besides, $A_s \in \{0,1\}^{N_s\times T}$, T is the total number of frames in source and target views. $A_s(i,j) = 1$ indicates that the i-th feature point is located in the j-th frame in the source view. The matrix $A_t \in \{0,1\}^{N_t\times T}$ is defined similarly in the target view. Thus, $X_s A_s$, $X_t A_t$ denote the histograms of all frames in source and target views, respectively.

The first term of Eq. (2) reflects the difference of the histograms of all corresponding frames in source and target views. The second and third terms of Eq. (2) are the representation errors of Y_s and Y_t, respectively.

It should be noted that Eq. (2) can be seen as a co-clustering problem, because codebooks C_s, C_t are generated simultaneously and the clustering on one of them induces that of the other, maintaining pairwise associations across source and target views. Specifically, for $i = 1, 2, \ldots, K$, the i-th columns of C_s and C_t are two codewords that correspond to each other.

2.2 Optimization

Since C_s, C_t contain both positive and negative entries, and the entries in X_s, X_t are nonnegative, Eq. (2) can be seen as a constrained joint semi-nonnegative matrix factorization. Inspired by [4], we propose an iterative algorithm to solve the problem as follows. Let $X_s = [x_{s1}, x_{s2}, \ldots, x_{sN_s}] \in R^{K\times N_s}$, $X_t = [x_{t1}, x_{t2}, \ldots, x_{tN_t}] \in R^{K\times N_t}$.

Step 1: Initialize X_s, X_t.

We first apply k-means clustering separately in the source and target views to obtain visual words in the two views. Then, we use these visual words as vertexes to build a bipartite graph for matching the visual words preliminarily across the two views. Afterwards, we can initialize X_s, X_t according to the matching result of visual words.

Step 2: Update C_s, C_t while fixing X_s, X_t as follows

$$C_s = Y_s X_s^T (X_s X_s^T)^{-1} \tag{3}$$

$$C_t = Y_t X_t^T (X_t X_t^T)^{-1} \tag{4}$$

Equations (3) and (4) are obtained by letting the partial derivatives of Eq. (2) with respect to C_s, C_t be zero, respectively.

Step 3: Update X_s column by column using Eq. (5) while fixing X_t, for $m = 1, 2, \ldots, K$,

$$x_{si(m)}^{(t+1)} = x_{si(m)}^{(t)} \sqrt{\frac{\alpha \left[X_t^{(t)} (A_t A_s^{\mathrm{T}})_{\bullet i}\right]_{(m)} + \left[(C_s^{\mathrm{T}} C_s)^{-} x_{si}^{(t)}\right]_{(m)} + \left[(C_s^{\mathrm{T}} Y_s)_{\bullet i}^{+}\right]_{(m)}}{\alpha \left[X_s^{*} (A_s A_s^{\mathrm{T}})_{\bullet i}\right]_{(m)} + \left[(C_s^{\mathrm{T}} C_s)^{+} x_{si}^{(t)}\right]_{(m)} + \left[(C_s^{\mathrm{T}} Y_s)_{\bullet i}^{-}\right]_{(m)}}} \tag{5}$$

To obtain Eq. (5), we use the auxiliary function approach as in [15] to find the auxiliary function of Eq. (2), which is the upper bound of Eq. (2) and is a convex function in X_s. Then, to find the minima of this auxiliary function, we set its partial derivative with respect to X_s to be zero. In Eq. (5), $x_{si(m)}^{(t+1)}$ is the updated value of the m-th entry in the i-th column of X_s at iteration $t+1$. $(\cdot)_{\bullet i}$ denotes the i-th column of the matrix in the parentheses. $[\cdot]_{(m)}$ is the m-th entry of the vector in the brackets. The matrix X_s^* is the result of X_s where the first $i-1$ columns have been updated in the iteration $t+1$. So we can see that, the updated result of the i-th column of X_s is related to the updated results of the first $i-1$ columns of X_s

Besides, in Eq. (5),

$$(C_s^T C_s)^{+} = \frac{1}{2}[\left|(C_s^T C_s)\right| + (C_s^T C_s)] \tag{6}$$

$$(C_s^T C_s)^{-} = \frac{1}{2}[\left|(C_s^T C_s)\right| - (C_s^T C_s)] \tag{7}$$

where $(C_s^T C_s)^{+}$ and $(C_s^T C_s)^{-}$ are the positive and negative parts of matrix $C_s^T C_s$, respectively. All superscripts "+" and "−" in Eq. (5) are defined similarly.

Note that $X_s^*(A_s A_s^T)_{\bullet i}$ in Eq. (5) corresponds to the histogram of the frame that contains the i-th feature point in the source view. And $X_t^{(t)}(A_t A_s^T)_{\bullet i}$ in Eq. (5) represents the histogram of a frame in the target view while the corresponding frame in the source view contains the i-th feature point. Consequently, the iterative process of updating each column of X_s (i.e., the cluster indicator of each feature point in the source view) is constrained by the interaction between the histograms of corresponding frames in source and target views. This means that Eq. (5) maintains the pairwise associations across source and target views.

Step 4: Update X_t column by column using Eq. (8) while fixing X_s, for $m = 1, 2, \dots, K$,

$$x_{ti(m)}^{(t+1)} = x_{ti(m)}^{(t)} \sqrt{\frac{\alpha \left[X_s^{(t+1)}(A_s A_t^{\mathrm{T}})_{\bullet i}\right]_{(m)} + \left[(C_t^{\mathrm{T}} C_t)^{-} x_{ti}^{(t)}\right]_{(m)} + \left[(C_t^{\mathrm{T}} Y_t)_{\bullet i}^{+}\right]_{(m)}}{\alpha \left[X_t^{*}(A_t A_t^{\mathrm{T}})_{\bullet i}\right]_{(m)} + \left[(C_t^{\mathrm{T}} C_t)^{+} x_{ti}^{(t)}\right]_{(m)} + \left[(C_t^{\mathrm{T}} Y_t)_{\bullet i}^{-}\right]_{(m)}}} \tag{8}$$

Eq. (8) is obtained in a similar way as Eq. (5).

We iteratively perform Step 1 to 4 until Eq. (2) converges. The convergence proof of the above algorithm can be found in the supplementary material. After the Dual-Codebook is obtained, we can replace traditional Bag-of-Visual-Words (BoVW) with Bag-of-Dual-Codes (BoDC), which contains the codebook-to-codebook correspondence across views.

3 Hierarchical Temporal-Structure Transfer

In this section, we firstly propose the action temporal-structure model that can effectively capture the information about atom actions within a full action. Then, based on this model, we propose the hierarchical temporal-structure transfer framework.

3.1 Action Temporal-Structure Modeling

The execution of an action is typically considered to be composed of several atom actions. Each of these atom actions corresponds to a short time interval and their sequential order forms the temporal pattern of an action. Thus, the categories and sequential composition of the atom actions can reflect the nature of an action [1]. More specifically, both the categories and sequential order of the atom actions will not change with viewpoints. For instance, the action "sit down" can be seen as an atom-action sequence "stand-stoop-sit" in whatever viewpoint it is observed. Consequently, we consider that these significant invariabilities should be fully utilized for solving the cross-view action recognition problem. Before doing this, it is necessary to construct a model that can effectively capture the atom actions and the temporal relationship among them within an action.

To exploit the temporal information, we divide actions into several segments along the time dimension. Each segment can be assumed to contain an atom action, which can be described by a BoDC. For example, when three segments are implemented, the action "sit down" can be divided into "stand", "stoop", and "sit". While for the action "stand up", the atom-action-sequence "sit", "stoop", and "stand" will be obtained instead. In this way, the sequential orders of the segment-BoDCs can be used to distinguish these two actions effectively.

Modeling Details: Based on the above analysis, we consider multiple timescales to construct the action temporal-structure model. For action videos that are approximately aligned in time dimension, at the l-th scale, $l = 1, 2, \ldots, L$, we divide an action into 2^{l-1} segments of equal duration along the time dimension. As a result, an action can be modeled as a sequence of increasingly finer segments at levels $1, 2, \ldots, L$. Based on this action temporal-structure model, we propose a novel hierarchical transfer framework in Sec. 3.2.

3.2 Hierarchical Transfer Framework

In this section, we propose a hierarchical transfer framework that exploits the previous temporal-structure model. In each level of proposed transfer framework, only the shared actions are used to construct the transferable relationship across different views, while orphan actions are utilized to test the effectiveness of this relationship.

In the training stage, we divide two action videos of the same class from source and target views into several segments as in Sec. 3.1. Then we construct the common representation of corresponding segments within these two videos. The basic idea is "pairwise dictionary learning". It has also been explored in cross-domain face recognition [32], and in cross-view action recognition for video-level correspondence [34].

Incorporating this basic idea into the temporal-structure model, we propose a novel hierarchical transfer framework. As illustrated in Fig. 1, both action videos are assumed to have a 2-level temporal structure. For each level of the models in two views, we aim to learn a transferable pairwise dictionary based on the Dual-Codebook. In other words, what we utilize to learn the transferable pairwise dictionary are the BoDCs of all pairs of corresponding segments from shared actions in the source and target views. This is different from existing implementations of dictionary learning strategies that are based on separately generated codebooks. In this hierarchical framework, each level has its own pairwise dictionary, i.e., $\{D_{si}, D_{ti}\}$ shown in Fig. 1, such that all pairs of corresponding segments across source and target views are converted to similar sparse representations, such as $x_{11}, x_{11}^{'}$ in Fig. 1. Thus, these sparse representations are view-invariant, and only depend on atom actions within the segments. This means that, the hierarchical framework is capable to transfer at the atom-level effectively. At last, for each action video in the source and target views, we obtain its full view-invariant sparse representation by concatenating the sparse representations of all segments at all levels of the temporal-structure model, i.e., $[x_{11}, x_{21}, x_{22}]$ and $\left[x_{11}^{'}, x_{21}^{'}, x_{22}^{'}\right]$ shown in Fig. 1.

In the following, we explain the procedure of learning a transferable pairwise dictionary at each level. Let $B_s, B_t \in R^{K \times N}$ denote the K-dimensional BoDCs of N segments of shared actions in the source and target views, respectively. The transferable pairwise dictionary $\{D_s, D_t\}$ is learned by solving the following optimization problem

$$\arg\min_{D_s, D_t, S} \{\|B_s - D_s S\|_2^2 + \|B_t - D_t S\|_2^2\}, \quad s.t.\ \forall i, \|s_i\|_0 \le T_0 \tag{9}$$

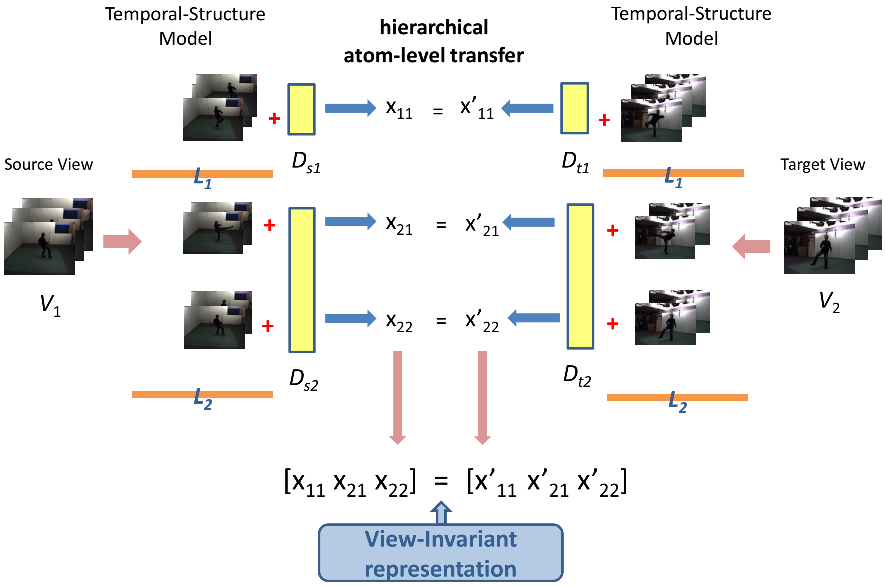


Fig. 1. The proposed hierarchical transfer framework, where we perform transfer at atom-level, converting the corresponding segments to similar sparse representations.

where $D_s, D_t \in R^{K\times J}$ denote, respectively, the dictionaries with J items in the source and target views. The matrix $S = [s_1, s_2, \ldots, s_N] \in R^{J\times N}$ denotes the common sparse representations of B_s and B_t, which satisfies the sparsity constraint $\|s_i\|_0 \leq T_0$. The terms $\|B_s - D_sS\|_2^2$ and $\|B_t - D_tS\|_2^2$ are the reconstruction errors of source and target views, respectively.

Furthermore, by constructing $B = \left[B_s^T \ B_t^T\right]^T$, $D = \left[D_s^T \ D_t^T\right]^T$, we formulate Eq. (9) equivalently as

$$\arg\min_{D,S} \{\|B - DS\|_2^2\}, \quad s.t.\ \forall i, \|s_i\|_0 \leq T_0 \tag{10}$$

The K-SVD algorithm can be used to solve Eq. (10) [2].

When the transferable pairwise dictionary $\{D_s, D_t\}$ is obtained, we calculate the sparse representations of all segments (from either shared actions or orphan actions) in source and target views by solving Eqs. (11) and (12), respectively,

$$S_s = \arg\min_{S_s} \{\|B_s^* - D_sS_s\|_2^2\}, \quad s.t.\ \forall i, \|s_{si}\|_0 \leq T_0 \tag{11}$$

$$S_t = \arg\min_{S_t} \{\|B_t^* - D_tS_t\|_2^2\}, \quad s.t.\ \forall i, \|s_{ti}\|_0 \leq T_0 \tag{12}$$

where M denotes the number of all segments both in the source and target views, $S_s = [s_{s1}, s_{s2}, \ldots, s_{sM}] \in R^{J\times M}$ and $S_t = [s_{t1}, s_{t2}, \ldots, s_{tM}] \in R^{J\times M}$ refer to the sparse representations of all the M segments in source and target views, respectively. The matrices $B_s^*, B_t^* \in R^{K \times M}$ denote the K-dimension BoDCs of all the M segments in source and target views, respectively. Both Eq. (11) and Eq. (12) can be efficiently solved by orthogonal matching pursuit (OMP) algorithm [24].

In our hierarchical transfer framework, what we obtain ultimately are the sparse representations at the atom-level of the action videos, which guarantees that the information of all the segments as well as the temporal relationship therein can be preserved. As a result, the invariance across different views are fully utilized during the transfer procedure, which contributes to the feature-to-feature correspondence at atom-level through our framework.

4 Experiments

4.1 Dataset and Experimental Setup

We use the multi-view dataset IXMAS [25] in our experiment. This dataset contains eleven action categories, each of which is observed by five different cameras, and is performed by twelve people for three times. We denote the five different camera views of this dataset as $C1, C2, \ldots, C5$ respectively.

We extract spatial-temporal interest-point-based features [5] to describe actions in each viewpoint. The protocol for calculating these features is the same as in [34] and [20]. Specifically, while constructing the codebook, the size of our Dual-Codebook is chosen from 50, 100, 250, and 500.

In our experiments, we observed that the 3-level temporal-structure model obtained relatively better results than other choices. Hence, we set our action temporal-structure model to 3-levels, and the action videos at these three levels have one, two, and four segments respectively.

The selection strategies of the parameters in our method are as follows. At each level of the hierarchical transfer framework, the number of dictionary atoms is set to be the same as that of training samples. The sparsity constraint T_0 is set to 36, since each action class has 36 samples in the IXMAS dataset, and we assume that each sample can be well represented by other samples of the same class. Moreover, α is empirically fixed to be 1 according to the experiments.

In order to have a fair comparison to [34] and [20], we follow their leave-one-action-class-out scheme, where each time we only consider one action category for testing (i.e., as an orphan action). Accordingly, we utilize all other action categories to learn the Dual-Codebook and the transferable pairwise dictionary at each level of our transfer framework.

In the classification phase, we take all action videos in the source view as training samples, and use the nearest-neighbor classifier to recognize the target-view video of the orphan action.

4.2 Experimental Results

Firstly, we conduct two controlled experiments to verify the effectiveness of the proposed framework. In the first experiment, we implement codebooks separately trained with k-means in the two views instead of Dual-Codebook, followed by the proposed hierarchical transfer framework, to verify the effectiveness of the proposed Dual-Codebook. In the second experiment, the Dual-Codebook is implemented, but action videos are not segmented, i.e., only video-level transfer is performed. The results are listed in Table 1.

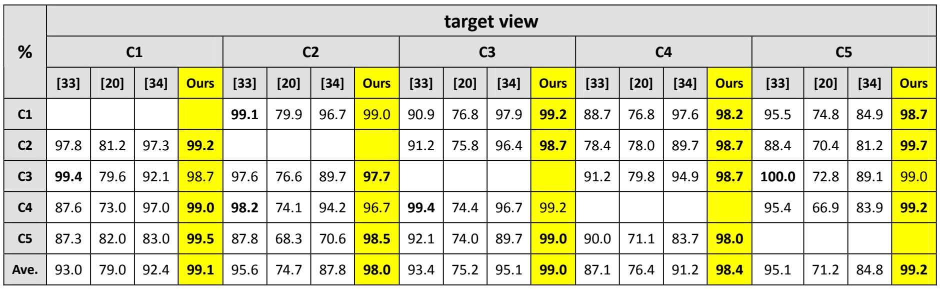
Table 1. The results of controlled experiments. Column A, B and "Ours" are the results of "*k*-means codebooks + hierarchical transfer framework", "Dual-Codebook + video-level transfer" , and our method, respectively.

%	target view														
	C1			C2			C3			C4			C5		
	A	B	**Ours**	A	B	**Ours**	A	B	**Ours**	A	B	**Ours**	A	B	**Ours**
C1				89.9	74.0	**99.0**	92.2	77.3	**99.2**	86.1	69.7	**98.2**	91.2	76.0	**98.7**
C2	89.6	67.4	**99.2**				92.7	66.4	**98.7**	87.6	68.4	**98.7**	91.2	77.0	**99.7**
C3	91.2	78.5	**98.7**	88.6	73.7	**97.7**				92.7	74.7	**98.7**	93.4	82.6	**99.0**
C4	86.6	76.0	**99.0**	87.4	70.5	**96.7**	92.4	80.1	**99.2**				91.2	74.0	**99.2**
C5	88.4	79.8	**99.5**	88.6	78.0	**98.5**	94.7	87.9	**99.0**	88.9	77.8	**98.0**			
Ave.	89.0	75.4	**99.1**	88.6	74.1	**98.0**	93.0	77.9	**99.0**	88.8	72.7	**98.4**	91.8	77.4	**99.2**

Comparing Column A to "Ours", we can see that our method performs better in all twenty pairwise view combinations. The average accuracy of our method is about 8.5% higher than that using separate k-means codebooks instead. This indicates that our Dual-Codebook is a better foundation than separate k-means codebooks for transferring actions models across pairwise views. Moreover, comparing Column B to "Ours", we can also see that our method performs better in all twenty pairwise view combinations. The average accuracy of our method is about 23.2% higher than that not splitting the action videos at all. This indicates that the atom-level transfer in our hierarchical framework is more discriminative than the usual video-level transfer in most existing work.

Additionally, in Table 2, we compare our method to three state-of-the-art approaches for all twenty pairwise view combinations on the IXMAS dataset. The size of Dual-Codebook is set to 500. As can be observed, our method has obtained recognition rates of higher than 98% in eighteen pairwise view combinations. Compared to the results of [20] and [34], our method performs better in all twenty pairwise view combinations. Compared to [33], the proposed method obtained higher recognition rates in fifteen pairwise view combinations. The reason is that, the proposed Dual-Codebook contains codebook-to-codebook correspondence across two views, while codebooks separately trained in each view cannot guarantee the same level of correspondence. Moreover, the atom-level transfer s-trategy in the hierarchical framework is more discriminative than the video-level transfer in [33].

Table 2. Performance comparison between our method and state-of-the-art approaches.



%	**target view**																			
	C1				**C2**				**C3**				**C4**				**C5**			
	[33]	**[20]**	**[34]**	**Ours**	**[33]**	**[20]**	**[34]**	**Ours**	**[33]**	**[20]**	**[34]**	**Ours**	**[33]**	**[20]**	**[34]**	**Ours**	**[33]**	**[20]**	**[34]**	**Ours**
C1					**99.1**	79.9	96.7	99.0	90.9	76.8	97.9	**99.2**	88.7	76.8	97.6	**98.2**	95.5	74.8	84.9	**98.7**
C2	97.8	81.2	97.3	**99.2**					91.2	75.8	96.4	**98.7**	78.4	78.0	89.7	**98.7**	88.4	70.4	81.2	**99.7**
C3	**99.4**	79.6	92.1	98.7	97.6	76.6	89.7	**97.7**					91.2	79.8	94.9	**98.7**	**100.0**	72.8	89.1	99.0
C4	87.6	73.0	97.0	**99.0**	**98.2**	74.1	94.2	96.7	**99.4**	74.4	96.7	99.2					95.4	66.9	83.9	**99.2**
C5	87.3	82.0	83.0	**99.5**	87.8	68.3	70.6	**98.5**	92.1	74.0	89.7	**99.0**	90.0	71.1	83.7	**98.0**				
Ave.	93.0	79.0	92.4	**99.1**	95.6	74.7	87.8	**98.0**	93.4	75.2	95.1	**99.0**	87.1	76.4	91.2	**98.4**	95.1	71.2	84.8	**99.2**

In Table 2, the recognition rates of [20] and [34] dropped dramatically when camera 5 was the source or target view. The reason might be that camera 5 is set above the actors. The action observations obtained in this camera is dramatically different from those in other cameras, which is very challenging. It is interesting to note that the proposed method obtained high accuracies under camera 5. We consider the reason is that, the key of our method is fully utilizing the information of the categories and sequential composition of the atom actions within an action during transfer, and this information is invariant to the viewpoint.

The average recognition accuracies of each action category for different target views are demonstrated in Fig. 2 (the size of Dual-Codebook is 500). We can see that the action "get-up" gains 100% recognition accuracies in all five target views. Besides, the action "kick" and "punch" achieve 100% recognition accuracies in four target views, although they tend to be mistaken for each other in some viewpoints. The recognition accuracies of the action "pick up" are relatively low compared to other actions, but still higher than 85%. We find that it is often mistaken for the action "sit-down". The possible reason is that, the observations of these two actions are extremely similar while seen in most of the viewpoints, since they contain the similar atom-action sequences "stand-stoop-squat" and "stand-stoop-sit". And our hierarchical transfer framework relies on the categories and sequential composition of atom actions, so it confuses "pick up" with "sit-down" in some cases.

We also conduct experiments studying recognition performance under Dual-Codebooks of different sizes. As shown in Fig. 3, in all source views, the recognition rates of our method increase quickly with the size of Dual-Codebook when the size is under 250. When the codebook size is greater than 250, the recognition rate curves tend to be flat. This indicates that, a larger Dual-Codebook generally contains more accurate codebook-to-codebook correspondence across two views and more discriminative information.

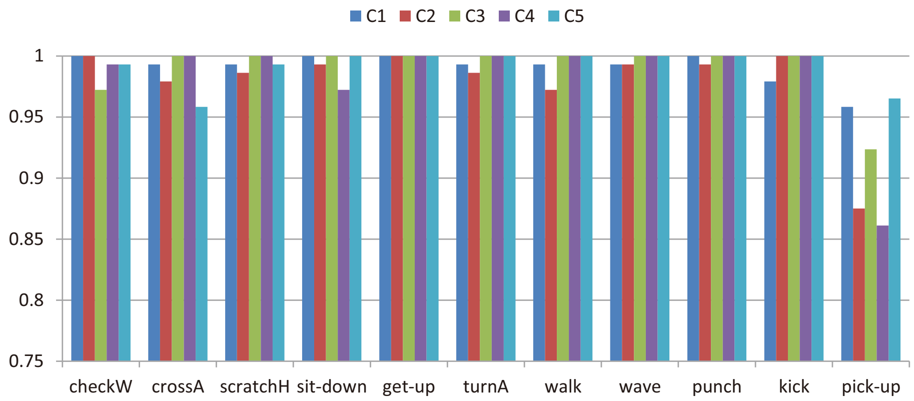


Fig. 2. The recognition accuracy of each action category in different target views (best viewed in PDF file).

5 Conclusion

In this paper, we explore the challenging cross-view action recognition problem. For this purpose, we firstly propose a Dual-Codebook that achieves codebook-to-codebook correspondence across different views. With the Dual-Codebook, each action can be represented based on Bag-of-Dual-Codes (BoDC). We further introduce a hierarchical transfer framework, which performs atom-level transfer on multiple timescales. This framework guarantees that each pair of corresponding video segments from pairwise views obtain similar sparse representations, and achieves feature-to-feature correspondence at atom-level. This contributes to a more accurate transfer relationship than the simple video-level transfer. At last, we conduct a series of experiments on the IXMAS dataset. The experimental results demonstrate that our method can achieve superior performance over state-of-the-art approaches.

Acknowledgement. This work is supported by National Natural Science Foundation of China (No. 61172141), Key Projects in the National Science & Technology Pillar Program during the 12th Five-Year Plan Period (No. 2012BAK16B06), and Science and Technology Program of Guangzhou, China (2014J4100092).

References

1. Aggarwal, J., Ryoo, M.: Human activity analysis: A review. ACM Computing Surveys **43** (2011)
2. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. TSP **54** (2006) 4311–4322
3. Cheung, G., Baker, S., Kanade, T.: Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In: CVPR. (2003)

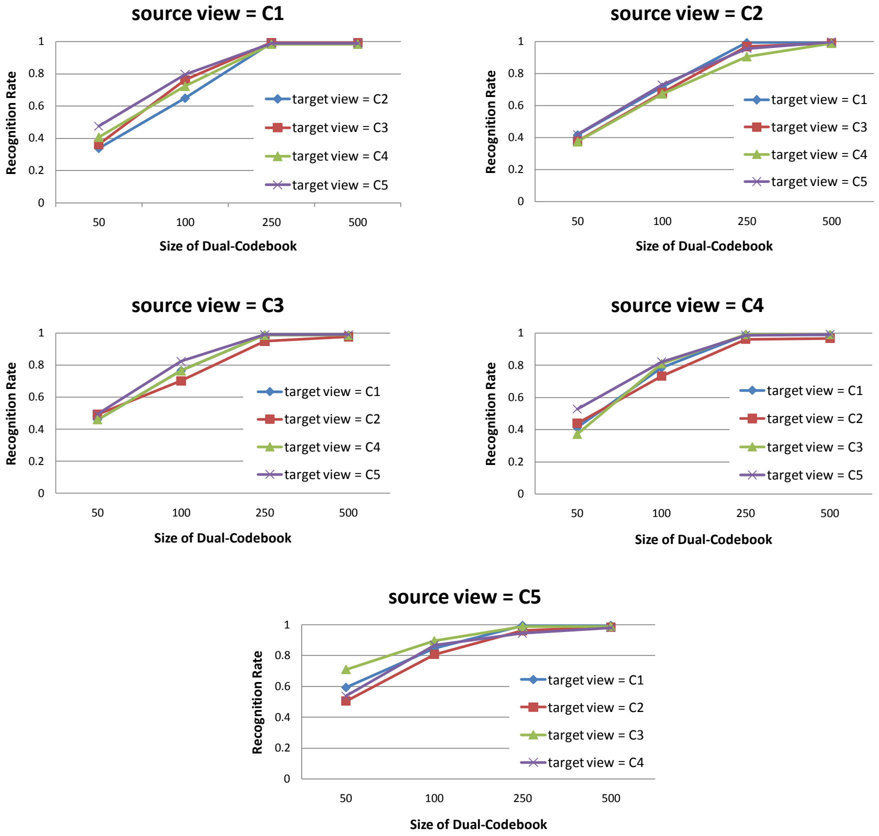


Fig. 3. Recognition performance under different size of Dual-Codebook (best viewed in PDF file).

4. Ding, C., Li, T.: Convex and semi-nonnegative matrix factorizations. PAMI **32** (2010) 45-55
5. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS. (2005)
6. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV. (2003)
7. Farhadi, A., Tabrizi, M.: Learning to recognize activities from the wrong view point. In: ECCV. (2008)
8. Farhadi, A., Tabrizi, M., Endres, I., Forsyth, D.: A latent model of discriminative aspect. In: ICCV. (2009)
9. Gavrila, D., Davis, L.S.: 3D model-based tracking of humans in action: a multi-view approach. In: CVPR. (1996)
10. Holte, M.B., Moeslund, T.B., Tran, C., Trivedi, M.: Human action recognition using multiple views: a comparative perspective on recent developments. In: HGBU. (2011)

11. Ji, X., Liu, H.: Advances in view-invariant human motion analysis: a review. TCSVT **40** (2010) 13–24
12. Junejo, I., Dexter, E., Laptev, I., Patrick, P.: View-independent action recognition from temporal self-similarities. PAMI **33** (2011) 172–185
13. Junejo, I., Dexter, E., Laptev, I., Perez, P.: Cross-view action recognition from temporal self-similarities. In: ECCV. (2008)
14. Laptev, I., Lindeberg, T.: Space-time interest points. In: ICCV. (2003)
15. Lee, D., Seung, H.: Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems 13. Cambridge, MA: MIT Press (2001)
16. Li, R., Zickler, T.: Discriminative virtual views for cross-view action recognition. In: CVPR. (2012)
17. Lin, Z., Jiang, Z., Davis, L.: Recognizing actions by shape-motion prototype trees. In: ICCV. (2009)
18. Liu, J., Ali, S., Shah, M.: Recognizing human actions using multiple features. In: CVPR. (2008)
19. Liu, J., Shah, M.: Learning human actions via information maximization. In: CVPR. (2008)
20. Liu, J., Shah, M., Kuipers, B., Savarese, S.: Cross-view action recognition via view knowledge transfer. In: CVPR. (2011)
21. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and viterbi path searching. In: CVPR. (2007)
22. Paramesmaran, V. Chellappa, R.: View invariance for human action recognition. IJCV **66** (2006) 83–101
23. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. IJCV **50** (2002) 203–226
24. Tropp, J., Gilbert, A.: Signal recovery from random measurements via orthogonal matching pursuit. TIT **53** (2007) 4655-4666
25. Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine recognition of human activities: a survey. TCSVT **18** (2008) 1473–1488
26. Valera, M., Velastin, S.: Intelligent distributed surveillance systems: a review. VISP **152** (2005) 192–204
27. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3D examplars. In: ICCV. (2007)
28. Weinland, D., Ozuysal, M., Fua, P.: Making action recognition robust to occlusions and viewpoint changes. In: ECCV. (2010)
29. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. PAMI **31** (2009) 210–227
30. Yan, P., Khan, S.M., Shah, M.: Learning 4D action feature models for arbitrary view action recognition. In: CVPR. (2008)
31. Yilmaz, A., Shah, M.: Actions sketch: A novel action representation. In: CVPR. (2005)
32. Zhang, Z., Wang, Y., Zhang, Z.: Face synthesis from near-infrared to visual light via sparse representation. In: IJCB. (2011)
33. Zheng, J., Jiang, Z.: Learning view-invariant sparse representations for cross-view action recognition. In: ICCV. (2013)
34. Zheng, J., Jiang, Z., Phillips, P., Chellappa, R.: Cross-view action recognition via a transferable dictionary pair. In: BMVC. (2012)