Learning Human Actions via Information Maximization

Jingen Liu
Computer Vision Lab
University of Central Florida
liujg@cs.ucf.edu

Mubarak Shah
Computer Vision Lab
University of Central Florida
shah@cs.ucf.edu

Abstract

In this paper, we present a novel approach for automatically learning a compact and yet discriminative appearance-based human action model. A video sequence is represented by a bag of spatiotemporal features called video-words *by quantizing the extracted 3D interest points (cuboids) from the videos. Our proposed approach is able to automatically discover the optimal number of video-word clusters by utilizing Maximization of Mutual Information(MMI). Unlike the k-means algorithm, which is typically used to cluster spatiotemporal cuboids into video words based on their appearance similarity, MMI clustering further groups the* video-words*, which are highly correlated to some group of actions. To capture the structural information of the learnt optimal* video-word *clusters, we explore the correlation of the compact* video-word *clusters. We use the modified correlgoram, which is not only translation and rotation invariant, but also somewhat scale invariant. We extensively test our proposed approach on two publicly available challenging datasets: the KTH dataset and IXMAS multiview dataset. To the best of our knowledge, we are the first to try the bag of* video-words *related approach on the multiview dataset. We have obtained very impressive results on both datasets.*

1. Introduction

Automatically recognizing human actions is critical for several applications such as video indexing, video summarization, and so on. However, it remains a challenging problem due to camera motion, occlusion, illumination changes and the individual variations of object appearance and postures.

Over the past decade, this problem has received considerable attention. We can model the human actions using either holistic information or part-based information. One way to compare two actions is to compute the correlation of their spatiotemporal (ST) volumes. Shechtman *et. al.*[14] proposed a method which measures the degree of consistency by computing the correlation using the local intensity variance. Similarly, Efros *et. al.*[3] extracted an optical flow field as a descriptor from the stabilized object ST volume, and computed the cross correlation between the model and the input optical flow descriptors. In another holistic approach, an action is considered as a 3D volume and features are extracted from this volume. For instance, Yilmaz *et. al.*[23] used differential geometry features extracted from the surfaces of their action volumes and achieved good performance. Yet this method requires robust tracking to generate the 3D volumes. Parameswaran *et. al.*[13] proposed an approach to exploit the 2D invariance in 3D to 2D projection, and model actions using view-invariant canonical body poses and trajectories in 2D invariance space. They assume the body joints are available. Bobick *et. al.* [1] introduced the motion-history images, which are used to recognize several types of aerobics actions. . Although their method is efficient, they still assume a well segmented foreground and background. Most holistic-based paradigms either have a limitation on the camera motion or are computationally expensive due to the requirement of pre-processing of the input data, such as background substraction, shape extraction, body joints extraction, object tracking and registration.

Due to the limitation of holistic models to solve some practical problems, the part-based models have recently received more attention. Unlike the holistic-based method, this approach extracts "bag of interesting parts". Hence, it is possible to overcome certain limitations such as background substraction and tracking. Fanti *et. al.*[4] and Song *et. al.*[15] proposed a triangulated graph to model the actions. Multiple features, such as velocity, position and appearance, were extracted from the human body parts in a frame-by-frame manner. Spatiotemporal interest points [8, 6, 2, 11] have also been widely successful. Laptev [8] computed a saliency value for each voxel and detected the local saliency maxima based on Harris operator. While Dollar *et. al.*[2] applied the separate linear filters in the spatial and temporal directions and detected the interest points, which have local maxima value in both directions. Then an action video is represented by the statistical distribu-

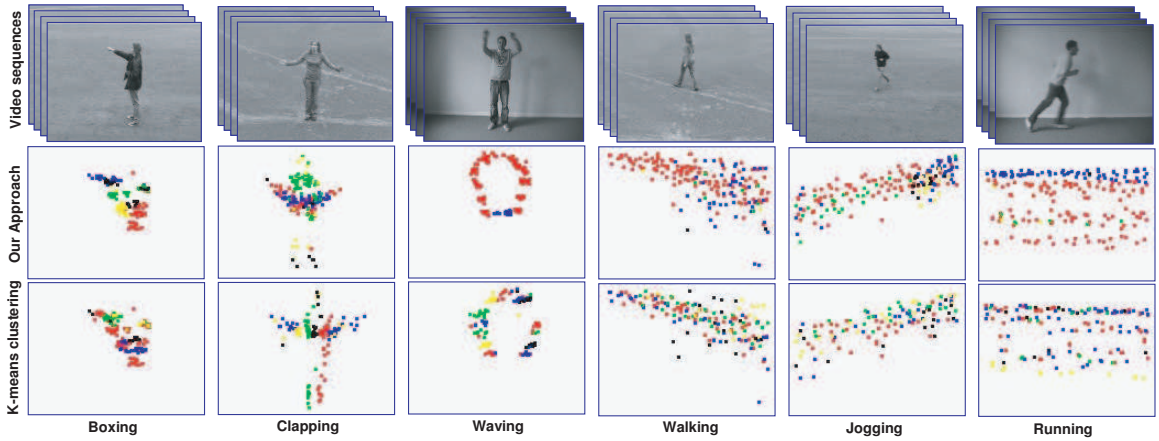

Figure 1. The first row shows the examples of six actions. The following tow rows respectively demonstrate the distribution of the optimal 20 *video-words* clusters using our approach and 20 *video-words* using k-means. We superimpose the 3D interest points in all frames into one image. Different clusters are represented by different color codes. Note that our model is more compact e.g. see 'waving' and' 'running' actions (Best viewed in color).

tion of the bag of video-words (BOV). Beyond the BOV, the discriminative models like SVM [2, 16] and the generative graphical models such as probalilistic Latent Semantic Analysis (pLSA)[12, 22] for action recognition have achieved inspiring performance.

One of the critical steps in the BOV-related approach is quantizing the local cuboids into *video-words* using k-means algorithm, which clusters the cuboids based on their appearance similarity. It has been noted that the size of the codebook affects the performance and that there is an optimal codebook size which can achieve maximal accuracy [12, 7]. In [21], Winn *et. al.* proposed a generative model to learn the optimal visual dictionary for object recognition in a **supervised** manner, and represented the visual words by the GMMs of pixel appearance.

In this paper, we propose an approach to automatically discover the optimal number of *video-words* clusters (*VWC*s) utilizing Maximization of Mutual Information (MMI) principal in an **unsupervised** manner. Our goal is to find a compact and yet discriminative *VWC*s by grouping the redundant *video-words*. The benefits of a compact representation are twofold: more effective and efficient classification due to lower dimension, and effectively capturing the spatiotemporal correlation of the *VWC*s. Specifically, we maximize the Mutual Information(MI) when merging two *VWC*s, which is **unsupervised**. The maximization of Mutual Information (MMI) has several available mechanisms[7][18]. We adopt the Information Bottleneck [18]. MMI based clustering has been successfully used for word clustering, where the words are grouped into semantic concept clusters (e.g. "pitching", "score","teams" etc. can be clustered into "baseball" concept; and "biker", "wheel", "ride" may be clustered into "motorcycle" concept.). This is effective due to the fact that words related to a particular concept have higher co-occurrence in the documents. Similarly, each cluster of *video-words* achieved by MMI method tends to correspond to a group of semantically related *video words*. For instance, one particular cluster may contain the cuboids related to "raising the hands" motion in different actions.

The *VWC*s are somewhat analogous to *hidden topics* in pLSA. However, there are significant differences between them. pLSA is a generative model, which employs hidden variables; while MMI clustering does not use hidden variables. Secondly, pLSA assumes conditional independence (i.e. given the latent variable, the document and word are independent), which is not required in MMI clustering. In [12], pLSA is used as a clustering method. The number of topics (clusters) is typically set to be the number of categories. Wong *et. al.*[22] set the number of topics to be three times the number of categories. Nevertheless, our approach aims to automatically discover the optimal number of *VWC*s, such that the action can be represented by a compact while discriminative model. .

Although BOV has achieved very good performance, it ignores any spatial and temporal information between the *video-words*. The cuboids representing the parts of human body motion have strong correlation to each other, due to the fact that they all belong to the same body. Fig.1 shows some examples of spatial distribution of the cuboids. In our work, we apply the correlogram which has been successively applied for image and scene classification [7, 17]. The modified correlogram is able to somewhat cope with the translation, rotation and scale problems. Besides, we also explore the spatiotemporal pyramid approach in order to capture both spatial and temporal information.

1.1. Proposed Framework

The major steps of the training phase in our framework are described in table 1. The videos are feed into the system, and the appropriate number of cuboids (3D interest points) are extracted from each video. The K-means algorithm is

Objective: Action recognition using the learnt optimal number of *VWC*s and their structural information.

- **Extracting Cuboids**. Apply separate linear filters in spatial and temporal direction, and extract cubiods around the local maxima.
- **Learning Codebook**. Quantize the cuboids into N *video-words* using k-mean algorithm based on the appearance similarity.
- **Compressing Codebook**. Apply MMI clustering to find the optimal number of *video-word* clusters.
- **Capturing Structural Information**. Extract translation, rotation and scale invariant spatial correlogram and spatial temporal pyramid.
- **Training SVM models**. Training using feature vectors extracted as described above.

Table 1. Major steps for the training phase of our framework

applied to obtain a large number of *video-words*. Then MMI clustering automatically discovers a compact representation from the initial codebook of *video-words* and efficiently captures the correlation. Furthermore, we use spatial correlogram and spatiotemporal pyramid models for capturing structural information. Finally, we use a SVM as a classifier to train and test these models.

We tested our approach on two publicly available datasets: KTH dataset [8] and IXMAS multiview dataset[20]. To the best of our knowledge, we are the first to try the bag of *video-words* related approach on the the multiview dataset, and have obtained very inspiring results. Our results also show that, the performance of MMI clustering is not only much better than that of the k-means algorithm, but also better than the performance carried on the initial large codebook from which the optimal number of *VWC*s are learnt. This is because MMI clustering generates fewer but more meaningful clusters of *video-words*. We also found that the models with the structural information of *VWC*s can further improve the recognition performance.

2. Extracting Optimal Number of *VWC*s

In this section we first briefly describe the spatiotemporal interest point detection, then we discuss how to discover the optimal number of *video-words* using MMI clustering.

2.1. Feature Detection and Representation

In this paper, we adopt the spatiotemporal interest points detector proposed by Dollar[2]. This detector produces dense feature points and performs better on the action recognition task [12, 2, 22]. Instead of using a 3D filter on the spatiotemporal domain, it applies two separate linear filters respectively to spatial and temporal dimensions. A response function can be represented as follows:

$$R = (I(x,y,t) * g_\sigma(x,y) * h_{ev}(t))^2 + (I(x,y,t) * g_\sigma(x,y) * h_{od}(t))^2,$$

where $g_\sigma(x,y)$ is the spatial Gaussian filter with kernel σ, h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied along the time dimension. They are defined as $h_{ev}(t;\tau,\omega) = -cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{od}(t;\tau,\omega) = -sin(2\pi t\omega)e^{-t^2/\tau^2}$, where $\omega = 4/\tau$. They give a strong response to the temporal intensity changes. The interest points are detected at locations where the response is locally maximum. The ST volumes around the points are extracted and the gradient-based descriptors are learnt using PCA. All descriptors are quantized into *video-words* using k-means algorithm.

2.2. Clustering of Video-words by MMI

Consider two discrete jointly distributed random variables X and Y, where $X \in \mathcal{X} = \{x_1, x_2, ..., x_n\}$ and $Y \in \mathcal{Y} = \{y_1, y_2, ..., y_m\}$. In our work, $\mathcal{X}$ represents a set of *video-words*, and $\mathcal{Y}$ is a set of action videos. We can simply build a histogram of the *video-words* to model each action video y_i. Then the similarity of two action samples can be measured by their conditional distributions $p(x|y)$. However, the size of $\mathcal{X}$ is difficult to choose. If the codebook size is too small, it may cause over-clustering with higher intra-class distortion. Therefore, it is common to choose an appropriately large value of codebook size. But that may cause a sparse histogram and introduce more noise. So, we seek to find a more compact and yet discriminative representation of X, say $\hat{X}$ which groups the *video-words* with higher co-occurrence relationship together, and also preserves the information about Y. Our criteria for $\hat{X}$ is to maximize the mutual information $I(\hat{X};Y)$.

2.2.1 Mutual Information

Given two discrete random variables X and Y, the Mutual Information (MI) between them is defined as:

$$I(X;Y) = \sum_{y\in Y, x\in X} p(x,y) log \frac{p(x,y)}{p(x)p(y)}, \qquad (1)$$

where $p(x,y)$ is the joint distribution of X and Y, $p(x)$ and $p(y)$ are probability distributions of X and Y respectively. MI tells how much information from variable X is contained in variable Y. Using Kullback-Leibler divergence, it also can be expressed as:

$$I(X,Y) = D_{KL}(p(x,y) \parallel p(x)p(y)), \qquad (2)$$

where D_{KL} computes the distance between two distributions. In the context of this paper, X and Y represent *video-words* and actions respectively.

2.2.2 MMI clustering Algorithm

Our goal is to find an optimal mapping of the *video-words* X, say C(X) into a more compressed representation $\hat{X}$ such

that the MI between $\hat{X}$ and Y, say $I(\hat{X};Y)$, is as high as possible, given the constraint on the MI between X and $\hat{X}$, say $I(\hat{X};X)$. $I(\hat{X};X)$ signifies how compact the new representation $\hat{X}$ is. Obviously, lower value means a more compact representation, and the most compact representation will correspond to the merging of all *video-words* into one single cluster. However, that representation may not be discriminative, because it may not give any information regarding Y from $\hat{X}$. Therefore, we also need to keep higher value of $I(\hat{X};Y)$, which provides the discrimination of the new representation or quality of the clustering. There is a tradeoff between the compactness and discrimination. Given the mapping $p(\hat{x}|x)$, this problem can be mathematically expressed as:

$$max(I(\hat{X};Y) - \lambda^{-1} I(\hat{X};X)), \tag{3}$$

where λ^{-1} is the Lagrange multiplier. The solution of formula 3 gives three self-consistent equations on $p(\hat{x}|x)$, $p(y|\hat{x})$ and $p(\hat{x})$. The details of the solution of this minimization problem are given in [19]. When $\lambda = 0$, the solution of 3 assigns all x to one cluster, and when $\lambda \to \infty$, it gives solution for hard clustering as follows: $p(\hat{x}|x) = 1$ if $x \in \hat{x}$, otherwise $p(\hat{x}|x) = 0$; $p(y|\hat{x}) = \frac{1}{p(\hat{x})} \sum_{i=1}^{|\hat{x}|} p(x_i, y)$ and $p(\hat{x}) = \sum_{i=1}^{|\hat{x}|} p(x_i)$. If one specified clustering $C(X)$ always has $I(C(X);Y) \geq I(C'(X);Y)$ where $C'(X)$ is an arbitrary mapping, $C(X)$ is one of the optimal solutions.

This problem can be solved by a greedy algorithm based on a bottom-up pair-wise merging procedure. The algorithm starts with a trivial partition, where each element of X is a singleton cluster. In order to keep $I(\hat{X};Y)$ as high as possible, at each step we greedily merge two components into one, which causes minimal loss of mutual information $I(\hat{X};Y)$. Let $\hat{x}_1$ and $\hat{x}_2$ be the two candidate clusters to be merged, the cost of this merge is defined as the loss of MI due to the merge, which is expressed as:

$$\Delta I(\hat{x}_1, \hat{x}_2) = I(\hat{X}_{bef};Y) - I(\hat{X}_{aft};X), \tag{4}$$

where $I(\hat{X}_{bef};Y)$ and $I(\hat{X}_{aft};Y)$ denote the MI before and after the merging step respectively. x is a *video-word* which is represented by a normalized vector with its frequency in the training videos, specifically it is a vector of $p(y|x)$ with $y \in Y$. Similarly, every cluster has a "prototype" say $p(y|\hat{x})$. Assume $\hat{x}_1$ and $\hat{x}_2$ are merged into $\hat{x}^*$, the new "prototype" is updated as:

$$p(y|\hat{x}^*) = \frac{p(\hat{x}_1)}{p(\hat{x}^*)} p(y|\hat{x}_1) + \frac{p(\hat{x}_2)}{p(\hat{x}^*)} p(y|\hat{x}_2), \tag{5}$$

where $p(\hat{x}^*) = p(\hat{x}_1) + p(\hat{x}_2)$. This prototype is like the centroid of a cluster. Now the loss of MI can be derived from 4 and 5 as:

$$\begin{aligned}
\Delta I(\hat{x}_1, \hat{x}_2) &= I(\hat{X}_{bef};Y) - \sum_y p(\hat{x}^*) p(y|\hat{x}^*) log \frac{p(y|\hat{x}^*)}{p(y)} \\
&= \sum_{y,i=1,2} (p(\hat{x}_i) p(y|\hat{x}_i) log \frac{p(y|\hat{x}_i)}{p(y)} - p(\hat{x}_i) p(y|\hat{x}_i) log \frac{p(y|\hat{x}^*)}{p(y)}) \\
&= \sum_{y,i=1,2} p(\hat{x}_i) D_{KL}(p(y|\hat{x}_i) || p(y|\hat{x}^*)).
\end{aligned} \tag{6}$$

As we see, $\Delta I(\hat{x}_1, \hat{x}_2)$ is the weighted distance of two original "prototypes" to the merged "prototype". Also, we can consider the loss of MI due to the merging of clusters $\hat{x}_1$ and $\hat{x}_2$ as the distance between $\hat{x}_1$ and $\hat{x}_2$. At each step, we greedily merge the closest ones. The algorithm is summarized as follows:

1. Initiate $C(X) \equiv X$, which means regard each point as a singleton cluster;
2. At each step, compute the distance (actually $\Delta I(\hat{x}_1, \hat{x}_2)$) between all pair of elements using formula 6.
3. Pick the pair which gives the minimum loss of MI $\Delta I(\hat{x}_1, \hat{x}_2)$.
4. Continue the merging operation until the loss of MI $\Delta I(\hat{x}_1, \hat{x}_2)$ is larger than the predefined threshold ϵ or number of clusters.

In summary, the motivation to learn the optimal number of clusters of *video-words* is twofold. The compact features with lower number of dimensions are efficient and effective to learn. Besides, compact features are easier to be encoded with spatiotemporal structure information. Here,we apply two steps to achieve this. We first use k-means algorithm to cluster the cuboids into *video-words*. Since the criterion for k-means is based on appearance similarity, cuboids belonging to one *video-word* are visually similar. Further, we group the *video-words* into some more compact but discriminative clusters via MMI clustering.

3. Spatiotemporal Structural Information

The bag of *video-words* approach ignores the spatial and temporal structural information of the features. In our work, we explore two approaches to capture this information, namely spatial correlogram and spatial temporal pyramid matching. In this section we describe the modified correlogram.

Assume n local cuboids denoted as $\mathcal{P} = \{p_1, p_2, ..., p_n\}$ extracted from video $\mathcal{V}$, and quantized into m labels $\mathcal{L} = \{l_1, l_2, ..., l_m\}$, which corresponds to the *VWC*s. We quantize the distance into K distance levels $\mathcal{D} = \{D_1, D_2, ..., D_K\}$, where $D_i = [d_{i1}\ d_{i2}]$ ($[x_1\ x_2]$ denotes an interval). The distance between two cuboids p_1 and p_2

is defined as a function $d(p_1, p_2)$, which could be the L_∞-norm or Euclidean distance. Consequently, the correlogram of two labels l_i and l_j with distance interval D_k is defined as a probability $\mathcal{R}$,

$$\mathcal{R}(D_k, l_i, l_j) = \mathbf{Pr}\big(l(p_2) = l_j | l(p_1) = l_i, d(p_1, p_2) \in D_k\big),$$

where $p_1, p_2 \in \mathcal{P}$, $1 \leq i, j \leq m$ and $1 \leq k \leq K$. From the correlogram of two labels l_i and l_j, we can know the probability of finding a cuboid p_2 with label l_j at D_k distance away from the given cuboid p_1 with label l_i.

Assume $\mathcal{R}_1$ and $\mathcal{R}_2$ represent correlgoram of videos $\mathcal{V}_1$ and $\mathcal{V}_2$ respectively, then the similarity between them is computed as,

$$S(\mathcal{R}_1, \mathcal{R}_2) = \sum_{k=1}^{K} \sum_{i,j=1}^{L} min(\mathcal{R}_1(D_k, l_i, l_j), \mathcal{R}_2(D_k, l_i, l_j)).$$

As the correlogram gives the local correlation of two *VWC*s , it is translation and rotation invariant. However, it may not be scale invariant due to the quantization of distance. Instead of using fixed absolute distance quantization, we use the relative distance quantization. Given a video, we get a bounding box around the object with diagonal length of L_d^{sub}. Then the quantization of relative distance can be computed as:

$$D_k^{rel} = D_k^{abs} \frac{L_d^{sub}}{L_d^{frm}}, \tag{7}$$

where L_d^{frm} denotes the diagonal length of the frame.

4. Experiments

We have applied our approach to two datasets: the KTH dataset [8] and the IXMAS multiview dataset [20]. The default experiment settings are as follows. From each action video 200 cuboids are extracted. All the results reported in this paper are obtained using the gradient-based feature descriptor. The initial codebook is generated by k-means algorithm, where 5 randomly selected videos of actors are used for training. We use SVM with Histogram Intersection kernel as the multi-classifier, and adopt the Leave One Out Cross Validation (LOOCV) and 6-fold cross validation (CV) strategy on KTH and IXMAS respectively. Specifically, we use 24 videos of actors as training and the rest as testing videos for KTH dataset, and 10 actors as training for multiview dataset. The results are reported as the average accuracy of 25 runs on KTH and 6 runs on IXMAS. In the following, the initial codebook and the optimal codebook refer to *video words* and *VWC*s respectively.

4.1. KTH dataset

The KTH dataset contains six actions. They are performed by 25 actors under four different scenarios of illumination, appearance and scale changes. In total it contains 598 video sequences. Fig. 1 shows some examples.

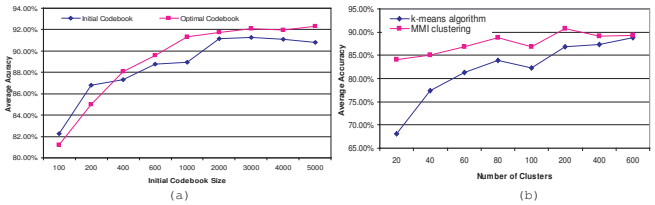

Figure 2. (a)A comparison of classification performances of the initial and the optimal codebooks using different initial codebook sizes. (b) The performance comparison of MMI clustering and k-means algorithms. MMI clustering reduces the initial dimension of 1,000 to the corresponding number.

4.1.1 Action recognition using orderless features

We investigate the gain of MMI clustering by comparing the classification performance before and after learning the optimal number of *VWC*s. Fig.2(a) shows the performance comparison between the initial codebook (before learning) and the optimal codebook (after learning) with different sizes of initial codebook. As we see, the optimal codebook can consistently improve the performance when the size of the initial codebook is large. This improvement is very significant. As Liu *et. al.*[7] observed in their experiments, SVM is strong classifier which can cope with higher dimensional features. Therefore, it is not easy to observe the gain of dimension reduction using SVM. It also shows that increasing the size of the initial codebook decreases the performance in case of k-means clustering, while using *VWC*s the performance increase slightly.

Instead of doing dimension reduction, we can directly get lower dimension using k-means clustering. Here, we also investigate the gain of MMI clustering compared to directly applying k-means. First, we create an initial codebook with size 1,000, which achieves 88.95% average accuracy. MMI clustering preserved 177 as the optimal number of *VWC*s, with an average accuracy of 91.31%. Further, we also performed eight different clustering with $\{20, 40, 60, 80, 100, 200, 400, 600\}$ clusters using MMI clustering and k-means algorithm respectively. Fig. 2 (b) shows the results. From the figure, we can see that MMI clustering can significantly improve the performance when the number of clusters (N_c) is small. This is due to better clustering or more compact data representation. The K-means algorithm groups the cuboids into *video-words* based on the appearance of the cuboids. When N_c is small, the intra-cluster variance is large, which hurts the performance. However, when MMI clustering groups the 1,000 *video-words* into new clusters, it tries to preserve the mutual information between the *video-words* and the actions, so the *video-words* in the same cluster may have strong correlation. Note that they are not necessarily similar in visual appearance. Although in MMI clustering intra-cluster variance of appearance may be large, it can preserve some meaningful concept correlations. Therefore, MMI clustering can still achieve better classification performance even

Figure 3. (a) Confusion table for the classification using the optimal number of *VWC*s (N_c=177, average accuracy is 91.31%). (b) Confusion table for the classification using the *VWC* correlogram. The number of *VWC* is 60, and 3 quantized distances are used (average accuracy is 94.15%).

with small N_c. While increasing N_c will probably cause the performance curves in the figure to meet at some point. It is logical, because MMI clustering starts with an initial codebook of size 1,000, which is the result of the k-means algorithm. Therefore, when N_c=1,000 there is no difference between them.

Another observation from our experiments is that the size of the training examples have little effects on the performance. We try different x-fold CV, where x=$\{$3, 5, 8, 12, 25$\}$ in our experiments with $N_c = 200$, and we obtain the corresponding average accuracy $\{$90.58, 89.97, 90.37, 90.67, 90.80$\}$(%). Hence our LOOCV (25-fold CV) training scheme, which has about 570 training videos, is reasonable. In another words, the performance is not significantly affected by changing of the number of training examples in our case.

In [12] and [22], the authors use pLSA to do unsupervised classification. More precisely, it is pLSA clustering which groups the videos to the topics (clusters). Each cluster is assigned to one action. Given a test video, pLSA will assign it to one major topic based on the probability. In order to check the unsupervised classification capability of our approach, we perform the double clustering scheme[18]. It has two phases. In the first phase, we use MMI clustering to get the optimal number of video-words clusters (*VWC*s). In the second phase, each video is represented by the *VWC*s, and we apply MMI clustering again to the new representations of the action videos. But this time, we only group the videos. We pick the optimal number of *VWC*s of $C = 177$ and set the number of "topics"(action clusters) to 10, which is a slightly larger than the number of actions (six). Our average accuracy is 84.13% which is slightly better than 81.50% [12] (they use 6 topics) and 68.53% [22] (they choose 10 topics) by pLSA.

Fig.3 (a) shows the confusion table for the classification using the optimal number of *VWC*s (N_c=177). From this table, we can see the "hand" related actions ("boxing", "hand clapping", "hand waving") are confused with each other. The "leg" related actions (e.g. "jogging","running" and "walking") are confused, especially for "jogging" and "running". In Fig.4 we show two example testing videos from each category with their corresponding *VWC* histograms to

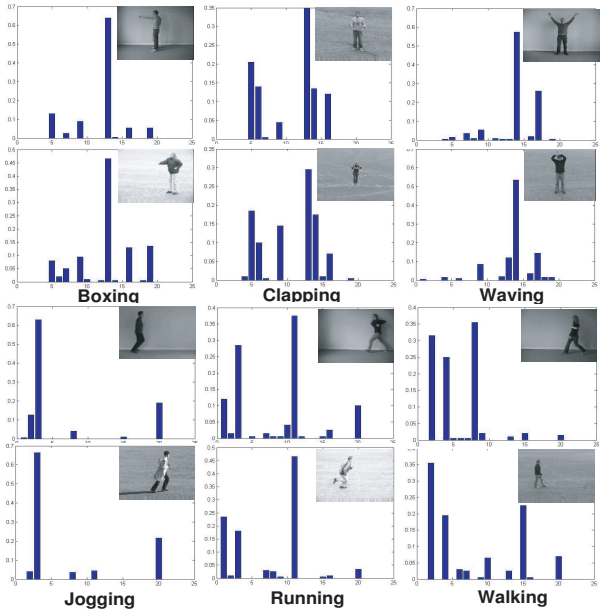

Figure 4. Example histograms of the *VWC*s (N_c=20) for two selected testing actions from each action category. It demonstrates that actions from the same category have similar *VWC* distribution, which means each category has some dominating *VWC*s.

demonstrate discrimination of the distribution of the learnt *VWC*s. Actions from the same category share the similar *VWC* distribution. It is also clear to see from the peaks of these histograms that some *VWC*s are dominating in one action but not others. When specifically looking into "jogging" and "running", one might note that they may have some overlap bins (e.g. bin no. 3 and 20). That is why "running" is confused with "jogging", which is consistent with the observation from the confusion table 3. Furthermore, in Fig. 1 we see the distribution of *VWC*s is more compact, while that of *video-words* is not.

4.1.2 Classification using spatiotemporal structural information

In order to encode the spatiotemporal structural information of the cuboids, we have two alternatives. One way is to encode the structural information into *video-words*, then use the learning tools (e.g. pLSA) to train. For instance, pLSA-ISM [22] performs pLSA clustering on the structural *video-words* by ISM model (In fact, we can say that pLSA-ISM does dimension reduction on the ISM model). We encode the structural information in a more straightforward way. Specifically, our model captures the structural information of the optimal *VWCs* instead of the *video-words*. As we discussed, one benefit of performing MMI clustering on the *video-words* is that we can capture more complicated structural information by fusing the compact and yet discriminative *VWC*s. When computing the correlgoram, we used a small number of *video words*(*VWs*) or *VWCs*, and adopted three absolute quantized distance intervals [8 16 32], from which we can estimate the relative distance by us-

dimension	20	40	60	80
VW(%)	68.09	77.42	81.27	83.94
VWC(%)	84.11	85.13	86.79	88.80
VW_Correl(%)	82.28	84.92	**86.61**	85.45
VWC_Correl(%)	87.09	90.47	**94.16**	91.29
STPM(%)	88.21	92.90	93.79	**93.81**

Table 2. The performance comparison between different models. *VW* and *VWC* respectively denote *video-words* and *video-word-clusters* based method, and VW_Correl and VWC_Correl are their corresponding correlgoram models. *STPM* denotes the Spatiotemporal Pyramid Matching approach. The dimension denotes the number of *VW*s and *VWC*s.

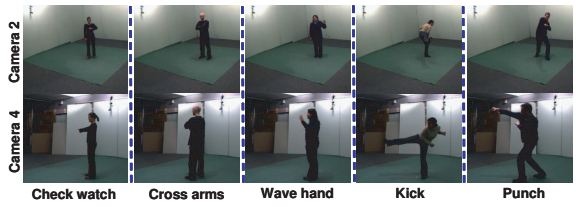

Figure 5. Two views of five selected action examples from IXMAS dataset.

	cam1	cam2	cam3	cam4
1,000 VWs	75.6	73.77	69.13	70.41
186 VWCs	76.67	73.29	71.97	72.99

(a)

	cam1	cam2	cam3	cam4
Ave. Accuracy	72.29	61.22	64.27	70.59

(b)

Figure 6. (a)Performance (%) comparison between the original 1,000 *video-words* and the optimal 189 *video-word-clusters*. (b)Average accuracy (%) using three views for training and single view for testing.

ing Equation 7. Table 2 shows the performance comparison between the *VW* correlogram (*VW_Correl*) and the *VWC* correlogram (*VWC_Correl*). As shown in the table, both *VW_Correl* and *VWC_Correl* achieve a large improvement compared to the corresponding orderless models, *VW* and *VWC* model. *VWC_Correl* also outperforms *VW_Correl*, which further verifies *VWC*s are more discriminative. When the dimension (the number of *VW*s or *VWC*s) is larger than 60, we do not observe much performance improvement.

So far little research work has been reported on exploiting the temporal structure of the *video-words*. Nowozin *et. al.*[10] represent the action as overlapping sub-clips and perform subsequences mining and matching to capture the temporal information of *video-words*. We extend the Spatial Pyramid Matching [9] to the time dimension. Specifically, we perform action matching at multiple resolution along the time dimension. We also quantize the position of the points. In our model, we have 15 bins quantization for spatial information and three levels of pyramid in time dimension. We show the results of Spatiotemporal Pyramid Matching (*STPM*) in table 2. It also obtains better performance compared to *VWC* model. Although it is difficult to directly compare our approach to other approaches due to different experiment settings, we summarize all the results of the bag of *video-words* related approaches in table 3 as a reference.

Methods	Accuracy (%)	Structural Inf.
Our SVM VWCs	91.31	No
Our VWC Correl.	94.16	Yes
pLSA_ISM*[22]	83.92	Yes
WX_SVM [22]	91.6	Yes
pLSA [12]	81.50	No
Nowozin *et. al.* [10]	84.72	Yes
Dollar *et. al.*[2]	80.66	No
Schuldt *et. al.*[16]	71.71	No

Table 3. The performance of the different bag of *video-words* related approaches. pLSA_ISM is the major contribution of [22].

4.2. IXMAS Multiview dataset

We also applied our approach to IXMAS multiview dataset. It contains 14 daily-live actions performed three times by 12 actors (Fig.5 shows some examples.). 13 action videos were selected for our experiments. Most current approaches applied to this dataset need some pre-procssing such as background substraction or 3D model construction. We are the first to use the data for the bag of *video-words* approach, which does not require background substraction. We select four views excluding the top view. We generate the codebook with 1,000 *video-words* using k-means algorithm on four actor' actions. Though it is difficult to directly compare our approach with [20] and [24], we obtained competitive performance, noting that our approach does not require 3D model construction.

Learning from four views: We adopt 6-fold CV scheme, namely 10 videos of actors for learning and the rest for testing. In the testing phase, we designed two testing schemes: recognition using single view and multiviews. Our experimental setting is similar to that of [20]. Fig.6(a) provides the single view recognition accuracy comparison between models learnt from the original 1,000 *video-words* and 189 *VWC*s. It shows the VWC achieves better results than the original *video-words*. In the following, all reported results are achieved by using the optimal *VWC*s. Our average performance of each view outperforms that of [20], where $\{65.4, 70.0, 54.3, 66.0\}(\%)$ were reported as average accuracy for four views, and they only tested on 11 actions. Fig.8 plots the detail of recognition accuracy for each action.

In the recognition from multiviews, we adopt simple voting method. Fig. 7 shows the confusion table of the recognition using voting from four views. The average rate is 82.8%, which is slightly better than the one reported in [20] (81.27%) and [24] (80.6%). It is interesting to note that our approach works much better on large motions, like "walk", "pick up" and "turn around", yet it somewhat gets confused with small hand motions such as "point", "cross arms", "wave hand" and "scratch head". One possible reason is that our features are orderless, which mean they do not have any view constraints between them. The hand re-

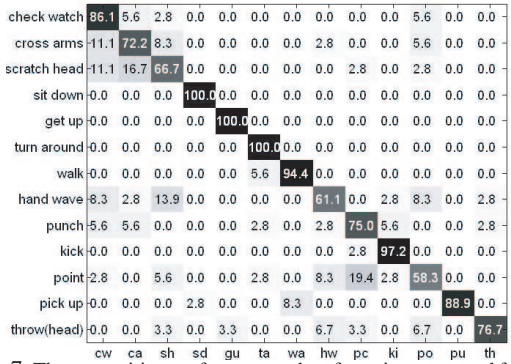

Figure 7. The recognition performance when four views are used for training and single view is used for testing. The average accuracy is 82.8%

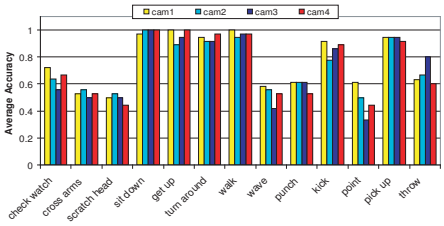

Figure 8. The recognition performance when four views are used for training and single view is used for testing.

lated actions share some basic motions, which makes it difficult to distinguish by only using the orderless features. For instance, "wave hand" is easy to be confused with "scratch head".

Learning from three views: We trained actions from three selected views, and tested on the fourth view. Therefore there is no information from the fourth view when learning the models, which includes the codebook generation. Fig. 6 (b) lists the average accuracy of this experiment. The results are still satisfactory. We also tried one more complicated experiment. We still train the models using three views, but when testing the model, from the fourth view we only select the subjects which are not included in the training phase. In this experiment, the learning process is totally blind to testing examples. The average accuracy is $\{42.6, 38.08, 58.3, 62.48\}$. The third and fourth views get better results, which means the other three views can provide enough information when testing on this view. To increase the performance, we conjecture more views are necessary.

5. Conclusion

In this paper, we propose the MMI clustering approach to find the compact and yet discriminative *VWC*s. Since the bag of *video-words* ignores the spatial and temporal structural information, we also use spatial correlogram and temporal pyramid match to make up it. Our approach have been extensively tested on two public datasets: KTH and IXMAS multiview datasets, and we obtain very impressive performance on both dataset. We are the first to apply the bag of *video-words* related approach on mutliview dataset, and have obtained competitive results.

6. Acknowledgements

This research was funded by the US Government VACE program.

References

[1] A. Bobick and J. Davis. The recognition of human movement using temporal templates, PAMI 23(3):257-267,2001.

[2] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie. Behavior recognition via sparse spatio-temporal features. ICCV workshop: VS-PETS 2005.

[3] A. Efros, A. Berg, G. Mori and J. Malik. Recognizing action at a distance, ICCV 2003.

[4] C. Fanti, L. Zelnik-Manor and P. Perona. Hybrid models for human recognition, ICCV 2005.

[5] R. Fergus, L. Fei-Fei, P. Perona and A. Zisserman. Learning object Categories from Google's Image Search, ICCV 2005.

[6] Y. Ke, R. Sukthankar and M. Hebert. Efficient visual event detection using volumetric features, ICCV 2005.

[7] J. Liu and M. Shah. Scene Modeling using Co-Clustering, ICCV 2007.

[8] I. Laptev. On space-time interst points, IJCV, 64(2-3):107-123, 2005.

[9] S. Lazebnik, C. Schmid and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, CVPR 2006.

[10] S. Nowozin, G. Bakir and K. Tsuda. Discrinative subsequence mining for action classification, ICCV 2007.

[11] J. Niebles and L. Fei-Fei. A hierarchical model of shapes and appearance for human action classification, CVPR 2007.

[12] J. Niebles and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words, BMVC 2006.

[13] V. Parameswaran and R. Chellappa. View Invariance for Human Action Recognition, IJCV, 66(1), 2006.

[14] E. Shechtman and M. Irani. Space-time behavior based correlation, CVPR 2005.

[15] Y. Song, L. Goncalves and P. Perona. Unsupervised learning of human motion, PAMI, 25(25):1-14,2003.

[16] C. Schuldt, I. Laptev and B. Caputo. Recognizing human actions: A local svm approach, ICPR, 2004.

[17] S. Savarese, J. Winn and A. Criminisi. Discriminative Object Class Models of Appearance and Shape by Correlatons, CVPR 2006.

[18] N. Slonim and N. Tishby. Document Clustering using Word Clusters via the Information Bottleneck Method,ACM SIGIR 2000.

[19] N. Tishby, F. Perira and W. Bialek. The Information Bottleneck Method. Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing.

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

[21] J. Winn, A. Criminisi and T. Minka. Object Categorization by Learned Universal Visual Dictionary, ICCV 2005.

[22] S. Wong, T. Kim and R. Cipolla. Learning Motion Categories using both Semantics and Structural Information, CVPR 2007.

[23] A. Yilmaz and M. Shah. Action sketch: A novel Action Represetantion, CVPR 2005.

[24] F. Lv and R. Nevatia, Single View Human Action Recognition using Key Pose Matching and Viterbi Path Searching, CVPR 2007.