MULTI-VIEW DESCRIPTOR MINING VIA CODEWORD NET FOR ACTION RECOGNITION

Jingyu Liu[1], *Yongzhen Huang*[1], *Xiaojiang Peng*[2] *and Liang Wang*[1*]

[1]National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing 100190, China
[2]Hengyang Normal University
{jingyu.liu,yzhuang,wangliang}@nlpr.ia.ac.cn, xiaojiangp@gmail.com

ABSTRACT

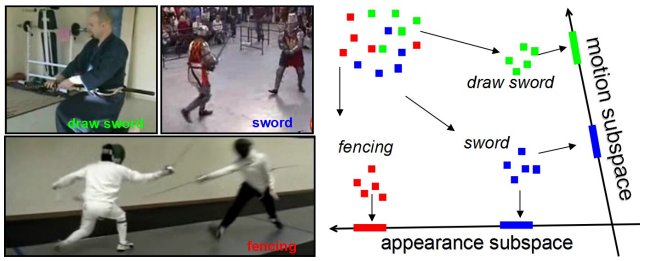
Action recognition is an important yet challenging task in computer vision. A successful and widely used framework in this field is the Bag of Visual Words (BoVW), wherein the first step is to extract local features. One critical property of local features is that they are often multi-view, e.g., dense trajectory feature includes both appearance and motion properties. Different types of features are aligned together in coding and pooling thus leading the process to be heavily entangled. Our motivation is to disentangle each sub-descriptor and let them contribute to the maximum extent. To achieve this, a codeword net is constructed via exploiting the relation between features and codewords. Based on the codeword net, features from the same viewpoint are pooled together. Experiments on two large scale action recognition datasets, UCF50 and HMDB51, demonstrate that our approach can enhance the state-of-the-art algorithms.

Index Terms— multi-view descriptor mining, codeword net, BoVW, action recognition

1. INTRODUCTION

The decreasing cost of camera equipment results in huge volumes of video data nowadays. This excessive amount requires automatic methods of video processing and analysis, among which human action recognition is a fundamental part for applications such as event detection, video indexing, human interface, video description, etc. Though remarkable improvements have been made, action recognition still remains challenging due to camera motions, view point changing, large intra-class variations, etc.

The Bag of Visual Words (BoVW) [1] is a popular framework which generates good representation for action recognition. Leveraging BoVW, researchers mainly explore discriminative representation along two approaches: Sophisticated hand-crafted local features and elaborately designed encoding methods. As for local feature, early efforts include local spatio-temporal features [2] and Spatio Temporal Interest Points (STIP) [3]. Later, Dense Trajectory Feature (DTF) [4], outperforms the STIP. DTF densely samples trajectory-aligned cuboids at regular positions in space and time, then extracts traditional HOG [5], HOF [6] and MBH [7] features. For coding methods, LLC [8] based methods and Fisher Kernel [9] methods have been successful. An efficient super vector based method called VLAD [10] also achieves promising results.



(a) (b)

Fig. 1: An illustration of our motivation. (a) three similar action categories from HMDB51 all related with sword-like weapons. (b) features are more distinctive when projected to different subspaces of appearance and motion.

The BoVW pipeline usually contains five steps: local feature extraction, codebook generation, feature encoding, pooling and classification. At the coding stage, each descriptor generates responses on its related codewords. Then pooling accumulates responses of similar features into one value. However, due to the multi-view property of local descriptor, features located at the same codeword are not necessarily similar. One codeword's neighboring features may share different affinities with it if projected to different subspaces. This property accords with descriptors applied for action recognition, e.g., the dense trajectory descriptor includes both appearance and motion characteristics. More explanation is illustrated in Figure 1. Figure 1(a) shows "draw sword", "sword" and "fencing" from the HMDB51 dataset. They are similar in that people all act with sword-like weapons. But in particular, "fencing" is characteristic in appearance compared with "sword", and "draw sword" is distinguishable in action compared with "sword", as illustrated in Figure 1(b). To utilize

*The corresponding author

the multi-view property of action descriptors, features from the same view should be pooled together. To this end, we use a codeword net to separate the feature space into local regions. Accordingly, features located in different local regions are pooled separately. We call such a process "multi-view pooling". One critical step of multi-view pooling is constructing the codeword net. We propose a data-driven approach to construct the codeword net. The details will be discussed in Section 2.

Multi-view pooling can be embedded into different encoding methods. In particular, we embed it with two representative coding strategies: LLC [11] and FK [9]. We choose them because LLC is fast and FK is the state-of-the-art coding method. Our approach is evaluated on two challenging datasets: UCF50 and HMDB51 with 50 and 51 action classes respectively. Experimental results demonstrate that our approach obtains superior recognition performance over current coding methods.

2. OUR METHOD

To utilize the multi-view property in the BoVW framework, two main stages, namely "Codeword net construction" and "Multi-view pooling", are embedded into the traditional BoVW framework. Details will be respectively discussed in Section 2.1 and Section 2.2.

2.1. Construction of codeword net

Let $\mathcal{A} \in \mathbb{R}^{M\times M}$ be the adjacency matrix used to represent the codeword net, where M is the number of codewords. $\mathcal{A}(i,j) = 1$ means the i^{th} and the j^{th} codewords are linked and $\mathcal{A}(i,j) = 0$ otherwise. Our aim is to construct an optimal $\mathcal{A}$ which best describes local features. Let $\mathcal{X}$ be a set of N features in the D-dimensional feature space, i.e., $\mathcal{X} = [x_1, x_2, \cdots, x_N] \in \mathbb{R}^{D\times N}$, and $\mathcal{C}$ be a codebook consisting of M codewords, i.e., $\mathcal{C} = [c_1, c_2, \cdots, c_M] \in \mathbb{R}^{D\times M}$.

For each local feature, we use its neighboring codewords to describe the displacement between this feature and its nearest codeword. The optimization problem is formulated as:

$$\min_{w_i} \|r_i - \mathcal{Z}w_i\|_2^2 + \lambda \sum_{k=2}^{K} (w_{i,\widetilde{k}} d_{i,\widetilde{k}})^2 \tag{1}$$

$$s.t.\ \|w_i\|_1 = 1 \tag{2}$$

where

$$r_i = x_i - c_{i^*}$$

$$\mathcal{Z} = [z_2, z_3, \cdots, z_K], z_k = c_{\widetilde{k}} - c_{i^*}$$

$$d_{i,\widetilde{k}} = \|r_i - z_k\|_2 = \|x_i - c_{\widetilde{k}}\|_2,$$

where r_i denotes the displacement between x_i and its nearest codeword c_{i^*}. $\widetilde{k}$ is the index of the k^{th} closest codeword of x_i. The constraint term $\sum_{k=2}^{K} (w_{i,\widetilde{k}} d_{i,\widetilde{k}})^2$ indicates that the reconstruction tends to choose the codewords with small $d_{i,\widetilde{k}}$, i.e., the nearby codewords of x_i. This is called "local" constraint and demonstrated to be useful to enhance the robustness of reconstruction [12, 11]. Thus, the optimization result, i.e., w_i, reflects a set of robust codeword links used to solve r_i.

The optimization problem described in Eqn. (1)∼(2) has an analytical solution:

$$w_i = [(\mathcal{Z} - r_i\mathbf{1}^T)^T(\mathcal{Z} - r_i\mathbf{1}^T) + \lambda diag^2(d_i)]^{-1}\mathbf{1}. \tag{3}$$

After obtaining the coefficient matrix w, the adjacency matrix $\mathcal{A}$ is solved in an iterative manner. $\mathcal{A}$ is firstly set as a zero matrix, i.e., no codewords are linked. Then, it is updated after we solve a reconstruction problem for x_i. This process is iterated N times, and each iteration is formulated as:

$$\mathcal{A}(i^*, I_k) = \begin{cases} 1, & if\ w_{i,k} \geq T \\ unchanged, & otherwise \end{cases}, \tag{4}$$

$$I = [I_2, I_3, \cdots, I_K], I_k = 1, 2, \cdots, M$$

where i^* denotes the index of the closest codeword of x_i. I is the index of the $K-1$ nearest codewords. T is a threshold parameter, which controls the final number of links per codeword.

Since enumerating each local feature to solve Eqn.(1)∼(2) is time-consuming, we solve it with respect to codewords. For each codeword c_i, we find all the local features whose nearest codeword is c_i, thus we can optimize the reconstruction procedure in batches. The algorithm for constructing codeword net can be summarized as follows:

Algorithm 1 Codeword Net Construction

Input: codebook: $\mathcal{C} = [c_1, c_2, \cdots, c_M] \in \mathbb{R}^{D\times M}$; a set of features: $\mathcal{X} = [x_1, x_2, \cdots, x_N] \in \mathbb{R}^{D\times N}$

Output: optimal adjacency matrix: $\mathcal{A}$

1: initial $\mathcal{A} = 0_{M,M}$;
2: **for** each codeword c_i **do**
3: find the local feature set F_i whose nearest codewords are c_i;
4: compute the coefficient matrix W_i using F_i via Eqn.(3);
5: update $\mathcal{A}$ via Eqn.(4);
6: **end for**

2.2. Multi-view pooling (MVP)

In this subsection, we explain how to perform MVP with the learned codeword net. Figure 2 depicts the difference between traditional pooling and MVP. Figure 2(a) shows a toy example of traditional pooling in a 2-D feature space. $c_1 \sim c_5$

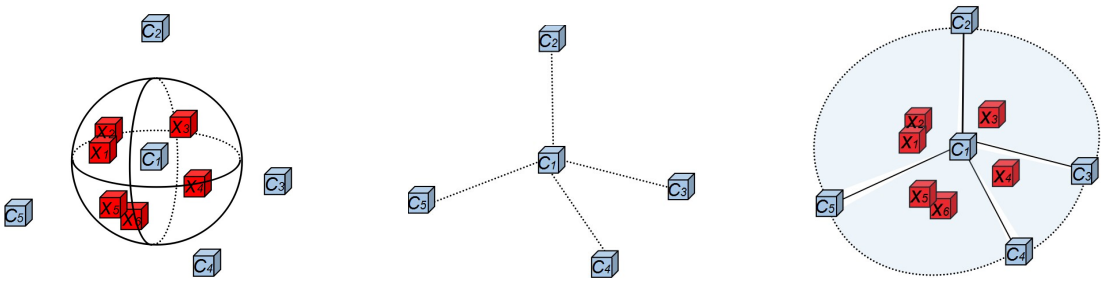


Fig. 2: An illustration of the proposed MVP.

are five codewords, and $x_1 \sim x_6$ are six features around c_1. In traditional codeword-oriented pooling, these features will be assigned to c_1, i.e., all coding responses will be pooled to one value. Figure 2(b) shows the codeword net resolved by Eqn.(1)∼(2). Figure 2(c) illustrates the subarea related with features assigned to c_1, which is further divided by the codeword links into four local regions. Features assigned to each local regions are pooled separately. We take MAX pooling as an example to formulate this process. After applying MVP, the response on the region q of c_p is:

$$r_{p,q} = \max_i \{v_{i,p}\}, \tag{5}$$

$$i \in \{\underset{j \in \mathcal{R}(p)}{\operatorname{argmax}} \frac{V_{c_p x_i} \cdot V_{c_p c_q}}{\|V_{c_p x_i}\| \|V_{c_p c_q}\|} = q\} \tag{6}$$

where c_p is an active codeword of x_i, $v_{i,p}$ is the coding value of x_i on c_p, $\mathcal{R}(p)$ is the set that contains all codewords linking with c_p in the codeword net, and V_{xy} denotes the vector from x to y.

3. EXPERIMENT

In this section, we evaluate our approach on two challenging datasets, i.e., UCF50 [13] and HMDB51 [14]. Datasets and settings will be detailed in Section 3.1. Experimental results will be shown and analyzed in Section 3.2.

3.1. Datasets and settings

UCF50 [13] is an action recognition dataset with 50 action categories, consisting of 6681 realistic videos from YouTube. HMDB51 [14] is an action recognition dataset with 51 action categories, consisting of 6766 manually annotated clips from movies and YouTube. It is more challenging due to large inter-class ambiguity and intra-class diversity. Examples of some categories of the two datasets are shown in Figure 3.

We conduct three experiments on the two datasets. Details of the three experiments are as follows: (i) MVP with different codebook sizes on UCF50. (ii) MVP with different codebook sizes on HMDB51. (iii) MVP with different average linking number per codeword on UCF50. In the first and the second experiment, for a consistent comparision, we tune the threshold T in Eqn.(1) to assure that the average linking number per codeword is two. In the third experiment, we fix the codebook size to 400 and test the accuracy with different average linking numbers per codeword of 2,4,8 and 16.

For UCF50, We follow the recommended evaluation protocols of Leave One Group Out cross validation (LOGO) and report the average accuracy. For HMDB51, We follow the experimental settings in [14] with three train/test splits provided, and report the mean average accuracy over three splits. For both datasets, we use Wang's code [15] to extract HOG, HOF and MBH features along the improved dense trajectories. Linear SVM is implemented with LIBSVM [16] and the *one-vs-rest* strategy is used for multi-class classification.

3.2. Experimental results

EXPERIMENT I: Results of LLC and LLC+MVP with different codebook sizes on UCF50 are shown in Table 1. The experimental results show that LLC+MVP obtains a significant improvement on LLC itself. It is reasonable to expect that Multi-view Pooling can also work well with other sparse coding based methods. However, one can also observe that the improvement becomes a little slighter with the codebook size becoming larger. We believe that this is mainly due to the over-fitting effect, i.e., with a larger codebook size, pooling views of each codeword would be overlapped to some extent.

Table 1: Results of LLC and LLC+MVP with different codebook sizes on UCF50.

Codebook size	LLC	LLC+MVP
400	64.03%	71.37%
800	72.47%	78.25%
1,600	78.09%	82.68%
3,200	81.71%	85.77%
6,400	83.86%	87.19%

EXPERIMENT II: Results of LLC and LLC+MVP with different codebook sizes on HMDB51 are shown in Table 2.

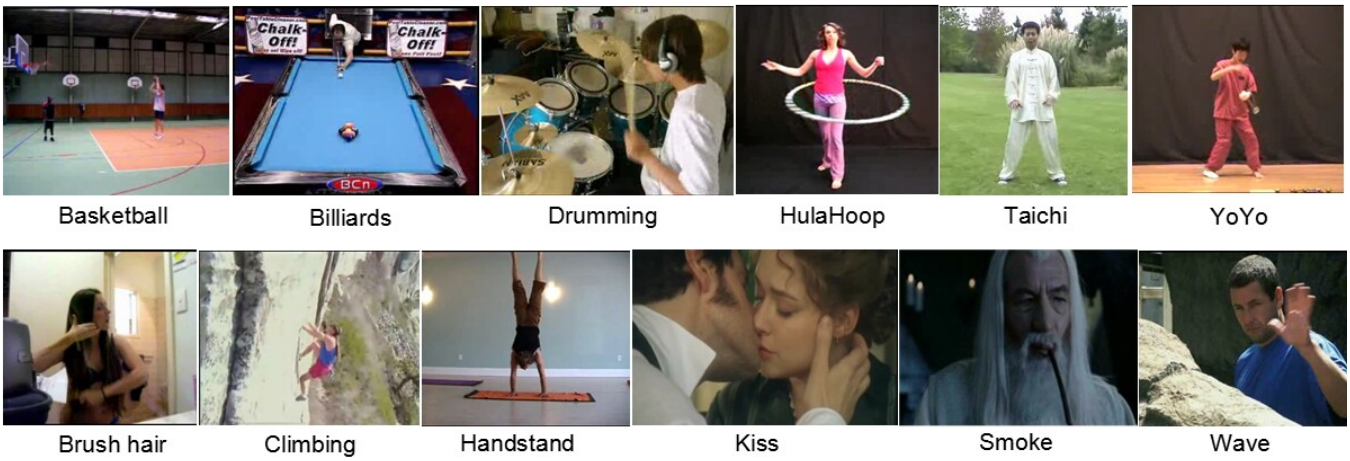


Fig. 3: Samples from two datasets. The top row is UCF50 and the bottom row is HMDB51.

The experimental results show that on a much more challenge dataset, LLC+MVP also obtains significant improvement on LLC itself.

Table 2: Results of LLC and LLC+MVP of different codebook sizes on HMDB51.

Codebook size	LLC	LLC+MVP
1,000	35.42%	41.87%
2,000	41.22%	45.90%
4,000	44.42%	48.76%
8,000	47.63%	49.95%

EXPERIMENT III: Results of LLC+MVP with different average linking numbers are shown in Tabel 3. The results show that accuracy can be improved by separating more views for each codeword. We believe that this is mainly due to the high dimensional representation of action descriptors, leading to multiple views of each codeword.

Table 3: Results of LLC+MVP of different average linking numbers on UCF50.

Linking number	2	4	8	16
accuracy	71.37%	73.33%	77.20%	78.60%

4. CONCLUSION

In this paper, we have analyzed the multi-view property of local descriptors in action recognition. To make it more compatible with the current Bag of Visual Words (BoVW) framework, we have proposed multi-view pooling (MVP), which accumulates responses of features from the same view. A codeword net is constructed to guide the process of MVP by exploiting the relations between features and codewords. We have applied our algorithm in action recognition on two standard datasets of UCF50 and HMDB51. Results show that the proposed method improves the state-of-the-art algorithms.

5. ACKNOWLEDGMENTS

This work is jointly supported by National Basic Research Program of China (2012CB316300), and National Natural Science Foundation of China (61175003, 61202328, 61420106015, U1435221).

6. REFERENCES

[1] G. Csurka, C. Dance, L.X. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *European Conference on Computer Vision (ECCV), 2004*.

[2] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*.

[3] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[4] H. Wang, A. Klaser, C. Schmid, and C.L. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011*.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR), 2005*.

[6] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *European Conference on Computer Vision (ECCV), 2006*.

[7] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition (CVPR), 2008*.

[8] J.C. Yang, K. Yu, Y.H. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2009*.

[9] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition (CVPR), 2007*.

[10] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *Pattern Analysis and Machine Intelligence,*, vol. 34, no. 9, pp. 1704–1716, 2012.

[11] J.J. Wang, J.C. Yang, K. Yu, F.J. Lv, T. Huang, and Y.H. Gong, "Locality-constrained linear coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010*.

[12] K. Yu, T. Zhang, and Y.H. Gong, "Nonlinear learning using local coordinate coding," in *Advances in Neural Information Processing Systems (NIPS), 2009*.

[13] K.K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, vol. 24, no. 5, pp. 971–981, 2013.

[14] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *International Conference on Computer Vision (ICCV), 2011*.

[15] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *International Conference on Computer Vision (ICCV), 2013*.

[16] C.C. Chang and C.J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST),*, vol. 2, no. 3, pp. 27, 2011.