

# Fudan University at TRECVID 2009: High-Level Feature Extraction and Copy Detection

Xiangyang Xue, Yingbin Zheng, Hong Liu, Zhaohui Wen, Xiaochuan Guo,  
Renzhong Wei, Hong Lu, Qi Zhang

Shanghai Key Laboratory of Intelligent Information Processing  
School of Computer Science, Fudan University, China

## 1 Introduction

In this notebook paper we describe our participation in the NIST TRECVID 2009 evaluation. We took part in two tasks of benchmark this year, i.e., high-level feature extraction and content-based copy detection pilot.

For high-level feature extraction, we submitted 4 automatic runs:

**Fudan.Global:** this run is based on global features of keyframes.

**Fudan.Local:** this run is based on local features of keyframes.

**Fudan.Rerank:** this run is based on local features and spatial information of keyframes.

**Fudan.Fusion:** this run is based on the fusion of global and local features of keyframes.

Focus of our system was on the effective utilization of global and local visual features.

For content based video copy detection, we submitted 2 runs for video-only type:

**FudanVID.v.balanced.fudan09Lo1:** this run is optimized in “balanced”.

**FudanVID.v.nofa.fudan09Lo2:** this run is optimized in “no false alarms”.

The main steps in our copy detection system include: filtrate a part of the query videos with the global method; retrieve the rest of the queries with the local method.

## 2 High-Level Feature Extraction

For high-level feature extraction task, we principally focus on:

(1) Local descriptors are experimented besides the color, texture, edge features. To add spatial information to the basic local features, we also explored the spatial layout partition on fixed grids and the spatial relation between two visual words.

(2) Effective fusion of texture, edge and color features, and effective fusion of global and local features.

## 2.1 Feature Extraction

This year we explore both global visual features, i.e. the MPEG-7 descriptors [1], and local features, i.e. SIFT feature [2] and bag-of-visual-words [3].

### 2.1.1 Global Feature Extraction

We extract six MPEG-7 visual features [1] for each keyframe of the video shots as our HLF system last year [4]. To reduce the complexity, all these low-level features are extracted at global scale. The features are:

- (1) Color Layout Descriptor (CLD): 12 dims;
- (2) Color Structure Descriptor (CSD): 256 dims;
- (3) Scalable Color Descriptor(SCD): 64 dims;
- (4) Homogenous Texture (HT): 62 dims;
- (5) Edge Histogram Descriptor(EHD): 80 dims;
- (6) Region Shape (RS): 35 dims.

### 2.2.2 Local Feature Extraction

For each keyframe in the dataset, we extract salient image patches using Harris-Laplace detector [5] and SIFT descriptor [2]. We use the local feature implementation of [6]. A codebook vocabulary  $V=\{v_1, v_2, \dots, v_n\}$  of SIFT points is constructed through clustering of the local features. In our experiment we choose  $n=1000$ . Then the keyframe can be described as a bag of visual words (BoW) [3]. A codebook histogram is obtained for each keyframe with each bin representing a codeword  $v_i$  in  $V$ .

Besides the standard BOW model, we also incorporate spatial information. Spatial layout partition scheme [7] is a simple but helpful method. We select the  $2*2$  grid (Gr22) and  $1*3$  grid (Gr13) to represent the layout for our system(Fig. 1). Another approach to represent the spatial information is to consider spatial correlogram (Corr) between codewords in vocabulary [8]. This method works by augmenting the vocabulary histogram with the distance distribution of pairwise visual words.



Fig. 1. Spatial layout partition. Form left to right: original image, Gr22, Gr13.

## 2.2 Experiments

**Experimental Setup.** We use TRECVID 2009 collaborative annotation organized by LIG and LIRIS for training our models. Each classifier is trained by SVM (we use the libSVM package [9]). The kernel for global features is RBF, while the kernel for local features is chi-square kernel [10]. For each video shot in the testing set, we extract 5 keyframes to capture more information from the video shot. The maximum score of the 5 keyframes is the final score of the shot. In the experiments last year [4], we found that the fusion of too many classifiers leads to worse performance, so we use only a few classifiers to fuse. For fusion of different features, we apply linear weighted fusion method for its simple and efficiency.

We submitted a total of 4 automatic runs. The description and infAP of each run are shown in the following Table 1.

**Table 1.** Description and infAP of our HLF runs.

Run	infAP	Description
Fudan.Global	0.063	Global feature baseline
Fudan.Local	0.063	Local feature baseline
Fudan.Rerank	0.083	Local feature reranking
Fudan.Fusion	<b>0.105</b>	Fusion of Fudan.Rerank and Fudan.Global

The fusion run is our best submission run with infAP 0.105. This result is expected since the fusion is able to represent both global visual features and local saliency regions. For Fudan.Local run and Fudan.Rerank run, we can find a significant improvement for adding the spatial information.

Table 2 summarizes the components of Fudan.Global run. EHD seems to be the most helpful features in the six MPEG-7 features we used. The result with single global feature is poor; however, when we apply linear fusion, the infAP value increases. This may due to the complementary abilities of these features to express texture, edge and color information of the keyframes.

**Table 2.** Comparison of infAP within our global feature methods

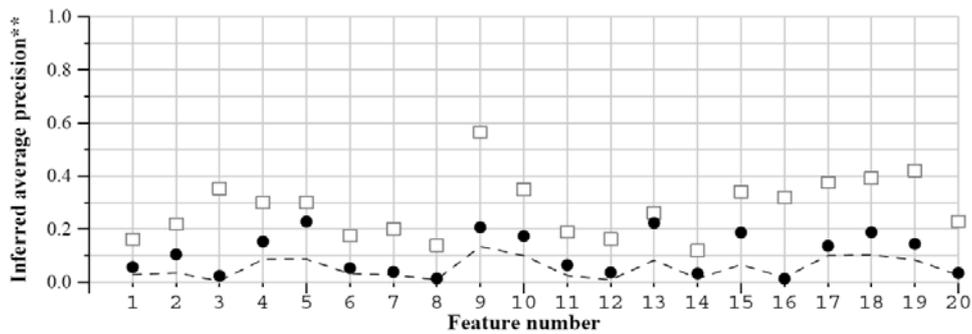
Method	EHD	CSD	HT	CLD	SCD	RS
infAP	0.023	0.011	0.009	0.008	0.004	0.004
Method	EHD+CSD	EHD+CSD+CLD	EHD+CSD+CLD+SCD+HT+RS			
infAP	0.035	0.048	<b>0.063</b>			

Table 3 summarizes the components of Fudan.Rerank run. Methods which incorporate spatial information outperform the vocabulary histogram method. Among the spatial-incorporated methods, Gr13 achieves the highest infAP. Another observation is that the vocabulary histogram method is comparable with the fusion of global features.

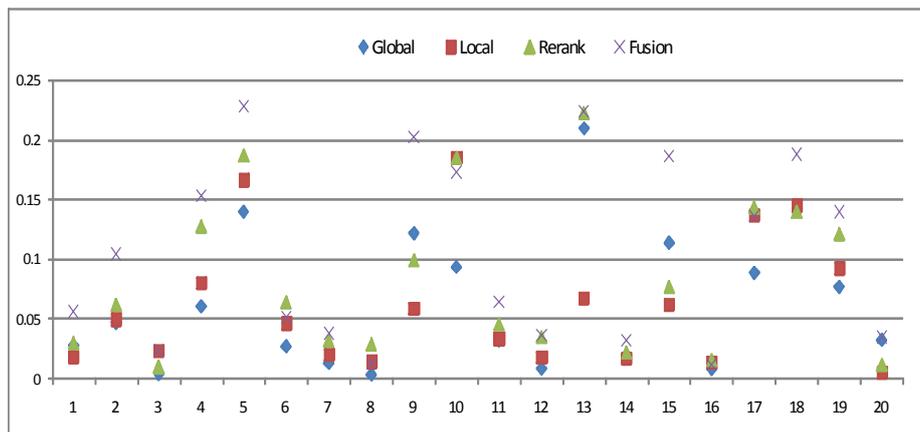
**Table 3.** Comparison of infAP within our local feature methods

Method	Hist	Corr	Gr22	Gr13
infAP	<b>0.063</b>	0.070	0.067	0.077
Method	Hist+Gr22	Hist+Gr22+Corr	Hist+Gr22+Corr+Gr13	
infAP	0.076	0.078	<b>0.083</b>	

Figure 2 and Figure 3 compare the performance of our runs and others with regard to each high-level feature in the evaluation. Generally, automatic concept detection still faces great challenges when dealing with most of the concepts. But some concepts, e.g. “005 - Doorway”, “013 - Person-eating”, “015 - Hand”, show good performance in our best run.



**Fig. 2.** Our best run score (dot) versus median (---) versus best (box) by feature.



**Fig. 3.** Comparison within our 4 submissions.

### 3 Content Based Copy Detection

In video copy detection, much more data has to be processed than in image copy detection, images (video frames) feature selection becomes a key point to develop a specific approach to video comparison. Usually, the features employed are simple, distinctive and easy to compute. Hampapur et al. [11] examined several sequence-matching methods based on the motion, ordinal, and color features, and reported that the ordinal signature achieves the best video copy detection performance. We also compare several image low-level features for CBCD and also find the ordinal signature having better performance for specific cope type (especially for Change of gamma).

In addition, in large scale video copy detection, the computational costs are also an important criterion for measuring the comparison method. Commonly, sparse comparison methods require less computational resources during the comparison process. On the other side, dense comparison approaches are more robust. To get a trade-off between computational costs and detection precision, we cluster the consecutive video frames in advance, it is different to the common shots segmentation, because video information has a strong temporal redundancy, and our aim only makes the consecutive similar video frames become a cluster and reduces the computational costs. Furthermore, clustering the similar video frames brings an advantage, namely, it can cope with video sequences with different resolution, frame rate and bit rate [12].

The framework of our content-based video copy detection system this year is described in Fig. 4.

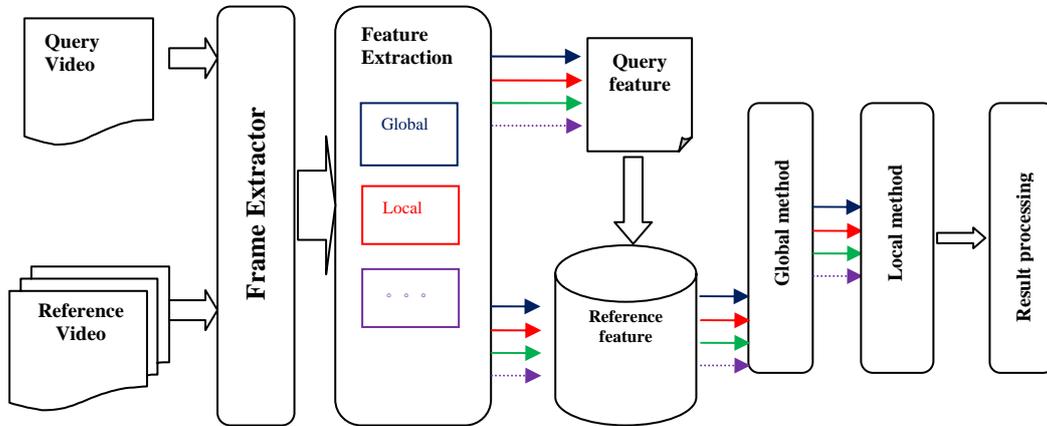


Fig. 4. System framework

#### 3.1 Framework Description

In this part, we will describe our content-based video copy detection system. The system includes three steps:

- Key Frame Extraction

- Feature Extraction
- Video Sequence Matching

We will now describe three steps blow detailedly.

### 3.1.1 Keyframe Extraction

The first step of our CBCD system is extracting frames from videos; different frame extraction methods are used for query videos and reference videos.

For reference videos, a fixed number of frames per second are extracted. In our runs, 0.5 frames per second are extracted on average.

For query videos, we extract one frame per 0.2 second. Then we clustering all the frames with time constrain (in the temporal order). Final, three frames are chosen in every class.

### 3.1.2 Feature Extraction

It is hard for our system to retrieve some transform styles (picture-in-picture style) in TRECVID 2008, because we used only the global features. Therefore, we used two different kinds of features this year. One is global feature: the GG feature, another one is SIFT.

#### *Global Feature*

Our global feature, GG feature, which is based on the OIS feature [11] (Ordinal Intensity Signature), is described in Fig. 5 below:



150	170	190	155	180	170	152	175	195	160	175	195
90	120	130	92	125	138	93	126	132	89	121	132
125	100	70	120	100	75	126	101	73	123	99	72

7	8	9	7	8	9	7	9	8	7	8	9
2	4	6	2	4	6	2	5	6	2	4	6
5	3	1	4	3	1	4	3	1	5	3	1

Fig. 5. the OIS feature

Each keyframe is divide into several blocks ( $3 \times 3$  in Fig. 5), calculate the average intensity value of each block, and then sort the average intensity values of the blocks. Thus, we get a vector of 9 dimensions for each keyframe.

### Local Feature

For dealing with the difficult styles of transformed videos (picture-in-picture style), local feature is needed. SIFT feature [2] is used widely in recent years. We compare the some local features, and find that SIFT has the best performance. So we choose SIFT feature.

Although the SIFT feature is widely used, it also has some disadvantages. First, too many interest points are detected with the Standard SIFT (for a  $300 \times 300$  picture, there are about 800 interest points detected); second, the dimension of SIFT is too high.

Therefore, we use the DOG (Difference of Gaussian) detector of standard SIFT to detect the interest points, but only consider four octaves. So the number of interest points is only about 300. For each interest point, we calculate a  $12 \times 12$  local area. Therefore, we get a 72-dimension vector for each SIFT interest point.

### 3.1.3 Video Sequence Matching

We use a new method to make the video sequence matching: graph-based video sequence matching.

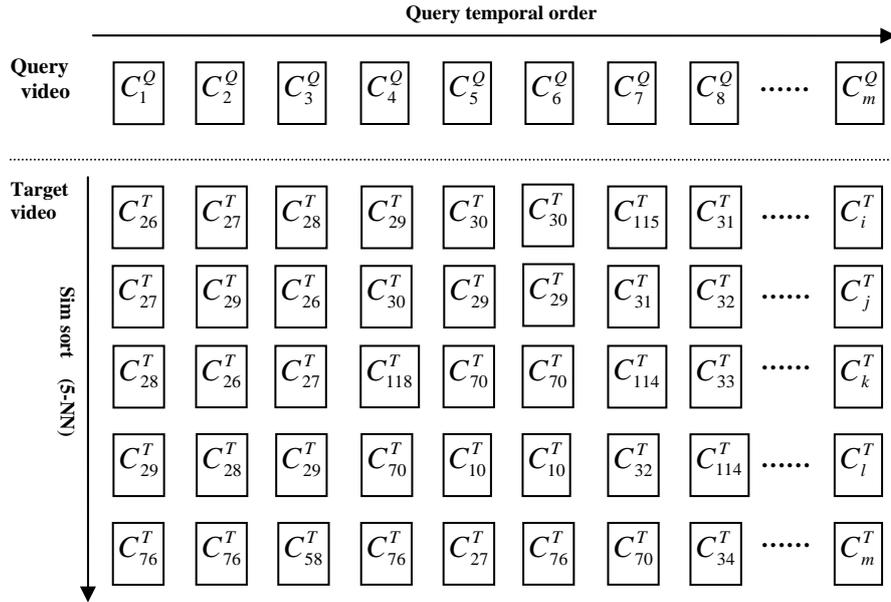


Fig. 6. Matching progress

For two feature sequences, query feature sequence  $C^Q = (C_1^Q, C_2^Q, C_3^Q, C_4^Q, \dots)$ , and target feature sequence  $C^T = (C_1^T, C_2^T, C_3^T, C_4^T, \dots)$ ,  $C_i^Q$  presents the feature extracted from the  $i$ th keyframe of the query video.  $C_k^T$  presents the feature extracted

from the  $k$ th key frame of the target video. First, for a query feature  $C_i^Q$ , we calculate the  $t$  most similar target frames. And sort them by their descending similarities. Take Fig. 6 for example, for  $C_1^Q$ , the 5 most similar target frames are  $C_{26}^T, C_{27}^T, C_{28}^T, C_{29}^T, C_{76}^T$ .

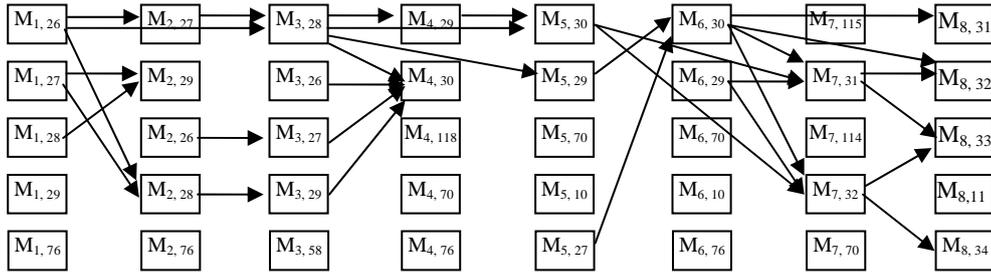


Fig. 7. Matching result

We get a digraph for this query and target frames as described in Fig. 7. Then we look for the longest path of the digraph. And the path should be satisfied the time constrain ( take Fig. 7 for example,  $M_{1,26}$  stand the first frame of the query and the 26th target video frame is a match, then the second frame of the query should match the 27th target frame or the frames behind it, we can get a path  $M_{1,26}, M_{2,27}, M_{3,28}, M_{4,29}, M_{1,26}, M_{5,30}, M_{7,31}, M_{8,32}$ ).

### 3.1.4 Evaluation

We submitted 2 runs to the TRECVID 2009 for evaluation. They are FudanVID.v.balanced.fudan09Lo1 and FudanVID.v.nofa.fudan09Lo2, comparing with our system last year. Firstly, for NDCR(The minimal normalized detection cost rate), we get better performance on some transform types (T2, T8,T10), on others transform types; however, the performance stays unsatisfactory, because for the difficult types like T2,T8, local method but global method is helpful. Secondly, as we used the local feature method and do not use indexing, the process time is too long. Thirdly, the copy location accurate is also far from satisfactory; it may be caused by the keyframe extraction method and the graph-based method.

The performance of FudanVID.v.balanced.fudan09Lo1, which is optimized by the ‘balanced’, is described in Fig. 8-10: Fig. 8 describes the NDCR; Fig. 9 describes the actual mean F1; and Fig. 10 describes the mean processing time.

TRECVID 2009: copy detection results (balanced application profile)

Run name: FudanVID.v.balanced.fudan09Lo1  
Run type: video-only

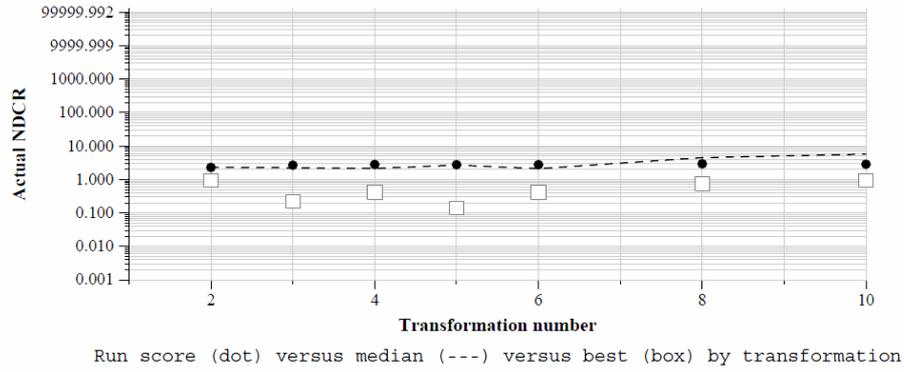


Fig. 8. Actual NDCR result

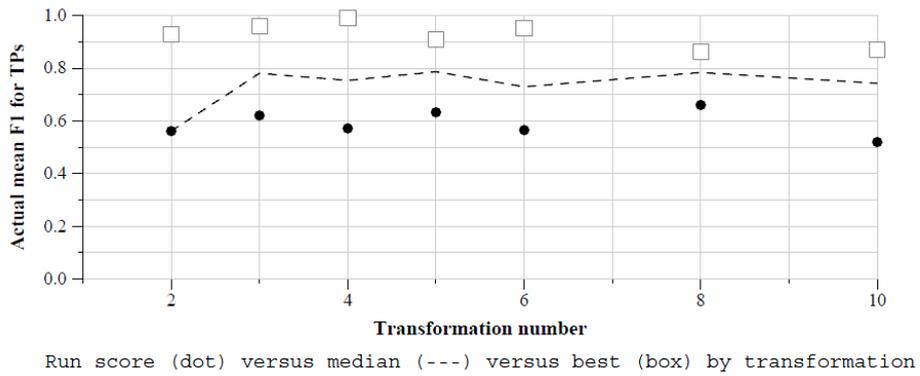


Fig. 9. Actual mean F1

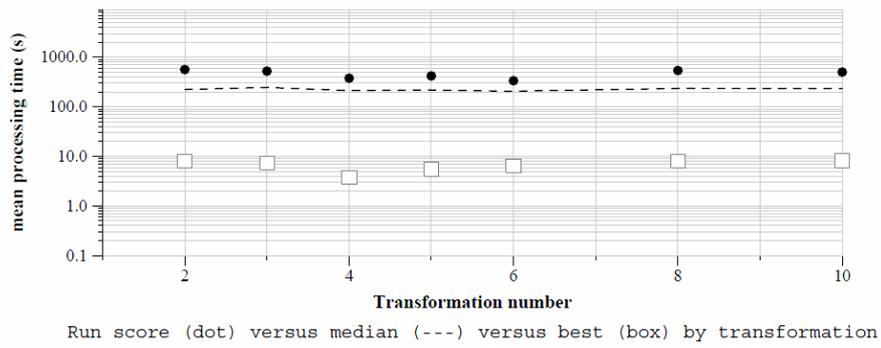


Fig. 10. Mean processing time

### 3.2 Summary

For content-based video copy detection, we submitted 2 runs; both runs filtrate some query videos with the global method, and then deal with the leaving query videos with local method. For the global method, we use the OIS (Ordinal Intensity Signature) feature, and for the local method, we use the SIFT method. After the features are extracted, the video sequences are matched with the graph-based matching method.

Since we use the local feature method, and do not adopt the index method, the processing time needs much more improvement. In addition, the graph-based matching method needs to be improved to make the copy location accuracy better.

**Acknowledgments.** This paper is supported by Natural Science Foundation of China (No. 60873178 and 60875003), National Science and Technology Pillar Program of China (No. 2007BAH09B03), and MSRA Young Faculty Innovation Fund.

### References

1. Smith, T.F., Waterman, M.S.: Color and Texture Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology* 11(6): 703--715 (2001)
2. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(2): 91--110 (2004).
3. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos"; In *International Conference on Computer Vision* (2003).
4. Xue, X., Zhang, W., Guo, Y., *et al.*: Fudan University at TRECVID 2008. In: *TRECVID Workshop* (2008)
5. Mikolajczyk, K., Schmid, C.: Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision* 60(1): 63--86 (2004)
6. van de Sande, K. E. A., Gevers, T., Snoek, C. G. M.: Evaluating Color Descriptors for Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (in press)
7. Liang, Y., Liu, X., Wang, Z., *et al.*: THU and ICRC at TRECVID 2008. In: *TRECVID Workshop* (2008)
8. Zheng, Y., Lu, H., Jin, C., Xue, X.: Incorporating Spatial Correlogram into Bag-of-Features Model for Scene Categorization. In: *Asian Conference on Computer Vision* (2009)
9. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
10. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International Journal of Computer Vision* (2006)
11. Hampapur, A., Hyun, K.H., Bolle, R.: Comparison of sequence matching techniques for video copy detection. In: *Proceedings of SPIE - The International Society for Optical Engineering*. vol. 4676, pp. 194--201 (2002)
12. Guil, N., Gonzalez-Linares, J. M., Czar, J. R., Zapata, E. L.: A clustering technique for video copy detection. In *IbPRIA* (1), volume 4477 of *Lecture Notes in Computer Science*, pages 451--458 (2007)