THU-IMG at TRECVID 2009

Yingyu Liang, Binbin Cao, Jianmin Li, Chenguang Zhu, Yongchao Zhang, Chenhao Tan,
Ge Chen, Chen Sun, Jinhui Yuan, Mingxing Xu, Bo Zhang
Intelligent multimedia Group,
State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Computer Science and Technology, Tsinghua University, Beijing, P. R. China

Abstract

Results of content based copy detection task

ID	Profile	Brief description
IMG.v.nofa.CN	NOFA	Video only, ***ScoreTimes***=1.5
IMG.v.balanced.CB	BALANCED	Video only, ***ScoreTimes***=1.3
IMG.m.nofa.NOFA	NOFA	Video + audio, ***ScoreTimes***=1.5, AND(video_result, audio_result)
IMG.m.balanced.BALANCED	BALANCED	Video + audio, ***ScoreTimes***=1.3, OR(video_result, audio_result)

Results of high level feature extract task

ID	MAP	Brief description
run1	0.134	Simple sum fusion for region based features
run2	0.118	Weighted sum fusion for region based features
run3	0.128	Simple sum fusion for keypoint based features
run4	0.100	Simple sum fusion for all features
run5	0.113	Weighted sum fusion for all features

1. Introduction

Intelligent multimedia group in Department of computer science and technology, Tsinghua University took part in TRECVID 2009 and submitted the results for content based copy detection and high level feature extraction. In this paper, the approaches for these two tasks are presented.

2. Content based copy detection

The copy detection system for video-only queries consists of two subsystems. The first one uses block based feature while the second one uses keypoint based feature to deal with those that cannot be processed easily in the first subsystem. As illustrated in Fig. 1, each subsystem consists of two parts: offline part and online part, dealing with the database and the query respectively. From the view of algorithmic procedure, each subsystem can be divided into three steps: feature extraction, feature indexing and determination.

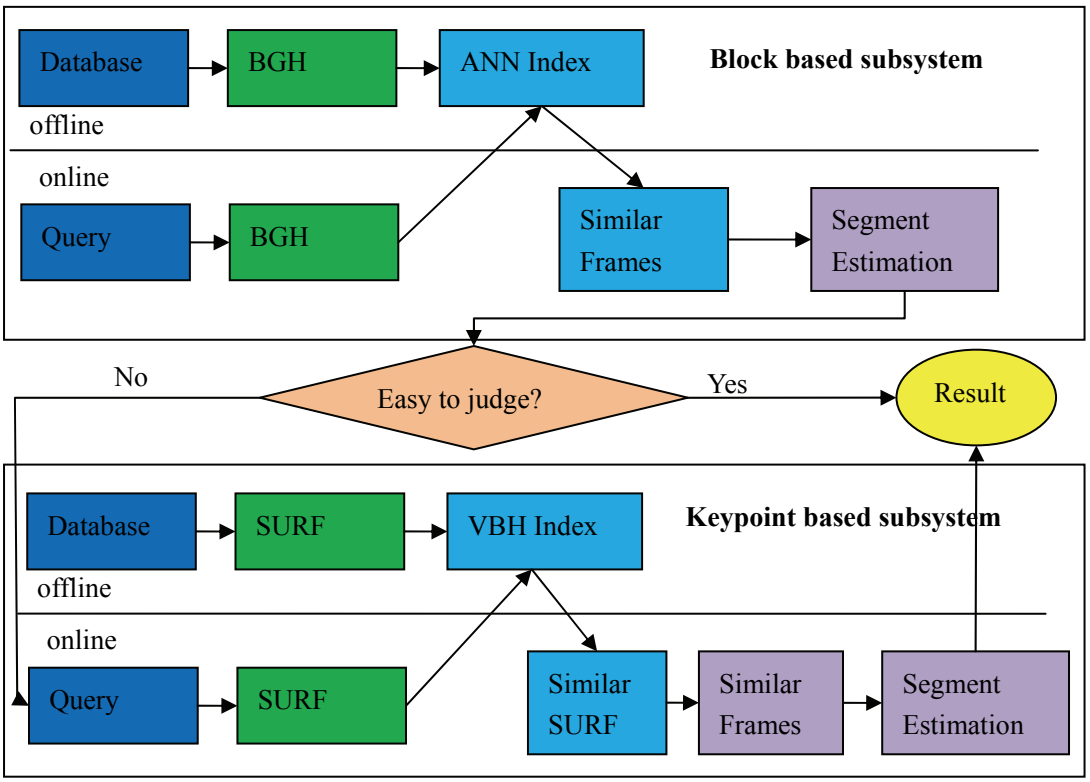


Fig 1. System overview for video-only copy detection

As illustrated in Fig. 2, the copy detection system for audio+video queries combines the video-only queries results and the audio-only queries results.

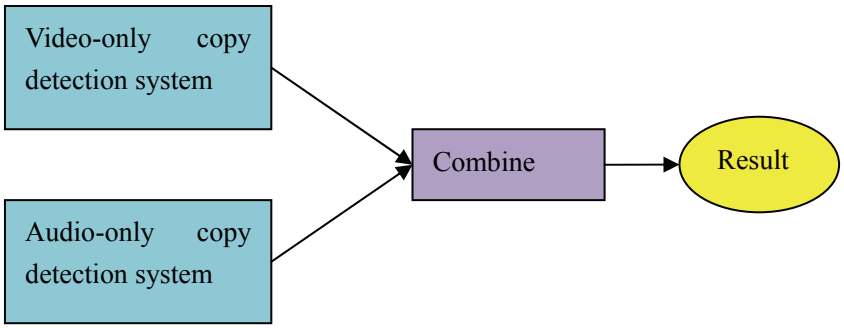


Fig 2. System overview for audio+video copy detection

2.1 Block based feature subsystem in video-only copy detection

2.1.1 Feature

In this part, three kinds of features are extracted from key frames which is selected every ***Nsample***=20 frames from the videos. First of all, each key frame is converted to a gray image. Then 3 features, Median [5], Ordinal Measure [6] and Block Gradient Histogram (BGH) [1] are tested. It is observed that among 3 features, the BGH feature is generally more effective than the others, so we adopt this feature in the TRECVID task finally.

Dealing with Picture-in-picture The Picture-in-picture transformation in the task is difficult for either block based features or keypoint based features. In order to deal with this special transformation, we extract BGH from small windows which locate in the top-left, top-right,

bottom-left, bottom-right and middle position and are of 30%, 40% and 50% size of the original image. Along with BGH for the whole image, 16 BGH features for different windows are extracted. For each of these features, we use it through the following indexing and determination step and get a list of reference videos and corresponding scores for a query video. The list with the maximum score reference video is selected as the final result list.

2.1.2 Index

Once the BGH features are extracted, ANN [2] index is built for all the BGH features extracted from videos in the database. The feature extracted from each key frame of a query video is used as query in the index, and the retrieved neighbors are treated as similar frames.

2.1.3 Determination

The possible copy segment between the query and a reference video in the database is determined in this step by utilizing the similar frames from the reference videos. An adjusted version of the time sequence consistency method [1] is adopted to locate the matched segments between the query and the reference video. Like the original method, our version finds the time consistent similar frame path with maximum weight. However, in the original method, the weight of the path is defined as the sum of all the weights of the similar frames in the consistent path, which benefits the paths with many low-weighted frames. In our version, the weight is defined as the sum of the logarithms of all the weights of the similar frames in the consistent path, which favors the paths with a session of significant-weighted frames.

As mentioned above, we extract BGH, build index and determine the possible copy segment for each of the 16 windows, and then select the result list with the maximum-scored reference video as the final result list. If in this list, the score of the first-ranked reference video is greater than ***ScoreTimes*** times the score of the second-ranked reference video, the list is output as the result of the video-only copy detection system; otherwise, the query is further processed by the keypoint based feature subsystem.

2.2 Keypoint based feature subsystem in video-only copy detection

2.2.1 Feature

In this part, Speeded Up Robust Feature (SURF) [3] is extracted from keyframes which are selected every ***Nsample***=20 frames from the videos.

2.2.2 Index

Once the features are extracted, Vocabulary-Based Hashing (BGH) [4] index is built for the features from videos in the database. VBH index combines the popular Bag-Of-Features (BOF) and Locality Sensitive Hashing (LSH) [7] approach. Each BOF vocabulary defines a hashing function, which maps the input feature point into the ID of the nearest visual word in the vocabulary. Different vocabularies define different hashing functions, and form a hashing function family. The hashing function family is incorporated into the LSH approach and results in the VBH index. Details can be found in [4]. For this task, we use vocabularies of size 4000×2, and the LSH approach parameters $k = 8$, $L = 1$.

Features extracted from each key frame in a query video are used as queries in the index. Once the similar features for those in the key frame are obtained, a voting is performed and the key frames in the database are ranked according to their scores. The images with the top ***Nsimilar***=3 maximum scores are treated as similar frames for the query key frame.

2.2.3 Determination

As in global feature subsystem, the possible copy segment between the query and reference

videos in the database is determined by utilizing the similar frames from the reference videos. The same approach (the adjusted version of the time sequence consistency method) is employed.

2.3 Audio-only copy detection system

Our Content Based Copy Detection solution consists of four main parts: feature extraction, threshold generation, calculate distance and match and adjust results.

Step 1. We use an open source tool named HTK to extract a 26-dimensional mfcc (Mel-frequency cepstral coefficients) feature to represent each frame.

Step 2. According to the TRECVID Data document file of 2008 named "o.gt", for each query wav we align it with the matched test wav, then we calculate the average frame Euclidean distance of the two copied episodes if existed. Because there may be some inaccuracy when align the two wave file, we move the aligned position forward and backward 10 frames, then we take the min distance. So for every query wav-file if it has a core in another test wav-file, we'll have a distance, among all the distances, we multiply the biggest one with a coefficient, that's the Threshold we'll use later.

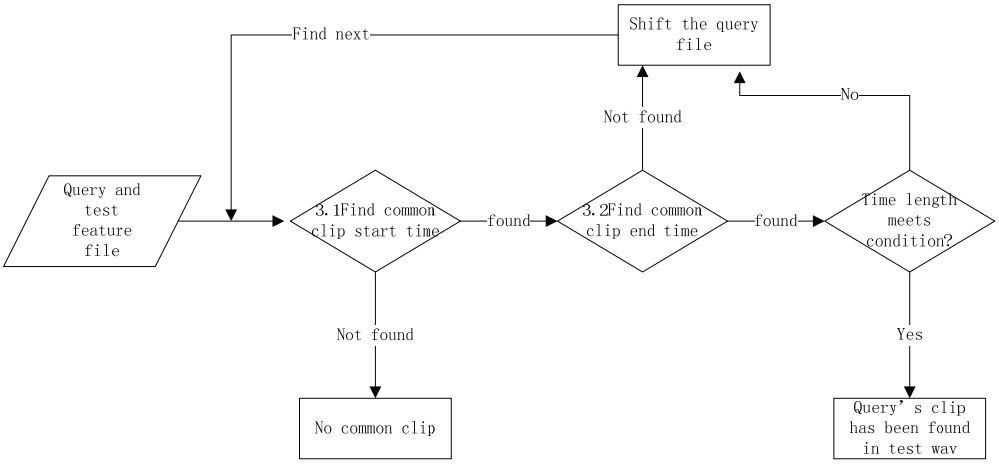


Fig 3. Step 3, calculate distance and match

Step 3 is the most important and time consuming. The kernel part of this step is to find the start-time and the end-time.

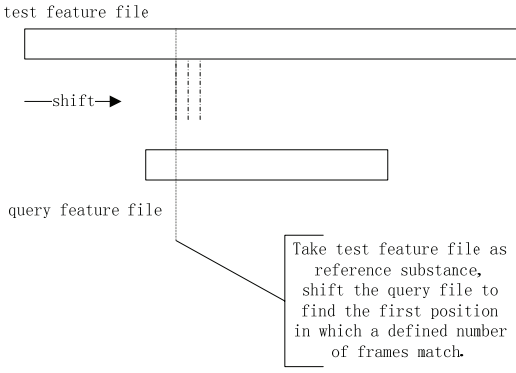
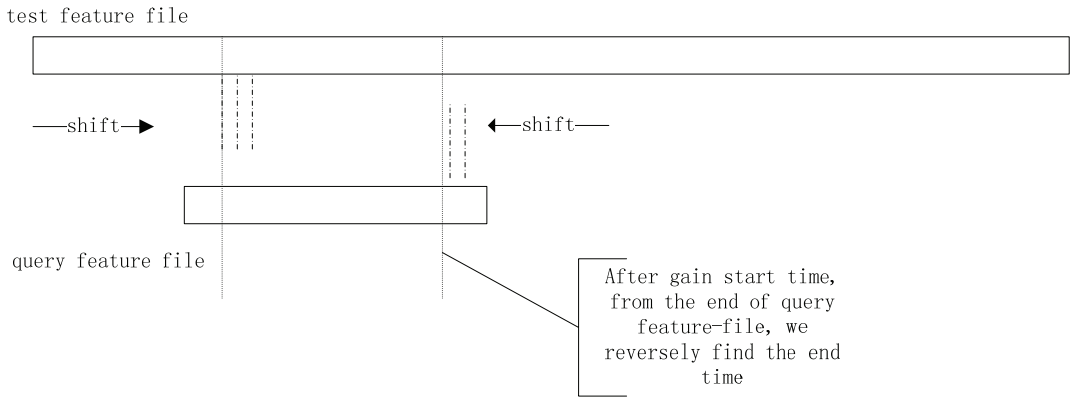


Fig 4. Step 3.1, find start time

To find the start time the two wav-files match (step 3.1), we take the test-wav feature file as a

reference substance, then align the start of query and test feature file. We use two pointers which are initialized as zero to point to the position to start processing. Then we use a pre-defined number of frames to calculate the average distance which will be compared with the threshold formed in step 2. If it's less than the threshold, we take the two pointers as the start time of the clips in test wav and query wav that match each other, and then we come to the state to find the end time. If the distance is bigger than the threshold, because for a query wav we don't know whether the original query is one whole query or a clip inserted into another clip, we shift a certain number frames (because we assume the common clip time length has a lower limit) and continue this calculating distance and compare until to find a position that meets the condition. If the either pointer comes to the end of file, then we have a conclusion that the corresponding two wav-files have no clip in common. If we gain the positions, then we shift forward each time one frame to locate the exact positions that first meet the condition. Suppose we have found the start time in the two wav-files.



3.2Find End Time

Fig 5. Step 3.2, find end time

The way we get the end time (step 3.2) is similar with the way we gain the start time. The only difference is that the direction is reverse. After we have the end time, we calculate the time length, if it's less than the lower limit (we defined it according to the 2008 data), then we are back to the step 3.1, if not, the time information is valid.

Step 4. Since we know that there're 201 groups of queries which means there should be 201 groups of results, so we refine our results to 201 groups. First we sort the 1407 files by files length. Then we use the file length to regroup, the files with a similar length will be assigned to one group (use this method one group may contain 7 or 14, 21 or more files). If the file number in one group is bigger than 7, we use a clustering method by which the Euclidean distance is least to generate a sub-group of which the audio file number is 7. After that we'll use the longest clip result to replace the others' results.

2.4 Audio+video copy detection system

We design two combining strategies, one for the no-false-alarm profile and the other for the balanced profile. For the no-false-alarm profile, a reference video is treated as the result of audio+video copy detection if it appears in the result list of the audio-only copy detection **AND** the result list of the video-only copy detection. For the balanced profile, a reference video is treated as the result of audio+video copy detection if it appears in the result list of the audio-only copy detection **OR** the result list of the video-only copy detection.

2.5 Experiments

We submit 4 runs for the task: the no-false-alarm profile and the balanced profile for video-only copy detection; the no-false-alarm profile and the balanced profile for audio+video copy detection.

In video-only copy detection, ***ScoreTimes***=1.5 for the no-false-alarm profile and ***ScoreTimes***=1.3 for the balanced profile.

In audio+video copy detection, for the no-false-alarm profile ***ScoreTimes***=1.5 and **AND** strategy is used to combine results from video-only and audio only copy detection; for the balanced profile ***ScoreTimes***=1.3 and **OR** strategy is used to combine results from video-only and audio only copy detection.

Due time limitation, we only do some experiments on a small training dataset selected from TRECVID2008 video copy detection task. The parameters are set manually, according to observation of mean F1 in the training experiments. The more proper adjustment of parameters is for future works.

2.5.1 Video-only copy detection

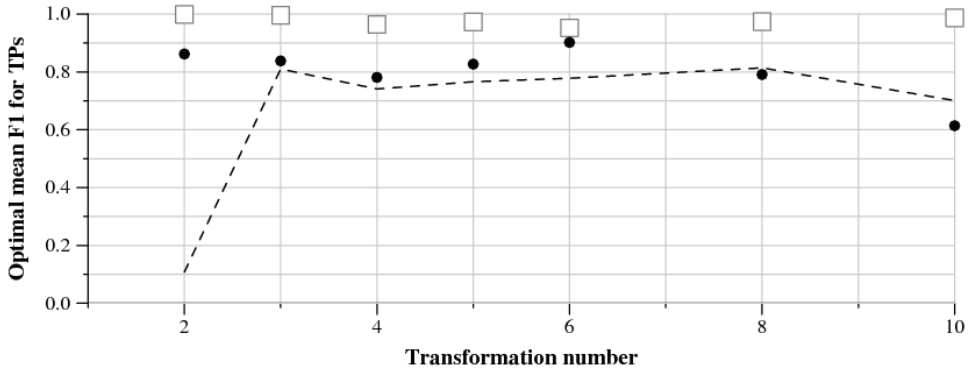


Fig 6. Performance of no-false-alarm profile for video only copy detection

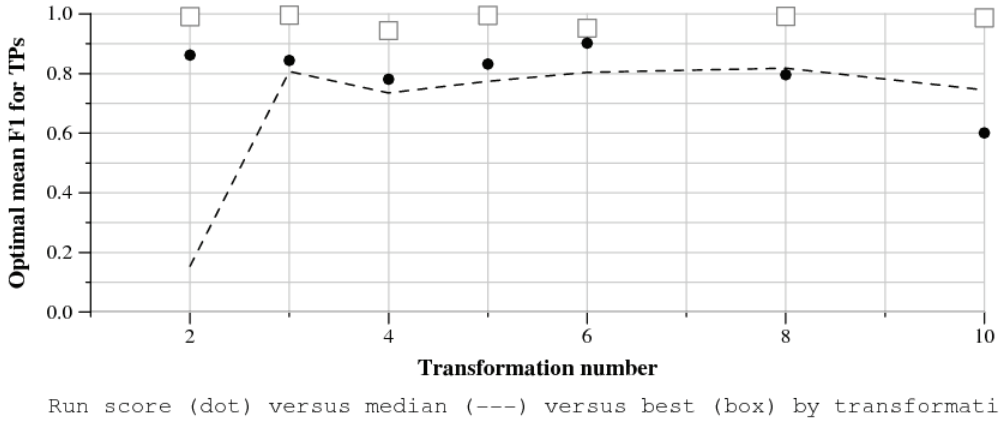


Fig 7. Performance of balanced profile for video only copy detection

As observed in Fig.6 and Fig. 7, the performance of our method is better than average. Especially for transformation 2 (Picture-in-picture), we achieve much better results than average, which suggests that our sub-window approach in video-only copy detection is effective. Very similar results are obtained for both profiles, so we aim to design more specific approaches for the two different profiles.

2.5.2 Audio+video copy detection

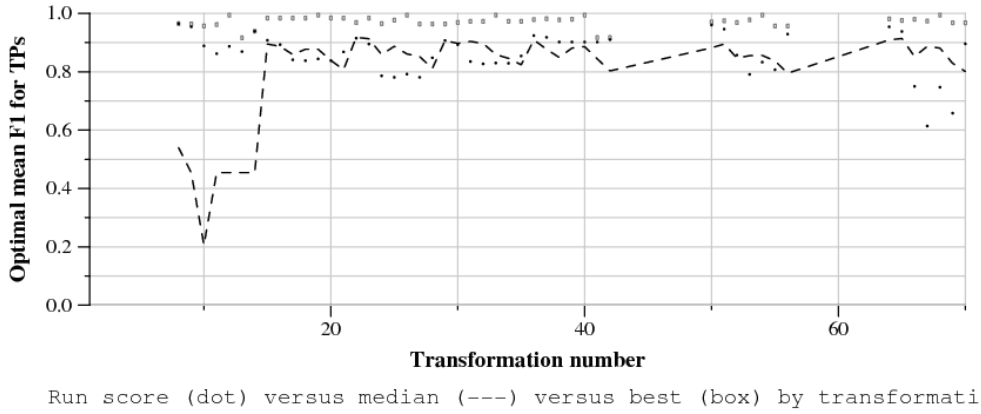


Fig 8. Performance of no-false-alarm profile for video+audio copy detection

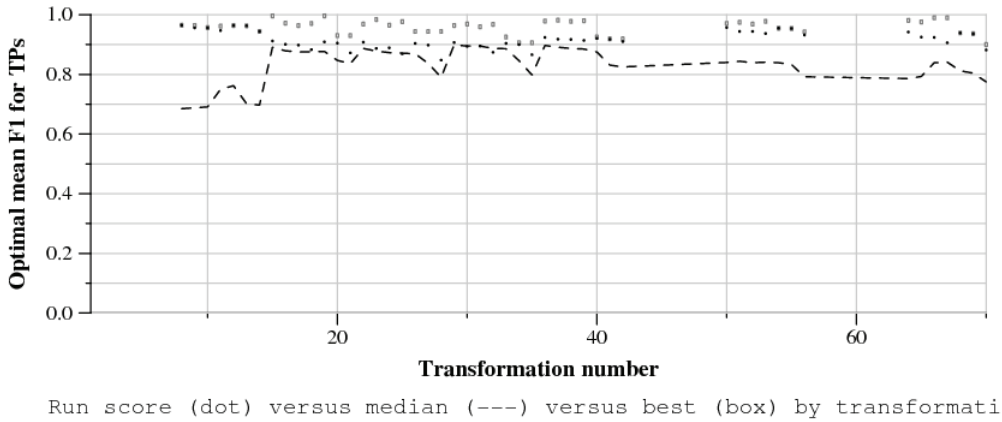


Fig 9. Performance of balanced profile for video+audio copy detection

As observed in Fig. 8 and Fig. 9, our results are generally better than average. Compared with video-only results, the audio+video results are slightly better, which shows that the audio information helps to improve the performance but the benefit is limited.

3. High Level Feature Extraction

This year, our video indexing system follows the same framework as our Video Driver system since TRECVID 2006. Large codebooks are adopted with the same feature extraction algorithm implementation in TRECVID 2007. 36 types of features are involved to represent the visual content of the videos. Confronting the large scale of data (40k images, 10k for training set, 30k for testing set) and intensive computations, parallelization during low level feature extraction phase, training phase, testing phase are achieved in task level on multi-core system.

3.1 Annotation and keyframe extraction

Annotation provided by LIG and LIF [11] is used in the learning phase. Several keyframes are extracted to represent the visual content for each shot. Average number of keyframes per shot is set to 3. Within a shot, keyframes are extracted by temporal equally sampling. Number of

keyframes in each shot is proportional to the log of the corresponding duration. During the training phase, each training image is tagged by the annotation of the corresponding shot. During the testing phase, each shot is scored by the sum of all images within it.

3.2 Feature representation

Low level feature representation is the major factor in high level feature extraction task. To classify objects, human, scenes and activities, various robust and discriminative feature representation methods are required. This year, 36 types of features are adopted during our experiments. To compare the performance of each different feature, experiments with the same configurations of this year has also been done on TRECVID 2007 dataset. Evaluation of the results is described in the result section.

To categorizing the feature representation methods involved this year, two aspects are considered. One is the content of the feature for representing the visual content, e.g. color, texture, edge, the other is the layout of the feature representing the sample methods and region schemes while extracting.

3.2.1 Feature content

Types of feature content can be generalized to 4 categories, color, gabor, gradient and edge.

3.2.2 Sampling

Local feature descriptor representations have been proved to be more discriminative due to additional spatial information. To make better feature representations, common feature extraction methods are combined with local patch sampling over images. Half-overlapped 20x20 pixel grid patches are sampled in our experiment. Thus, over 200 patches are sampled from each image according to the TRECVID dataset.

3.2.3 Region schemes

Using region in images is another method to add spatial information to common features. It is simple for implementation and efficient for computation. It is helpful when videos are captured from the similar viewpoint. Several region schemes involved in our experiments are shown in Fig 10.

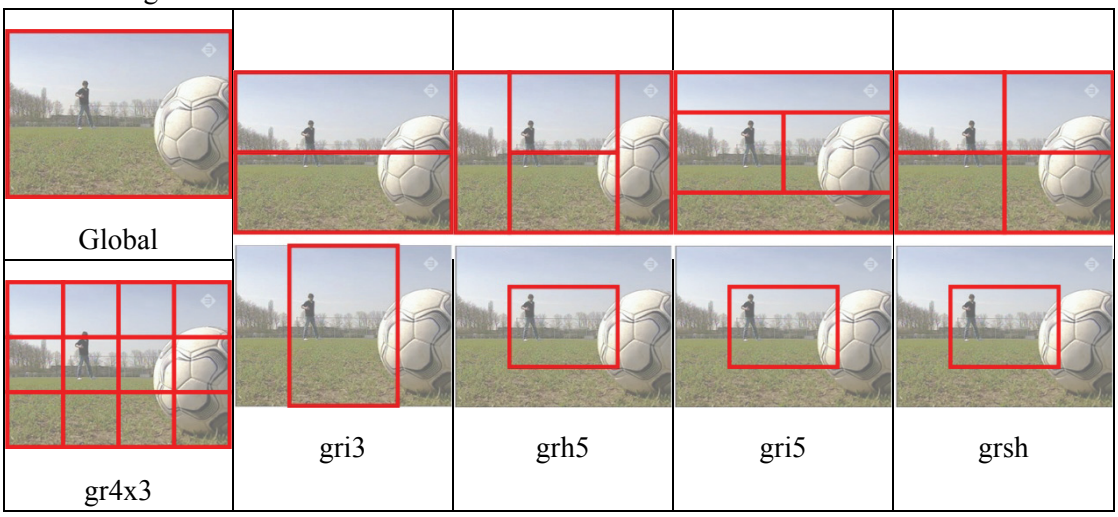


Fig 10. Feature representation region schemes

3.2.4 Learning and fusion

We follow the Boost method [12, 13] for training this year. 5 SVMs are trained for each high level feature and low level feature pair with χ^2 kernel for keypoint histogram features and RBF kernel for common features. Parameters of SVMs are determined in a coarse-to-fine grid

search manner by cross validation.

Output by all SVM models is fused by simply sum for each image in late fusion manner. Output for each shot is fused by sum of all output for images within it.

3.2.5 Results

To compare effectiveness of various types of features, experiments with same configurations have been done on TRECVID 2007 dataset and TRECVID 2009 dataset. The 07 (09) develop data is used for training while 07 (09) test data is used for testing. Table 1 shows the mean average precision (MAP) of all the 36 features in descending order of TRECVID 09.

Table 1 MAP of 36 low level features on TV07 and TV09

Feature	Layout	Description	MAP@07	MAP@09
ceh64	grsh	Canny Edge Histogram	0.0566	0.0322
ceh32	grh5	Canny Edge Histogram	0.0525	0.0313
ceh64	patch20r20	Canny Edge Histogram	0.0678	0.0284
eccv32	gri5	Edge CCV	0.0455	0.0278
ec64	grh5	Edge Correlagram	0.0465	0.0255
ctn48	patch20r20	Grey Level Co-occurrence Matrix (energy entropy contrast homogeneity)	0.0539	0.0225
cm9	gr4x3	Color Moment	0.0422	0.0211
ctn48	gri5	Grey Level Co-occurrence Matrix (energy entropy contrast homogeneity)	0.0389	0.0207
scntx270	g	Shape Context	0.0447	0.0205
sift	g	SIFT	0.0557	0.0191
hm10	gr4x3	Haar Wavelet Moment	0.0394	0.0182
ccv72	gri3	Color Coherence Vector	0.0396	0.0166
wt20	grh5	Wavelet Texture	0.0332	0.0165
mluvb36t1	gri3	Moment Band on LUV space	0.0240	0.0151
fc27	gri3	Fuzzy Color	0.0134	0.0148
ctmhsv48	gri3	Color Texture Moment Feature on HSV space	0.0365	0.0147
glcm48	gri3	Grey Level Co-occurrence Matrix	0.0266	0.0145
ccv72	patch20r20	Color Coherence Vector	0.0449	0.0140
bac36t1	gri3	Band Auto Correlagram	0.0275	0.0137
ac166t1	g	Auto Correlagram	0.0115	0.0135
hsv36	patch20r20	Color histogram on HSV space	0.0462	0.0134
gabor48	patch20r20	Gabor filter	0.0408	0.0132
cm9	gri3	Color Moment	0.0338	0.0130
gabor48	grh5	Gabor filter	0.0367	0.0130
mhb60	g	Moment Haar Band	0.0291	0.0124
surf	g	SURF	0.0005	0.0121
hsv36	gri3	Color histogram on HSV space	0.0346	0.0117

hm10	gri3	Haar Wavelet Moment	0.0220	0.0117
spin	patch20r20	Spin	0.0426	0.0115
cc	patch20r20	Cross Correlation	0.0438	0.0115
mff18	gri3	Multi-feature Feature	0.0132	0.0110
hsv166	g	Color Histogram on HSV space	0.0071	0.0105
ac64t1	g	Auto Correlagram	0.0077	0.0098
bc36t1	gri3	Band Correlagram	0.0310	0.0056
chc314	g	Color Histogram Haar Correlogram	0.0258	0.0039
wt20	patch20r20	Wavelet Texture	0.0398	0.0003

We submitted 5 runs totally this year. All of the runs are trained and tested with official training set, testing set and annotations. Different fusion methods are evaluated by different runs. The parameters for weighted sum at feature level come from the MAP evaluated on TRECVID 2007 dataset.

Table 2. MAP and description of each run

ID	MAP	Brief description
run1	0.134	Simple sum fusion for region based features
run2	0.118	Weighted sum fusion for region based features
run3	0.128	Simple sum fusion for keypoint based features
run4	0.100	Simple sum fusion for all features
run5	0.113	Weighted sum fusion for all features

Acknowledgement

This work is supported by the National Natural Science Foundation of China under the grant No. 60621062 and 60605003, the National Key Foundation R&D Projects under the grant No. 2007CB311003, and the Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).

Reference

[1] Yongdong Zhang, Ke Gao, etc. TRECVID 2008 Content-Based Copy Detection By MCG-ICT-CAS. In Proceedings of TRECVID 2008 workshop.

[2] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. J. ACM, 45:891–923, 1998.

[3] H. Bay, T. Tuytelaars, L. Gool. Surf: Speeded up robust features. ECCV, May 2006.

[4] Y. Liang, J. Li, B. Zhang. Vocabulary-based hashing for image search. ACM MM, 2009.

[5] B. Thomee, M. J. Huiskes, E. Bakker and M. S. Lew. Large Scale Image Copy Detection Evaluation.

[6] X. S. Hua, X. Chen and H. J. Zhang. Robust Video Signature Based on Ordinal Measure.

[7] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In SCG, 2004.

[10] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in

videos. In ICCV, 2003.
[11] TRECVID 2009 Collaborative annotation. http://mrim.imag.fr/tvca2009/
[12] D. Wang, X. Liu, L. Luo, J. Li, B. Zhang. Video Diver: Generic Video Indexing with Diverse Features. MIR workshop at ACM Multimedia, 2007.
[13] Jie Cao, et al. Intelligent Multimedia Group of Tsinghua University at TRECVID 2006. In Proceedings of TRECVID 2006 workshop