BUPT at TRECVID 2008 *

Zan Gao, Zhicheng Zhao, Tao Liu, Xiaoming Nan, Mei Mei, Bin Zhang,
Xiaodan Liu, Xu Peng, Hui Zheng, Yanyun Zhao, Anni Cai

Multimedia Communication and Pattern Recognition Labs,
School of Telecommunication Engineering, Beijing University of Posts
and Telecommunications, Beijing 100876, China

Abstract

High-level feature extraction

We describe our system for the task of HLF extraction in TRECVID 2008. Features at different granularities are extracted to describe visual content of keyframes, and four classifiers for each concept are trained by SVM, and different fusion strategies are used to combine final results. The brief introduction to each run is shown in the Table 1.1.

Table 1.1. MAP and description of HLF

HLF Run	infMAP	Description
BUPT_Sys1	0.047	The classifiers are trained by annotation-III, and maximum fusion scheme is used
BUPT_Sys2	0.043	The classifiers are trained by annotation-III, and average fusion scheme is used
BUPT_Sys3	0.039	The classifiers are trained by annotation-II, and maximum fusion scheme is used
BUPT_Sys4	0.037	The classifiers are trained by annotation-II, and average fusion scheme is used
BUPT_Sys5	0.037	The classifiers are trained by annotation-I, and maximum fusion scheme is used
BUPT_Sys6	0.034	The classifiers are trained by annotation-I, and average fusion scheme is used

The best results of our system are slightly better than the average of participant groups, but it does not work as well as our previous testing.

Copy detection

Content-based copy detection (CBCD) aims at retrieving a test collection in all the transformations. The visual features we used are robust to intensity offset and color distortion between video clips and test collection to be searched. On the other hand, different search strategies are proposed for long query video clips and short ones. This can reach a tradeoff between accuracy and speed. Evaluation results showed that our CBCD algorithm can achieve a high precision and mean F1, but lots of miss detections happen.

This year, Multimedia communication and pattern recognition labs in School of telecommunication engineering, Beijing University of Posts and Telecommunication (BUPT) takes part in three tasks of TRECVID 2008, they are high-level features extraction, copy detection and event detection.

1. High-level feature extraction

1.1 Different annotations

* This work was supported by China National Natural Science Foundation under Project 60772114.

Three kinds of annotations based on TRECVID-2007 development dataset and test dataset are used this year, and the differences of each annotation are as follows:

- Annotation-I: It is supplied by MCG-ICT-CAS [1].
- Annotation-II: In order to test the difference of different annotations, we re-annotate the Trecvid-2007 development corpus. 20 concepts are divided into two groups according to the concept description of the Trecvid-2008. The first one is region-related concepts, and the rectangles are used to locate the local objects, such as dog, bus, telephone. This group includes 13 concepts: 002 Bridge, 003 Emergency_Vehicle, 004 Dog, 006 Airplane_flying, 007 Two people, 008 Bus, 009 Driver, 010 Cityscape, 012 Telephone, 013 Street, 015 Hand, 018 Boat_Ship, 019 Flower.

  The second one is global scene-related concepts, and the whole frame is used to describe the concepts, such as classroom, kitchen. This group includes: 001 Classroom, 005 Kitchen, 011 Harbor, 014 Demonstration_Or_Protest, 016 Mountain, 017 Nighttime, 020 Singing.
- Annotation-III: We re-annotate the Trecvid-2007 development corpus by IBM MPEG-7 Annotation tool [2], and the whole frame is regarded as the concept object.

1.2 Features extraction

It is difficult to develop a single approach which is not only invariant to various disturbances but also sensitive enough to capture the details of content. Integrating some complementary features to describe the content is probably a promising way. Therefore, features at different granularities are extracted: for scene-related concepts, global features are selected, and for region-related concepts, some local features are available. At the same time, in order to increase the stability of features, we consider the feature of group of frames. In addition, different color spaces have different characteristics, so we extract features from different color spaces, such as YUV, RGB and HSV.

- SIFT

SIFT feature achieved good performance in video analysis [3, 4, 5, 6, 7], so it is taken into account this year. At first, we build a visual vocabulary of SIFT points detected from keyframes based on Difference of Gaussian (DoG) [8], and then more than 50 positive samples from Trecvid-2007 development corpus are chosen for each concept. Secondly, by using of K-means cluster algorithm, about 270,000 SIFT points are clustered into 1000 classes, and each class represents a visual keyword. Thirdly, for every test keyframe, SIFT points are extracted, and then every point is assigned to the nearest class. Finally, the number of occurrences of each visual word is recorded as a histogram.

- Gabor Wavelet

Gabor wavelet usually is used at 5 different scales $\nu \in (0,1,....,4)$ and 8 orientations $\mu \in (0,1,.....7)$. In order to speed up the runtime, we only extract 3 different scales $\nu \in (0,2,4)$ and 6 orientations $\mu \in (0,1,...5)$. For each frame, we divide it into 3*3 blocks, and then the mean and standard deviation of the blocks are calculated to represent the block. Finally, each keyrame is represented by a 324-dimension feature vector.

- Edge Orientation Histogram [14]

The edge histogram descriptor represents the spatial distribution of five types of edges (0^o, 45^o, 90^o, 135^o, non-direction). Since edges play an important role for image perception, it can retrieve images with similar semantic meaning, especially for natural images with non-uniform edge distribution.

- Color Feature

Several color detectors which recommended by MPEG-7 are extracted, and some have been proved effective in testing.

(1) RGB Color Moment (9 dim)
(2) HSV Color Auto-Correlogram (512 dim)
(3) HSV Color Histogram (256 dim)
(4) HSV Group of Frame (256 dim)
(5) RGB Histogram of Block (576 dim)
(6) Average Brightness (1 dim)

Features are complementary, so before training, a linear feature fusion scheme is first used to normalize and combine them into a feature vector.

1.3 The framework of HLF

The framework plays a very important role in the high-level feature extraction. It decides how to choose the classifiers and how many classifiers need to be trained. According to the state of the art, the general classifier [10, 11] could not obtain good performances, but hundreds of classifiers such as [1, 2, 6] would be time-consuming and complicated. In order to balance the performance, complicated and runtime, we propose our HLF framework which is shown in Fig.1.1.

Due to the good performance SVM achieved in the past few years [3, 4, 5, 6, 7], we adopt it to train our classifiers. For each concept, four different SVM classifiers based on different feature granularities are trained and total 80 classifiers are obtained. LibSVM tool [9] with RBF kernel is used. During the course of testing, the keyframes of each shot is first extracted, and then several fusion strategies (Vote, maximum probability, average probability) are performed to generate the final results.

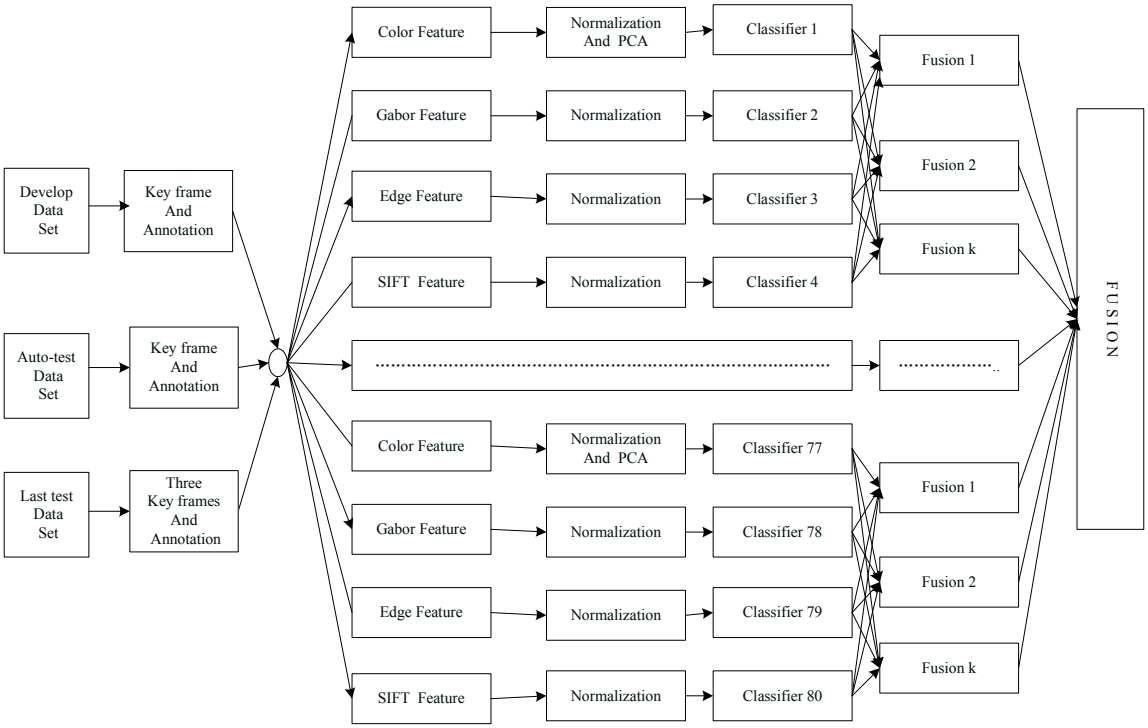

Fig. 1.1. The framework of HLF 2008.

1.4 Fusion strategy

The fusion of classifiers can improve the performance of HLF. There are a number of fusion ways and can be roughly divided into two types: non-heuristic methods and heuristic methods. Non-heuristic methods do not need

fusion training, such as Max, Vote, Average and so on, and the later need fusion training and Adaboost, Linear weighted are usually adopted. In the experiments, we train four basic classifiers for every concept respectively, they are SIFT Classifier, Gabor Classifier, Edge Classifier and Color Classifier, and several fusion schemes are defined:

$$prob_{\max}(i) = \max\{prob_{sift}(i), prob_{gabor}(i), prob_{edge}(i), prob_{color}(i)\} \tag{1}$$

$$prob_{aver}(i) = \{prob_{sift}(i) + prob_{gabor}(i) + prob_{edge}(i) + prob_{color}(i)\}/4 \tag{2}$$

$$prob_{vote}(i) = \begin{cases} \max\{prob_{sift}(i), prob_{gabor}(i), prob_{edge}(i), prob_{color}(i)\} & if \ \ n >= 2 \\ \min\{prob_{sift}(i), prob_{gabor}(i), prob_{edge}(i), prob_{color}(i)\} & else \end{cases} \tag{3}$$

$$prob_{apweight}(i) = \begin{matrix} \{prob_{sift}(i) * apw_{sift}(i) + prob_{gabor}(i) * apw_{gabor}(i) \\ + prob_{edge}(i) * apw_{edge}(i) + prob_{color}(i) * apw_{color}(i)\} \end{matrix} \Big/ \begin{matrix} \{apw_{sift}(i) + apw_{gabor}(i) \\ + apw_{edge}(i) + apw_{color}(i)\} \end{matrix} \tag{4}$$

Where i is the concept index $i \in (1, 2, ...20)$ $prob_{sift}(i)$, $prob_{gabor}(i)$, $prob_{edge}(i)$, $prob_{color}(i)$ are the probability of four classifiers of concept i. For the Vote scheme, we consider the sample as a positive sample if more than two classifier probabilities are larger than 0.5, otherwise, negative sample. The number of positive sample is n. if $n \geqslant 2$, then we choose the maximum fusion scheme. For linear weighted scheme, we use the precision and average precision as the weight, the Eq.(3) and (4) show above two situations respectively.

1.5 Parallel computing

The HLF task is highly compute intensive, thus we use parallel computing techniques to accelerate the SVM classifier training.

1.6 Experiments and discussion

In the Trecvid-2008, as every team would submit no more than 6 runs, so 6 runs of our results are randomly selected and submitted for evaluation. The performance and description are shown in Table 1.1.

From the Table 1.1, we can see that among our 6 runs, the first run achieve the best MAP (0.047) and this result is slightly better than the average of all participant groups. Fig.1.2 shows the detection results of 20 concepts.

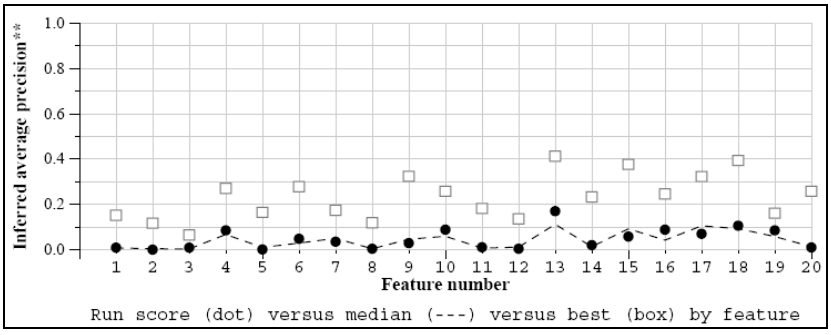

Fig. 1.2. HLF results of BUPT_Sys1.

We also find that the performance of classifiers trained by annotation-III is better than annotation-I and annotation-II, and annotated by whole frame are also better than that annotated by rectangle regions.

1.7 Conclusions

From the HLFexperiments in TRECVID 2008, we found that:

1. Features selection is crucial.
2. Annotation also affects the final results.
3. An enhanced SVM training algorithm needs to improve the performance of classifier.
4. More classifiers would perform better results.
5. Large-scale parallel computing plays an important role in HLF extraction.

2. Copy detection

2.1 System overview

As shown in Fig.2.1, the CBCD system. The system consists of five parts: the 1st one is an initial process which includes the decoding and features extraction. The 2nd one is video segmentation module, which divided the query into a series of sub-segments with similar contents. The 3rd part is keyframes and new features extraction. Searching and matching modules are the fourth and fifth parts of our system. The modules will be described in next sections in details.

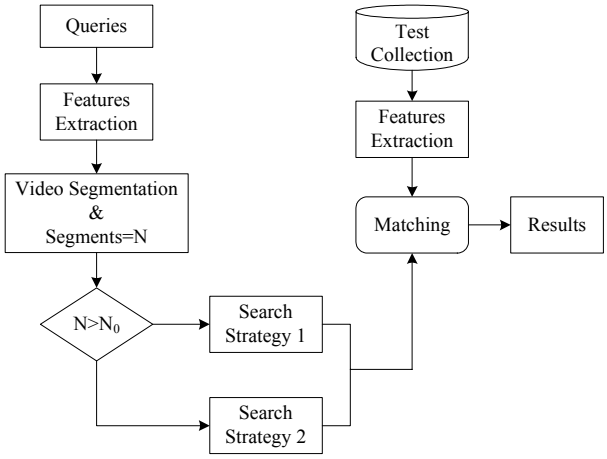

Fig. 2.1. The flowchart of CBCD system 2008.

2.2 Features extraction

The visual features we used in this paper are global intensity histogram (IGH), intensity ordinal measurements (IOM) [13], enhanced local color histogram (ELCH), local edge histogram descriptor (LEHD) [14], Canny edge, SIFT and Gabor color moment [15]. Among them, IGH, IOM and ELCH are used to segment query clip into sub-segments, and the last four features for searching.

2.3 Video segmentation

For each feature, feature difference of neighboring frames is calculated and then subtraction is implemented among adjacent frame differences. It is defined as the difference of frame difference. It avoids to segment in the positions of gradual change frames and single frame with sharp changes, which is helpful for sequent matching module. The difference of frame difference is compared with corresponding threshold of each feature to decide whether to segment. If there are two features satisfying partition condition among three features, this segmenting position is recorded. Proven through our experiments, this method can mark most cut frame positions. But when segmenting condition of Intensity Ordinal Measurements is satisfied and the rest two are not satisfied, this position is considered non-ideal segmenting. In the process of video segmenting, we can obtain two kinds of

non-visual assistant features, including length of each segment and marks at non-ideal segmenting position. After this, the query video was divided into N segments.

2.4 Search strategy

Two different search strategies are proposed for long query video clips and short ones respectively. Here a short query clip is defined as the one with the number of segments, $N<N_0$ threshold. For long query video clips, we will start searching at least from the second segment rather than the first one of the query video clip since the beginning and ending of a query clip often lack a couple of frames compared with its corresponding part in test collection. However, this strategy is not suitable to short query video clips because short video clips have few segments.

2.4.1 Search for Long Query Video Clips

The search for long query clips consists of five stages: feature analysis of the query video clip, search stage 1, search stage 2, frame-by-frame search at non-ideal segment positions, and search for the starting and ending frames. Among them, search stage 1 and frame-by-frame search at non-ideal segment positions are performed in parallel. Fig.2.2 shows the flow chart of the whole process. Search stage 1 is to estimate possible corresponding positions of a query video clip in dataset, and Search stage 2 is setup to eliminate the influences caused by editing effects and frame rate variations.

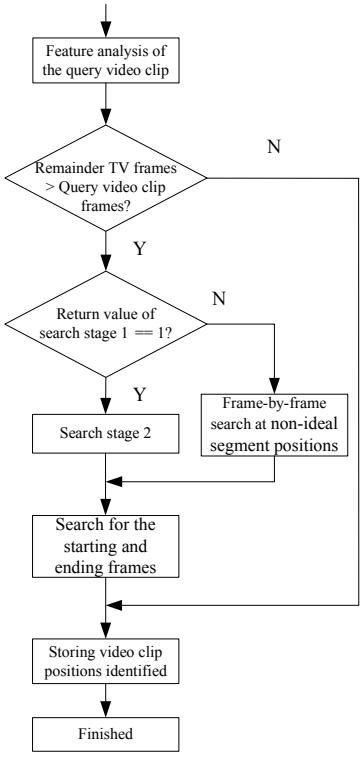

Fig. 2.2. Search flow chart for long query clips.

2.4.2 Search for Short Query Video Clips

The search for short query video clips consists of four stages: locate possible positions, search the range of still frames, search the optimal position and locate starting and ending frames. Fig.2.3 shows the flow chart of the search for short query video clips.

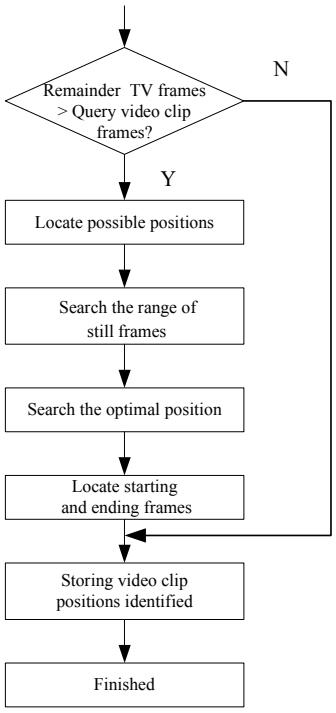

Fig. 2.3. Search flow chart for short query video clips.

2.5 Experiment and Evaluation

We submitted one run for video-only this year. The detailed performance of is show in Fig. 2.4. and Fig. 2.5. From the evaluation, we can see that our CBCD system can achieve a high precision and mean F1 scores, but at the same time, much more miss detections appear in some video such as "video-in-video" and great transformations.

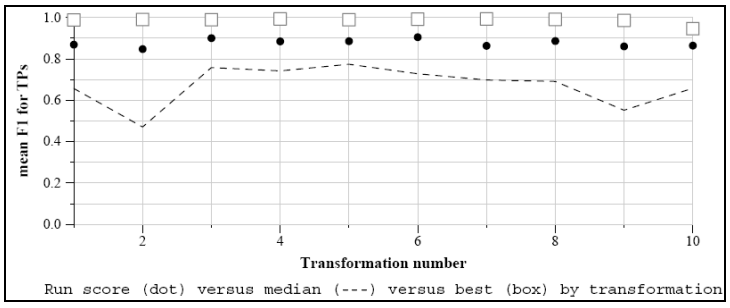

Fig. 2.4. Copy detection results: mean F1.

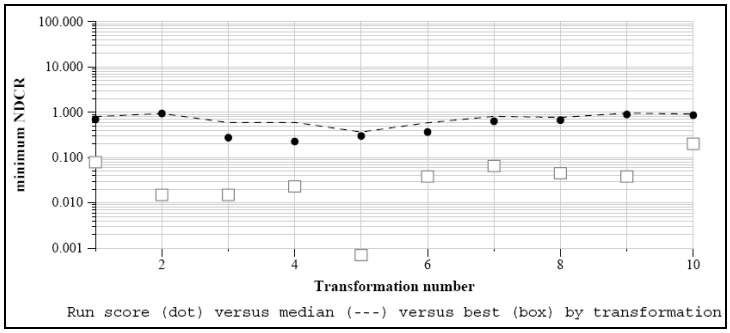

Fig. 2.5. Copy detection results: minimum NDCR.

2.6 Conclusions

From the CBCD experiments in TRECVID 2008, we found that:

1. Features selection is vital and the tradeoff between invariance and sensitivity of features should be taken into account to achieve a good performance.
2. The result of video segmentation is also very important to affect initial search.
3. Indexing document should be reconstructed to enhance the speed of retrieval.

References

[1] MCG-ICT-CAS, "Annotation of TRECVID 2008 Development Key frames"

[2] IBM VideoAnnEx MPEG-7 Video Annotation Tool, http://www.research.ibm.com/VideoAnnEx/

[3] J. Cao et al. Tsinghua University at TRECVID 2006. In: Proceedings of TRECVID 2006 Workshop.

[4] Jinhui Yuan, Zhishan Guo, Li Lv et al. THU and ICRC at TRECVID 2007. In: Proceedings of TRECVID 2007 Workshop.

[5] Shih-Fu Chang, Wei Jiang, Akira Yanagawa et al. Columbia University TRECVID2007 High-Level Feature Extraction. In: Proceedings of TRECVID 2007 Workshop.

[6] C.G.M. Snoek, I. Everts, J.C. van Gemert et al. The MediaMill TRECVID 2007 Semantic Video Search Engine. In: Proceedings of TRECVID 2007 Workshop.

[7] Yu-Gang Jiang, Chong-Wah Ngo, Jun Yang. Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval. Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR 2007, 2007, p 494-501.

[8] D. Lowe. Distinctive image features from scale-invariant key points. Int. Journal on Computer Vision, 60(2):91-110, 2004.

[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] Naphade M R, Smit hJR,et al. IBM Research TRECVID-2005, Video Retrieval System. TRECVID, 2005

[11] Chang SF, Hsu W, et al. Columbia University TRECVID-2005 Video Search and High-level Feature Extraction. TRECVID, 2005

[12] Emine Yilmaz, Javed A. Aslam, Estimating average precision with incomplete and imperfect judgments, Proceedings of the Fifteenth ACM International Conference on Information and Knowledge Management (CIKM), November, 2006.

[13] D. N.Bhat, S.K.Nayar. Ordinal measures for image correspondence, IEEE Trans. on Pattern analy. and machine Intell.,vol.20, no.4, pp: 415-423, 1998.

[14] C.S. Won, D.K. Park, S.J. Park. Efficient Use of MPEG-7 Edge Histogram Descriptor. ETRI Journal, vol.24, no.1, Feb. pp:23-30, 2002.

[15] H. Yu, M.J Li, H.J. Zhang. Color Texture Moments for Content-Based Image Retrieval. ICME'03, 2003.