Fudan University at TRECVID 2005

Xue Xiangyang, Lu Hong, Wu Lide, Guo Yuefei, Xu Yuan, Mi Congjie, Zhang Jing
Liu Shenggui, Yao Dan, Li Bin, Zhang Shile, Yu hui, Zhang Wei, Wang Bei
Department of Computer Science and Engineering, Fudan University, Shanghai, China

Abstract

In this notebook paper we describe our participation in the NIST TRECVID 2005 evaluation. We took part in all four tasks of the benchmark including shot boundary determination, low-level feature extraction (camera motion), high-level feature extraction and search (manual). We describe the different runs submitted for each task and analyze their performance.

For shot boundary determination, we submitted 10 runs:

sh01~sh05: different thresholds, post-processing in HSV color space

sl01~sl05: different thresholds, post-processing in Lab color space

Evaluation result shows that detection performance differs not much between these ten runs. Our system does not have good performance. The main reason is that we define short gradual boundary as cut, which usually lasts less than five frames and do not have remarkably gradual transition in vision, and such definition is different from the one adopted in TRECVID, which regards the transition that may just have one transition frame as gradual boundary.

For low-level feature extraction, we submitted 3 runs:

D_DT14: 14-d feature, rule-based decision tree, motion accumulation

D_SVM14: 14-d feature, support vector machine (SVM), motion accumulation

D_SVM18: 18-d feature, SVM, motion accumulation

From evaluation, it seems 18-d feature is better than 14-d feature and SVM performs better than decision tree. And due to fixed threshold, our motion accumulation method sometimes failed to detect human perspective camera motion and needs more improvement.

For high-level feature extraction, we submitted 7 runs:

B_D_PCA_BC_1: multi-model, principle component analysis (PCA), Borda count

B_D_PCA_LR_2: multi-model, PCA, logistic regression

B_D_LPP_BC_3: multi-model, supervised LPP（locality preserving projection）, Borda count

B_D_LPP_LR_4: multi-model, supervised LPP, logistic regression

A_D_MC_5: clustering, multi-model, Borda count

B_D_SPE2_6: specific methods for specific concepts

A_D_ASR_7: ASR-based

We find that PCA and supervised LPP almost have the same performance in dimension reduction. And supervised fusion method's performance relies on its training process. Region-based method works better for object concepts.

For search, we submitted 7 manual runs:

M_A_2_D_MM_BC_1: multi-model, relation expression, MC.

M_A_2_D_MM_2_BC_2: multi-model, twice fusion.

M_A_2_D_AOH_LR_ONLINE_3: textual feature, visual concepts, linear fusion, pseudo relevance feedback, logistic regression, online.
M_A_2_D_MM_LR_OFFLINE_4: multi-model, linear fusion, logistic regression, offline.
M_A_2_D_AO_5: textual retrieval.
M_A_2_D_AOH_LR_OFFLINE_6: textual feature, visual concepts, linear fusion, logistic regression, offline.
M_A_1_D_A_7: baseline
Evaluation results illustrate that the relation expression fusion method is better than the linear fusion methods and training weights online is superior to training weights offline. We also find that the fusion and rank method is very important for multi-model video retrieval.

1. Introduction

Content-based video retrieval is an interesting but challenging work. It draws more and more attentions to develop effective techniques for analysis, indexing, and searching of video database. TRECVID provide a standard dataset and evaluation criterion for comparing different algorithms and systems. This year, we took part in all four of TRECVID tasks---shot boundary determination, low-level feature extraction, high-level feature extraction and search (manual). We submitted the maximum number of runs for each task except for low-level feature extraction.

2. Shot Boundary Determination

Our shot boundary detection is performed directly on compressed MPEG video stream. Many useful features including temporal frame structure, number of inter-coded macro-blocks and residue energy are extracted without fully decompressing video stream. Based on these features, SVM method [1] is performed to detect both cut and gradual shot boundaries.

2.1. Features from MPEG Stream

MPEG video stream comprises a frame sequence which is hierarchically organized by I, P, and B frames. This frame structure has good temporal scale and can improve detection efficiency, thus our system is based on this hierarchical frame structure.

Different frame has different coding modes and can provide different information for shot boundary detection. For I frame, it is intra coded and has no prediction information. Feature extracted from I frame is DC image which keep most content information while the size is only 1/64 of the original picture. Interval between I frames is large, larger than 10 frames generally. Thus if there is a shot boundary between two I frames, either cut or gradual, picture content will change prominently. As shown in Figure.1, in a gradual transition, difference between the two I frames is large, just like a cut. We use a window on I frame sequence to capture the content change characteristic. Distances between every two I frames in the window form a feature vector which could be sent to SVM classifier as training or testing sample. Feature vectors are calculated in YUV color space, so for each color dimension there is one feature vector.

Figure.1 I frame difference in gradual transition

For P frames, features can be extracted are: number of intra coded macro-blocks and residue energy after prediction. When there is shot transition over a sequence of P frames, the number of intra coded macro-blocks and residue energy will become larger than those of sequence inside a shot. We use sliding window to form P frame feature vector which will be used for both cut and gradual boundary detection. So there are totally four feature vectors which will be sent to SVM classifier, i.e. vectors of number of intra coded macro-block, Y residue energy, U residue energy, and V residue energy.

For B frames, three features are extracted: number of forward prediction macro-block, number of backward prediction macro-blocks and number of bidirectional prediction macro-blocks. When there is cut between two B frames or a B frame and a P/I frame, macro-block prediction mode in B frame will change. Distribution of number of three types of prediction macro-block will change prominently. A sliding window is added on B frame sequence to form feature vector to represent this characteristic. For each extracted features, there will be one feature vector accordingly. We concatenate these three vectors into one feature vector which will be used as sample for SVM classifier.

2.2. Framework of SBD

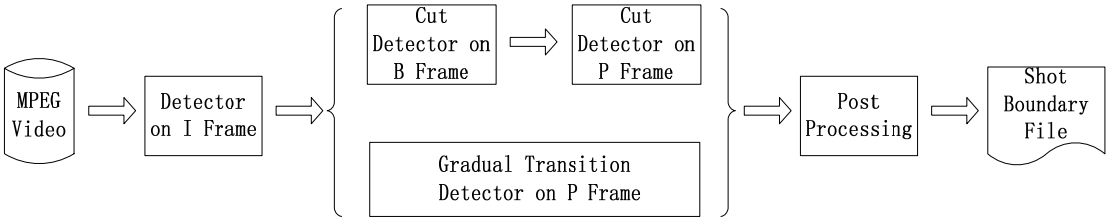


Figure.2 Overview of shot boundary detection framework

Figure.2 shows the system flowchart of SBD. Our real time SBD system is performed directly on MPEG video stream. Extracting feature and shot boundary detection are performed at the same time. Since interval between two I frames is large (always more than 10 frames), potential boundary position is first detected on I frames. The recall on I frame will be very high, and precision will be lower. Then, cut and gradual shot boundary detection are performed respectively. As for gradual shot boundary, visual difference between two successive frames is not noticeable, so gradual boundary detector only performs on P frames. As for cut boundary, they are first detected on B frames, then on P frames. In the end, post processing module is performed to eliminate false detection brought by flashlight and intensive motion. Post processing module also discriminates the ambiguous boundaries which are detected by both cut and gradual boundary detector.

Each detector adopts SVM classifier to judge whether there is a boundary. This is a two-class

classification question. What should be highlighted is that for detector on I frame or cut/gradual detector on P frame, feature vectors received by them are more than one. So for each feature vector there should be one SVM classifier, i.e. three SVM classifiers on I frame and four on P frame. Confidence output of these classifiers will form a new vector again and will be sent to a final SVM classifier to decide whether a shot boundary exists.

2.3. Experiment and Evaluation

We annotate 40 segments of development videos for system development. All SVM training samples are collected from development videos. Ten runs are submitted. Their differences are mainly in the post-processing module. Run sh01~sh05 post process in HSV color space and run sl01~sl05 post process in LAB color space. In each color space five thresholds are set. Evaluation result shows that detection performance differs not much between these ten runs.

The evaluation shows our performance is not very high. Figure.3 compares our performance with the best and median of all submitted runs in two aspects, recall and precision.

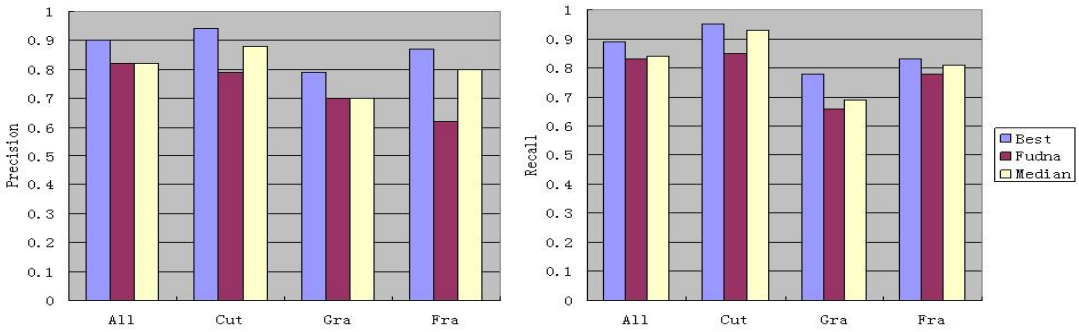


Figure.3 Fudan shot boundary detection performance

There are two reasons degrade our performance:

1. Some encoding modes of test videos are different from those of development videos. The SVM classifier needs more samples to be trained.
2. There are large amount of gradual transitions whose duration is less than five frames. In our system, we regard these transitions as cut, because such boundaries do not cause gradual transition visually. We believe that we should define cut or gradual boundaries from our visual perception. But this is different from the shot definition adopted in TRECVID, which regards the transition that may just have one frame as gradual boundary.

Advantage of our system is that it can run on compressed domain and can achieve high detection speed. Also many kinds of gradual transition can be detected by our system. The main disadvantage lies in that the detection performance depends too much on the training sample of SVM classifier.

3. Low-level Feature Extraction (Camera Motion)

We use a feature-based camera motion detection approach, pictured in Figure.4. The approach utilizes the motion vectors in P-frames from compressed video stream (e.g., MPEG stream). The median filter is

used first to smooth the raw motion vector field to remove some unreliable motion vectors. Then, a motion feature vector is constructed to characterize the statistical motion information. The feature vectors are input to a classifier for camera motion classification. Finally, a motion accumulation method is used to ensemble the camera motion of consecutive P-frames to detect human perceptive camera motions in an individual shot.

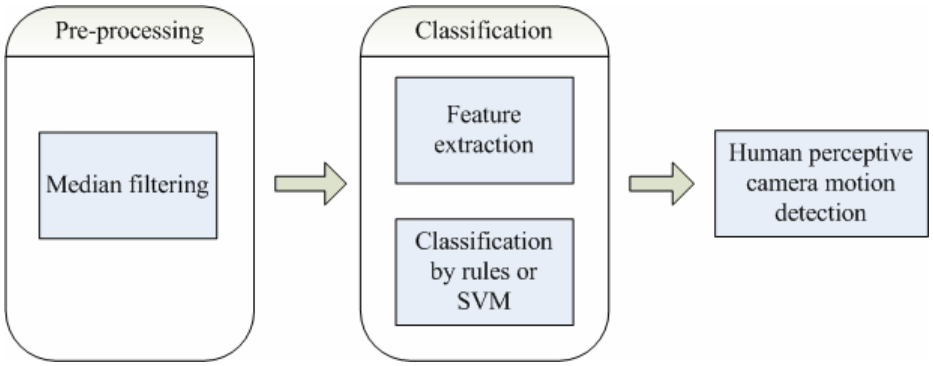


Figure.4 Flow chart of camera motion detector

For run D_DT14, the camera motion classification algorithm described in [2] is applied. It extracts a 14-d feature vector by exploiting the mutual relationship between motion vectors to characterize the motion vector field, and a rule-based decision tree (DT) method is used for classification. The mutual relationship between motion vectors are defined to be approach, diverging, parallel, and rotation by setting threshold.

For run D_SVM14, the same 14-d feature vector was extracted. However, SVM, instead of rule-based DT, is adopted for classification.

For run D_SVM18, we extract an 18-d feature vector by describing the mutual relationship between motion vectors in a different way. We redefine the mutual relationship between motion vectors by separating the rage equally instead of setting threshold manually. And SVM is used for classification again.

There are many small camera movements which are hard for human to see. For example, handheld camera results in a minor camera movement in many directions. However these small movements can be found by computer based on each P-frame. In the applications, human perceptive instead of computer recognizable camera motions are needed. In order to detect human perceptive camera movements, we propose a motion accumulation method. After camera motion classification, P-frames are classified into 8 categories: still, pan left, pan right, tilt up, tilt down, zoom in, zoom out, rotation. Scan the P-frame sequence, if the number of continuous same type (e.g. pan left) P-frames is larger than T_{time}, accumulate the continuous P-frames' mean magnitude of valid motion vectors and get E. If E is larger than T_{space}, a pan left camera motion is detected. T_{time} and T_{space} represent the human perceptive camera motion's temporal and spatial characters respectively and are set manually in our realization.

Experimental results show that the 18-d feature produces a better performance than the 14-d feature and SVM can classify camera motion more accurately and efficiently for each P-frame than rule-based decision tree. Figure.5 compares our best camera motion detection performance with the best and median performance of all 63 submitted runs in two aspects, precision and recall. By analyzing the

evaluation result, we believe that our motion accumulation method produces a low precision but higher recall. So how to detect a human perspective camera motion from video shots based on the classified P-frames deserves a further research work.

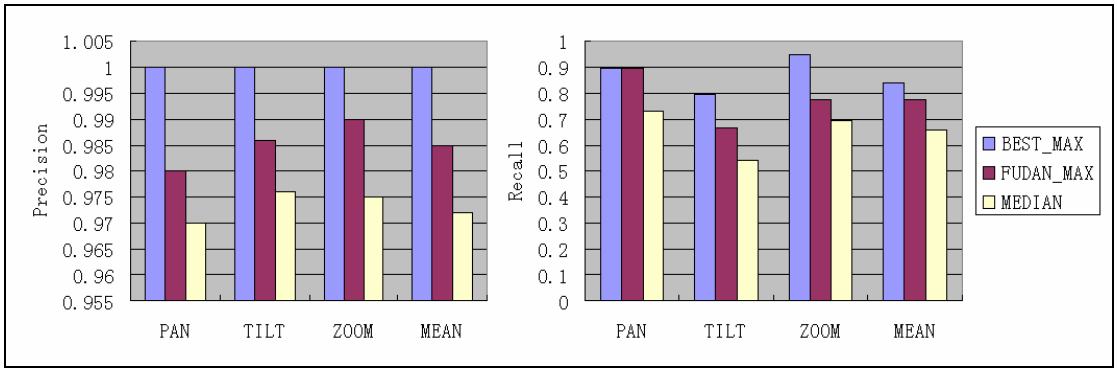


Figure.5 Fudan camera motion detection performance

4. High-level Feature Extraction

4.1. ASR-based Method

Text information retrieval approach is applied on ASR to detect specific concepts. For one specific concept, we consider that all words have two different properties: helpful to find the right shot or helpless. We have to use the training data to find out these properties. And then with these properties we make a judgment whether a shot with some words contains the specific concept. The training and testing procedures are described as follows.

For one shot on one concept, if the shot is positive, the words (after word stemming) related to this shot are added into a positive word list. If the shot is negative, the words related to this shot are added into a negative word list. Every word has a counter to record how many times this word totally appears. After the statistical work of all training videos, we normalize the counters of each word, obtaining the weight of each word. For some word that appears not only in the positive word list but also in the negative word list, we take the value in the positive word list as its weight, and make some modification on the word's weight. Assume set $X = \{x_1, x_2,...,x_n\}$, x_i appears in positive and negative word lists, with its counters c_{i1}, c_{i2} respectively, and the weight of x_i now is w_i. We compute the $S = \Sigma c_{i2}$, and the final weight of x_i is w_i - c_{i2} / S. Having the weights of all words appear in the training data, given a piece of words, we can compute its score. If setting a threshold, we divide the results into two types: positive and negative.

However, during our experiment, we find that there is great inconsistent between the concept words and concept appearance. So the performance of this method is lower than others.

4.2. Visual Information based Methods

4.2.1. Multi-Model

Since a shot is the basic unit, we use key-frame and sub key-frame in each shot as the representative images. Five different low-level visual features are extracted globally from each keyframe and

sub-keyframe. They are color layout descriptor (CLD), scalable color descriptor (SCD), Lab color histogram (LAB), edge histogram descriptor (EHD), and Gabor texture feature (GAB). The extraction of CLD, SCD and EHD is described in MPEG-7 [3]. For LAB and GAB, we split the image into a 5 by 5 grid in order to reflect spatial information in some degree. Then for each grid, Lab color histogram and Gabor texture feature are calculated. The features from grids are put together to form a high dimension feature vector. LAB feature is 4800-d and GAB feature is 1200-d.

Though each semantic concept has large diversity, we find they are still visually similar in some degree. So we try to classify the 10 high-level semantic features by using a multi-model classification strategy. The pipeline is described in Figure.6. Due to the high dimension of LAB and GAB feature, we employ PCA [4] and supervised locality preserving projection (LPP) [5][6][7] to reduce the dimension separately. SVM is adopted to build classifier for each low-level feature since it has achieved excellent empirical success. Then classifier combination methods are used to fuse the multiple classification results. We try two kinds of combination strategies: Borda count and logistic regression [8].

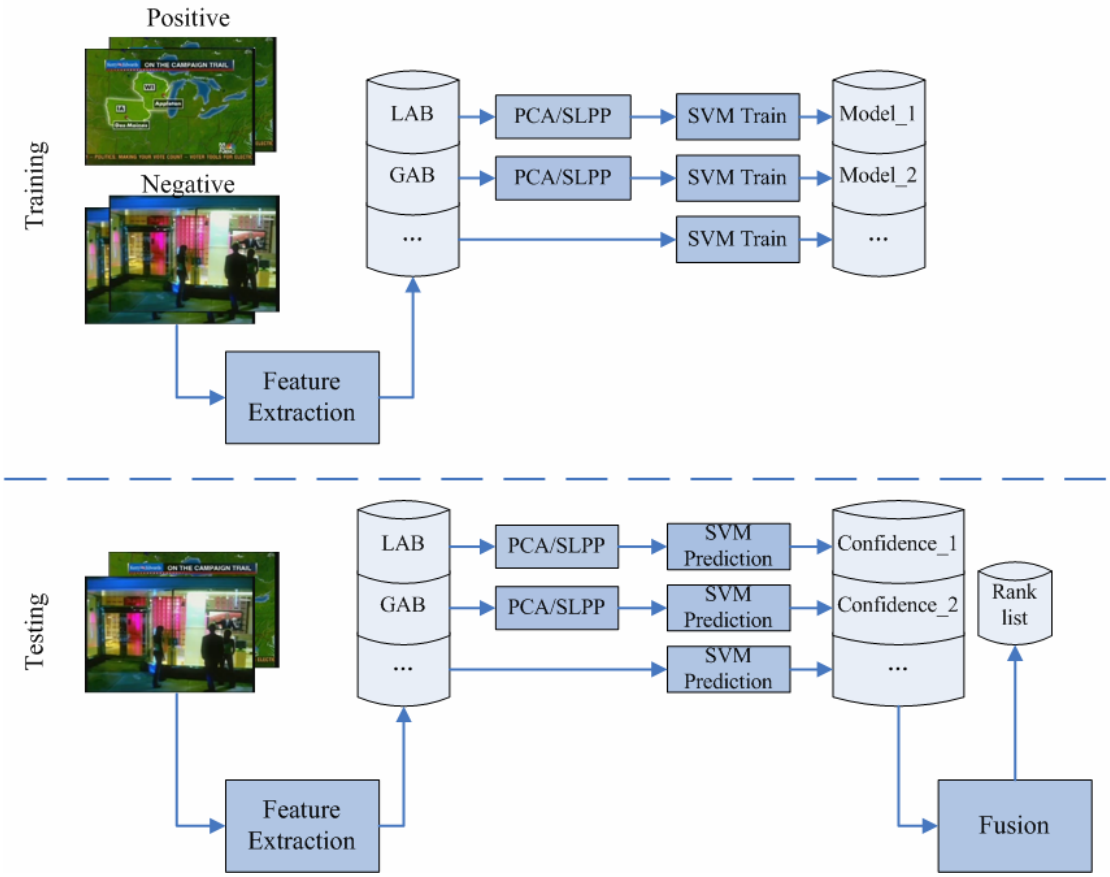


Figure.6 Multi-model concept detection pipeline

4.2.2. Multi-Cluster Multi-Model

For most of the concepts, we can find out that the low level features of the positive samples vary a lot. However, we can summarize them into several types, of which of each the samples have similar low level features. So we divide the positive samples into several types using clustering method. For each

cluster, we train a two-class SVM. Every classifier has a result and we make an “or” operation on these results. The flow chart is shown in Figure.7.

That is done on one feature. Different features get different results. So there still needs a fusion. The fusion methods we use here is Borda count due to our empirical study that it has a better performance than logistic regression in our experiment.

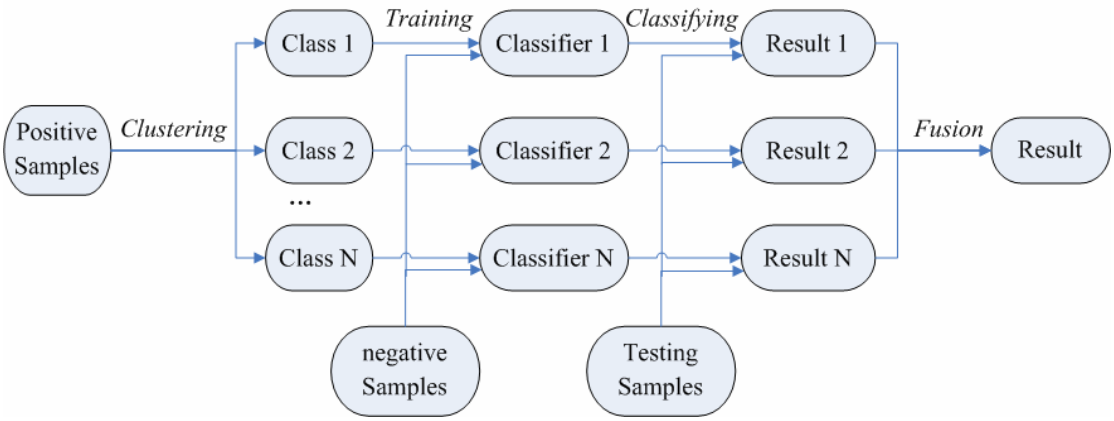


Figure.7 The flow chart of multi-cluster multi-model method

4.2.3. Specific Method for People walking/running

We analyzed the relationship of this feature to human bodies, motion vectors, and some closely related scenes like football, basketball, etc. First we did body detection, and then dealt with the positive results and negative results, respectively. To positive results, we did motion analysis and filtered those images which are still, or contain little motion, or contain large global motion as more as possible. To negative results, we built simple football, basketball and tennis ball model to extract those sports image in which the human bodies are too small to be detected. Finally, we combine the two parts to generate the final results.

4.2.4. Specific Method for Mountain and Water

We focus on the region distribution of natural scenes. The main idea of this special method is to recognize the regions containing those high level features, such as mountain, water, sky, and analyze the rationality of their layout, namely, it is region-based. There are two phases, offline phase and online phase. Figure.8 is the flowchart which demonstrates the main steps.

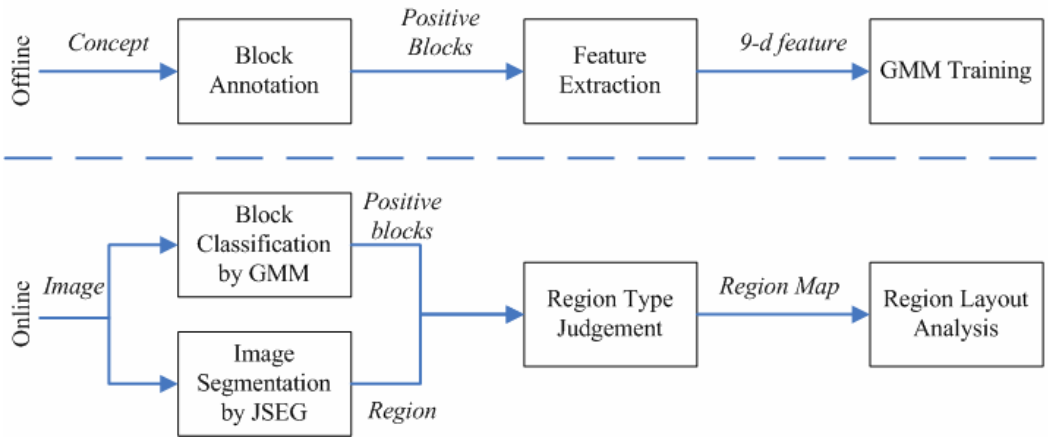


Figure.8 The flow chart of the specific method for mountain and water

Offline Phase

- Manually Annotation: Manually annotate three types (Mountain, Water, and Sky) of 16x16 blocks, and store them respectively as positive samples.
- Feature Extraction: Extract a 9-D feature vector from each block (256 pixels), 3-D for YUV color, and 6-D for Haar coefficients.
- Modeling: Train three GMMs for mountain, Water, and Sky using corresponding features.

Online Phase

- Blocks Detection: Detect 16x16 blocks by moving a 16x16 window on the input frame using GMMs obtained in the offline phase. Three types of blocks are marked respectively.
- Image Segmentation: Segment the input frame by JSEG algorithm [9].
- Semantic Region: For each region segmented by JSEG, if its area is covered by one type of blocks over a threshold, the region is marked as that type. Then three types of region maps are obtained.
- Classification: Input three region maps into five classifiers to analyze the layout of natural scenes, and apply Borda Count or Logistic Regression for fusing the five confidences.

4.2.5. Specific Method for US-flag

Through observation, we find the US-flag consist of parts of similar color and texture. In our system, the key-frame is divided into 16*16 blocks. Then color feature and texture feature are extracted from these blocks. By using SVM for training, color model and texture model are constructed. For each testing key-frame, possible US-flag parts are first detected by SVM models. Connected regions are segmented and judged whether they are similar to the US-flag. The similarity is the possibility of the existence of US-flag concept.

4.3. Evaluation

We submitted 7 runs for high-level feature extraction task:

B_D_PCA_BC_1: multi-model, PCA, Borda count

B_D_PCA_LR_2: multi-model, PCA, logistic regression

B_D_LPP_BC_3: multi-model, supervised LPP, Borda count

B_D_LPP_LR_4: multi-model, supervised LPP, logistic regression

A_D_MC_5: clustering, multi-model, Borda count?

B_D_SPE2_6: specific methods for specific concepts

A_D_ASR_7: ASR-based

Figure.9 compares our best performance with the best and median performance across all runs for all the 10 concepts. Most of our results are above the median except feature 40, map. The poor performance is caused by our manually dropping of some positive samples, which reduce the range of detection.

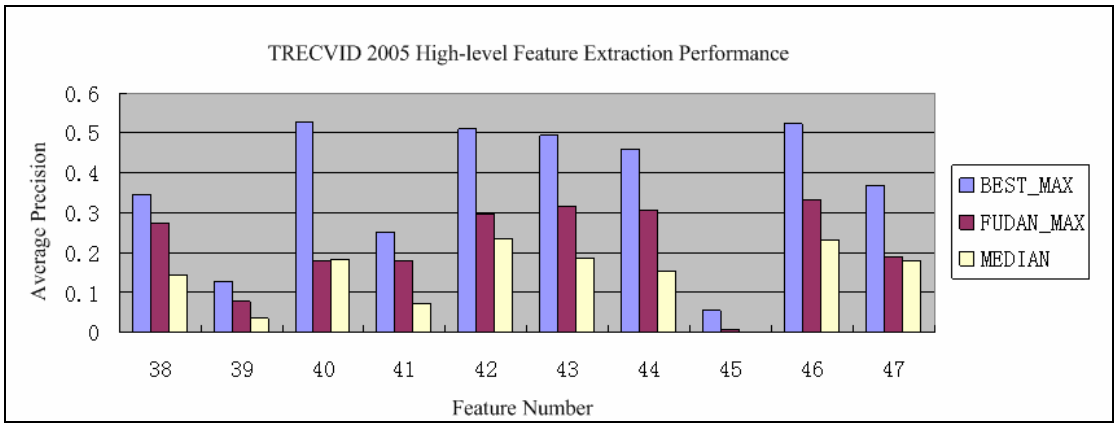


Figure.9 TRECVID 2005 high-level feature extraction performance

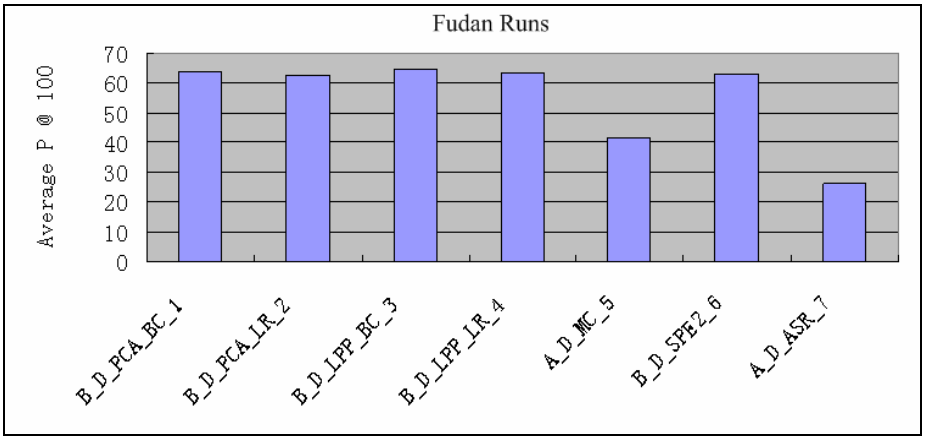


Figure.10 Average precision at a depth of 100 of Fudan runs

Figure.10 compares our submitted runs' precision at a depth of 100. We find that:

- A lower dimension improves the classifier performance when the original feature dimension is very high. For dimension reduction, PCA and supervised LPP almost have the same performance.

- Multi-model fusion works better than any single modality significantly. For supervised fusion method, the training process affects the fusion result greatly.
- Region-based method works better than global method for specific object concepts.
- Clustering the positive samples first has not improved the performance in our system. We believe that clustering performance is a key factor.
- Only use text information for high-level feature extraction has a worse performance than visual based method.

5. Search

For the search task of TRECVID 2005, we adopt multi-model fusion to realize the manual search. We proposed a new method of multi-model fusion which merges the multi-model query results using relation expression, and ranked the query by merge confidence (MC). In addition, the traditional method of linear fusion is also used to complete several runs. In the sub-section we will introduce these methods in details.

5.1. Multi-Model for Video Retrieval

The aim of video retrieval is to find a set of video shots for a given query, which is formulated in multi-modalities including text description, global features, visual concepts, and camera motion features. Every model only searches the video database using one kind of features which can't obtain satisfied query results, but the multi-model fusion can achieve better performance.

5.1.1. Text Retrieval

We mainly adopt two kinds of text information in text retrieval: the results of ASR/MT and the results of VOCR. We distill the text information corresponding to every shot and store them in an invert file, then query keywords on it. Because the original query is short and contains little contextual information, it is hard to just make use of this query to retrieve most relevant video stories. In our system, we distill the keywords from the topic and the ASR/MT results of example video and extend the query keywords by WordNet [10]. An IR search engine based on TF*IDF weighting scheme is adopted to generate the text retrieval scores for shots. Because relevant shot does not always have keyword hit on itself; more often the keyword hit is on its neighboring shots. Hence in order to overcome this temporal mismatch, we extend the query window and try to obtain more relevant shots.

5.1.2. Global Features

The similarity of the key-frame of the shot to the example images and the key-frames extracted from the example video can be used to retrieve similarity shots. In our system, four global features are used to retrieval similar video by QBE method. They are 192-d HSV color histogram, 81-d LAB color histogram, 1200-d GABOR texture features, and 80-d edge direction histogram descriptor (EHD). For data consistency reason, the example images used are the key- frames of the provided video examples, while the provided (external) image examples are discarded as they are not from news video.

5.1.3. Visual Concepts

Visual concepts are the high-level features which have been distilled from one of the TRECVID tasks. Here we used 14 visual concepts. They are anchor, airplane, boat-ship, building, car, fire, map, marching, meeting, military, office, road, sports and waterfront. Most of the visual concepts are distilled by the method in Section 4.2.1 except anchor shots. We design an anchor shot detection method based on clustering, which detect the anchor shot by several specific characteristic of anchor shots, such as the position of anchor is comparatively fixed, repeatedly appear many times in one video, the duration of a shot comparatively longer, and so on. First we use the information of face detection to filter the shots which don't have faces and the face position does not fit the set threshold. Then we extract their HSV color histogram and perform clustering using the single-link clustering algorithm. The clusters which have the number of shots larger than the threshold are regard as anchor shots candidates. Finally, we filter the shots whose durations are less than 3 seconds from the anchor shots candidates and get the anchor shots.

5.1.4. Camera Motion

We adopt the query results of Section 3, and divide the camera motion features to seven aspects, pan left, pan right, tilt up, tilt down, zoom in, zoom out, still.

5.2. Systems of Video Retrieval

For the manual search task, we design and implement two video retrieval systems to complete the manual search task. These two systems adopt the method of multi-model fusion based on relation expression and multi-model fusion based on linear fusion respectively.

5.2.1. Multi-Model Fusion Based on Relation Expression

We propose a new search method which adopt multi-model query respectively and merge the results by relation expression. Figure.11 shows the overall framework of the retrieval system. We take every feature as an atom search engine and use them to search in video database respectively. Then the relation expression of every topic which we set beforehand is used to merge these query results and MC (Merge Confidence) is used to rank the final results. The method of MC is to add all the confidence value of merge in relation expression and use the sum to rank the shots. Experimental results show that this method can achieve higher performance than linear fusion.

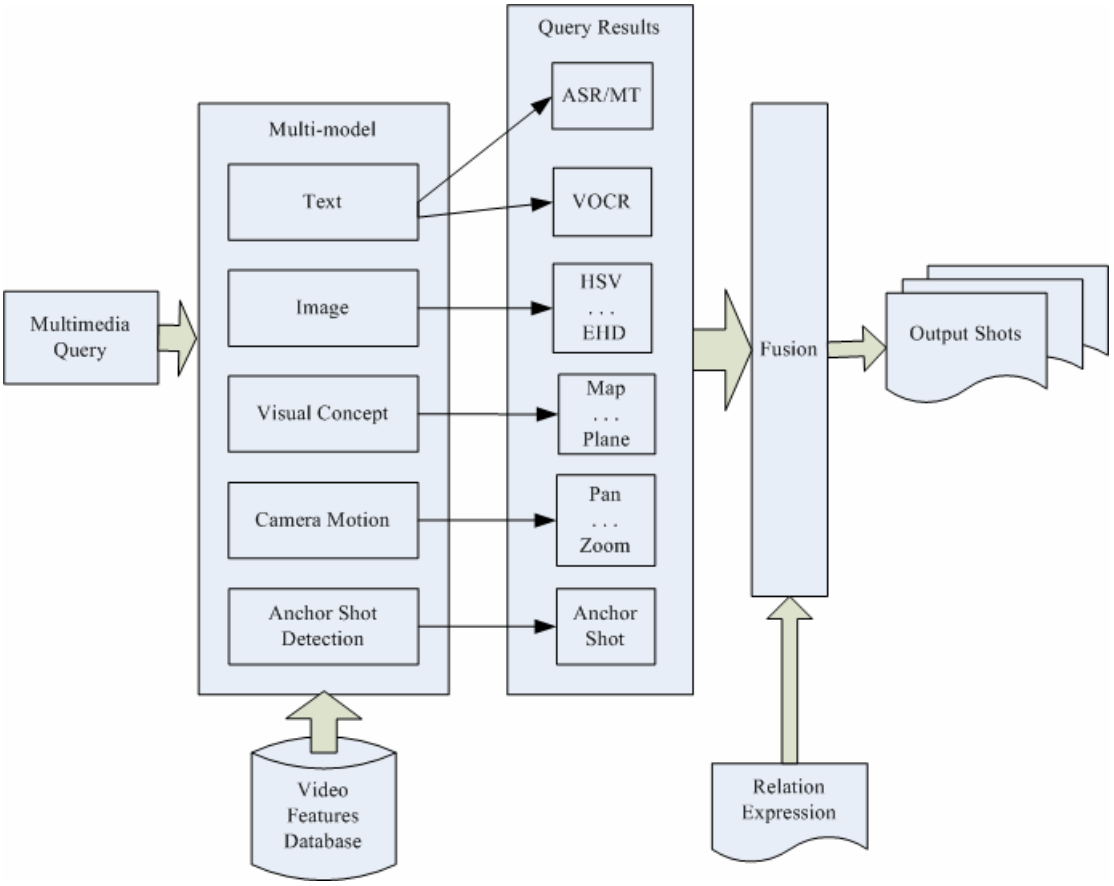


Figure.11 Framework of video retrieval system

5.2.2. Multi-Model Fusion Based on Linear Fusion

This system based on linear fusion is a general method which has been used widely. In this method we partition all the features to four classes, including text, global feature, visual concept and camera motion. All the visual concepts are represented by a model vector which was proposed by IBM [9], the same as camera motion.

We use the method of linear fusion to merge the query results of multi-models. Each model defines a different set of coefficients α_i to model the importance of different modality features for that query-class. After obtaining the results form different modality feature detectors, we combine the scores for shot S using Equation (1)

$$Final\_Rank(R) = \sum_{all\ models} \alpha_i^M * Rank_i \qquad where \sum \alpha_i^M = 1 \tag{1}$$

For the weights of multi-model, we adopt the query-specific weights learning by Logistic Regression which was proposed by CMU [12]. In this method we train the weights of models for every topic. Online training and offline training are used in our system. For the offline method, we train the weights on develop data by logistic regression and collect relevant shots (as many as possible) in 15 minutes. For the online method, we train query-specific weights from the pseudo relevance feedback, which is done

automatically without human effort. Since text retrieval provides the most important clues, our strategy is: among the top-400 shots ranked by text retrieval scores, label the first 100 shots as relevant and the rest 300 as irrelevant. Then weights are trained on these labeled examples by logistic regression.

5.3. Evaluations

We submitted 7 manual runs to TRECVID 2005 for evaluation. They are:

M_A_2_D_MM_BC_1: Use all the models and fusion by relation expression, rank by the method of MC.

M_A_2_D_MM_2_BC_2: Use all the models and twice fusion.

M_A_2_D_AOH_LR_ONLINE_3: Employ textual feature and visual concepts, fusion by linear fusion. Adopt information of pseudo relevance feedback to train the weights of every model for every topic by logistic regression.

M_A_2_D_MM_LR_OFFLINE_4: Use all the models and fusion by linear fusion. Train the weights of every model for every topic on develop data by logistic regression.

M_A_2_D_AO_5: Only use textual retrieval.

M_A_2_D_AOH_LR_OFFLINE_6: Employ textual feature and visual concepts, fusion by linear fusion. Train the weights of every model for every topic on develop data by logistic regression.

M_A_1_D_A_7: based only on the text from the provided ASR output and on the text of the topics.

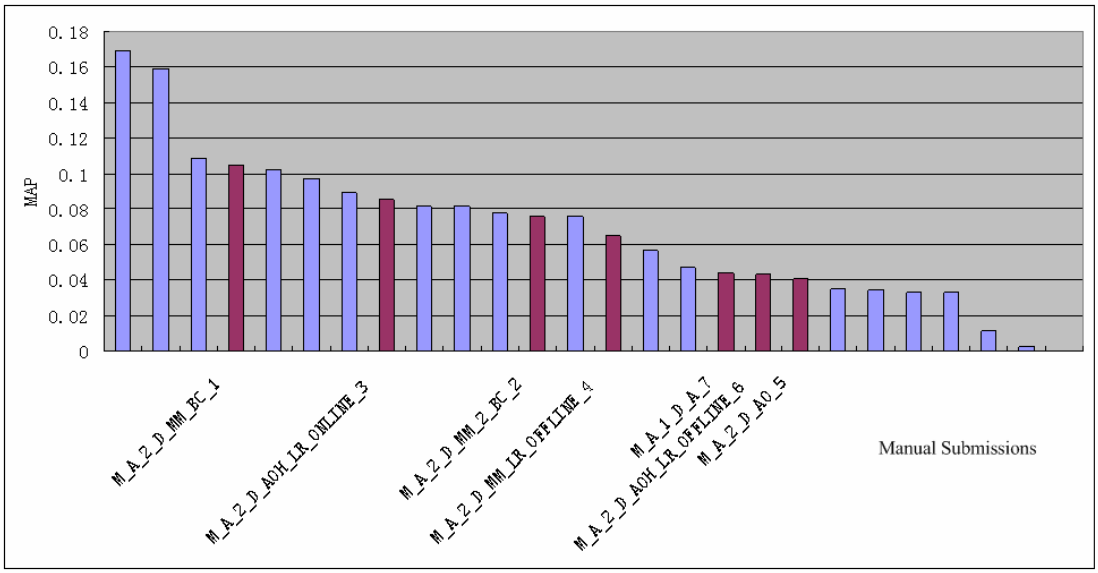
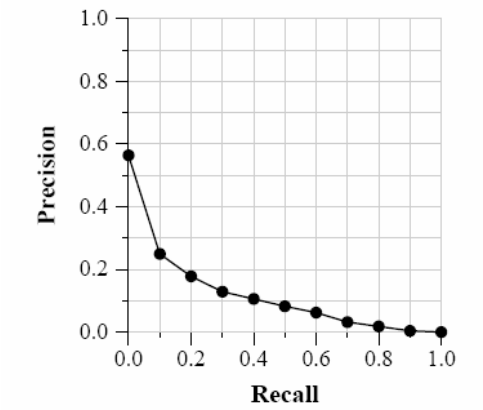


Figure.12 Performance of Fudan manual search submissions versus other manual submissions

Experimental results of the MAPs (mean average precision) of our submissions against other submissions are shown in Figure.12. We find that Run M_A_2_D_MM_BC_1 gives the best MAP in all seven runs. It illustrates that the relation expression fusion method is better than the linear fusion methods. In addition, the performance of run M_A_2_D_AOH_LR_ONLINE_3 is better than other methods which use the linear fusion, which shows that training weights online is superior to training weights offline. From Figure.12, we also find that the fusion and rank method is very important for

multi-model video retrieval. Figure.13 gives the recall-precision and the precision at *n* shots of Run M_A_2_D_MM_BC_1.



Precision at n shots	
5	0.4000
10	0.3375
15	0.3139
20	0.3042
30	0.2972
100	0.2221
200	0.1644
500	0.1118
1000	0.0825

Figure.13 Recall-Precision of Run M_A_2_D_MM_BC_1

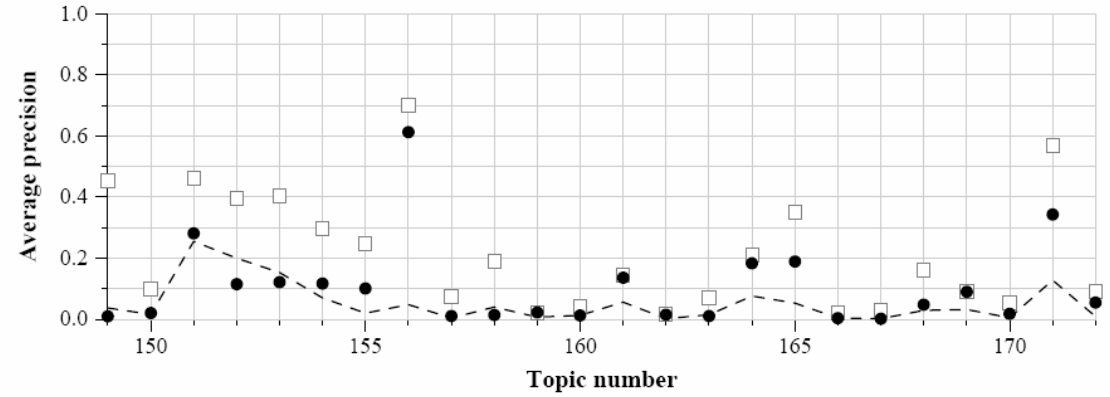


Figure.14 Run M_A_2_D_MM_BC_1 results breakdown

Figure.14 shows the result of our best run, which has achieved a mean average precision (MAP) of 0.105. We have achieved best or closed to best results for some of the queries. From the evaluations, our systems can deal with some topics well such as objects and scenes.

6. Summary

In shot boundary determination task, the performance of our system is not satisfying. In our system, the shot graduals are regarded as cut which is inconsistent with the reference answer. So a cut insertion error and a gradual deletion error results low performance.

Camera motion detection task is a new individual task of TRECVID. We find mutual relationship information between motion vectors is useful for single P-frame camera motion classification. However, human perceptive camera motion detection based on the classified P-frames is a problem that we have not dealt well with.

For high-level feature extraction task, we try a scale of different methods, including visual-based or text-based. For visual-based methods, global feature or regional feature, different dimension reduction,

different classifier combination algorithms are applied for separate runs. And the evaluation shows that we have more work to do to improve the performance.

For search task, we adopt multi-model fusion to realize the manual search. We proposed a new method which merges the multi-model query results using relation expression, and ranked the query by MC. In addition, the traditional method of linear fusion is also used to complete several runs. The evaluation shows that our method achieved better performance in all the submitted runs, and we can observe that the method of multi-model fusion by relation expression is better than the method of multi-model fusion by linear fusion from all of our runs.

Acknowledgement

This work was supported in part by Natural Science Foundation of China under contracts 60373020, 60402007 and 60533100, and Shanghai Municipal R&D Foundation under contracts 03DZ15019 and 03DZ14015, MoE R&D Foundation under contract 104075.

Reference

[1] C.C. Chang, and C.J. Lin, LibSVM: a library for support vector machines, 2004.

[2] X. Zhu, A.K. Elmagarmid, X.Xue, L.Wu, and A.C.Catlin, InsightVideo: Toward Hierarchical Video Content Organization for Efficient Browsing, Summarization and Retrieval, IEEE Trans. on Multimedia, 2005.

[3] MPEG-7 Overview, http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

[4] D.L. Swets, and J. Weng, Using Discriminant Eigenfeatures for Image Retrieval, IEEE Trans. on PAMI, 1996.

[5] Hwann-Tzong Chen; Huang-Wei Chang; Tyng-Luh Liu, Local Discriminant Embedding and Its Variants, CVPR, 2005.

[6] X. He and P. Niyogi**,** Locality Preserving Projections**,** NIPS 16, 2003.

[7] X. He, S. Yan, Y. Hu, and H.-J. Zhang, Learning a Locality Preserving Subspace for Visual Recognition, ICCV, 2003.

[8] O. Melnik, Y. Vardi, and C.H. Zhang, Mixed Group Ranks: Preference and Confidence in Classifier Combination.

[9] Y. Deng, B. S. Manjunath, “Unsupervised Segmentation of Color-Texture Regions in Images and Video,” IEEE Trans. on PAMI, 2001.

[10] WordNet, http://wordnet.princeton.edu/.

[11] J.R. Smith, M. Naphade, A. Natsev, Multimedia Semantic Indexing using Model Vectors, IEEE Intl. Conf. on Multimedia and Expo (ICME), Baltimore, MD , July, 2003.

[12] A.Hauptmannn, M.-Y. Chen, M. Christel, C. Huang et.al, Confounded Expectations: Informedia at TRECVID 2004.