Fudan University at TRECVID 2003

Lide Wu, Yuefei Guo, Xipeng Qiu, Zhe Feng,Jiawei Rong, Wanjun Jin, Danlan Zhou, Rongrong Wang, Ming Jin
Media Computing & Web Intelligence Group, Dept. of Computer Science and Engineering
Fudan University, Shanghai 200433, P. R. China

Abstract

This year we have participated in all the four tasks including shot boundary determination, story segmentation, high-level feature extraction and video search task. Several low-level visual, audio and text features are extracted for each of the tasks. In order to reduce the semantic gap between low-level features and high-level concepts, we have tried to use some learning algorithms, such as Adaboost and Maximum Entropy model, to automatically select and fuse low-level features. In this paper, we present the approaches for each task and discuss some of the evaluation results.

1. Introduction

This year, we participated in four tasks: (1) shot boundary determination, (2) story segmentation, (3) high-level feature extraction and (4) search task.

In Shot Boundary Determination Task, we submitted 6 runs using our old shot segmentation system, which is tuned on this year's data. In Story Segmentation and Classification task, we submitted 10 runs for type A, B and C task. In the high-level feature extraction, we divided 17 high-level features to four categories: scene, object, audio and motion. In search task, we submitted 8 manual runs, which use four methods.

This paper is organized as follows: section 2 describes the system of shot boundary determination. Then the methods for story segmentation are given in section 3. In section 4, we present our four sub-systems of high-level feature extraction task. In section 5, four types of methods for search task are described. Finally, section 6 gives some conclusions.

2. Shot Boundary Detection

This year we use almost the same of TREC-11 shot segmentation system [Wu02]. FFD (*Frame-to-Frame Difference*) calculated by luminance difference and color histogram similarity are used to detect the shot changes. We use two thresholds θ_C and θ_G, which are calculated automatically according to the FFD value histogram in 500 frames, to detect if there is a clear FFD value change caused by shot changes. Then flashlight detection and motion detection are applied on candidate shot changes to remove the false alarms of cut and gradual. Fade in/out detection is applied to all candidate gradual changes. If a black screen chain exists in the candidate duration, we think it is a fade. Otherwise, it will be labeled as dissolve. The parameters used in the system are trained and adjusted based on the TRECVID2003 Development Data.

Evaluation results show that performance of our system decreases a lot this year. We found that there are a lot of short graduals which duration is smaller then 3 frames in reference answer. In our system, such short graduals will be considered as cut because they are too short. Thus, a cut insertion error and a gradual deletion error will happen so that the cut precision and gradual recall are low.

3. Story Segmentation

News story segmentation is a new task of TRECVID2003, which requires segmenting news videos into story units and classifying them into two categories, namely news and miscellaneous. Considering the structural characteristic of news videos, there are some differences between news story segmentation and general scene segmentation. The latter mainly depends on similarity-based cluster algorithms while the former one involves anchorperson detection and topic detection, etc.

3.1 Anchorpersion(AP) Detection

In news video, anchorperson shot tends to be the beginning of a news story. According to our survey, almost half of the story boundaries are anchor person shots in this year's corpus. Therefore, anchorperson shot detection is crucial for news story segmentation.

Since anchorperson shows up in a news video repeatedly, and the anchorperson shots are similar so that we can detect them by a clustering procedure described below:

1. Initialization: $GroupNum$=1, $Group_1$={$Shot_1$}, i=2.
2. For $shot_i$, calculate the similarity with all existing groups and find the $group_k$ with the maximum similarity.

$$Lum(pixel) = 0.31 pixel_{red} + 0.10 pixel_{green} + 0.59 pixel_{blue} \tag{3.1}$$

$$ShotSimilariry(Shot_i, Shot_j) = \sum_k \left| Lum_{keyframe_i}(pixel_k) - Lum_{keyframe_j}(pixel_k) \right| \tag{3.2}$$

$$GroupSimilarity(Shot_i, Group_j) = \min_{Shot_k \in Group_j} \{ShotSimilarity(Shot_i, Shot_k)\} \tag{3.3}$$

$$k = \arg\max_j \{GroupSimilarity(Shot_i, Group_j),\ j = 1, \cdots, GroupNum\} \tag{3.4}$$

3. If $GroupSimilarity(Shot_i, Group_k)$ exceeds a threshold, we add $shot_i$ into $group_k$.
   Otherwise, create a new group which consists of $shot_i$, $GroupNum = GroupNum + 1$.
4. i=i+1, goto step 2.

After the clustering procedure, all the video shots are put into several groups. To decide whether a group is an anchorperson shots group, we filter them step by step. At first, we discard the groups consisting of only one shot. Then several filters are designed to improve the performance. Since commercials also occur in TV repeatedly, they cause several false alarms. We use an advertisement filter as the first one to remove groups containing ads. The second filter used is the non-face filter. We use face detection result to remove those groups not containing face. When adopting the two filters above, we tried two strategies: strict and loose. In strict strategy, we discard the group as long as it has one commercial shot (or non-face shot) while in loose one we discard a group only when all of its shots are commercial shots (or non-face shots). Apparently, strict strategy guaranteed precision at the sacrifice of recall and vice versa. However, as long as F-value is concerned, loose strategy performed a bit better than the strict one.

The third and the fourth filters are motion filter and range filter. Since anchorperson shots are supposed to be still shots, we removed the groups with great motion. We consider a shot as a still shot if 65% of its frames are still and a group as still if 80% of its shots are still shots. We also discard the groups whose range is less than ten shots concerning the fact that anchorperson shots tend to scatter all along the video. By adopting the

latter two filters, the F-value rises several percents.

Nevertheless, in TRECVID 2003 corpus, anchorperson shots are characteristic of complex background and variant anchorperson position (see the key frames shown in Figure.1). This affects the performance of our methods. Our anchorperson detection algorithm is capable of detecting only 60% of all the anchorperson shots, which is the cause for the poor recall of our final news segmentation system.

Figure.1 Some examples: keyframes of anchorperson shot

In order to make the clustering more robust when background and anchorperson position are variant, we split the keyframe of each shot into three parts with same size: Left, Middle and Right. Equation 3.2 in clustering algorithm is replaced with:

$$ShotSimilariry(Shot_i, Shot_j) = \max_{m,n\in\{left,middle,right\}} \left\{ \sum_k \left| L_{keyframe_i,m}(pixel_k) - L_{keyframe_j,n}(pixel_k) \right| \right\} \quad (3.5)$$

However, we found that this method caused a lot of false alarms with a little improvement on recall. Finally, we still use equation 3.2 to measure the similarity between two shots.

3.2 Text Segmentation

Text segmentation has been studied for decades. However, for news story segmentation, it has to change somewhat, since what we get is not the close caption but ASR transcription. We mainly consider the text similarity at some candidate boundaries. First, we establish a vocabulary containing only nouns and verbs from the news video ASR transcription made by LIMSI [Gauvain02]. Secondly, for each candidate boundary, we build two word histograms based on certain account of words before and after this boundary (in experiments, we chose 50 words). Then, we calculate the histogram intersection as the similarity at each candidate boundary and found the valley of the similarity curves. If the similarity value of this valley is less than a threshold, it is determined to be a story boundary. There is a problem for this method: if there are story boundaries in the first or the last 50 words, they cannot be detected.

3.3 Other useful information

There are some other information which is probably useful to story boundary detection, such as commercial, speaker change, audio type change, etc. Change from non-commercial to commercial may indicate a change from news to miscellaneous, therefore creates a story boundary. Actually, a combination of anchorperson shot and commercial renders the precision decreasing and the recall increasing about ten percents compared with using anchorperson information only.

3.4 News Story Segmentation

We used several types of methods. For type A, sentence boundary and shot boundary are selected as candidate boundaries for text segment algorithm mentioned in section 3.2. After experiments, we figure it out that text segmentation at shot boundary performs better than at sentence boundary. This is because sentence boundary detection in ASR is based on silence detection, which is not so reliable. It will merge several sentences into one and result in much lower recall for our story boundary detection system. The final evaluation of our text segmentation system shows that its performance is much poorer than the methods that have been reported so far. We believe it is to some degree due to the complex structure of news text: some long stories are made up of tens of sentences while some short one consists of only one sentence.

For type B and type C, we use both audio (i.e. text) and visual features described above. Rules and maximum entropy model are selected as classifiers to decide whether a shot boundary is a news story boundary. We do not consider sub-shot because about 92% of all the story boundaries occurs at shot boundary according to our investigation on TRECVID Development data.

For type B with rules they are simple but turn out to be efficient. We consider a shot boundary to be a story boundary if it is a change from non-AP shot to AP shot or it is a change from ads to non-ads.

For type C with rules, we combine the results of type A run and type B run in intersection way and union way. Evidently, intersection way can get high precision with low recall and vice versa. In our submission, we choose the intersection run though the two F-values are of little difference.

We use maximum entropy classifier because it is efficient when the features are discrete and can be effectively selected automatically. However, maximum entropy is beaten in type B task by rules, but interestingly surpasses the rule classifier in type C that involves text information. This may be caused by the low recall of intersection operation when applying rules in type C.

Table 1 News Story Segmentation Evaluation

	Type A		Type B		Type C		Best	Median
	Shot	Sent	Rule	ME	Rule	ME		
Recall	0.587	0.270	0.500	0.427	0.258	0.378	0.790	0.378
Precision	0.272	0.287	0.584	0.644	0.825	0.773	0.844	0.584
F-Value	0.372	0.278	0.539	0.514	0.393	0.508	0.775	0.386

Table 1 shows our evaluation results. We can find that visual information is more effective than text information. Meanwhile, visual information can improve the system which only using text information. In our experiments on audio features, we found that speaker change and audio type change are not helpful. It is probably because news story boundary is not necessarily the speaker change point and music is rare in news that sometimes co-occurs with speech.

3.5 News Story Classification

In video type classification, we use the three ratios below as features:

$$Ratio_{AD} = \frac{\#\ of\ \ commercial\ \ shots\ in\ story\ i}{\#\ of\ \ shots\ in\ story\ i} \tag{3.6}$$

$$Ratio_{Speech} = \frac{\#\ of\ \ speech\ shots\ in\ story\ i}{\#\ of\ \ shots\ in\ story\ i} \tag{3.7}$$

$$Ratio_{Music} = \frac{\#\ of\ \ music\ shots\ in\ story\ i}{\#\ of\ \ shots\ in\ story\ i} \tag{3.8}$$

We use TRECVID 2003 Development data as the training data and use GMM and maximum entropy model as classifiers. From the results shown in Table 2, we can find maximum entropy model gives more satisfactory classification results.

Table 2 News Story Classification Evaluation

	Type B Rule		Type B ME		Type C ME		Best	Median
	GMM	ME	GMM	ME	GMM	ME		
Recall	0.836	0.935	0.672	0.799	0.632	0.767	1.000	0.899
Precision	0.873	0.848	0.893	0.831	0.875	0.819	0.965	0.842
F-Value	0.854	0.889	0.767	0.815	0.734	0.792	0.944	0.808

For Type A runs, only ASR transcription can be used for news story classification. We use a text classification algorithm based on VSM (Vector Space Model) [Huang01].

4. High-level Feature Extraction

We classify the high-level features into four categories:

- Scene feature: High level features concern about the scene information, such as outdoor, building, road, vegetation, non-studio setting, sport events and weather news. They can be extracted through the low-level features of the whole images.
- Object feature: High level features concern about the object in the video, such as car, aircraft, animal, face, people and person X. They can be extracted through the low-level features of the regional images.
- Audio feature: High level features relevant to audio information, such as monologue and female speech. They can be extracted from audio features.
- Motion feature: High level features relevant to camera motion, such as zoom-in and physical violence. They can be extracted from motion information.

In section 4.1, we describe a general approach to the scene feature detection and a special system for vegetation detection. In section 4.2, we introduce our methods for the object feature detection, such as face detection, car detection etc. Audio feature detection and motion feature detection are described in section 4.3 and 4.4 respectively. Finally, an ASR based method is discussed in section 4.5.

4.1 Scene Feature Detection

For the scene features such as outdoor, non-studio setting, weather news, sports event, building, vegetation and road, we collect the positive and negative training images from the TRECVID2003 Development Data with TRECVID2003 Annotation, and then use the following system to extract these features. Besides the general system, we used a special system for vegetation detection.

4.1.1 General Scene Feature Detection System

The system calculates the confidence for each key frame instead of the whole shot. Each key frame is divided into 4*4 sub-blocks and the following low-level features are extracted on these sub-blocks. There are

totally nine features used in our system.

- LAB color histogram (3 features: each feature corresponding to 64 bins for single color channels)
- Edge direction histogram (1 feature: 73 bins, the first 72 bins corresponding to the 72 edge directions and the last bin corresponding to the ratio between the non-edge pixels and total pixels)
- Co-occurrence texture: (5 features: entropy, correlation, contrast, uniformity, inverse difference moment) [Haralick73]

We calculate the confidence as a fuzzy KNN classification. For the incoming image x, we find its k nearest neighbors in training samples that are labeled $\{x_i, i = 1,2, \ldots, k\}$, the confidence is defined as follow:

$$conf(x) = \frac{\sum_{i=1}^{k} conf(x_i)\left(SIM_{total}(x, x_i)^{b-1}\right)}{\sum_{i=1}^{k} SIM_{total}(x, x_i)^{b-1}}, \quad b > 1 \tag{4.1}$$

where SIM_{total} is the similarity between two images with the low-level features mentioned above, which are calculated by summing the similarities of 16 sub-blocks. $conf(x_i)$ is the confidence of training samples, which can be 1 or 0 depending on whether x_i is positive or negative. The parameter b is used to represent the importance of the similarity.

In equation 4.1, we can get the different similarities of two images with their different low-level features used. For different scene feature, different low-level feature has different discriminative abilities. Therefore, feature selection and fusion are important for the scene features detection. Simply combining every feature by experience may not get the optimal results. In this paper, we use a AdaBoost based algorithm [Freund96] to automatically adjust the features weight. The mechanism of boosting [Schapire98] is combining many weak classifiers to a final strong classifier by giving bigger weights to the better classifiers and smaller weights to the worse classifiers. It can automatically re-weighting the training samples and adjust the weights of the classifiers. In our algorithm, each weak classifier is trained use one feature. Therefore, the feature selection can be realized by adjust the weight of the respective weak classifier.

In our system, each weak classifier is a fuzzy classifier as follow:

$$h_j(x) = conf_j(x) = \frac{\sum_{i=1}^{k} conf_j(x_i)\left(SIM_j(x, x_i)^{b-1}\right)}{\sum_{i=1}^{k} SIM_j(x, x_i)^{b-1}}, b > 1 \tag{4.2}$$

where, $conf_j(x)$ is the confidence calculated with j-th feature of the sample x. After automatic fusion of all the classifiers, we can get the optimal results. The confidence got by final classifier is used as the ranking value for each keyframe.

The variant AdaBoost algorithm is described below:

- For the training samples set (x_1,y_1) , … , (x_n,y_n) , where y_i = 1, 0 respective represent the positives and the negatives.
- Initialize weights $w_{1,i} = \begin{cases} 1/2m & \text{when } y_i = 1 \\ 1/2l & \text{when } y_i = 0 \end{cases}$ for training example i, where m and l are the number of positives and negatives respectively.

- Suppose we extract T low-level feature from images，For t = 1 … T, repeat the step 1~4.
  1) Normalize weights of training examples
  2) For each feature j train a classifier h_j, where $SIM_j(x,x_i)=SIM_j(x,x_i)*w_{t,i}$, and estimate its error ε_j with respect to w_t as follow:

$$\varepsilon_j = \sum_i w_{t,i} \times \begin{cases} 1 - conf_j(x_i) & \text{if } y_i = 1 \\ conf_j(x_i) & \text{if } y_i = 0 \end{cases}$$

  Merge to:

$$\varepsilon_j = \sum_i w_{t,i} \times \left| y_j - conf_j(x_i) \right|$$

  3) Choose classifier $h_{t,j}$ with lowest error ratio as the classifier h_t.
  4) Update weights according to:

$$w_{t+1,i} = w_{t,i} \beta_t^{\left(1-\left| y_i - conf(x_i) \right|\right)}$$

  where, $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$， $conf(x_i)$ is the confidence of sample x_i calculated by the weak classifier h_t.

- The final strong classifier is:

$$h(x) = \sum_{t=1}^{T} \alpha_t h_t(x), \text{ where } a_t = \log\left(\frac{1}{\beta_t}\right)$$

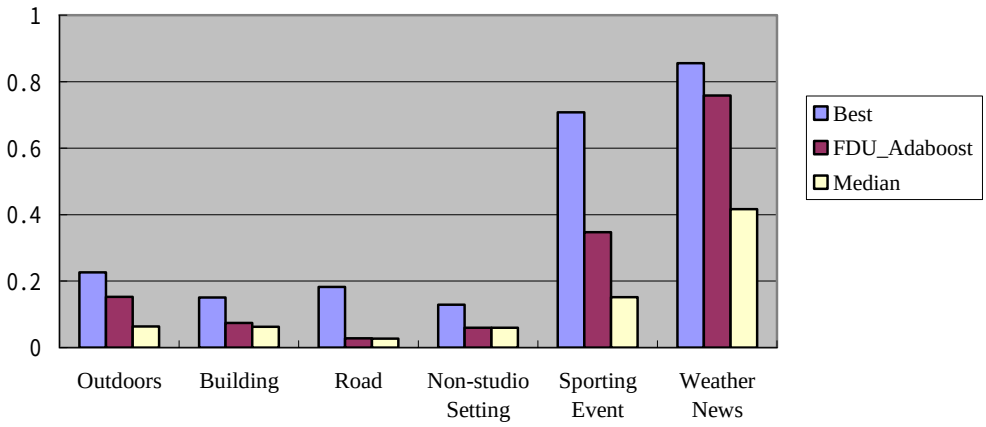

Figure 2. General Scene feature extraction Evaluation (Average Precision)

4.1.2 Special Scene Feature Detection System for Vegetation Detection

We also apply a specific method for vegetation detection using some special features.

In vegetation detection task, color and texture information are used in training stage. Firstly, some keyframes containing vegetation are extracted from video (typically 20 keyframes). Then for each keyframe, a vegetation patch (typically 48pixel*48pixel) is randomly selected manually as our training samples. The training samples should be chosen as variously as possible so that different illumination conditions are considered. Secondly, we discretize the RGB color space into 4096 color values. Then a color lookup table is

constructed in which each item represents the importance of a certain color for vegetation. A novel metrics CF*IRF (Color Frequency and Inverse Region Frequency) is introduced to calculate the weight of each item in the color lookup table, i.e. color importance. Meanwhile, a texture classifier is trained using the Gabor feature. We choose Support Vector Machine (SVM) as the classifier and choose 4 scales, 6 orientations for the Gabor filter bank.

When estimating if any vegetation exists in a keyframe, we firstly divide each keyframe to 8*6 rectangle patches. Then, trained color lookup table and texture classifier are used to estimate the possibility of vegetation presence in each patch. Patch with large possibility is named “vegetation-like patch”. Whether the keyframe has the vegetation or not depends on the number and presence possibility of vegetation-like patches. Here we simply sum up the possibility of all vegetation-like patches as the final ranking value for each keyframe.

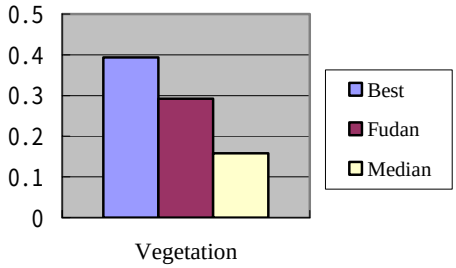

Figure 3. Evaluation on special vegetation detection system (Average Precision)

4.2 Object Feature Detection

4.2.1 Detecting Object Feature using Scene Information: Aircraft and Animal

One of the biggest difficulty in object detection is there are too many uncertainty in the appearance of an object. Individuals of same kind of object often have different color, shape and texture. However, the scene where a kind of object often appears is usually similar. Thus, we can use scene detection instead of detecting object itself.

For aircraft detection, we use the method described in section 4.1.1.

As to the animal detection, we apply similar approach as we did for vegetation in section 4.1.2. However, there are some differences. The first is that we use nearest neighborhood instead of SVM as the texture classifier, since the appearance of different animals vary greatly. The second is that when establishing the color lookup table, we use some negative samples as well as positive ones, so that the false alarm introduced by color diversity can be reduced as much as possible. The third difference is that in test stage, we consider the spatial distribution of animal-like patches. The possibility of one animal-like patch increases if it is adjacent to other animal-like patches.

4.2.2 Car Detection

Our car detection system uses the method proposed by Schneiderman [Schneiderman02]. It is a general object detecting system and can detect cars with different pose and size. The detector consists of multiple classifiers. Each classifier is specific to detect an object of fixed size at a specific pose. In detection, we apply all classifiers on the input keyframe and combine their results. For each classifier, we perform a wavelet transform on the input window using a linear phase 5/3 perfect reconstruction filter bank. The key of each classifier is to establish several statistical tables describing the probability of every input window being object

and non-object. In detection, we just look up the tables to get the confidence of each candidate window *w*.

$$Conf(w) = log(Pr(w|object)) - log(Pr(w|non-object)) \tag{4.3}$$

Car detection runs only on the keyframes of video shots. The ranking value is calculated as follows:

$$RV_{Car} = \frac{1}{|\{w|Conf(w) > th\}|} \sum_{Conf(w)>th} Conf(w) \tag{4.4}$$

In our system, we developed three different view classifiers: right, frontal and 45-angled right profile. To detect left and back profile cars, we just run the corresponding classifiers on the mirror-reversed images.

4.2.3 Face / People Detection

A real-time face detection method proposed by Viola [Viola01] is applied to detect face. For the people detection, a video shot is a shot with "people" feature if the number of detected faces is larger than or equal to 3. For the news subject face detection, we remove those shots with anchorpersons.

4.2.4 Person X detection

For the person X detection, we use the following face recognition algorithm only based on positive samples [Guo03].

- Training procedure:

Let x_1, x_2, …, x_N are the training face samples of one person. The within-class scatter matrix of them can be presented as follows:

$$S_w = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \tag{4.5}$$

where μ is the mean of training samples. Suppose $\{\varphi_1, \varphi_2, \cdots, \varphi_k\}$ is the orthonormal set of vectors which is orthogonal with the set of vectors $\{x_i - \mu, i = 1, \cdots, N\}$. Then the following equations are hold.

$$\varphi_i^T S_w \varphi_i = 0,\ i = 1,2,\cdots,k \tag{4.6}$$

It is easy to see that the within-class distance of training samples equals to zero on the direction φ_i. That is to say that the projection vector of every training sample on $\varphi_1, \varphi_2, \cdots, \varphi_k$ equals to the one of the sample mean.

- Testing procedure:

For each detected face α on keyframe, we project vector $\alpha - \mu$ on directions $\varphi_1, \varphi_2, \cdots, \varphi_k$. Then k features a_1, a_2, …, a_k are obtained. Consider the distance $d_\alpha = \sqrt{a_1^2 + a_2^2 + \cdots + a_k^2}$, it is reasonable that a face α with smaller d is more possible to be person X. Thus, we define the ranking value of person X as:

$$RV_{PersonX}(Shot_i) = \max_{\alpha\ is\ detected\ in\ keyframe\ of\ shot_i} \left\{ \frac{1}{d_\alpha + 1} \right\} \tag{4.7}$$

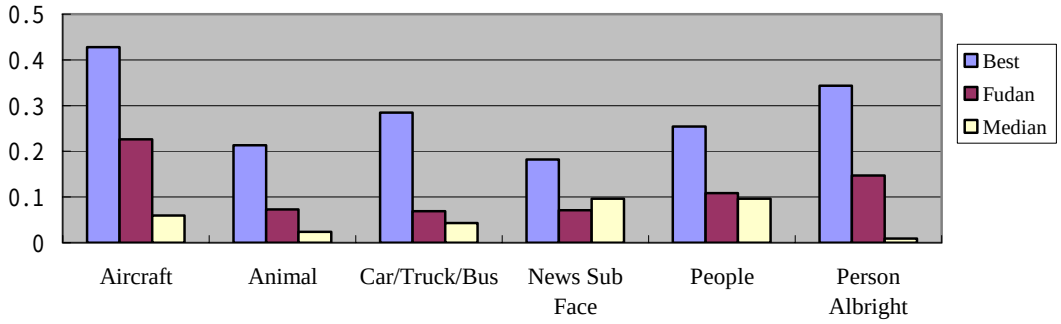

Figure 4. Object feature extraction Evaluation (Average Precision)

4.3 Audio Feature Detection

In TRECVID 2003, there are two high-level features relevant to audio information: Female Speech and News Subject Monologue. The ranking value for these two features combines the results of audio type classification, monologue detection, gender detection, and news subject detection.

An audio classification system based on maximum entropy model is applied to give the ranking value of speech for each video shot [Feng03]. In this system, we use some widely used audio features. Maximum entropy model can select features that are more effective and determine audio type for each 1-second clip. The maximum entropy model we used is trained from TRECVID2003 Develop Data using TRECVID2003 Annotations. The ranking value of speech is calculated by:

$$RV_{speech} = \frac{\#of\ \ clips\ whose\ type\ is\ speech}{\#of\ \ clips\ in\ a\ shot} \tag{4.8}$$

We have not developed a special monologue detection system. We just use the information from face detection. If a video shot contains only one face and the duration of this face in the shots is long enough, we think this shot a monologue.

$$RV_{monologue} = RV_{speech} \times RV_{face} \tag{4.9}$$

Gender detection is applied on the 1-second window. We have tried three kinds of features: 12-MFCC, 10-LPC and Pitch. Gaussian mixture models are trained on TRECVID2003 Development Data using TRECVID2003 Annotation for female speech and male speech. Applying these trained models on 1-second clip, we can get the gender of each audio clip. The ranking value of female speech is calculated by:

$$RV_{female} = \frac{\#of\ \ clips\ whose\ type\ is\ \ female\ speech}{\#of\ \ clips\ in\ a\ shot} \times RV_{monologue} \tag{4.10}$$

In our submission, Run01 and Run02 use 12-MFCC feature, Run03 and Run04 use 10-LPC feature, Run05, Run06 and Run07 use Pitch feature. Run08 combines the results of Run01~Run07 by multiply their ranking value. Evaluation results show that MFCC feature is better than Pitch and LPC. And our female speech detection get a good performance in submitted 34 runs.

In Section 4.2.3, we have mentioned that news subject face is detected by removing the anchorperson face

from face detection results. Combine the ranking value of speech and news subject face, the ranking value of news subject monologue is given as follows:

$$RV_{NewsSubMonologue} = RV_{NewsSubFace} \times RV_{speech} \tag{4.11}$$

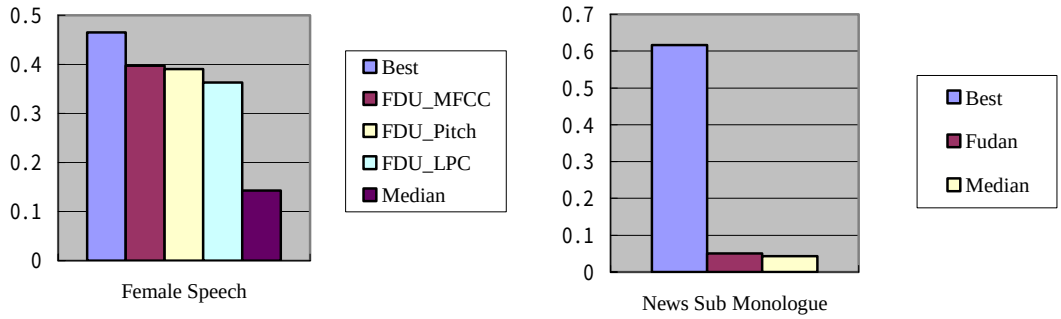

Figure 5. Audio feature extraction Evaluation (Average Precision)

4.4 Motion Feature Detection

In our system, we analyze camera motion of each frame by the motion vectors obtained from MPEG stream. Each motion is composed of motion magnitude and motion direction. The system tries to catenate them into sub-shots automatically. We define sub-shot as continuous frames in one shot with similar camera motion. The rank value of zoom-in feature can be calculated by:

$$RV_{zoom-in} = \frac{length\ of\ \ subshots\ \ which\ type\ is\ \ zoom\ in}{length\ of\ \ shot} \tag{4.12}$$

In IBM annotation tool, we learn that physical violence refers to riot, bomb etc. We use acceleration of motion to evaluate physical violence. We obtain motion vector for each 16*16 macro block from DC coefficients, and then decompose them into x and y axis. The acceleration of motion is calculated by:

$$acce(i) = \sum_{h=1}^{Height/16} \sum_{w=1}^{Width/16} |mv(i,h,w).x - mv(i+1,h,w).x| + |mv(i,h,w).y - mv(i+1,h,w).y| \tag{4.13}$$

Here, *mv(i,h,w).x* denotes the x-component of motion vector at position (*h*,*w*) in frame *i*. We submitted three runs. Run01 is sorted based on average acceleration of all frames in a shot, Run02 is based on the maximum acceleration of frames in a shot, and Run03 is based on the product of the average acceleration and maximum acceleration. The global acceleration metric has the shortcoming that it is incapable of differentiating fast camera motion from fast object movement. Therefore, some results returned with high acceleration are fast camera motion, or even fast shot transitions.

4.5 ASR-based Feature Detection

We have tried to use LIMSI's ASR transcription [Gauvain02] to extract high-level features. For each high-level feature, we have selected some keywords from TRECVID2003 Development Data to represent them. This method is useful only for those features containing a clear semantic concept that can be extracted from speech.

Figure 6. Motion feature extraction Evaluation (Average Precision)

5. Search

We have submitted eight manual runs in Search Task. They come from four different searching systems: ASR System (Run04), Color Histogram System (Run08), Multi Feature System (Run06) and Special Searching System (Run01). After combining high-level feature confidence with the results generated by Run04, Run06 and Run08, we got another 3 runs (Run03, Run 05 and Run07). Finally, Run02 is gotten from the combination of Run03 and Run 01.

5.1 ASR System

In ASR system, we generate invert table from LIMSI's ASR transcription [Gauvain02]. After analyzing the topics, document retrieval is applied based on the query words selected for each topic by user.

5.2 Color Histogram System

Color Histogram similarity is used to compare the image example and keyframe of each shot. It can provide us the similarity between image example and keyframe. In TRECVID 2003, we calculate the histogram in RGB space. During the calculation and comparison, two modes are used:

- Whole Image Mode: For both keyframe and image example, the histogram is calculated on the whole image.
- Block mode: For keyframe, we split it into several blocks with different size. The histogram is calculated on each block. Then the histogram comparison is processed between the histogram of each block and image example. The maximum similarity will be selected as the final similarity.

In searching, Block mode is used on the topics which is concerning about a certain object. User can select the mode.

5.3 Multi Feature system

The system extracts the following low-level features from the keyframe for each video shot based on 4*4 sub-blocks. Then we calculated the similarity between query image examples and keyframes of each shot.

- LAB color histogram (3 features: each feature corresponding to 64 bins for single color channels)
- Edge direction histogram (1 feature: 73 bins, the first 72 bins corresponding to the 72 edge directions and the last bin corresponding to the ratio between the non-edge pixels and total pixels)

- Co-occurrence texture: (5 features: entropy, correlation, contrast, uniformity, inverse difference moment) [Haralick73]

5.4 Special Searching System

Special Searching System consists of six different subsystems. After analyzing the search topic, user should determine which subsystem is the best to fit this query. For each subsystem, the query format is also different.

5.4.1 Human Face Recognition subystem

The subsystem is same with the system for person X detection mentioned in section 4.2.4. It can be used on the queries searching a special people. According to the evaluation results, this subsystem is not effective because there are not enough image examples for each query. This will decrease the performance much.

5.4.2 Adaboost Classifier subsystem

The subsystem is the same as the general scene feature detection system mentioned in section 4.1.1. It can be used on the queries which concern about scene, such as flame (Topic 112).

5.4.3 Multi Feature subsystem

The subsystem is the same as the multi feature system mentioned in section 5.3. However, it is based on the whole images instead of the 4*4 sub-blocks. Similar to Adaboost classifier subsystem, it is also can be used on the queries concerning about scene.

5.4.4 Motion subsystem

In motion subsystem, searching is based on motion information analysis mentioned in section 4.4.

After user selects the motion type they concerned, system can give the confidence of this motion type for each shot. The confidence is the percentage of the length of sub-shots with selected motion type in a shot.

This subsystem can be used to detect the events which contain specific motions, such as aircraft taking off (Topic 104).

5.4.5 Color Texture subsystem

This subsystem is the same as the special vegetation detection system mentioned in section 4.1.2. It can be used to detect the scene or object with specific color and texture, such as cat (Topic122).

5.4.6 Color Region subsystem

This subsystem is based on some rules about colors and regions. Only when the mean RGB value of some regions in an image is within a pre-calculated range, we can say that the image is the possible search result.

For some topics, we can find that different image examples have some commonness. For example, all image examples of snow mountain with blue sky contain blue color in the top and white color in some other areas.

During searching, users select some regions with specified colors from image examples, such as blue sky, white snow, and green up-arrow in rising Down Jones Graphics etc. Then system calculates their color histograms to get the range for color matching. According to the rules defined by user, system calculates average RGB value in each specified region of the keyframes and judge whether it satisfies the rule user specified.

For Topic 120, searching the graphic of Dow Jones Industrial Average showing a rise for one day, we first

run our OCR system [Hu02] over all keyframes and establish the index of caption text in the videos. Then color region system only runs on the keyframes containing the text “Dow Jones”.

Table 3 list the subsystem user selected for each query.

Table 3. Subsystems selected by user for each topic

Subsystem	Topics
Human Face	Arafat (T103), Laden (T114), Souder (T118), Freeman (T119), Pope (T123)
Adaboost Classifier	Helicopter (T105), Rocket (T107), Flame (T112)
Multi Feature	Aerial City (T100), Bench logo (T108), Tank (T109), Diving (T110) Locomotive (T111), Car (T115), People (T117)
Motion	Airplane (T104)
Color Texture	Cat (T122)
Color Region	Basketball (T101), Baseball (T102), Tomb (T106), Snow mountain (T113), Sphinx (T116), Dow Jones (T120), Coffee (T121), White House (T124)

5.5 High-level Feature Confidence Combination

It is clear that high-level feature detection results sometimes can help the search. Therefore, in four searching systems above, user can specify what high-level features are probably useful for a topic. Multiplying the confidence of these features with the confidence generated by searching system, final confidence for each shot is gotten. Except for 17 high level features of TRECVID2003, we also extract some additional high-level features: Basketball, Grass Sport and Motion (Include still, move, pan, tilt, zoom and rotate of different directions.) Table 4 lists the high-level features selected by user for each topic.

Table 4. High-level features selected by user in Search Task

100	Building / Road	101	Basketball	102	Grass Sport
103	News Subject Face	104	Aircraft / Pan	105	Aircraft
106	Outdoors	107	Outdoors / Tilt_Up	108	Car_Truck_Bus
109	Outdoors	110	Sporting Event / Tilt_Down	111	Outdoors / Pan
112	Outdoors	113	Outdoors	114	News Subject Face
115	Car_Truck_Bus / Road	116	Outdoors	117	Outdoors / People
118	News Subject Face	119	News Subject Face	120	---
121	---	122	Animal	123	News Subject Face
124	Outdoors / Building				

Evaluation results show that high level features can improve the average precision a lot because it can remove some false alarms.

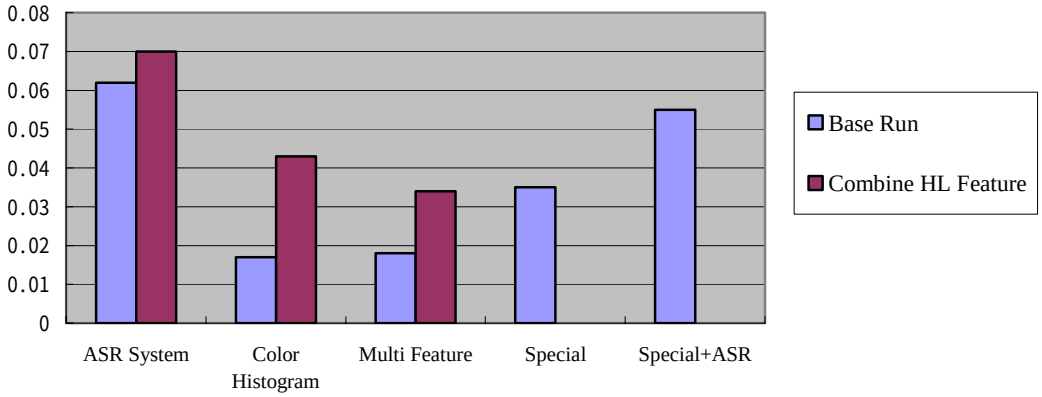

Figure 7. Comparison between Base run and High-level Feature Run (Mean Average Precision)

NIST's evaluation shows that our searching system is not efficient for several topics, especially for those searching an object. In the future work, we should pay more attention on region-based features and image segmentation.

6. Conclusion

This is the third time we participate in TRECVID Evaluation.

In Shot Boundary Determination Task, the performance of our system decreases a lot on this year's dataset. In the future, we should concentrate on the robustness so that it can work well on real video data.

Story Segmentation and Classification is a new task of TRECVID. From the evaluation results, we find visual information really helpful. We also tried some methods, such as maximum entropy model, to fuse visual, audio and text information. However, effective feature fusion is still an open problem because several methods proposed before are not so effective on real news video data.

In High-level Feature Extraction task and Search task, we extract some low level features based on whole image. Generally, objects are more interesting than whole images to users. In the future, we should pay more attention on the region. On other hand, we have not use motion information in most of tasks. So much work is based on keyframe. We hope motion information can improve our system in the future.

From the evaluation results of other participants, we gladly find an inspiring improvement on several tasks in recent 3 years. We believe research in this area will continue improving in the future.

ACKNOWLEDGMENTS

This research is supported by NSF of China under contract of 69935010.

Reference

[Feng03] Zhe Feng, Yaqian Zhou, Lide Wu, Zongge Li, “Audio Classification Based on Maximum Entropy Model”, *Proceeding of the 4th International Conference on Multimedia and EXPO*, 2003

[Freund96] Y. Freund, R. E. Schapire, “Experiments with a new boosting algorithm”, *Proc. of International Conference of Machine Learning*, pp.148-156, 1996

[Gauvain02] J. L. Gauvain, L. Lamel, G. Adda, “The LIMSI Broadcast News Transcription System”, Speech Communication, Vol 37: 1-2, pp.89-108, 2002. ftp://tlp.limsi.fr/public/spcH4_limsi.ps.Z

[Guo03] Yuefei Guo, Shijin Li, Jingyu Yang, Tingting Shu, Lide Wu, “A generalized Foley-Sammon transform based on generalized fisher discriminant criterion and its application to face recognition”, Pattern Recognition Letters, Vol 24:1-3, pp. 147-158, 2003

[Haralick73] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 3, No. 6, pp. 610-621, 1973

[Hu02] Jianming Hu, Jie Xi, Lide Wu, “Automatic Detection and Verification of Text Region in News Video Frames”, International Journal of Pattern Recognition and Artificial Intelligence, Vol.16:2, pp. 257-271, 2002

[Huang01] Xuanjing Huang, Lide Wu, Guowei Xu, Hiroyuli Ishizaki, “Language Independent Text Categorization”, *Proceeding of 6th Natural Language Processing Pacific Rim Symposium*, 2001

[Schapire98] Schapire R. E., Freund Y. et. al., “Boosting the margin: a new explanation for the effectiveness of the voting methods”, *Annual Statistics*26(5), pp.1651-1686, 1998

[Schneiderman02] H. Schneiderman and T. Kanade, “Object Detection Using the Statistics of Parts”, *IJCV*, 2002

[Viola01] P. Viola and M.J. Jones, “Robust real-time object detection”, *Technical Report Series*, Compaq Cambridge Research Laboratory, CRL 2001/01, 2001

[Wu02] Lide Wu et al., “FDU at TREC-11: Filtering, QA, Web and Video Tasks”, *Proceeding of the 11th Text Retrieval Conference*, pp.227-232, 2002