CLIPS-LIS-LSR-LABRI experiments at TRECVID 2004

Georges M. Quénot[1], *Daniel Moraru*[1], *Stéphane Ayache*[1],
Mbarek Charhad[1], *Mickaël Guironnet*[2], *Lionel Carminati*[4], *Philippe Mulhem*[1]
Jérôme Gensel[3], *Denis Pellerin*[2] *and Laurent Besacier*[1]

[1] CLIPS-IMAG, BP53, 38041 Grenoble Cedex 9, France
[2] LIS, B46 avenue Feélix Viallet, 38031 Grenoble Cedex, France
[3] LSR-IMAG, BP53, 38041 Grenoble Cedex 9, France
[4] LABRI, 351, cours de la Libération, 33405 Talence Cedex, France

Georges.Quenot@imag.fr

Abstract

This paper presents the systems used by CLIPS-IMAG and its partners, LSR-IMAG, LIS and LABRI laboratories, to perform the tasks proposed in the TRECVID 2004 workshop. SBD was performed using a system based on image difference with motion compensation and direct dissolve detection. This system gives control of the silence to noise ratio over a wide range of values and for an equal value of noise and silence (or recall and precision), the F1 value is 0.83 for all types of transitions. Story segmentation was performed using a combination of multi-modal detectors and the F1 value for the optimal system configuration was 0.48. Feature extraction was achieved using a combination of lexical context based classification, a color and texture based classification and face recognition. The search system uses a user controlled combination of five mechanisms: keywords, similarity to example images, semantic categories, similarity to already identified positive images, and temporal closeness to already identified positive images. The mean average precision of the system (with the most experienced user) is 0.24.

1 Introduction

The CLIPS-IMAG laboratory and his partners, LSR-IMAG, LIS and LABRI laboratories have participated to the four tasks proposed at the TRECVID 2004 workshop.

2 Shot Boundary Detection

The system used by CLIPS-IMAG to perform the TRECVID SBD task is almost the same as the one used for the previous TREC video evaluations [2][1][15] This system detects “cut” transitions by direct image comparison after motion compensation and “dissolve” transitions by comparing the norms of the first and second temporal derivatives of the images. It also contains a module for detecting photographic flashes and filtering them out as erroneous cuts and a module for detecting additional cuts via a motion peak detector. The precision versus recall or noise versus silence tradeoff is controlled by a global parameter that modifies in a coordinated manner the system internal thresholds. The system is organized according to a (software) dataflow approach and Figure 1 shows its architecture.

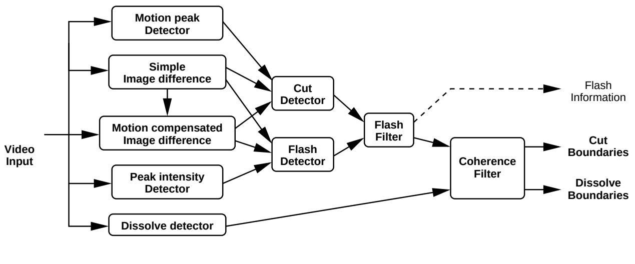


Figure 1: Shot boundary detection system architecture

The original version of this system was evaluated using the INA corpus and the standard protocol [3] (http://clips.imag.fr/mrim/georges.quenot/-OT10.3/aim1/) developed in the context of the GT10 working group on multimedia indexing of the ISIS French research group on images and signal processing.

This test protocol was partly reused (with different test corpora) for the TREC-10, TREC-11 and TRECVID SBD tasks. The reference segmentation for test collections of the TRECVID 2004 corpus was built with the TRECVID 2003 version of this system.

Very little modification was made relatively to the previous version of the system, only minor adjustments of control parameter. The main additional work was an attempt to get a precise control of the noise to silence ratio.

2.1 Cut detection by Image Comparison after Motion Compensation

This system was originally designed to evaluate the interest of using image comparison with motion compensation for video segmentation. It has been complemented afterward with a photographic flash detector and a dissolve detector.

2.1.1 Image Difference with Motion Compensation

Direct image difference is the simplest way for comparing two images and then to detect discontinuities (cuts) in video documents. Such difference however is very sensitive to intensity variation and to motion. This is why an image difference after motion compensation (and also gain and offset compensation) has been used here.

Motion compensation is performed using an optical flow technique [4] which is able to align both images over an intermediate one. This particular technique has the advantage to provide a high quality, dense, global and continuous matching between the images. Once the images have been optimally aligned, a global difference with gain and offset compensation is computed.

Since the image alignment computation is rather costly, it is actually computed only if the simple image difference with gain and offset compensation alone has a large enough value (i.e. only if there is significant motion within the scene). Also, in order to reduce the computation cost, the differences (with and without motion compensation) are computed on reduced size images (typically 88×60 for the NTSC video format). A possible cut is detected if both the direct and the motion compensated differences are above an adaptive threshold.

In order for the system to be able to find shot continuity despite photographic flashes, the direct and motion compensated image difference modules does not only compare consecutive frames but also, if needed, frames separated by one or two intermediate frames.

2.1.2 Photographic flash detection

A photographic flash detector feature was implemented in the system since flashes are very frequent in TV news (for which this system was originally designed for) and they induce many false positives. Flash detection has also an interest apart from the segmentation problem since shots with high flash densities indicates a specific type of event which is an interesting semantic information.

The flash detection is based on an intensity peak detector which identify 1- or 2-frame long peaks on the average image intensity and a filter which uses this information as well as the output of the image difference computation modules. A 1- or 2-frame long flash is detected if there is a corresponding intensity peak and if the direct or motion compensated difference between the previous and following frames are below a given threshold. Flash information is used in the segmentation system for filtering the detected cut transitions.

2.1.3 Motion peak detection

It was observed from TREC-10 and other evaluations that the motion compensated image difference was generally a good indicator of a cut transition but, sometimes, the motion compensation was too good at compensating image differences (and even more when associated to a gain and offset compensation) and quite a few actual “cuts” were removed because the pre- and post-transition images were accidentally too close after motion compensation. We found that it is possible not to remove most of them because such compensation usually requires compensation with a large and highly distorted motion which is not present in the previous and following image-to-image change. A cut detected from simple image difference is then removed if it is not confirmed by motion compensated image difference *unless* it also corresponds to a peak in motion intensity.

2.2 Dissolve detection

Dissolve effects are the only gradual transition effects detected by this system. The method is very simple: a dissolve effect is detected if the L_1 norm (Minkowski distance with exponent 1) of the first image derivative is large enough compared to the L_1 norm of the second image derivative (this checks that the pixel intensities roughly follows a linear but non constant function of the frame number). This is expected to detect dissolve effects between constant or slowly moving shots. This first criterion is computed in the neighborhood (± 5 frames) of each frame and a filter is then applied (the effect must be detected or almost detected in several consecutive frames).

2.3 Output filtering

A final step enforces consistency between the output of the cut and dissolve detectors according to specific rules. For instance, if a cut is detected within a dissolve, depending upon the length of the dissolve and the location of the cut within it, it may be decided either to keep only one of them or to keep both but moving one extremity of the dissolve so that it occurs completely before or after the cut.

2.4 Global tuning parameters

The system has several thresholds that have to be tuned for an accurate detection. Depending upon their values, the system can detect or miss more transitions. These thresholds also have to be well balanced among themselves to produce a consistent result. Most of them were manually tuned as the system was built in order to produce the best possible results using development data.

For the TREC-11 and following evaluations, as well as for other applications of the system, we decided to have all the threshold parameters be a function of a global parameter controlling the recall versus precision tradeoff (or, more precisely, the silence to noise ratio). We actually used two such global parameters: one for the cut transitions and one for the gradual transitions. A function was heuristically devised for each system threshold for how it should depend upon the global parameters.

Ten values were selected for the global parameters. These values were selected so that they cover all the useful range (outside of this range, increasing or decreasing further the global parameter produces a loss on both the silence and noise measures) and within that range they set targets on a logarithmic scale for the silence to noise ratio.

2.5 Results

Ten runs have been submitted for the CLIPS-IMAG system. These correspond to the same system with a variation of the global parameter controlling the silence versus noise (or precision versus recall) tradeoff.

Table 2.5 shows the performance of the system for the tradeoff values selected for the evaluation. The CLIPS-IMAG system appears to be quite good for gradual transitions both for their detection and location. This indicates that the chosen method (comparison of the first and second temporal derivative of the images) is quite good even if theoretically suited only for sequences with no or very little motion.

All		Cut		Gradual		Frame	
Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.
.894	.446	.927	.435	.824	.474	.749	.596
.887	.614	.925	.610	.805	.623	.757	.700
.877	.732	.921	.743	.784	.704	.761	.764
.857	.806	.900	.834	.765	.744	.754	.792
.819	.851	.858	.874	.737	.798	.745	.813
.777	.883	.809	.904	.709	.835	.736	.825
.698	.912	.718	.931	.655	.871	.723	.839
.606	.932	.605	.943	.608	.910	.716	.851
.491	.942	.474	.952	.528	.924	.731	.868
.312	.948	.241	.952	.461	.943	.725	.885

Table 1: Results for Shot Boundary Detection

3 Story Segmentation

3.1 Introduction

Among the different TRECVID tasks, the story segmentation task is defined as: given a test collection, identify the story boundaries with their location (time) and optionally their type (miscellaneous or news) in the given video clip(s), see [17] for details.

We describe here the multi-modal features used and their respective performance for the story segmentation task. These features are based on the audio, video and text modalities. The preliminary system, which has the advantage to be relatively free with respect to the use of training data, is also presented.

3.2 Multi-modal Features

Our approach to story bound detection is to use a range of different feature detectors and in this section we describe each of them in turn. Their measured performance are given on the TRECVID 2003 test set to illustrate their relative performance. These are gathered in Table 2 in section 3.4. The evaluation metrics used are also detailed in section 3.4.

3.2.1 Long pauses detection

A silence detection is applied on the audio channel. It is only based on an energy bi-Gaussian distribution and on a detection threshold between the two Gaussians. The silence segment minimal length is set to 1 second in order to only catch relatively long silence segments. It is interesting to note that this basic feature alone is already interesting for story segmentation. We have tested it on the reference boundaries and found its F1 measure to be **0.44**, when all the long pauses were assigned to a boundary in the story segmentation system output.

3.2.2 Shot boundary detection

Shot boundaries have been detected using the system described in section 2. The recall versus precision global control parameter has been set to obtain a high recall value.

Using this feature alone lead to a F1 measure of **0.25** with a recall of 0.934. This recall result, different from 1, confirms that a single video shot can contain multiple story boundaries. Thus, selecting all the shot boundaries as candidate points for story boundaries is not sufficient. Therefore, we take the union of shot boundaries and long pauses as candidate points for the story segmentation task, but we remove duplications within a 5s fuzzy window. A similar proposal was made in [12] and yielded 100recall rate. Our union however leads only to 0.963 recall rate.

3.2.3 Audio change detection

Audio change detection may be a useful feature since many story boundaries correspond to an audio change on the audio channel. Examples of "audio change" are: speaker changes, speech to music transitions, speech to speech-over-music transitions, etc... These audio changes can be automatically obtained by detecting abrupt changes on the audio channel.

At the moment, the CLIPS-IMAG audio change system is based on a BIC Bayesian Information Criterion (BIC) [8] detector. It is important to note that the BIC criterion has been often used for speaker change detection whereas it should be able to detect any other abrupt change on the audio signal. Thus, we called our feature "audio change detection" instead of "speaker change detection", even if a large part of the changes found with the BIC criterion may actually be speaker changes.

The signal is characterized by 16 mel cepstral features (MFCC) computed every 10ms on 20ms windows using 56 filter banks. Then the cepstral features are augmented by energy. No frame removal or any coefficient normalization is applied. The idea of the audio change detection is to find audio signal discontinuities that will help us to distinguish between two consecutive audio sources (speech followed by music ; speaker X followed by speaker Y ; ...). We can use two adjacent windows and a similarity measure between them. For the similarity measure we use the Bayesian Information Criterion.

In order to apply the BIC we consider that the sound signal is a Gaussian process in the space of acoustic parameters. This kind of approach is based on the decision theory. Let us consider two consecutive segments of speech, each of them being characterized by a sequence of spectral acoustic parameters (ex: coefficients MFCC, LPCC, etc) denoted by x_n $(n = 1..N_1)$ and respectively by y_n $(n = 1..N_2)$. We suppose that every sequence could be modeled by a multidimensional Gaussian distribution and that the vectors are statistically independent.

The question that we are asking regarding the two consecutive sequences is: do they belong or not to the same fundamental model or do both sequences correspond to the same acoustic source or not.We must test the next two hypotheses:

H_0 : the two sequences correspond to the same acoustic source,

H_1 : the two sequences correspond to two different acoustic sources.

We can evaluate these two hypotheses using the generalized likelihood ratio. We will compute the ratio using maximum likelihood estimated models for the two sequences. Let's say that $L(x; \mu_1, \Sigma_1)$ is the probability that the sequence x was generated by the Gaussian model characterized by the mean vector μ_1 and covariance matrix Σ_1, and $L(y; \mu_2, \Sigma_2)$ is the same probability for the sequence y; then the probability L_i of the two sequences being generated by two different models is:

$$L_1 = L(x; \mu_1, \Sigma_1).L(y; \mu_2, \Sigma_2) \tag{1}$$

The probability of the two sequences being generated by the same model is:

$$L_0 = L(z; \mu, \Sigma) \tag{2}$$

where z is the joint sequence of x and y, and μ and Σ are the parameters of the model estimated from sequence z. If we consider that λ is the generalized likelihood ratio, then $\lambda = L_0/L_1$ and if we use log-likelihood $R = -\log \lambda$ then we have:

$$R = \log L(x; \mu_1, \Sigma_1) + \log L(y; \mu_2, \Sigma_2) - \log L(z; \mu, \Sigma) \tag{3}$$

It was proved that for mono-Gaussian distributions we have:

$$R = \frac{N_1 + N_2}{2} \log \mid \Sigma \mid - \frac{N_1}{2} \log \mid \Sigma_1 \mid - \frac{N_2}{2} \log \mid \Sigma_2 \mid \tag{4}$$

We compute the ratio for all available data and we obtain a sequence of values R(t). The estimate of the audio change point is the value that maximizes R(t).

$$\hat{t} = \arg\max_t R(t) \tag{5}$$

Looking for audio change points means looking for maximum points of the curve R(t), called the BIC curve.

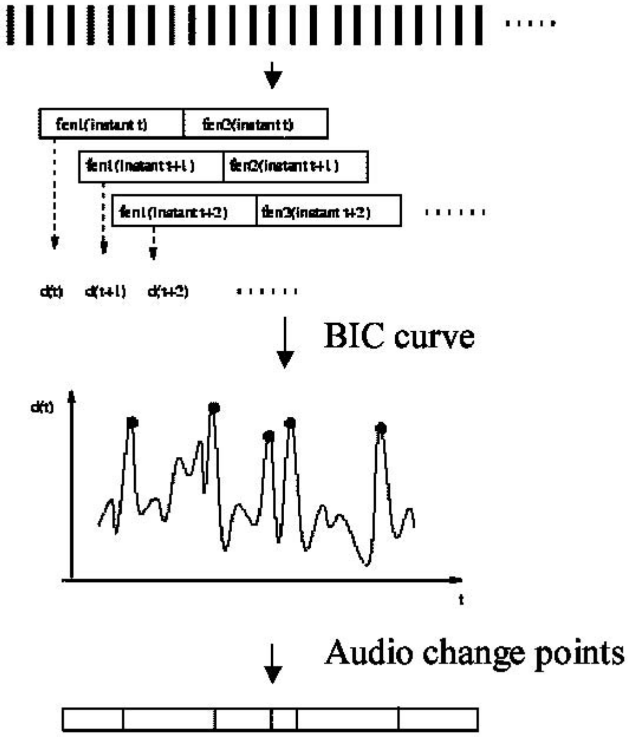


Figure 2: The audio change detection process

Figure 2 illustrates the whole audio change detection process. To select the maximum points of the BIC curve we use a sliding window that goes along the curve. The window is centered on the potential speaker change point. The point is selected if it has the highest BIC value in the window and if its BIC value is superior to 1.3 x average BIC curve value. The size of the window is 0.3 seconds. The use of the average BIC curve value gives us a data independent threshold. The use of the sliding window selects only the highest maximum among multiple close maximum points giving us a better precision.

For the story segmentation task, this feature alone gives a F1 score of **0.29** with 0.78 recall rate. That confirms our hypothesis that many story boundaries correspond to audio changes, but of course there are much more audio changes than story boundaries which explains the relatively low precision rate obtained with this feature alone (0.18).

3.2.4 Speaker segmentation

From the list of informative features that are provided in the ASR transcript, speaker information is available: for each speaker turn, a speaker label is assigned. This kind of output is generally called speaker segmentation [7][13].

This speaker segmentation output may be useful: for instance, since we do not have yet a visual anchor face detector at CLIPS, a complete speaker segmentation output could be interesting to retrieve anchor person shots which are known to be very useful for story segmentation [12].

To illustrate the interest of speaker segmentation, Figure 3 shows as example the speaker segmentation of a complete 30mn video file, that can be obtained from the LIMSI ASR XML files. Each line corresponds to a speaker occurring on the audio channel.

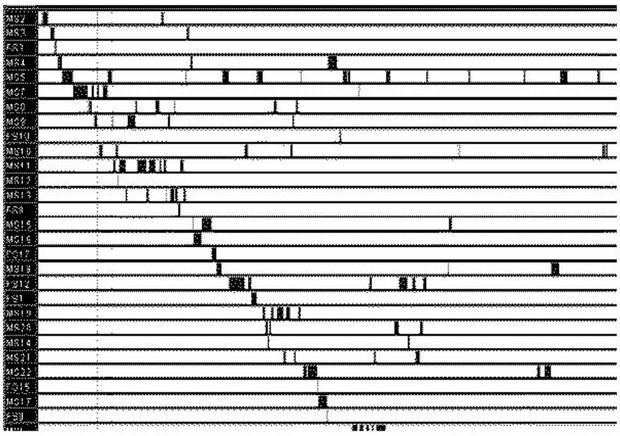


Figure 3: Speaker segmentation of a complete 30mn video

Each speaker intervention is given by the black segments on each line. We clearly see here that most of the speakers occur occasionally and on a limited period, except for the news presenter (on the 5th line) whose segments are spread over the whole video file. Thus, such segmentation output is interesting for finding the anchorperson shots without using the image channel. Of course, an automatic segmentation makes some errors on the speaker interventions. For instance, best speaker segmentation systems obtain around 15% of speaker segmentation error on broadcast news data, as shown in [13] and [16].

From the speaker segmentation output obtained with LIMSI ASR files, we extract automatically the news presenter line. There is no speaker training data available for any of the news presenters. The extraction is entirely based on the following empirical rules:

- The news presenter always speaks in the first five minutes of a broadcast news document.
- The news presenter interventions are spread over the entire news document. This is different from a usual reporter who speaks only during his news story.
- Finally, the news presenter is generally the main speaker (the speaker who has the highest total speech duration).

The first rule is applied as is, meaning that we first select all the speakers that speak during the first five minutes as "presenter candidates". For the second rule we split the audio document in fixed length intervals (3 minutes each). For each speaker we count the number of intervals where he actually does not speak. We select the speaker(s) with the minimum number of "no speech" intervals.. Finally if at this point we still have more than one speaker as a potential news presenter, the final criterion is the quantity of uttered speech (third rule).

When we have extracted the news presenter line, we know the start and the end of each of his interventions. The story boundaries then correspond to each start point of the news presenter interventions. On the contrary, the end points of the news presenter interventions correspond generally to a reporter start inside the same story ; thus, these end points are used as "anti-story-boundaries" to remove story detection points that could be found by other multi-modal features at the same time.

For the story segmentation task, this feature alone gives a F1 score of **0.32** with 0.205 recall rate and 0.702 precision rate. It is not as efficient as a complete anchor face detector (the one from [12] leads to 0.51 F1 score on CNN data for instance), but this "presenter information", extracted only from the audio channel helps to extract the general structure of a video document. It could also be very useful for story segmentation on radio broadcast, for instance, where no video channel is available.

3.2.5 Jingle detection

Detecting some key sounds (so called jingles) on the audio channel can reveal the beginning or the end of a particular sequence or announce it. This may be useful for the story segmentation part. Though most jingles include music, our jingle detector is not a music detector.

To detect and locate such jingles we used only one example of each jingle, taken on a separate video set. Each reference jingle was described by low level descriptors based on a spectral analysis while dissimilarity was measured between the target jingles and the whole video test set with an Euclidian distance, as done in [14] for instance.

More precisely, our low level descriptors were 8 coefficients corresponding to the spectral flatness feature computed on 8 frequency bands. This spectral flatness feature is part of the MPEG-7 low level description and has shown to give interesting results for audio fingerprinting [10].

We have selected 10 jingles from ABC and CNN (CNN headline news jingles, CNN top stories, CNN sport, ABC short jingles, etc...) which have a length between 2s and 10s.

Of course, for the story segmentation task, this feature gives very few boundaries which results in a very low recall rate when used alone (0.028). However, the boundaries obtained generally correspond to effective story changes, since the precision obtained with this feature alone is 0.735.

3.2.6 ASR Text output

The LIMSI laboratory provided to all participants of TRECVID the output of their automatic speech recognition (ASR) system [9] for the whole 2003 and 2004 database. Our ASR-based feature was based on the selection of a set of lexical sequences likely to correspond to story transitions. To extract our list of lexical sequences, we calculated on the development data the most frequent N-grams (N=1 to 5) computed from ASR outputs located around reference story boundaries. From this, we manually made a list of 27 transition word sequences. Examples of word sequences extracted are : "A. B. C. News", "C. N. N.", "Just before we leave", "Back with more news", "Coming up in two minutes", ... This feature is rather similar to the "cue phrases" proposed and used in [11].

To find the story boundaries using the ASR output, we have selected all the speaker turns containing at least one of our "transition word sequences". Then, the story boundaries were obtained by selecting the beginning or the end of each selected speaker turn according to the transition sequence concerned.

The use of this feature alone gives a F1 score of **0.41** with a relatively good precision (0.73).

3.3 System Overview

3.3.1 Candidate Points

A good candidate set should have a very high recall rate on the reference boundaries. As seen previously, we decided not to use only shot boundaries, but the union of shot boundaries and long pauses which lead to 0.963 recall rate.

3.3.2 Overall strategy

At the moment, our strategy is very basic, but it has the advantage to be free of any development set which is not the case when some SVM-based combination schemes are used, for instance. It could however benefit from training when there is time and opportunity for it. This will be the case for additional features currently considered.

The general idea is to evaluate, for each candidate point, the output of each separate detector (described in section 2) which indicates the presence or not of a story boundary. For audio change (AC) and Pauses (P) features, a boundary is considered to be detected if it is found inside a 2s fuzzy window around a candidate point. For ASR, Speaker Segmentation (SS) and Jingles (J), a boundary is considered to be detected if it is found inside a 4s fuzzy window around a candidate point. Then, the combination of features is based on logical operations between each separate detectors. For instance, in Table 3.4, (AC $\wedge$ P) $\vee$ J means that a candidate point is considered to be a story boundary if one of the following cases is encountered:

- the audio change and pause detectors both found a boundary around it,
- the jingle detector found a boundary around it.

3.4 Results

3.4.1 Story boundary detection metric

We used the official TRECVID precision P and recall R measures for the story segmentation task. Since there is no ranking considered in this task, it is not possible to compute the classical Mean Average Precision (MAP) for system ranking. Since there are two values with very variable P versus R tradeoffs between system, it is not easy to compare systems. In order to obtain a single measure to permit such comparison and ranking, we chose the classical F-measure (harmonic mean between P and R) and, more precisely, the F1 measure (giving equal weight to P and R in the mean).

3.4.2 Characterization on TRECVID 2003

We have made these experiments using the data and methodology proposed by TRECVID but we did so after the official evaluation period and the results presented here were not submitted to NIST. Therefore our results should not be directly compared to the official TRECVID 2003 official results because a) we would compare our system to systems that are older and could have evolved in the interval and b) though we have followed the methodology and we have made an appropriate use of development and test data respectively, we had knowledge of the results of other systems (which feature worked and didn't work for instance).

The story boundary detection performance on the TRECVID 2003 test set for the different detectors alone, as well as for their combination with logical operators are given in Table 2.

	Rec.	Pre.	F1
Pauses (P)	0.613	0.344	0.44
Shots (S)	0.934	0.142	0.25
P $\vee$ S (candidate points)	0.963	0.146	0.25
Audio Change (AC)	0.782	0.176	0.29
Speaker Segmentation (SS)	0.205	0.702	0.32
Jingles (J)	0.028	0.735	0.05
ASR	0.280	0.734	0.41
AC $\wedge$ P	0.495	0.382	0.43
(AC $\wedge$ P) $\vee$ J	0.516	0.394	0.45
(AC $\wedge$ P) $\vee$ SS $\vee$ J	0.567	0.405	0.47
(AC $\wedge$ P) $\vee$ SS $\vee$ J $\vee$ ASR	0.616	0.450	0.52
(AC $\wedge$ P) $\vee$ SS $\vee$ J $\vee$ ASR + commercials detection	0.613	0.467	0.53

Table 2: Story boundary detection performance on TRECVID 2003 evaluation data (105*30mn files)

The comments concerning the detectors alone are to be found in section 3.2. The association of audio change and pause feature (AC $\wedge$ P) is slightly disappointing since it leads only to a F1 score of **0.43** which is approximately the same performance as the pauses used alone. However, we kept the boundary points found because the precision is improved in that case. Adding the jingle detector (AC $\wedge$ P) $\vee$ J improves the overall performance which shows the interest of this detector. It seems to be able to find boundaries that are not redundant with the boundaries found by other detectors. Adding now the news presenter information obtained from speaker segmentation (AC $\wedge$ P) $\vee$ SS $\vee$ J improves again the overall performance since we reach 0.567 recall and 0.405 precision rate. It is also interesting to note that the association of these 4 features (AC, P, SS and J) leads to a system with acceptable performances without using the ASR text output (this corresponds to condition 1 of TRECVID 2003 evaluation plan).

If we add the ASR-based boundary detector, we reach a F1 score of **0.52**. At this point, an analysis of the errors shows some false alarms occurring during commercial sequences. We have done a final experiment to detect and remove candidate points which are inside a sequence of commercials. The sequences of commercials were detected by applying a black frames detector on the video channel, since we noticed that commercials are generally separated by a variable number of consecutive black frames. This final process allowed to slightly increase the precision rate, leading to a F1 score of **0.53**.

3.4.3 Results on TRECVID 2004

Table 3 shows the results of two variants of the above described segmentation systems for the three tests conditions specified for the task (removing when applicable some features in the logical combination). A small loss is observed between the test on TRECVID 2003 (F1 at 0.52) and the test on TRECVID 2004 (F1 at 0.48) in similar conditions. The system was not trained on the TRECVID 2003 test set and the difference may come from thefact that the TRECVID 2003 development collection is a bit closer to the TRECVID 2003 test collection than to the TRECVID 2004 test collection.

System	cond.	Rec.	Pre.	F1
Primary	1	0.539	0.404	0.46
Primary	2	0.585	0.407	**0.48**
Primary	3	0.265	0.677	0.38
Secondary	1	0.539	0.400	0.46
Secondary	2	0.585	0.403	0.48
Secondary	3	0.265	0.676	0.38

Table 3: Story boundary detection performance on TRECVID 2004 evaluation data (128*30mn files)

3.5 Future work

In the near future, we notably plan to use our own speaker segmentation system [7][13] instead of the LIMSI one and to improve our commercial detection system in order to reduce false alarms. We also plan to include more features from the image track and from ASR analysis. We are currently developing a multi-modal story classification tool (politics, sports, weather, commercials, ...) and to integrate story segmentation and story classification together with a feedback to each other. We finally consider the integration of external feature detectors (developed elsewhere than at CLIPS) and the use of a more flexible, more analog (non-Boolean) and less ad'hoc fusion procedure.

4 High-Level Feature Extraction

High-level features were extracted in three different methods and the output of the various detectors were merged. The first method used the ASR transcript provided by LIMSI to build lexical models for the features (CLIPS feature detector). The second method classified the reference key frames using a SVM on color and texture vector descriptors (LIS feature detector). The third method was used only for the person features and used a face detector followed by a face classifier (LABRI feature detector). Two simple strategies were used for the fusion of detector outputs: the estimated relevance of two outputs are either multiplied or linearly combined to produce the final relevance.

4.1 Lexical context

Detection of high-level features in video documents is usually done by categorizing key images from signal information. These approaches use low-level extraction process for color, texture and motion features and a supervised learning phase such as KNN, SVM, NN methods. The speech flow can also help to distinguish categories since the different classes are semantically far enough. Existing approaches show that lexical context perform well for emotion detection ‘citeDevi04 and topic classification [19]. Thus, we have developed and experiment a classifier based on a lexical analysis of speech transcription from LIMSI.

Since our approach is supervised, we must predefine a set of classes and build a model for each one in 3 steps:

- Extract text from ASR around apparitions of visual feature: In order to catch lexical context of visual features, we define temporal offsets around apparition of a shot containing visual features. We choose offsets for each class by computing cross validation in the development data.
- Textual analysis: This process tags the text using our specific knowledge base by finding named-entities, or applying stemming and stop- lists. We also define a set of entities referring the same concept, such as “Madeleine Albright” and “secretary

of state", or very closed entities such as "train" and "locomotive".

- Compute probability pwe of each term, entity, or concept w being in the class e.

Learning a semantic class by lexical analysis aim to perform a co-occurrence like process between semantic and lexical information. In this way, the following lines are the top 5 entries in the "Madeleine Albright" model, with offsets 4.5 and 1.5, respectively before and after apparition of visual feature:

0.042693	Madeleine Albright
0.032841	United States
0.026273	Iraq
0.016420	Balkans
0.013136	Saddam Hussein

Next, during the detection process, our system extract textual information aligned with the shot bounds and offsets, find named-entities and assigns a score value Vse for each shots s being in semantic class e:

$$V_{se} = \frac{\Sigma_{w \in s | p_w > \gamma} \log \frac{\lambda p_w e + (1-\lambda) p_w}{p_w}}{\Sigma_{w \in s | p_w > \gamma} 1} \qquad (6)$$

Where p_w is the probability of a word w to be in the general model (computes on development set). According to the Zipf law, we define a threshold γ and only terms that have $p_w > \gamma$ are computed. We define γ experimentally using cross validation. We notice that γ value depends of the kind of the semantic class.

A shot could be in several classes since the classes are not exclusive. For instance, it is possible that Madeleine Albright and Bill Clinton occur together in a shot.

4.2 Key frame classification

4.2.1 Extraction of color and texture

It is well-known that color and texture are visual cues in the classification of images. Color is the most widely used feature in content based retrieval and texture is an important feature in the perception of images. Moreover, color indexing methods are limited to retrieve images which have a similar color composition as the query image but they can have a completely different content. Color indexing is then combined with texture indexing methods.

Color feature

Among the color descriptors, we retain color histogram which offers a great simplicity. A color histogram captures global color distributions in an image. Selected color space is YCbCr space, which is used in compression MPEG. However, we do not use an uniform quantization of the color space which gives the same weight to the pixels near the centre of a bin as those that are located at the edges. The use of the fuzzy sets makes it possible to associate a membership degree at each pixel according to each bin. Each dimension of YCbCr following is quantified following figure 4 and a fuzzy 3D histogram with 8×8×8 components is computed.

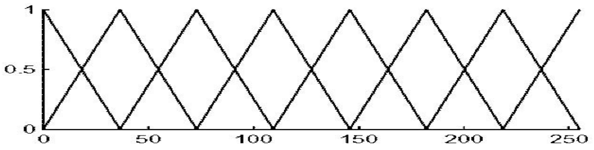


Figure 4: A membership degree of pixels according to each bin of one dimension

Texture feature

First, a retinal filter [20] is applied on each key frame. The photoreceptors of the retina perform an adaptive compression of brightness intensity. This adaptation leads to provide more dynamic for the values corresponding to dark-colored zones. Then, the neuronal circuits carry out high-pass filtering, which corresponds to a spectral whitening because of $1/f$ image amplitude spectrum. Finally, the filtering enhances the local variations of contrast and the details.

Some cells of the primary visual cortex are sensitive to stimuli having a certain orientation and a certain frequency in a specific position of the visual field; we modeled this using two-dimensional Gabor function. A Gabor filter is defined like a Gaussian with spatial extent s_x and s_y modulated by a complex exponential with frequency f_k in a direction θ_i. We chose 7 frequency bands $f_k = 2^k f_0$ and 7 orientations $\theta_i = i\pi/7$ (figure 5).

We carried out this filtering by directly multiplying the retina output with the Gabor filter in the Fourier domain. Before achieving the Fourier transform, we multiplied it by an Hanning window to remove edge effects. Finally, we obtain maps $E(f_k, \theta_i)$ depending on the frequency and the orientation. We carry out a normalization of the characteristic vector (49 dimensions) as described in [21]. The blur is an isotropic function of the $G(f)$ frequency and the normalization carried out by frequency band removes this term. Each key frame is characterized by a matrix 7×7 which corresponds to energy according to an orientation and a frequency.

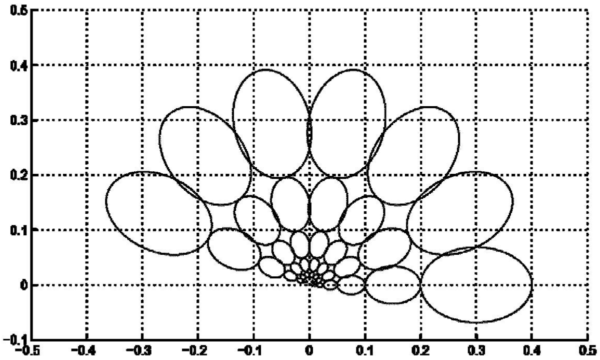


Figure 5: Bank of 49 Gabor filters

4.2.2 Classification

First, Principal Component Analysis (PCA) has been used to reduce feature dimensionality. We reduce the number of components from 512+49 to 128. Then, we apply SVM (Support Vector Machine) to learn each TRECVID concept. SVM is successfully used in a variety of pattern recognition tasks. We use SVM in the binary classification. Let $\{x_1...x_n\}$ be a set of training data which are feature vectors of labeled images. We are also given their labels $\{y_1...y_n\}$ where $y_i \in \{-1,+1\}$. SVM are simply hyperplanes that separate the training data by a maximal margin. A ground truth of each concept is carried out on TREC 2003 development set. Finally, this classifier is applied on the key frames of the 2004 test set.

4.3 Face recognition

We resort to Support Vector Machines (S.V.M.) because they offer state-of-the-art capabilities in the context of supervised detection and recognition. Their effectiveness resides in part in the manner they address the fundamental issue of generalization [22].

In [23], LABRI applied M.I.T. SVM quadratic optimized classifier [24] for the face detection problem in video at various scales. We proposed also a set of pre-process tools in order to alleviate problems of illumination variations and noise produced by background for example. In TREC Vid 2004 feature extraction task we use and train a S.V.M. for Bill Clinton and Madeleine Albright's face recognition problem. CLIPS-IMAG used OpenCV [25] face detector to supply face detection results. The performances of S.V.M. classifier are dependent of the quality of the input data with regard to training conditions. OpenCV [25] face detector did not supplied perfectly centered on face images, thus we developed an automatic face centering process as a preliminary step to face recognition.

4.3.1 Face Detection & Recognition Cooperation

The method we suggest improves face localization by determining face location respectively to the center of the picture. We divide process into two steps: Assuming face color is modeled by a Gaussian on each RGB component, the first step, called "training step", is based on detection of face color pixels and estimation of normal distribution parameter. First the face color space segmentation is first performed by the well known K-Mean algorithm on each RGB component. Then, assuming that faces occupy the major part of the image, we consider the most representative cluster as the dominant color in thumbnails images. An Open Close filtering is next applied to regularize dominant color mask in order to alleviate false detections. Filtering gives convincingly better result of object integrity and elimination of false detections as you can see in Figure 6. All pixels values of the thumbnails in dominant cluster are finally used to determine normal distribution parameter. The Gaussian parameters were calculated on a set of 5000 pictures extracted from TREC Vid 2003 corpus. RGB Gaussian parameters deducted are $\mu_R = 180.078$ and $\sigma_R^2 = 36.68$, $\mu_G = 142.259$ and $\sigma_G^2 = 30.74$, and $\mu_B = 114.847$ and $\sigma_B^2 = 30.80$.

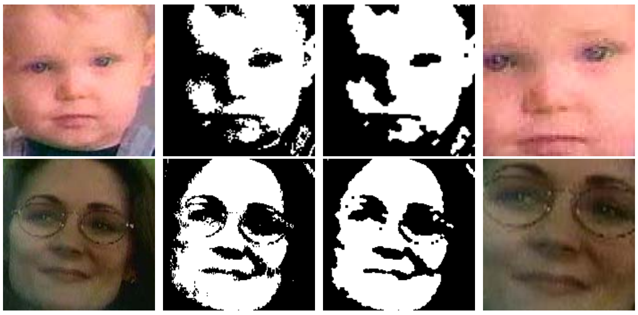


Figure 6: The picture in the left gives an example of the OpenCV Face Detection. The middle pictures represent "face-color" mask returned by 4-Means algorithms and next post-processed by Open Close Algorithm. Finally the picture in the right shows centered faces extracted from the left picture.

The second step, "generalization", classify each pixel of thumbnails images extracted from TREC Vid 2004 into two classes, "face color" and " no face-color" with estimated Gaussian parameter. We next extract the box centered on face where we get feature mainly rectangular. The new picture thus obtained is next processed by our S.V.M. Face recognition.

4.3.2 Support Vector Machine Face Recognition

Here we will not exhaustively describe S.V.M. theory, more details can be found in [22][26][27]. Applied to the problem of face recognition in video frames to Feature Extraction task, the problem can be formulated as follows. Let us consider fixed size windows selected from image signal containing or not a face of interest. The training step will consist in construction of a set of classification surfaces from labeled examples by a "one-against-all" method. The generalization step consists in classifying windows from input images into three classes: "Bill Clinton's face", "Madeleine Albright's face" and "Other face" class.

Taking into account the large variability of image content of real scenes, we chose a polynomial second order classifier in this case. The second difficulty consists in variable lightening conditions as we aim to recognize faces in a natural video stream. Therefore, a pre-processing step has been realized before training and generalization. The pre-processing consisted in two steps as follows:

- histogram equalization in order to alleviate differences in brightness between two images,

- illumination normalization. It consists of computation of mean intensity over all data base and compensation for each picture of the training and testing data set.

Training is realized by selecting and labeling $N * N$ windows on Bill and Madeleine and Other faces in full resolution video frames. The generalization step consists in classifying given picture into Bill Clinton' s face, Madeleine Albright' s face and other face class by trained classifier.

4.4 Fusion

In order to take advantage of multiple information sources we submitted runs of systems combining key frames classification, lexical analysis classification and face recognition outputs. Since this was our first participation for the TRECVID features extraction task, we used a very simple way to combine these information: after normalizing the classifiers outputs, we perform linear combination or simple product. The main goal is to obtain best results than just one classifier output, moreover, the difficulty grow when classifiers provide bad results.

4.5 Results

4.5.1 Lexical context alone

We submitted one run with the lexical analysis detector alone and we obtained with it the best results for our set of high-level features extraction submissions. Despite of the relatively low quality of our submissions, we judge acceptable and promising the accuracy of the lexical analysis approach. Especially, considering the little amount of time we had, we weren't able to tune our system for each of the features. For the rest, we are actually working on that aspect, and observe much better results.

In accordance with official results and our new experiments, we conclude that lexical analysis for detecting visual and semantic features could be appropriate, according to the contextual specificity of the feature. Features appearing in a specific context should be detected quite efficiently.

4.5.2 Face recognition alone

We evaluate first classifier on 64 videos extracted from TRECVID 2003 where we classified manually 22920 faces into our three classes containing 600 Clinton's faces and 116 Albright's faces. We divided sets of faces in two sub sets, training and testing set. Also we trained classifier with 411 Clinton's faces and 90 Albright's face and 7500 other faces chosen randomly in faces database. We obtain, after classification on remaining pictures, the results presented in table 4.5.2.

	Recall	Precision	F1
"Clinton"	0.985	0.959	0.97
"Albright"	0.962	0.898	0.93

Table 4: Performance of face recognition on properly extracted faces

These results were very encouraging but many faces of Bill Clinton & Madeleine Albright were selected from the same shot for training and thus were very similar.We test, next, classification on overall TRECVID 2004 video stream. We trained classifier with 11571 Clinton's faces, 1920 Albrigth's faces and 21632 other faces extracted manually from TRECVID 2003 video streams. Despite the encouraging results on TRECVID 2003, the final results on TRECVID 2004, given in the table 4.5.2 are much below what was expected.

There are two mains reasons for this: first of all only 10% to 35% of shot containing Bill or Madeleine's faces have been extracted by OpenCV face detector (mostly due to inappropriate face size and/or orientation). Secondly, the training set we used was too homogeneous

	Mean Average Precision
"Clinton"	0.0023
"Albright"	0.0001

Table 5: Performance of face recognition on properly extracted faces

for deducing the best inference principle. These results place us at about the two thirds in systems ranking for Bill Clinton and Madeleine Albright search.

4.5.3 Fusion

We submitted 3 runs from fusion of ASR analysis and key frames classification, and 2 runs from faces recognition and ASR analysis combination. In both cases, linear combination gives the best results, with weights 0.65 and 0.35 respectively for ASR and images analysis. These runs perform globally less than ASR only (lexical context) run, however we can notice that linear combination give better average precision for features 28, 32 and 37 (boat/ship, beach and roads) which are typically visual features.

5 Search

The CLIPS-LIS-LSR search system uses a user-controlled combination of five mechanisms: keywords, similarity to example images, semantic categories, similarity to already identified positive image, and temporal closeness to already identified positive image (Figure 7).

5.1 Keyword based search

The keyword based search is done using a vector space model. The words present in the ASR transcription are used as vector space dimensions. Stemming ans stopword list are used. Relevance is first assigned to speech segments (as provided in the LIMSI transcription [9]) and projected onto overlapping shots.

5.2 Similarity to image examples

Visual similarity between key frames and image examples is looked for using color and texture characteristics. The same primary vector descriptors than for the feature extraction task are used (8×8×8 color histograms and 7×7 Gabor transforms. Distance are computed, normalized and then turned into a relevance value for each characteristic. A 65% color and 35% texture linear combination is then used.

5.3 Feature based search

The goal of this part is to help focusing on specific categories of the video shots, according to a non-crisp labeling of their keyframes. All keyframes are automatically labeled according to 15 categories (table 6). These categories differ from the feature extraction task ones. They have been chosen because of their availability from the TRECVID 2003 collaborative annotation effort [28]. We picked the top categories from the annotation hierarchy, and added the "Studio Setting" category because of its expected usefulness on the TRECVID 2004 collection.

1. Animal
2. Cartoon
3. Graphic And Text
4. Human
5. Man Made Object
6. Outdoors
7. Outerspace
8. People Event
9. Person Action
10. Physical Violence
11. Sport Event
12. Studio Setting
13. Transportation
14. Transportation
15. Weather News

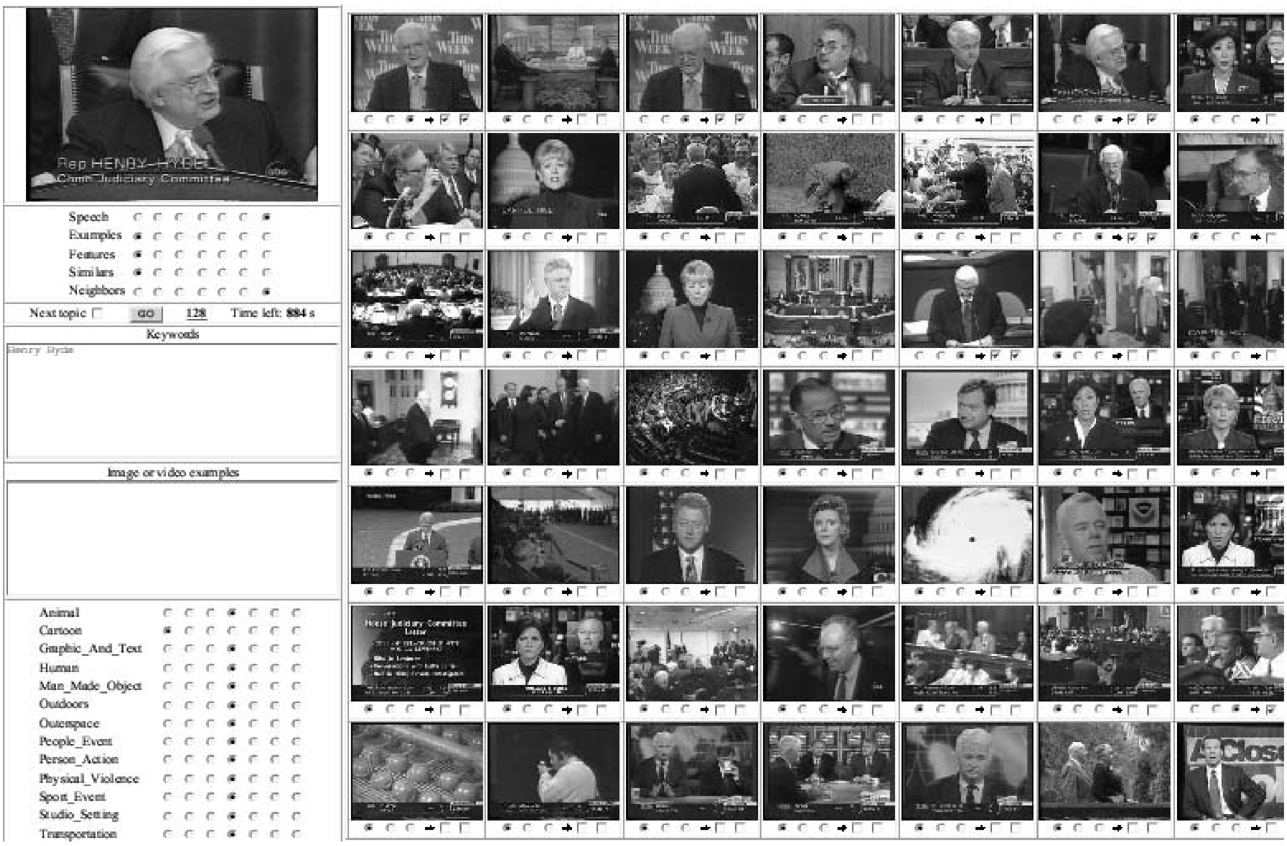
Table 6: Categories used for feature based search

A different approach than the one of the feature extraction task was used because of the different goals. The background of this labeling comes from indexing and retrieval of photographs and videos [29] [30], but with major differences due to the kind of images processed. The learning of the labels is defined as follows:

1. Extraction of features. The images are segmented spatially according to a predefined pattern:

a) blocks of $n \times n$ pixels (overlapping from $n/2$ pixels on both directions) are extracted from the images,

b) on each block, according to what is described in [31] for image indexing, colors (in YIQ space) features and texture features (Gabor energy, on 6 directions and 5 scales) are extracted. The stored data about colors for one block are the Y characteristic mean and standard deviation, the I characteristic mean and standard deviation, and the Q characteristic mean and standard deviation. So, a 6-tuple is extracted for the color information of one block. The stored data about textures for one block are, for one direction D and one scale S, the mean and standard deviations of the feature. So, a 60-tuple is extracted for the texture information of one block. For the color features extraction we take the original images pixels, and for the gabor features extraction we apodize the borders of the



Speech	○	○	○	○	○	○	●
Examples	●	○	○	○	○	○	○
Features	●	○	○	○	○	○	○
Similars	●	○	○	○	○	○	○
Neighbors	○	○	○	○	○	○	●

Next topic □ GO 128 Time left: 884 s

Keywords

Henry Hyde

Image or video examples

Animal	○	○	○	●	○	○	○
Cartoon	●	○	○	○	○	○	○
Graphic_And_Text	○	○	○	●	○	○	○
Human	○	○	○	●	○	○	○
Man_Made_Object	○	○	○	●	○	○	○
Outdoors	○	○	○	●	○	○	○
Outerspace	○	○	○	●	○	○	○
People_Event	○	○	○	●	○	○	○
Person_Action	○	○	○	●	○	○	○
Physical_Violence	○	○	○	●	○	○	○
Sport_Event	○	○	○	●	○	○	○
Studio_Setting	○	○	○	●	○	○	○
Transportation	○	○	○	●	○	○	○

Figure 7: View of the CLIPS-LIS-LSR search system

images. The descriptor of one block B of an image I is a 66-dimensional vector, called $Vinit_{I,B}$ in the following.

2. Learning set processing. The learning set is composed, for each category label L, of a set IP_L of x positive images and a set IN_L of y negative images. Each of these set was built manually from TRECVID 2003 image data, because the labels are not exclusive. From the positive images, a subset of blocks BP_L is used as positive sample set. From this set of positive blocks, we compute the mean $\mu_L(i)$ and standard deviation $\sigma_L(i)$ of each $Vinit_{I,B}(i)$ of the 66 features extracted $(1 \leq i \leq 66)$. These means and standard deviations are used to generate the final feature vector $Vfin_{I,B}$ of each block using a zero mean normalization for any block feature considered (during the learning process for the positive and negative samples, and also during the recognition process).

3. Learning process. We use Support Vector Machines (SVM) to achieve the learning on the 15 labels. SVM have been successfully used for labeling of home photographs [32], [33]. The implementation SVM considered for this experiment is SVM_light [34]. For each $Vfin_{I,B}$ of a positive or a negative image, a polynomial kernel SVM-based classifier is learned. On average, the size of the learning sample blocks is 1665 (880 for positive set and 785 for negative), and the learning process takes on average 6 hours (Linux-based PC under RH9.0, Pentium IV at 3.2 Ghz, 1GB memory) per label, so 90 hours for the 15 labels.

4. Probabilization. We take for each category label a set of positive and negative images (different from the steps 2 and 3 above). Each block B of these images is extracted and the 15 corresponding final feature vectors (one per label) are computed. These vector are then input to the respective SVM classifiers, that give a 15-dimensional binary histogram $Hb_{B,I}$ with a 1 value if the block is classified as positive, and 0 otherwise. Each bin of $Hb_{B,I}$ is related to one and only one label L associated to its identifier id_L . If we consider that one image I is split into NB_I blocks, we generate a 15-dimensional histogram $Hinit_I$ that sums up for each dimension all the $Hb_{B,I}$ of I, having $0 \leq Hinit_I(id_L) \leq NB_I)$. A normalization of this histogram is then achieved . We probabilize these histograms by computing the probability of correct recognition having given values of the normalized $Hinit_I(id_L)$. More precisely, this is achieved through the computation of the mean of correct labeled images and the mean of wrongly labeled images for each label, and by defining a probability density function based on an approximation of a sigmoid function based on the means computed. As a result, we obtain a probability that an image is correctly labeled whith each of the 15 category labels. The computing of the probability density functions (one per category label) is achieved on average on 880 blocks for positive sets and 780 for negative sets per category label.

During the recognition process, similarly to the step 4 above, we first extract the blocks of one keyframe, then we compute the normalized histogram of the image and eventually we probabilize the results using the probability density functions defined in step 4. The result obtained for each keyframe is then a real number in $[0, 1]$ for each category label, indicating how likely the keyframe is to be relevant for the category. During the search task, each category can be assigned a positive or a negative importance, and the retrieval is processed through a combination of the assigned importance and the probability to be labeled by the selected categories (Figure 7).

5.4 Visual similarity to already identified positive images

Visual similarity to already retrieved images can be used for the search. These images have to be marked as positive examples for similarity based search by the user (relevance feedback). The search is performed in the same way as for the original image examples. Key frames are ranked according to their closeness to thes positive examples. The images selected for similarity-based search need not to be actually positive example for the current search.

5.5 Temporal closeness to already identified positive images

Temporal closeness (within the video stream) to already retrieved images can be used for the search. These images have to be marked as positive examples for similarity based search by the user (relevance feedback). Key frames are ranked according to their temporal closeness to thes positive examples. The images selected for similarity-based search need not to be actually positive example for the current search.

5.6 Combination of search criteria

The user can define dynamically his search strategy according to the topic and/or the looking of the retrieved images. Each search mechanism can be configured independently and each mechanism can be given a global weight for the search (Figure 7). Relevance are computed independently for each mechanism and for each key frame (or subshot). The per-mechanism relevances are then linearly combined according to the mechanism weight to produce the final key frame relevance. A relevance is computed for each shot at the maximum of the relevances associated to each key frame (or subshot). A ranked list of shots is the produced.

5.7 Search strategy

The system is designed for very fast response time and efficient user feedback. The user is encouraged to use whatever search mechanism seems best appropriate and to view and mark as many images as possible in the given time (900s). At each iteration, the system displays 49 images. By default they are marked as negative. The user only has to mark the positive that he sees by clicking on them. In case of doubt he can see them at actual size in a separate window just by mouse overlap and, if still necessary, he can play the shot by clicking below the images. By default also, the positive images are also positive examples for visual similarity and temporal closeness based search but this can be changed also by the user. Any key frame marked positive by the user receives a relevance of 1 and any key frame marked positive by the user receives a relevance of 0.

The same system has been used for manual and interactive submissions. Manual submissions are the results of the system at the first iteration (without any feedback). Interactive submissions are the results of the system after as many iteration as possible within the allocated time. The system keep track of the output (ranked list of 1000 shots) at each iteration as well as the time elapsed since the beginning of the topic processing. This allows to display the evolution of the Mean Average Precision (MAP) over time during the search.

5.8 Results

Four users have participated to the tests. Some of them did not have the time to process all topics and other (new) users completed the processing of the remaining topics. Each user processed each topic at most once. One user did the search using only the mechanism based on ASR (keywords). All three users used all available modalities. The visual similarity based search was not operational yet at the time at which the experiments were conducted. Table 5.8 shows the Mean Average Precision for each user for manual (a single iteration, no feedback) and interactive searches.

User	Type	Manual	Interactive
1	unlimited	0.0652	0.2471
2	unlimited	0.0581	0.2105
3	unlimited	0.0319	0.1306
4	ASR only	0.0555	0.1623

Table 7: Mean Average Precision for the search task

It can be noticed that there is a significant variability of the system performance according to the user. The relative user performance is consistent with the knowledge and the experience the user has of the system. It is also most probable that the mother language as well as the cultural background of the users significantly affect the system/user performance. None of the users here is an English native speaker. None of them either is much familiar with the politics and sports in the US.

Table 5.8 shows the evolution over time of the Mean Average Precision for user 1. M.A.P. 0 minutes corresponds to a random answer. M.A.P. 1 minutes corresponds to the manual search (the first iteration is usually done in less tha 1 minute). M.A.P. 5, 10 and 15 minutes were obtained using the system output trace.

Elapsed time	M.A.P.
0 minutes	0.0002
1 minutes	0.0652
5 minutes	0.1593
10 minutes	0.2243
15 minutes	0.2471

Table 8: Evolution of M.A.P. over time for user 1

6 Conclusion

We have presented the systems used by CLIPS-IMAG and his partners, LSR-IMAG, LIS and LABRI laboratories, to perform the tasks proposed in the TRECVID 2004 workshop. SBD was performed using a system based on image difference with motion compensation and direct dissolve detection. This system gives control of the silence to noise ratio over a wide range of values and for an equal value of noise and silence (or recall and precision), its F1 value is 0.83 for all types of transitions. Story segmentation was performed using a combination of multi-modal detectors and the F1 value for the optimal system configuration was of 0.48. Feature extraction was achieved using a combination of lexical context based classification, a color and texture based classification and on face recognition. The search system uses a user-controlled combination of five mechanisms: keywords, similarity to example images, semantic categories, similarity to already identified positive image, and temporal closeness to already identified positive image. The mean average precision of the search system (with the most experienced user) is 0.24.

7 Acknowledgments

This research has been sponsored by a BQR grant from the Institut National Polytechnique de Grenoble (INPG) and an EPML grant from the Centre National de la Recherche Scientifique (CNRS).

References

[1] Quénot, G.M.: CLIPS at TREC-11: Experiments in Video Retrieval, In em 11th Text Retrieval Conference, Gaithersburg, MD, USA, 19-22 November, 2002.

[2] Quénot, G.M.: TREC-10 Shot Boundary Detection Task: CLIPS System Description and Evaluation, In em 10th Text Retrieval Conference, Gaithersburg, MD, USA, 13-16 November, 2001.

[3] Ruiloba, R., Joly, P., Marchand, S., Quénot, G.M.: Toward a Standard Protocol for the Evaluation of Temporal Video Segmentation Algorithms, In *Content Based Multimedia Indexing*, Toulouse, Oct. 1999.

[4] Quénot, G.M.: Computation of Optical Flow Using Dynamic Programming, In *IAPR Workshop on Machine Vision Applications*, pages 249-52, Tokyo, Japan, 12-14 nov. 1996.

[5] Magrin-Chagnolleau, I., Gravier, G, Blouet, R. for the ELISA consortium: Overview of the 2000-2001 ELISA consortium research activities, In *2001: A Speaker Odyssey*, pp.6772, Chania, Crete, June 2001.

[6] Przybocki, M., Martin, A.: NIST's Assessment of Text Independent Speaker Recognition Performance, The Advent of Biometrics on the Internet, A COST 275 Workshop in Rome, Italy, Nov. 7-8 2002

[7] Moraru, D., Meignier, S.,Besacier, L., Bonastre, J.-F., Magrin-Chagnolleau, Y.: The ELISA Consortium Approaches in Speaker Segmentation during The NIST 2002 Speaker Recognition Evaluation In *Proceedings of ICASSP*, Hong Kong, 6-10 Apr. 2003.

[8] Delacourt, P., Wellekens, C.: DISTBIC: a speaker-based segmentation for audio data indexing, In *Speech Communication*, Vol. 32, No. 1-2, September 2000.

[9] Gauvain, J.L., Lamel, L., Adda, G.: The LIMSI Broadcast News Transcription System, In *Speech Communication,* 37(1-2):89-108, May 2002.

[10] Allamanche E., Herre J., Hellmuth O., Fröba B., Kastner T., Cremer M.: Content-based identification of Audio Material Using MPEG-7 Low Level Description, In *ISMIR*, 2001.

[11] Chaisorn L., Koh C., Zhao Y., Xu H., Chua T.-S., Qi T.: Two-level multi-modal framework for news story segmentation of large video corpus, In *TRECVID'2003 Workshop*, Gaithersburg, MD, USA, 17-18 November, 2003.

[12] Hsu W., Kennedy L., Huang C.-W., Chang S.-F., Lin C.-Y. and Iyengar G.: News video story segmentation using fusion of multilevel multi-modal features in TRECVID 2003, In *Proceedings of ICASSP*, Montréal, Canada, Mai 2004.

[13] D. Moraru, S. Meignier, C. Fredouille, L. Besacier, and J.-F. Bonastre, "The ELISA consortium approaches in Broadcast News speaker segmentation during the NIST 2003 Rich Transcription evaluation". ICASSP'04, Montreal, Canada, May 2004.

[14] Pinquier J. and André-Obrecht R.: Jingle detection and identification in audio documents, In *Proceedings of ICASSP* Montréal, Canada, Mai 2004.

[15] Quénot G.-M., Moraru D., Besacier L.: CLIPS at TRECVID: Shot Boundary Detection and Feature Detection, In *TRECVID'2003 Workshop*, Gaithersburg, MD, USA, 17-18 November, 2003.

[16] http://nist.gov/speech/tests/rt/rt2004/spring/

[17] Smeaton A., Kraaij W. and Over P.: TRECVID 2004: An introduction, In *TRECVID'2004 Workshop*, Gaithersburg, MD, USA, 15-16 November, 2003.

[18] L. Devillers and I. Vasilescu: Détection des émotions à partir d'indices lexicaux, dialogiques et prosodiques dans le dialogue oral. In *TALN-JEP*, Maroc (2004).

[19] L. Chen, J. L. Gauvain, L. Lamel, and G. Adda: Unsupervised language model adaptation for broadcast news. In *Proceedings of ICASSP*, pages I-220-223, Hong Kong, April 2003.

[20] W. H. A. Beaudot: Sensory coding in the vertebrate retina: towards an adaptive control of visual sensitivity, In *Network: Computation in Neural Systems*, vol. 7, pp. 317-323, 1996.

[21] N. Guyader, J. Hérault: Représentation espace-fréquence pour la catégorisation d'images, In *Human Neurobiology*, GRETSI'01, Toulouse.

[22] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, ISBN 0-387-94559-8, 1995.

[23] L. Carminati, J. Pineau, and M. Gelgon: Human detection and tracking from video surveillance applications in ow density environment, In *SPIE VCIP'2003 SPIE 0277 -786X* **5150**, pp. 51–60, 2003.

[24] R. Rifkin: M.I.T. SvmFu Version 3. Software developed by the Center for Biological and Computational Learning du M.I.T., http://five-percent-nation.mit.edu/SvmFu, 2001.

[25] OpenCV Project: Intel Image Processing Library, http://www.intel.com/research/mrl/research/opencv/.

[26] C. J. Burges: A tutorial on support vector machine for pattern recognition, In *Data-Mining and Knowledge Discovery* **2**, pp. 121–167, 1998.

[27] E. Osuna, R. Freund, and F. Girosi: Support vector machines: Training and applications, *Tech. Rep. AIM-1602*, Massachusetts Institute of Technology, 1997.

[28] C.-Y. Lin, B. L. Tseng and J. R. Smith: Video Collaborative Annotation Forum: Establishing Ground-Truth Labels on Large Multimedia Datasets, In *TRECVID'2003 Workshop*, Gaithersburg, MD, USA, 17-18 November, 2003.

[29] P. Mulhem and J.-H. Lim: Symbolic Photograph Content-Based Retrieval, In *ACM Conference on Information and Knowledge Management (CIKM)*, Virginia, USA, pp. 94-101, 2002.

[30] J.-H. Lim, Q. Tian and P. Mulhem: Home Photo Content Modeling for Personalized Event-Based Retrieval, In *IEEE Multimedia, Special Issue on Multimedia Content Modeling and Personalization*, 10(4), pp. 28-37, October-December, 2003.

[31] J.-H. Lim: Building Visual Vocabulary for Image Indexation and Query Formulation, In *Pattern Analysis and Applications (Special Issue on Image Retrieval)*, Vol. 4, nos. 2/3, 2001, pp. 125-139.

[32] S. Tong and E. Chang: Support Vector Machine Active Learning for Image Retrieval, In *Proc. 9th ACM Int'l Conf. on Multimedia*, pp.107-118, 2001.

[33] M. C. S. Paterno, F. S. Lim,and W. K. Leow: Fuzzy Semantic Labeling for Image Retrieval, In *Proc. Int. Conf. on Multimedia and Exposition*, 2004, to be published.

[34] T. Joachims: SVMlight Support Vector Machine, http://www.cs.cornell.edu/People/tj/svm_light