Research Article

Unsupervised Segmentation Methods of TV Contents

Elie El-Khoury, Christine Sénac, and Philippe Joly

IRIT Laboratory, Toulouse University, 118 route de Narbonne, 31062 Toulouse Cedex 9, France

Correspondence should be addressed to Elie El-Khoury, elie.el-khoury@irit.fr

Received 1 September 2009; Accepted 26 March 2010

Academic Editor: Ling Shao

Copyright © 2010 Elie El-Khoury et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a generic algorithm to address various temporal segmentation topics of audiovisual contents such as speaker diarization, shot, or program segmentation. Based on a GLR approach, involving the ΔBIC criterion, this algorithm requires the value of only a few parameters to produce segmentation results at a desired scale and on most typical low-level features used in the field of content-based indexing. Results obtained on various corpora are of the same quality level than the ones obtained by other dedicated and state-of-the-art methods.

1. Introduction

Nowadays, due to an explosive growth of digital video content (both online and offline and available by means of public or private databases and TV broadcasts), there is an increasing of accessibility for these data. Actually, the wealth of information raises the problem of an adapted access to video content which includes heterogeneous information that can be interpreted at different granularity levels, thus leading to many profiles of requests.

Under these conditions, automatic indexing of the structure, which provides direct access to the various components of the multimedia document, becomes a fundamental issue.

For this purpose, a temporal segmentation of audiovisual is required as a preprocessing operation. Results of this segmentation may be directly used for delinearization purposes such as providing a direct access to the content itself. They can also feed other analysis algorithms aiming at producing synoptical views of the content or exploiting temporal redundancy properties inside homogeneous segments to speed up the processing time.

Basically, temporal segmentation tools work on a low-level feature (or a small set of low-level features) extracted from the content along the time. Commonly, these low-level features express meaningful properties that can be observed or processed directly from the signal, such as spectrum/cepstrum features for an audio signal or color histograms for an image. They are expressed numerically and

represented through vectors whose dimensions depend on the number of those features.

Two kinds of segmentation strategies can then be applied. Some algorithms try to gather set of successive values which are supposed to belong to a same homogeneous segment. Some others are focusing on transitions detection between segments.

Such algorithms have been developed independently one with the others for different temporal segmentation problems.

Among the most addressed ones, we find the “audio turn” segmentation. An “audio turn” denotes a homogeneous audio segment related to basic semantic audio classes namely, speech, music, speech superimposed with music, ambient sounds, and silence. This is generally a preprocessing tool. This is also the case for shot segmentation algorithms. The goal here is to identify successive frames in an edited video content which are belonging to a same cinematic take. More recently, some algorithms have been also proposed for TV programs segmentation. They can be used for Electronic Program Guide (EPG) synchronization or simply to provide some entries in recordings.

All those algorithms are dedicated to specific tasks of segmentation. They are based on more or less explicit models and properties of the concepts associated to the segments or to transitions and so cannot be applied to any other segmentation tasks.

In this paper, we develop the idea that, on light processing architectures, a single operator able to produce audio turns, or shots, TV programs segmentation could be of interest if the results are of nearly the same quality than the ones obtained with dedicated tools.

So, after an overview of related works concerning video and audio segmentation methods in Section 2, we present, in Section 3, a generic unsupervised segmentation we firstly developed to process audio contents. Then we show how this segmentation method can be adapted for different granularity levels such as shot detection (Section 4), programs boundaries detection in days of television recordings (Section 5), and speaker segmentation (Section 6).

2. State of the Art of Video and Audio Segmentations

2.1. Video Segmentation. Video segmentation has been studied extensively. Traditionally, a four-layer hierarchical structure is adopted for video structure analysis which consists of a frame layer, a shot layer, a sequence layer, and a program layer. At the bottom of the structure, continuous image frames taken by a single camera are grouped into shots. A series of related shots are then grouped into a sequence. Shot change detection is performed as the first step for video segmentation by most previous approaches.

2.1.1. Video Shot Boundaries Detection. Historically, the first studied video segmentation task is the shot boundaries detection which aims at breaking the massive volume of video into smaller chunks. Shots are concatenated by editing effects such as hard cuts, fades, dissolves, and wipes. A reliable shot detection algorithm should identify such short breaks.

Because it is not an easy task, quite a lot of approaches were proposed in the literature [1–3]. See for example the report of TRECVID for a review and a comparison of state-of-the-art systems and [4] for an overview of the methods and evaluation of shot-segmentation algorithms.

Among the classical algorithms, one can cite the color histogram differences used to detect hard cuts, the standard deviation of pixel intensities for fade cuts, the edge-based contrast for dissolve cuts, and the edge-change ratio for hard, fade, and dissolve cuts. Parameters are often chosen in order to describe color or luminous intensity of the video. However the challenging problem is to distinguish shot boundaries from the following: fast object or camera motion, fast illumination changes, reflections, sudden change to explosion, and flash photography. Each potential artifact leads to develop an adhoc processing tool and explains the myriad of method.

2.1.2. Video Sequence Boundaries Detection. Some works have attempted to find an upper level structuring, mainly by grouping together adjacent shots semantically linked up in scene form [5] using more or less explicit rules used in the audiovisual production domain [6]. This task is tricky for movies because it obeys to subjective criterions. Furthermore

it is difficult to process heterogeneous corpora, but scenes can be detected from special programs having a quite stable structure such as broadcast news or sports [7, 8]. But the implemented methods use a lot of many a priori knowledge.

Reference [9] presented an approach with no models or decision rules to define “story units” according to the following method. When two shots are very similar and nearby temporally, they are grouped together with all the intermediary shots in order to form a segment. To calculate the similarity between two shots, intensity histograms of keyframes are used. Similar shots are grouped together in graph nodes form, and nodes are linked up when the two corresponding groups are temporally adjacent. Then segments are produced by partition of the graph, deleting the more minor links.

2.1.3. Program Boundaries Detection. Very few researches have been done for program boundaries detection on TV broadcast. Let us mention here those published by Liang et al. [10], Poli and Carrive [11] and Naturel et al. [12]. Here, a “program” must be understood as a regular television program such as news, weather broadcast, talk show, sports, or sitcoms.

Poli proposes to predict forthcoming TV programs by modelling the past ones in order to boil down the television stream structuring problem to a simple alignment one with the EPG thanks to Hidden Markov Models trained on data about television schedules collected over a full year. The stream is first segmented to find boundaries of programs which are labeled later.

In the same time, Naturel proposes a fast structuring of large TV streams using also program guides to label the detected programs. The method for segmenting a TV stream which is built on the detection of nonprogram segments (such as commercial breaks) uses two kinds of independent information. The first one is a monochromatic frame and silence detector appearing between commercials on French TV. The second kind of information comes from a duplicate detection process. Nonprograms are detected in this way because they are usually broadcasted several times and so already present in a labeled reference video dataset.

Liang proposes a less ambitious work, closer to our proposition, as it only detects programs boundaries without labeling them. He supposes that TV videos have two intrinsic characteristics. On one hand, for a TV channel, programs appear and end at a relatively fixed time every day. On the other hand, for programs of the same type, they have stable or similar starting and ending clips even when they appear in different days. As such, the approach consists of two steps: model construction and program segmentation. The program boundary models for the selected TV channel are constructed by detecting the repeating shots on different days. Then, based on the obtained models, videos recorded from the same TV channel can be segmented into programs. This approach is not valid for more complex streams and can not take into account any possible change of TV schedule.

2.2. Audio Segmentation. Auditory scene segmentation is an important step in the process of high-level semantic inference from audio streams, and in particular, a prerequisite for auditory scene categorization.

As opposed to single modal audio (e.g., pure speech in the context of speech recognition task), composite audio of multimedia databases usually contains multiple audio modalities such as speech, music and various audio effects, which can be mixed.

This is why, in the audio indexing context, first works were focusing on music/speech discrimination obtained directly with a set of low-level characteristics [13] or using multi-Gaussian models learned with huge corpora [14].

Other segmentation methods identify key sounds, such as whistle sound, crowd noise, or commentator voice in a soccer broadcast [15]. Here again, segmentation is possible only with the use of a priori knowledge.

In [16], authors first extract audio elements such as speech, music, various audio effects, and any combination of these in order to detect key audio elements and to segment the auditory scene to obtain a semantic description.

Some works, within a musical program, try to identify the musical type [17] or musical instruments used [18]. In [19] a microsegmentation of musical sequences is performed by detecting onsets of notes and percussive events.

This vast number of audio segmentation and classification methods is due on the one hand to heterogeneousness of the content and on the other hand to the aimed semantic level.

Reference [5] presents a method to detect the different audio scenes without a priori knowledge. An audio scene change occurs when a majority of the sources present in the data change. The dominant sources are assumed to possess stationary properties that can be characterized using a few features extracted from the signal [20]. In order to detect scene change, a local correlation function is then used.

2.3. Knowledge-Free and Generic Methods. We have seen that nearly all methods of audio or video segmentation perform with a priori knowledge. These approaches are based on a spatial-temporal modelling of the content and use decision rules. Currently, it is the only way to reach the semantic quality required by search engines. But only recording collections highly structured, such as broadcast videos of news and sports programmes, and homogeneous in terms of production can benefit from such methods. Furthermore, model or decision rules-based methods are limited because, for each new collection, they need either a new learning or a new expertise and often new tools have to be defined.

However some generic methods exist, permitting to process both audio and video documents. Foote and Cooper [21] show that a similarity matrix applied to well-chosen features allows a visual representation of the structural information of a video or audio signal. The similarity matrix can be analyzed to find structure boundaries. Generally, the boundary between two coherent segments produces a checkerboard pattern. The two segments will exhibit

high within-segment similarity, producing adjacent square regions of high similarity along the main diagonal of the matrix. The two segments will also produce rectangular regions of low between-segment similarity off the main diagonal. The boundary is the crux of this checkerboard pattern.

A supplementary approach was developed by Haidar et al. [22]. This approach is also generic because it is independent of the size and of the type of the document. Several similarity matrices, each one representing one feature, are accumulated, and the resultant matrix shows the temporal areas homogeneous in terms of the set of the different used features. But automatically inferring a document structure from such a matrix is not easy.

3. The Proposed Segmentation Method

Contrary to the methods seen in the above section, we present a priori knowledge-free segmentation approach relaying mainly on the hypothesis that it is possible to segment at some different granularity levels any audiovisual document and that is equivalent to segment into homogeneous segments at the adequate scale.

The segmentation we propose was firstly designed for audio segmentation in the context of speaker diarization [23]. Because the traditionally used metric approaches (symmetric Kullback-Leibler, Hotteling's T2-Statistic) did not give us sufficient results in presence of multiple simultaneous audio sources, we turned towards approaches based on model selection like the Generalized Likelihood Ratio (GLR) [24] and the Bayesian Information Criterion (BIC) [25]. Though the results are better, we observed that usual GLR and BIC methods present some weaknesses: too many parameters are required to tune the algorithm, and a bad precision is obtained in detecting boundaries when segments are small.

So, we propose some improvements to the general algorithm described hereafter.

3.1. Overview of the Basic Segmentation Algorithm. The basic method for detecting a change between homogenous zones is the GLR applied on a temporal signal in which each sample is a vector of several low-level features.

For genericity reasons, we will describe this method using an unknown signal that may be an acoustic signal, a video signal or an audiovisual signal.

Let $X = x_1, \dots, x_{N_x}$ be the sequence of observation vectors of dimension d to be modeled, M the estimated parametrical model, and $L(X, M)$ the likelihood function. The GLR introduced by Gish et al. [24] considers the two following hypotheses.

H_0 : This hypothesis assumes that the sequence X corresponds to only one homogeneous segment (in the case of audio signal, it corresponds to only one audio source). Thus, the sequence is modeled by only one multi-Gaussian distribution

$$(x_1, \dots, x_{N_x}) \sim N(\mu_X, \sigma_X). \quad (1)$$

H_1 : This hypothesis assumes that the sequence X corresponds to two different homogeneous segments $X_1 = x_1, \dots, x_i$ and $X_2 = x_{i+1}, \dots, x_{N_x}$ (in the case of audio signal, it corresponds to two different audio sources or more particularly to two different speakers). Thus, the sequence is modelled by two multi-Gaussian distributions

$$\begin{aligned} (x_1, \dots, x_i) &\sim N(\mu_{X_1}, \sigma_{X_1}), \\ (x_{i+1}, \dots, x_{N_x}) &\sim N(\mu_{X_2}, \sigma_{X_2}). \end{aligned} \quad (2)$$

The generalized likelihood ratio between the hypothesis H_0 and the hypothesis H_1 is given by

$$\text{GLR} = \frac{P(H_0)}{P(H_1)}. \quad (3)$$

In terms of likelihood, this expression becomes

$$\text{GLR} = \frac{L(X, M)}{L(X_1, M_1)L(X_2, M_2)}. \quad (4)$$

If this ratio is lower than a certain threshold Thr , we can say that H_1 is more probable, so a point of change in the signal is detected.

By passing through the log

$$R(i) = -\log \text{GLR} \quad (5)$$

and by considering that the models are Gaussian, we obtain

$$R(i) = \frac{N_X}{2} \log |\Sigma_X| - \frac{N_{X_1}}{2} \log |\Sigma_{X_1}| - \frac{N_{X_2}}{2} \log |\Sigma_{X_2}|, \quad (6)$$

where Σ_X , Σ_{X_1} , and Σ_{X_2} are the covariance matrices of X , X_1 , and X_2 and N_X , N_{X_1} , and N_{X_2} , are, respectively the number of the acoustic vectors of X , X_1 , and X_2 .

Thus, the estimated value of the point of change by maximum likelihood is given by

$$\hat{i} = \arg \max_i R(i). \quad (7)$$

If $\hat{i}$ is higher than the threshold $T = -\log \text{Thr}$, a point of speaker change is detected. The major disadvantage resides in the presence of the threshold T that depends on the data. That is why, Rissanen [26] introduced the Bayesian Information Criterion (BIC).

3.1.1. Bayesian Information Criterion. For a given model M , the BIC is expressed by

$$\text{BIC}(M) = \log L(X, M) - \frac{\lambda}{2} n \log N_X, \quad (8)$$

where n denotes the number of the observation vectors of the model. The first term reflects the adjustment of the model to the data, and the second term corresponds to the complexity of the data. λ is a penalty coefficient theoretically equal to 1. [26].

The hypotheses test of (3) can be viewed as the comparison between two models: a model of data with two

Gaussian distributions (H_1) and a model of data with only one Gaussian distribution (H_0). The subtraction of BIC expressions related to those two models is

$$\Delta \text{BIC}(i) = R(i) - \lambda P, \quad (9)$$

where the log-likelihood ratio $R(i)$ is already defined in (6) and the complexity term P is given by

$$P = \frac{1}{2} \left(d + \frac{1}{2} d(d+1) \right) \log N_X, \quad (10)$$

d being the dimension of the feature vectors.

The BIC can be also viewed as the thresholding of the log-likelihood distance with an automatic threshold equal to λP .

Thus if $\Delta \text{BIC}(i)$ is positive, the hypothesis H_1 is privileged (two different speakers). There is a change if

$$\left\{ \max_i \Delta \text{BIC}(i) \geq 0 \right\}. \quad (11)$$

The estimated value of the point of change can also be expressed by

$$\hat{i} = \arg \max_i \Delta \text{BIC}(i). \quad (12)$$

A well-known BIC segmentation method was proposed by Sivakumaran et al. to detect multipoints changes in audio recordings [27]. In our work we applied this method, and we figure out some limitations. Although the amount of parameters to be tuned is important, the penalty coefficient is not as stable as expected, and there is a possible cumulative error due to the sequential segmentation process: if a point is erroneously detected, the next point might be affected by this error and might not be detected correctly. All those limitations encouraged us to propose a new segmentation based on GLR and BIC.

3.2. Proposed Improvements. The proposed method for signal segmentation follows four main steps as explained below and in Figure 1.

Row a is a time line on which the expected segmentation points are shown.

- (1) This time line is split into fixed size temporal windows (shown on row b) of duration d . On each window, a GLR point detection is performed independently. Doing so, one potential segmentation point P_i^0 on each window W_i^0 is so obtained. Row c shows these intermediate results. At this step, actual segmentation points closed to a temporal window boundary have a poor probability to be detected. Furthermore, some of the candidate segmentation points may have only a local significance in their temporal window but not at a larger scale.
- (2) To overcome these problems, we now consider overlapping temporal windows whose boundaries correspond to one candidate segmentation point over two, obtained during the previous step of the process.

FIGURE 1: The proposed segmentation.

At the first iteration of the process, the first window W_1^1 goes from the beginning to point P_2^0 ; the second window W_2^1 goes from point P_1^0 to P_3^0 ; ...; the i th window W_i^1 goes from point P_{i-1}^0 to P_{i+1}^0 . On each window, a new GLR point detection is performed. We obtain after this step a new set of potential segmentation points P_i^1 . More generally, we note P_i^k the GLR point detected over the window W_i^k whose temporal boundaries are defined by points P_{i-1}^{k-1} and P_{i+1}^{k-1} at the k th iteration.

- (3) We then go for a readjustment step where the closest segmentation points are merged (two points are closed if the distance between them is less than 3 samples). Step (2) and step (3) are iterated until stable results are obtained, that is we look for k so that $P_i^{k+1} = P_i^k$.
- (4) At the last step, all the candidate segmentation points, obtained during the last iteration, are tested against the BIC Criterion.

One may have noticed that if there is N fixed size temporal window defined at the beginning of this process, we will obtain M segmentation points at the end with $M \leq N$. This means that the size of the window must be a priori fixed at a lower value than the minimal length of segments expected as a result.

Moreover, we found that a bidirectional segmentation of the signal (i.e., both forward and backward) may be useful in some cases where the transitions between two homogeneous regions are not very discriminative (interactive acoustic regions, fade or dissolve transition effects between shots,

etc.). Indeed, due to the shifted variable size window introduced in the segmentation method, processing from “left to right” may detect different points of change than processing from “right to left”, and therefore, there is a chance that a missed boundary in the first direction can be detected in the other direction and vice versa.

The purpose of all those steps is to generate an as stable as possible segmentation that gives homogeneous zones in terms of features distributions.

4. Application to Shot Boundaries Detection

We formulate now the hypothesis that shots are homogeneous video segments and that we may find features that can at the same time be modelled by Gaussian distributions. If we can assume that it exists a large range of such features, the first hypothesis (shot are homogeneous segments) is far from be always observed. Some lighting effects (such as flashes) or fast moving objects are strong limitations to this hypothesis. To take this specificity into account, once we have applied the previously described segmentation algorithm, we go for a rather simple postprocessing step aiming at removing false detections generated by those kinds of effects.

In this present work, a feature vector is extracted as follows. Each image provided every 40 ms (25 images/second) is divided into 4 equal parts.

The mean values of R, G, and B colour space descriptors are computed in each part. Therefore, the feature vector of dimension d ($d = 3 \times 4$) is composed of those values.

Then, the segmentation algorithm is applied as explained in Section 2. The window size W is fixed equal to 50 feature vectors because we took the assumption that the minimum duration in which a point of change can be detected is greater than 2 seconds. The penalty coefficient λ was tuned to 3.

In order to eliminate some false alarm detections due fast motion and lighting effects, a final step of histograms comparison is applied on the detected boundaries using the Manhattan (or City-Block) distance.

Suppose the video is composed of frames $I_1, I_2, I_3, \dots, I_n$ and the segmentation step returns the following frames $I_1, I_{k_1}, I_{k_2}, \dots, I_n$ as boundaries. For each boundary frame I_{k_j} , we consider the window of frames $[I_{k_j-a}, I_{k_j+a}]$ (a is fixed experimentally to 6), and we compute all the Manhattan distances between the histograms of the frames I_{p-1} and I_{p+1} where $p \in [k_j-a+1, k_j+a-1]$. Then, if all the distances are lower than a threshold, the frame I_{k_j} is withdrew from the boundaries set.

4.1. Experiments

4.1.1. The Corpus. We experiment our method on the corpus of the French ARGOS campaign [28]. The content set of the ARGOS campaign was made of various TV recordings, gathering TV news programs as well as commercials, weather forecast, documentaries, and fiction.

We used the two files of development (about 1 hour) to tune parameters. Then test was performed on 10 other hours.

FIGURE 2: Identification of matched segment intersections.

TABLE 1: Shot boundaries detection: evaluation of the proposed system with the two types of metrics.

	Recall	Precision	$F_measure$
ARGOS metrics	0.935	0.931	0.933
TRECVID metrics	0.893	0.918	0.905

4.1.2. *The Metrics.* Two types of metrics both from ARGOS and TRECVID campaign (<http://www.nlp.ir.gov/projects/tv2007/tv2007.html>) were used. The TRECVID metric highlights the ability to localize transitions in opposition with the ARGOS metric which highlights the ability of the segmentation tool to gather units belonging to a same segment.

(a) *The TRECVID Metric.* This is the traditional $F_measure$ computed from precision and recall as follows:

$$\begin{aligned}
 \text{precision} &= \frac{\text{number_of_correctly_detected_boundaries}}{\text{total_number_of_detected_boundaries}} \\
 \text{recall} &= \frac{\text{number_of_correctly_detected_boundaries}}{\text{total_number_of_boundaries_to_be_detected}} \\
 F_measure &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (13)
 \end{aligned}$$

(b) *The ARGOS Metric.* The reference and system outputs are transformed into a list of continuous segments. Each segment of the ground truth is matched with the longest overlapping segment obtained as a result. A segment of the results can be matched only once. The temporal intersection between matched segments is then identified (cf. Figure 2).

Dynamic programming is used to find an optimal matching. Once the optimal matching is found, the $F_measure$ is defined as

$$F_measure = \frac{2 \times |\text{matched_intersection}|}{|\text{Ground_truth}| + |\text{Results}|}, \quad (14)$$

where $|\cdot|$ represents the overall duration of the segments.

4.1.3. *Results.* Table 2 shows the results (with ARGOS metric) of the proposed system compared to the average system and the best system of the campaign. We can see that our system and the best system of ARGOS give quite similar results. The method of the best system is specific

TABLE 2: Shot boundaries detection: the proposed system versus the average one and the best one at the ARGOS campaign (ARGOS metric).

	Proposed Sys.	Average Sys.	Best Sys.
ARGOS metrics	0.933	0.87	0.94

to the task: it detects cuts by image comparisons after motion compensation. Then gradual transitions are detected by comparing norms of the first and second temporal derivations of the image

5. Application to Program Boundaries Detection

We consider here that hypotheses made for shot detection can be extended to program segmentation. It means that a selected set of features during a program behave in an homogeneous manner so that their values distribution can be modelled by a Gaussian law and that features of two consecutive programs follow two rather different Gaussian laws. The last hypothesis is that a segment is of a minimal duration (in order to fix the size of the window used at the beginning of the algorithm and to determine when fusion of boundaries must be operated—see Figure 1(d)).

In our work, the goal is to check if typical video and audio features could validate the above hypotheses.

5.1. *Program Boundaries Detection Using Visual Features.* Each TV program has a certain number of visual characteristics that make this program different from the others. For example, the luminance, the dominant colors, and the activity rate in a soap episode are different from those observed on a TV game or a TV News program.

As input for the system, a vector of features is originally provided as follows. Every k seconds where k denotes the approximate value of the most frequent shot duration in seconds (experimentally $k = 8$) for the tested content set, a frame is extracted and then, the three corresponding 2^m -dimension color histograms (R, G and B) are computed and their 3×2^m ($m = 8$ if the images are 8-bit images) values are concatenated in a vector. Furthermore, the Singular Value Decomposition (SVD) is applied in order to reduce the vectors dimension. Experimentally, an inertia ratio higher than 95% is reached with a vector dimension reduced to 12.

Finally, the segmentation method explained above is applied on the sequence of these 12-dimension feature vectors. Results in table 3 show a precision of about 78% on 5 days of television (120 hours).

The major errors appear when there are commercial breaks: it may be typically explained because in this type of programs, in addition to their short duration, the homogeneity hypothesis is not still verified.

The variation and the distribution of the first “video” feature (after SVD) on 3 consecutive programs are given on Figures 3 and 4. Figures 5 and 6 show the same phenomena for the third “video” feature obtained after SVD.

FIGURE 3: Variation of Feature F1 for 3 consecutive programs.

We can verify that both variation and repartition are different for the three programs.

5.2. Program Boundaries Detection Using Acoustic Features. In this subsection, we evaluate the ability of our segmentation method to detect program boundaries using only audio features.

The input feature vectors are provided as follows. The first p Mel Frequency Cepstrum Coefficients ($p = 16$) are extracted every 10 ms using a sliding window of 20 ms. Those coefficients are then normalized and quantified between 0 and $D - 1$ ($D = 48$). Every k seconds ($k = 8$), histogram vectors are computed for each MFCC coefficient and concatenated to build a super-vector of dimension $p \times D$. Then, the SVD is applied in order to reduce the dimension of those vectors. Practically, an inertia ratio of about 90% is obtained for a resulting vector dimension of 40. Finally the segmentation is applied.

Table 3 shows that scores are lower with the acoustic features (75%) than with the visual features.

5.3. Program Boundaries Detection Using Audiovisual Features. In order to exploit the complementary information brought by the two different modalities, the previous audio and video features are simultaneously used. Because we took the same temporal sampling to produce feature vectors ($k = 8$) with the same dimensions value ($3 \times 2^m = p \times D$) reduced by the same processing (SVD, histograms) for the above two methods, it is very easy to combine them using two kinds of fusion: fusion at the decision level and fusion at the feature level.

At the decision level, the fusion was done by computing

$$\Delta BIC_{AV} = \Delta BIC_A + \Delta BIC_V, \quad (15)$$

where ΔBIC_{AV} corresponds ideally to a change between two TV programs.

TABLE 3: Results of the program boundaries detection (120 hours of test).

	Visual System	Audio System	AV System
$F_measure$	78.04%	75.16%	80.72%

At the feature level, the early fusion aims to concatenate the visual vector of features (dimension = 3×2^m) and the acoustic vector of features (dimension = $p \times D$). SVD is then applied: a resulting vector of dimension 60 is obtained for an inertia ratio of about 90%. Finally, the segmentation is processed to detect the frames of change. Experimentally, the early fusion at the features level gives (about 80.7%) better results than the fusion at the decision level (about 78.5%).

5.4. Experiments. Tests were carried out on 120 hours of TV videos recorded continuously from a general French TV channel during 5 days (including various kinds of programs such as news, weather forecast, talk shows, movies, sports, and sitcoms) with a rate of 25 frames/second. The size of the programs is very variable: from few minutes for weather forecast to 3 hours for a film.

For the segmentation step, we had to define the length of the fixed size window W , the penalty coefficient which depends on W , and the dimension of the feature vectors (12 for video features, 40 for audio features and 60 for audio/video features). We chose a window size of 4 minutes (corresponding to 30 vectors) as the hypothesis on the minimal duration of a program. The penalty coefficient λ was tuned to 5 for the Video system, to 1.2 for the Audio system, and to 1 for the AV system.

To evaluate those systems, the ARGOS $F_measure$ metric, described above, was used. It highlights the ability of the segmentation tool to gather units belonging to a same segment.

Results in Table 3 show that the visual system is better (about 78%) than the audio one (about 75%). With audio features, the majority of errors appear especially when there are commercial breaks. This might be explained typically because this type of program does not follow the homogeneity hypothesis. We can see that the two modalities audio and visual bring complementary information because the results are better than those obtained with only one modality.

Many improvements can be done while taking into account some knowledge already identified in the state of the art. For example, on French TV, commercials are separated by a sequence of monochrome images (white, blue, or black). As this kind of effect can be easily detected, improvements of about 9% ($F_measure = 87.34\%$) can easily be reached while gathering advertisements in a single program.

Comparison of the above results with those obtained by the state-of-the-art systems is a difficult task because corpora, units, and metrics are different for each experience and cannot be shared. To our knowledge, there is no international campaign addressing this topic. In this case, the evaluation we provide here should be considered as a basic reference which can be used later to evaluate improvements of this method or to compare with other future approaches.

FIGURE 4: Distribution of Feature F1 for 3 consecutive programs.

As our system is almost knowledge-free, it can process any kinds of TV content without any prior training phase. In this way, it can be seen as a useful preprocessing step in the context of video indexing for example.

As part of the project ANR EPAC (<http://www.epac.univ-lemans.fr/>), the program boundaries detection was applied on 1700 hours of TV and Radio contents: the processing took less than 16 hours (lower than $(\text{recording duration} \times 10^{-2})$) with a nonoptimized version written in Matlab on a classical PC architecture.

6. Application to Speaker Diarization

In the context of speech processing on meeting data, with high interaction between speakers, one of the most difficult and unsolved problems is “speaker diarization”. Speaker diarization is the process that detects speaker turns and groups those uttered by the same speaker. It is based on a first step of segmentation that consists in partitioning the regions of speech into segments: each segment must be as long as possible and must contain the speech of only one speaker. The second step is speaker clustering which consists in giving the same label to all the segments corresponding to the same speaker.

6.1. Experiments. In order to evaluate our method applied on speaker segmentation, we compare it to a well-known state-of-the-art method based only on BIC as described in [27, 29, 30].

The test set is the one used in ESTER 2009 evaluation competition (<http://www.afcp-parole.org/ester/index.html>). This set contains 20 shows for a total duration of about 7 hours recorded from 4 French radio stations.

FIGURE 5: Variation of Feature F3 for 3 consecutive programs.

In these experiments, a centisecond approach is used that is, the soundtrack is firstly decomposed into 10 ms frames: the feature vectors used are the first 12 Mel Frequency Cepstrum Coefficients (MFCCs). The bidirectional segmentation is then directly applied on these vectors by fixing the size of the window equal to 2 seconds, and the penalty coefficient λ is tuned to 1.

As part of the speaker diarization task, the segmentation step is followed by a clustering step which consists in grouping all segments corresponding to the same speaker. The clustering step we use, presented in [31], allows adjusting the boundaries previously detected by the segmentation. Therefore, an additional improvement of 2.77% is

FIGURE 6: Distribution of Feature F3 for 3 consecutive programs.

TABLE 4: Evaluation of the output segments provided by the state-of-the-art segmentation method, our segmentation method, and our diarization system.

	Recall (%)	Precision (%)	F-measure (%)
State-of-the-art method	74.25	71.62	72.91
Our segmentation method	85.36	85.37	85.37
Our diarization method	87.95	88.26	88.10

obtained when we take into consideration this clustering process.

7. Conclusions

We have presented in this paper a temporal segmentation algorithm aiming at detecting stable boundaries between homogeneous segments. The parameters of this algorithm allow adapting it in order to address different types of segmentations problems. The size of temporal windows used at the first step of the algorithm allows controlling the size of the generated segments and the algorithm complexity. The penalty coefficient used in the DBIC criterion allows to adapt the decision sensitivity to the given problem parameters.

Applied on typical audiovisual data, the performances of this algorithm can only be compared with state-of-the-art methods if we apply the same kind of postprocessing tools which are already involved in these methods (lighting effects or fast motion detection, dedicated commercial breaks detection, etc.). This algorithm can be applied on a light processing architecture (such as a set top box for example) in order to produce segmentation results on a large variety of content and for a large variety of applications.

References

- [1] B. T. Truong, C. Dorai, and S. Venkatesh, "New enhancements to cut, fade, and dissolve detection processes in video segmentation," in *Proceedings of the 8th ACM International conference on Multimedia (ACM '00)*, pp. 219–227, New York, NY, USA, 2000.
- [2] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 90–105, 2002.
- [3] Z. Liu and Y. Wang, "Major cast detection in video using both speaker and face information," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 89–101, 2007.
- [4] A. M. A. Ahmad, "Multimedia content and the semantic web: methods, standards and tools: book reviews," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 3, pp. 457–458, 2007.
- [5] H. Sundaram and S.-F. Chang, "Video scene segmentation using video and audio features," in *Proceedings of the IEEE International Conference on Multi-Media and Expo (ICME '00)*, vol. 2, pp. 1145–1148, Beijing, China, 2000.
- [6] P. Aigrain, P. Joly, and V. Longueville, "Medium knowledge-based macro-segmentation of video into sequences," in *Intelligent Multimedia Information Retrieval*, pp. 159–173, MIT Press, Cambridge, Mass, USA, 1997.
- [7] M. Bertini, A. Del Bimbo, and P. Pala, "Content-based indexing and retrieval of TV news," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 503–516, 2001.
- [8] G. Piriou, P. Bouthemy, and J.-F. Yao, "Extraction of semantic dynamic content from videos with probabilistic motion models," in *Proceedings of the 18th European Conference on Computer Vision (ECCV '04)*, vol. 3023, pp. 145–157, 2004.
- [9] M. Yeung, B.-L. Yeo, and B. Liu, "Extracting story units from long programs for video browsing and navigation," in *Readings in Multimedia Computing and Networking*, pp. 360–369, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 2001.

- [10] L. Liang, H. Lu, X. Xue, and Y.-P. Tan, "Program segmentation for TV videos," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '05)*, pp. 1549–1552, Kobe, Japan, May 2005.
- [11] J.-P. Poli and J. Carrière, "Modeling television schedules for television stream structuring," in *Proceedings of the 13th International Multimedia Modeling Conference (MMM '07)*, vol. 2, pp. 680–689, Singapore, 1996.
- [12] X. Naturel, G. Gravier, and P. Gros, "Fast structuring of large television streams using program guides," in *Proceedings of the 4th International Workshop on Adaptive Multimedia Retrieval (AMR '06)*, vol. 4398 of *Lecture Notes in Computer Science*, pp. 222–231, Paris, France, 2007.
- [13] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '97)*, vol. 2, pp. 1331–1334, Munich, Germany, 1997.
- [14] J. Pinquier and R. André-Obrecht, "Audio indexing: primary components retrieval: robust classification in audio documents," *Multimedia Tools and Applications*, vol. 30, no. 3, pp. 313–330, 2006.
- [15] S. Lefevre, B. Maillard, and N. Vincent, "Deux niveaux et deux outils d'analyse pour une meilleure segmentation de données audio," in *Proceedings of the 19th Colloque GRETSI sur le Traitement du Signal et des Images*, Paris, France, September 2003.
- [16] L. Lu, R. Cai, and A. Hanjalic, "Audio elements based auditory scene segmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. 17–20, Orlando, Fla, USA, May 2006.
- [17] G. Tzanetakis and G. Essl, "Automatic musical genre classification of audio signals," in *Proceedings of the IEEE Transactions on Speech and Audio Processing*, pp. 293–302, New York, NY, USA, 2001.
- [18] T. Heittola and A. Klapuri, "Locating segments with drums in music signals," Tech. Rep., Tampere University of Technology, Tampere, Finland, August 2002.
- [19] O. Gillet and G. Richard, "Comparing audio and video segmentations for music videos indexing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '06)*, vol. 2, pp. 873–876, Toulouse, France, May 2006.
- [20] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '96)*, vol. 2, pp. 993–996, Atlanta, Ga, USA, 1996.
- [21] J. T. Foote and M. L. Cooper, "Media segmentation using self-similarity decomposition," in *Storage and Retrieval for Media Databases*, vol. 5021 of *Proceedings of SPIE*, pp. 167–175, Santa Clara, Calif, USA, January 2003.
- [22] S. Haidar, P. Joly, and B. Chebaro, "Style similarity measure for video documents comparison," in *Proceedings of the 4th International Conference on Image and Video Retrieval (CIVR '05)*, vol. 3568 of *Lecture Notes in Computer Science*, Springer, Singapore, July 2005.
- [23] E. El Khoury, C. Sénac, and R. André-Obrecht, "Speaker diarization: towards a more robust and portable system," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '07)*, vol. 4, pp. 489–492, Honolulu, Hawaii, USA, 2007.
- [24] H. Gish, M.-H. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '91)*, vol. 2, pp. 873–876, Toronto, Canada, 1991.
- [25] S. S. Chen and P. S. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '98)*, vol. 2, pp. 645–648, Seattle, Wash, USA, May 1998.
- [26] J. Rissanen, *Stochastic Complexity in Statistical Inquiry Theory*, vol. 2, World Scientific Publishing, River Edge, NJ, USA, 1989.
- [27] P. Sivakumaran, J. Fortuna, and A. Ariyaeinia, "On the use of the Bayesian information criterion in multiple speaker detection," in *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech '01)*, vol. 2, pp. 795–798, Aalborg, Denmark, 2001.
- [28] P. Joly, J. Benois-Pineau, E. Kijak, and G. Quénot, "The ARGOS campaign: evaluation of video analysis and indexing tools," *Signal Processing: Image Communication*, vol. 22, no. 7–8, pp. 705–717, 2007.
- [29] A. Tritschler and R. Gopinath, "Improved speaker segmentation and segments clustering using the Bayesian information criterion," in *Proceedings of the European Speech Processing (Eurospeech '99)*, vol. 2, pp. 679–682, Budapest, Hungary, 1999.
- [30] P. Delacourt, D. Kryze, and C. J. Wellekens, "DISTBIC: a speaker-based segmentation for audio data indexing," *Speech Communication*, vol. 32, no. 1, pp. 111–126, 2000.
- [31] E. El-Khoury, C. Sénac, and J. Pinquier, "Improved speaker diarization system for meetings," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 4097–4100, Taipei, China, 2009.