AVSST: an Automatic Video Stream Structuring Tool

Ibrahim Zein Al Abidin, Patrick Gros, Sébastien Campion

AVSST: an Automatic Video Stream Structuring Tool

Zein Al Abidin IBRAHIM[1], Patrick GROS[2], Sebastien CAMPION[3]

[1]Assistant teacher, Angers University, Angers, France; [2] Senior research scientist, INRIA-IRISA Rennes, France; [3] Research engineer, INRIA-IRISA, Rennes, France;

[1]zibrahim@info.univ-angers.fr, [2]Patrick.Gros@inria.fr, [3]Sebastien.Campion@inria.fr

Abstract: **The aim of this paper is to present the tool that we have developed to automatically structure TV streams. The objective is to determine precisely the start and the end of broadcasted TV programs (P). Usually, TV channels separate programs with breaks (B). These breaks can be commercials, trailers, station identification breaks (monochrome frames for example), or bumpers. They may be broadcasted several times in the stream. The detection of these repetitions is the key of our method to structure the TV stream. After the detection step, a classification method is applied to separate the program repeated content from breaks ones. The latter are used to segment the stream in Program/Breaks sequence. Finally, the segmented stream is aligned with the metadata provided with the stream such as the Electronic Program Guide (EPG) in order to provide labeled programs. Experimentations are made on 22-day long TV stream that show the effectiveness of our method.**

Keywords: TV Stream, Indexing, Structuring, Repetition detection, Classification, EPG, metadata, DTW.

1 INTRODUCTION

Nowadays, there is an increase of the available digital videos where some of them are recorded from TV channels. These videos may contain several different programs and may be of a long duration. The availability of this special type of videos, which are of rapid growth nowadays, makes the access and the analysis steps very difficult. The traditional browsing methods provided by the state of the art base on a hierarchy structure. The concept of video table of content (ToC) is proposed by Rui et al. in [1] where a five-level hierarchy (video, scene, group, shot and keyframe) is used to represent the video. Using this type of hierarchy to browse TV stream is not suitable since the number of scene will be very high. In the other hand, the TV stream is composed of heterogeneous programs and users tend to browse it by program.

Even for video analysis, most of the available methods are highly dedicated ones. They depend on the type of the video analyzed. Structuring a news video or detecting events in a soccer game can be as such examples. For thus, there is a big interest in separating the stream into homogenous units at the program level which may help the analysis step to be fully automatic in spite of segmenting the stream manually in programs and then applying analysis tools.

Several methods have already been proposed to delimit or detect some specific content items in TV streams. Some detect bumpers to find breaks or programs. Others are dedicated to commercials. Each of the existing methods solves only a part of the structuring problem. An example of the first complete solutions is Naturel's [2]. This approach requires a lot of manual annotation and cannot scale up. On the other hand, Manson [3] proposed a new technique based on a supervised learning algorithm which requires manual annotation in order to train the system. This method is validated on the most structured time slice in the day (18:00-24:00). Furthermore, Poli [4] proposed a top-down approach, which learns to predict a more accurate program guide. This approach in its turn requires an enormous learning set (several years of exact program guides in his case).

The aim of our work is to provide a tool that structures the TV stream automatically. Our objective is to limit the manual annotation phase in the work of Naturel et al. [2]. The availability sometimes of metadata, like Event Information Table (EIT) or Electronic Program Guide (EPG) that provide information about the structure of the stream may make our work with no sense. Unfortunately, they provide imprecise and incomplete information ([5], [6]). Most of the programs starts and ends earlier or later than the announced time especially the live ones since the boundaries cannot be predicted a priori. In addition to that, most of the small programs are not announced. The use of these metadata directly in real-world applications has no sense. In the other hand, these metadata carry valuable information about the structure of the content and ignoring them make the automatic step very difficult. Metadata information will help us to label automatically the structured stream.

In the next section, we will present the different modules of our tool.

2 SYSTEM OVERVIEW

Our system takes as input several broadcasted days of TV streams with their associated Electronic Program Guide (EPG) and provides as output more precise EPG. The following figure (Figure 1) gives an overview of the different system steps. In the first step, the set of repeated content are detected. These repetitions are then processed to fuse consecutive repeated content that belongs to the

same content sequence (a commercial sequence for example). In the third step, the set of repeated content are classified in order to separate P ones from B ones. The latter are used in the fourth step to provide a Program/Break segmentation of the stream. Finally, this segmentation is labeled using the metadata provided with the stream such as the EPG.

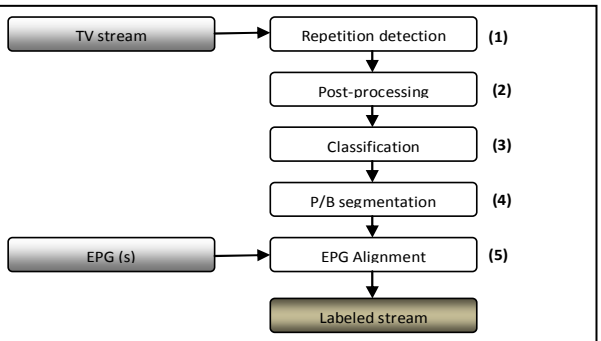

Figure 1: System overview: TV stream and EPG(s) are the inputs and labeled stream is the output

2.1 Repetition Detection

Detecting the boundaries of programs in a TV stream is a hard step. Usually, program segments (P) are separated by non-program ones that we will call breaks (B). Commercials, trailers, station identification breaks (monochrome frames for example), or bumpers can be such examples. The B segments have repetitive behaviors. They are frequently broadcasted several times with no modification except the broadcast and compression noise. Detecting these breaks (B) can help to recover the stream structure. Thus, the first step of our system consists in detecting the set of content in the TV stream that repeats more than once. They may be of break type and are used later to segment the stream in P/B sequence.

Our method of repetition detection is based on the work of Naturel et al. [2]. Naturel et al. uses a reference video database to retrieve the segments of the stream present in a database. The database contains a 24 hour of annotated TV stream and can be updated automatically. The aim of our method is to delete this manually annotated reference video database to provide a fully automatic method. We adopt the same method to detect the repeated content. The method is modified to take into account the new hypothesis (no reference video database).

To detect the repeated content in the stream, we firstly segment the stream in shots using an adaptative thresholding of luminance histogram [3] with improvements to detect dissolves and fades. A 64-bit visual signature is extracted from each frame. It is based on the 64 lower frequency coefficients (except the DC coefficient itself) extracted from the DCT of the frame luminance channel.

Then, we insert these signatures in a hash table with reference to the shot the corresponding frame belongs to. By this way, we have the possibility to compare the stream with itself in order to detect the set of content that repeats more than once. To do that, for each signature, we retrieve the shots that have this signature and approximately the same duration. A distance between each couple of shots is computed. This distance measures the mean of the hamming distance between the signatures of the shots. The hamming distance between two signatures u and v measures the number of different bits between them. It is defined as follows:

$Hamm(Sig_i , Sig_j) = \sum_k Sig_i [k] \oplus Sig_j [k]$, k=1,..64

The distance between two shots Sh_i and Sh_j having the same duration is defined by:

$D(Sh_i,Sh_j) = 1/N (\sum_k Hamm(Sig_{ik} , Sig_{jk}))$ for k=1..N

where Sig_{ik} (Sig_{jk} resp.) is the signature of the frame number k of the Sh_i (Sh_j resp.) and N is the number of frames in the two shots.

In the case where the two shots are of different duration, the middle frame of the first shot is aligned with the middle frame of the second one. The frames at the boundaries that don't have associated ones in the second shot are discarded when computing the distance.

Two shots are considered as having the same content if their distance is less than a fixed threshold.

2.2 Post-Processing Step

In the repetition detection process, we have taken the shot as the basic unit to compare if two shots are of same content or not (i.e repeated content). A repeated sequence (a commercial for example) may be composed of several shots. The aim of the post-processing step is to fuse contiguous shots in order to get repeated content of sequence type not of shot one.

Three rules for fusion are proposed. In the first rule, we deal with the case where the content of a segment (a commercial for example) is re-broadcasted entirely several times and that we have detected all its repeated content. In the second rule, we will address the problem of a segment that is not re-broadcasted entirely. For example, a commercial may be shortened after one or several broadcasts, an error in the shot segmentation or the repetition detection may have occurred. In the third rule, we drop short segments (less than one second) in favor of getting longer repeated content.

For simplicity, we denote $\sum$ the set of repeated content and $\sum_i$ is the ith repeated content in the stream. Each $\sum_i$ is composed of the set of segments that have this content. Let S_{ij} be the j^{th} segment of the i^{th} repeated content. Each segment S_{ij} is represented by its start time (S_{ij}.start) and end time (S_{ij}.end) in the stream. The set $\sum$ is sorted by the increasing start time of the first segment in each $\sum_i$. We will explain the proposed fusion rules directly on an example. We will suppose having a commercial segment C composed of N shots and that it appears M times in the stream. Other repeated content may also be detected but, for simplicity, we take a stream that contains only this repeated commercial.

The first case is where $\sum$ contains N repeated content ($\sum_i$) and each $\sum_i$ is composed of M segments. Let $|\sum_i|$ be the number of segments in the repeated content $\sum_i$. The objective here is to fuse all the $\sum_i$ in order to get a repeated content that represent the whole commercial C.

This repeated content should contain M segments that represent the repetitions of commercial C. Two consecutives repeated content are fused if they have the same number of segments ($|\sum_i|=|\sum_{i+1}|$) and their distance is less than a fixed threshold. The distance between $\sum_i$ and $\sum_j$ is defined as follows:

$d(\sum_i, \sum_j) = \{ [S_{jk}.start - S_{ik}.end]$ for each k=1..M$\}$

Then, the mean and variance (mean,var) of the distance is computed and compared to a threshold (α,β) to decide if the two repeated content should be fused.

This distance takes into account the case where some repeated content are not detected. In other words, some $\sum_i$ ($1<i<N$) are not detected as repeated content due to segmentation or repetition detection problem. The case taken into account here is when the whole content, some $\sum_i$ (all their segments), are not detected.

The second post-processing rule takes into account the case where two consecutive $\sum_i$ that are parts of the same content (the same commercial for example) doesn't have the same number of segments. In other words, for a specific $\sum_i$, the process was unable to detect all its repeated segments. This problem may be due to one of three situations. The first one is due a problem in the detection process. It can be also due to a shot segmentation problem that has differently segmented the repeated content. The third situation which happens usually is when the broadcasters have shortened the content after one or several broadcasts. In the $\sum$ set, it can be seen as two consecutives repeated content ($\sum_i$, $\sum_{i+1}$) that belong to the same content (the same commercial for example) and $|\sum_i| > |\sum_{i+1}|$. In this case, we search if there exists a $\sum_j$ with $j>i+1$ and $d(\sum_i, \sum_j) < (\alpha,\beta)$.

The third post-processing rule try to fuse two repeated content even if they don't have the same number of segments by dropping some segments in a $\sum_i$ if the duration of segments is less than one second. In this rule, we try to get the longer repeated content we can that will facilitate the analysis step afterward by dropping some short segments that will not have a significant impact on the structuring step as they are of short duration. In the $\sum$ set, two consecutives repeated content ($\sum_i$, $\sum_{i+1}$) are of different cardinality $|\sum_i| < |\sum_{i+1}|$. In this case, we search if there exists a $\sum_j$ with $j>i+1$ and $d(\sum_i, \sum_j) < (\alpha,\beta)$. The $\sum_k$ with $i<k<j$ that have $|\sum_k| > |\sum_{i+1}|$ should have segments of short duration (less than one second).

If two repeated content $\sum_i$ and $\sum_j$ should be fused, all the $\sum_k$ (k=i+1...j) are removed from $\sum$ and $S_{ik}.end$ are replaced with $S_{jk}.end$ (k=1,2,… $|\sum_i|=|\sum_j|$).

In the tool, the set of repeated content is presented as a tree where each node contains the whole set of segments that correspond to the same content.

2.3 Classification

The result provided by the post-processing step is a set of repeated content noted $\sum$ where each $\sum_i$ contains the set of segments S_{ik} that have the same content. Each segment is represented, as noted before, by its start time $S_{jk}.start$ and end time $S_{ik}.end$. Each repeated content $\sum_i$ may be of diverse nature: B segments or P segments like opening and closing credits, programs broadcasted twice and small programs like weather forecast. Considering $\sum$ as containing only break segments and use it to segment the stream will generate errors due to labeling the P segments as B ones. That is why a classification step should be applied to separate the P $\sum_i$ from B ones.

Two different methods can be used according to the data to be classified. In addition, the user has the possibility to choose one of the following classifiers: SVM (RBF), Classification tree (C. T.), Random forest (R. F.), KNN (10NN), C45, and Naïve Bayes (N. B.).

In the first method, each segment S_{ik} of each set $\sum_i$ in $\sum$ is described by a set of local and global features and then classified. In our system, we call this method the segment-based classification. Each segment may be of P or B type.

In the second method, each set $\sum_i$ in $\sum$ is described by a set of global features and then classified. This method bases on the fact that $\sum_i$ contain the same segments that have the same content. Thus, the segments S_{ik} in $\sum_i$ may have the same type except some cases that we will present later. We reference this method by RepSet-based classification. Each $\sum_i$ may be of P, B or T (trailer) type. The trailer type is emerged from the case where usually a trailer sequence announce a program that will appear later in the stream. This sequence contains extracts from the announced program. These extracts are broadcasted as B segments in the trailer sequence and then they appear in the program as part of program sequence. In this case, a repeated content may contain segments of P and B type.

To describe the segment in the first method, a set of local and global features are extracted from each one.

Four global features measure the number of times the content of this segment was broadcasted, the number of different calendar days where the content appears, the number of days of the week where the content appears and the mean duration of all the segments that have this content.

The local features are issued from two sources: the first is the presence or not (as binary feature) of a separation before and/or after a segment. The separation represents the simultaneous occurrence of monochrome frames and silence that happens usually before, after and between commercials. To detect the separations, we have used the method proposed by [2]. The second source of information describes locally a segment basing on the neighboring ones. Features that measure the number of segments that have repeated content in a window before and a window after and theirs repetitions are used. The application of the min, max and average operations are used to derive such features.

By the same way, each repeated content $\sum_i$ in the second method is described by a set of global features. In addition to the global ones used to describe a segment, we have proposed a set of features that are derived from the local features of the S_{ik} in $\sum_i$. The set of features are obtained by application of the min, max and average on the set of the local features, presented above, of the S_{ik} in $\sum_i$.

In our tool, users have the possibility to evaluate the classification of the repeated content. In this feature, the user provides the ground truth files of the annotated stream, chooses one or several classifiers and a random sampling method or a cross-validation one.

2.4 P/B Segmentation

In the previous step, the set of segments that have repeated content are classified as P or B segment by doing segment-based classification or RepSet-based one. Using a RepSet-based classification, each repeated content $\sum_i$ is labelled as P, B or T content. In this step, the stream is segmented in P/B sequences. This segmentation is done in three passes: pre-segmentation, classification and fusion.

In the first pass, all the segments that are classified as breaks are retrieved from the stream. The stream is then segmented in pre-segments where each one has a start time and end time.

Let Stm = $\{S_i / S_i = (S_i.start , S_i.end) \}$ represents the segmented stream.

The aim of the second pass is to classify each segment in Stm as being a program segment or B one. The classification bases on the length of segments. A fixed threshold d_{min} is used in this pass to label the segments longer that d_{min} as P segments and the others as B ones. Experimentations are done by Naturel et al. [2] and led them to fix the threshold to one minute.

In the third pass, we fuse the consecutives B segments into one segment. At the end of this step, we obtain a new segmentation of the stream as P/B sequence.

2.5 EPG Alignment

Once the stream is segmented, the next step is to add a label to each segment especially the program ones. Two types of analysis methods may be used here to label the segmented stream: content-based analysis or meta-data analysis. In our work, we have based on the metadata broadcasted with the stream, namely EPG (Electronic Program Guide). The EPG contain useful information about the programs broadcasted. We can find the title, genre and sometimes other information such a short description, the list of actors...

The method proposed to label the segmented stream aligns it with the EPG using the Dynamic Time Warping (DTW) algorithm. It is the well-known method that computes a path and a distance between two sequences X and Y. The distance may be interpreted as being the cost to transform X into Y by a set of weighted edition operations. The operations are the insertion, deletion and substitution. The best alignment is provided by the path with minimal cost. In our system, a distance is computed between segments which measure the similarity of durations, start and end time of the two segments.

$dist(seg_i, seg_j) = | d_i - d_j | + | s_i - s_j | + | e_i – e_j |$

where d_i (d_j resp.) is the duration of seg_i (seg_j resp.), s_i (s_j resp.) is the start time of seg_i (seg_j resp.) and e_i (e_j resp.) is the end time of seg_i (seg_j resp.).

The cost of the insertion, deletion and substitution operations are defined as:

$C_{del} = dist(seg_i, seg_j)$; $C_{ins} = dist(seg_i, seg_j)$; and

$C_{sub} = \alpha * dist(seg_i, seg_j)$ where $1 < \alpha < 2$

The α parameter is used to favor a substitution operation over a deletion followed by an insertion one.

For more information about the used method, you can refer to the work of Naturel et al. [2].

3 EXPERIMENTS

To evaluate our method, we used a corpus of 22 consecutive days of TV recorded from a French channel (France2) for the period from 9/5/2005 to 30/5/2005. The evaluation concerns only the classification, the segmentation and the alignment steps. The repetition detection and the port-processing steps cannot be evaluated because of the impossibility to manually annotate a database in terms of repeated content. In our experiments, we compare our method to the one proposed by Naturel et al. [2] for three reasons. The first is that our method tries to overcome the drawbacks of their work. The second is that to our knowledge, their results can be considered the best obtained in the stream structuring area. The third is that we could use the same database with the same ground truth. In this section, we present a summary of the results obtained due to lack of space. The full experimentation results will be published soon.

3.1 Classification Evaluation

As mentioned in 2.3, two different methods can be used in this step according to the data to be classified. It can be a segment-based classification (each S_{ik} in each $\sum_i$ is classified) or RepSet-based (each $\sum_i$ is classified).

3.1.1 Segment-based Classification

As we have already mentioned, each segment may be of P or B type. Table 1 and Table 2 show the precision and recall using several classification and sampling methods.

Table 1: Segment-based classification using cross-validation (C. V.) sampling method

C. V. 10-folds	P		B	
	Prec.	Rec.	Prec.	Rec.
Random forest	96.38%	98.01%	97.66%	95.76%
Classification Tree	97.29%	97.48%	97.09%	96.87%
C4.5	97.12%	97.29%	96.88%	96.68%
KNN(10NN)	95.98%	96.66%	96.12%	95.33%
Naïve Bayes	92.36%	94.93%	93.97%	90.95%
SVM(RBF)	92.31%	95.54%	94.65%	90.83%
CN2	95.77%	98.34%	98.03%	95.00%

Table 2: Segment-based classification using random sampling (R. S.) method (30% to train and 70% to test)

R. S. iterated 5 times	P		B	
	Prec.	Rec.	Prec.	Rec.
Random forest	96.17%	97.08%	96.59%	95.55%
Classification Tree	96.29%	96.76%	96.25%	95.71%
C4.5	96.46%	97.03%	96.56%	95.89%
KNN(10NN)	95.16%	96.19%	95.56%	94.36%
Naïve Bayes	92.41%	94.88%	93.92%	91.02%
SVM(RBF)	94.37%	96.44%	95.79%	93.38%
CN2	95.31%	97.71%	97.28%	94.46%

3.1.2 RepSet-based Classification

In this case, each repeated content $\sum_i$ may be purely P segments, purely B segments or Trailer (T). The latter contain segments of P and B type. Table 3 and Table 4 show the precision and recall using the same classification and sampling methods.

Table 3: RepSet-based content classification using cross-validation sampling method

C. V. 10-folds	Trailers		Programs		Breaks	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
R. F.	62.16	31.08	97.75	98.19	93.24	93.49
C. T.	43.14	29.73	98.13	97.45	91.31	94.20
C4.5	37.20	23.46	97.65	97.52	91.35	92.83
KNN	42.11	21.62	97.621	96.39	88.24	93.04
N. B.	0.89	78.38	99.89	13.08	95.29	73.10
SVM	72.73	10.81	97.88	97.12	90.00	94.56
CN2	64.86	32.43	97.02	97.94	92.82	91.66

Table 4: RepSet-based classification using random sampling method

R. S. iterated 5 times	Trailers		Programs		Breaks	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
R. F.	62.50	13.46	97.85	97.82	91.87	94.34
C. T.	29.91	13.46	98.00	97.23	90.49	94.28
C4.5	36.54	25.68	97.72	97.77	92.17	92.95
KNN	28.57	9.23	97.41	96.05	87.15	92.83
N. B.	0.92	76.54	99.94	35.23	93.96	77.69
SVM	72.73	10.81	97.88	97.12	90.00	94.56
CN2	27.16	8.46	97.89	97.07	90.22	94.59

3.1.3 Comparing Segment-based and RepSet-based classification

In order to compare the results obtained with RepSet-based to those obtained with segment-based, the former should be translated in terms of segments. To this aim, we compute the percentage of well-classified segments that are produced by the RepSet-based classification. Table 5 gathers the results in term of number of RepSet (column 1) or segments (columns 2 and 3) correctly classified. Columns 1 and 3 correspond to the classification of the RepSet and the segments respectively. Column 2 is the translation of column 1 in terms of correctly classified segments (only column 2 and 3 can be directly compared). As Table 5 shows, the accuracy of the classification is more than 96% especially in RepSet-based classification.

Table 5: comparison between RepSet-based and segment-based classification

C.T.	96.67%	95.57%	93.5%
R. F.	97.4%	**96.72%**	**94.73%**
SVM	95.34%	94.94%	91.32%
KNN	95.28%	94.09%	92.31%
Type	**RepSet -based**	**RepSet -based in terms of segments**	**Segment-based**

3.2 Segmentation Evaluation

In order to evaluate the segmentation step, we will consider it as a binary classification problem where each frame is classified as P or B. In the segmentation evaluation, we will consider the program segments because they are the most interesting for users. The problem is that the majority of the stream is composed of program frames, which lead to high score in the evaluation step. Consequently, we evaluate also the B segments also. Figure 2 and Figure 3 show the F-measure of the segmentation step.

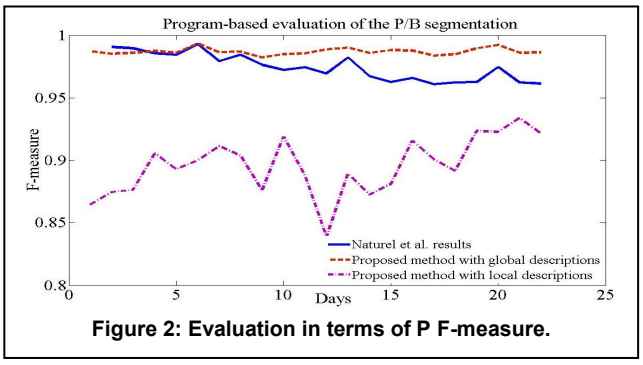

Figure 2: Evaluation in terms of P F-measure.

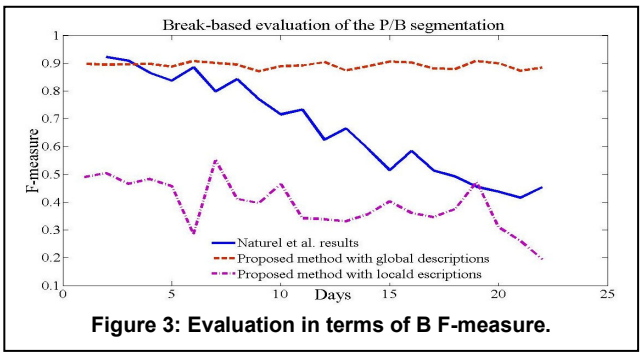

Figure 3: Evaluation in terms of B F-measure.

As we can notice in Figure 2 and Figure 3, our method is very stable over days contrarily to Naturel et al one. Their problem is probably because of the depreciation of the reference video database used in this method. Only few of the breaks of this database still have repeated behaviour after some days in the stream. The results of our method prove the efficiency of relying on repetitions detection rather than on a manually annotated reference video set.

3.3 Alignment Evaluation

To evaluate the alignment step, two measures are considered. The first takes the segment as the basic unit for evaluation. This measure is not sufficient since the short programs are usually not presented in the EPG and therefore the correct label cannot be retrieved from the EPG. Therefore, we consider a second measure where the frame is the basic unit for evaluation.

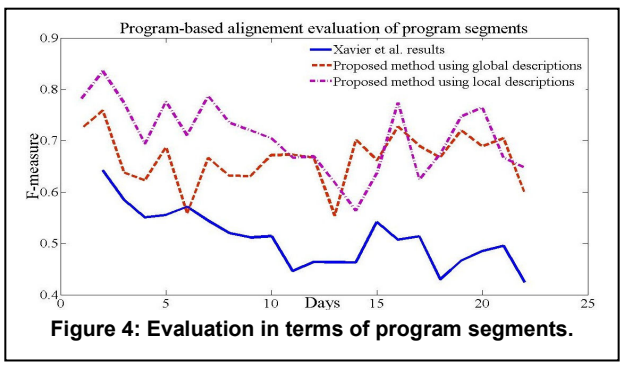

Figure 4: Evaluation in terms of program segments.

As shown in Figure 4 (program-based), the performance of our method is higher than the Naturel et al. one while they are quasi-identical in Figure 5 (Frame-based).

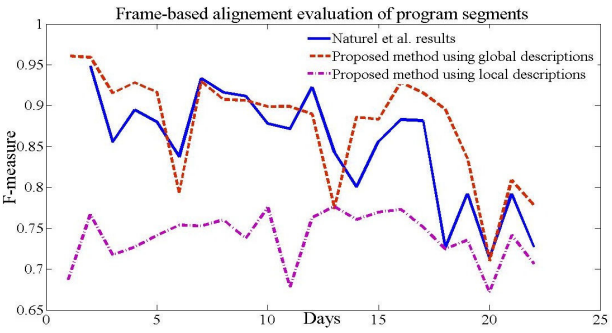

Figure 5: Evaluation in terms of program frames

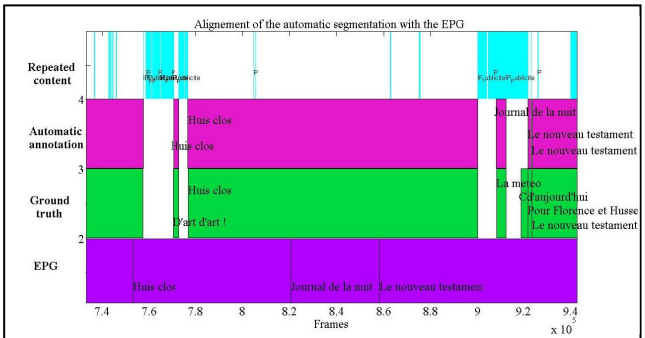

Figure 6: Portion of the alignment results of the first corpus day with its associated Electronic Program Guide.

3.4 Discussion

In this work, our aim is to overcome the drawbacks of the work of Naturel et al. in [2] in order to structure TV stream. The evaluation of the proposed method has shown its effectiveness.

In this work, we have faced three problems. The first comes from the video capturing process, the second is due to the ground-truth data and the third concern the EPGs. Concerning the first problem, we have noticed that two segments having the same content were segmented differently even when testing several shot segmentation tools. The cause of this limitation can be the video capturing process, which is affected by several broadcasting and capturing effects. This problem has impacts on the repetition detection and the post-processing steps.

The second problem is that human annotators have not considered the B segments broadcasted during a P segment. In other words, a publicity segment broadcasted in a program segment is annotated as Break segment. Consequently, some segments are well-classified as B segments in the classification step but counted as if wrongly-classified. This problem has also affected the evaluation of the segmentation step.

A third problem is encountered during the alignment process and comes from the fact that small program segments are not announced in the EPG. Our method detects the start and the end of these programs but their label cannot be predicted since it is not presented in the EPG. Most of the time, the alignment algorithm doesn't add labels to these segments. The use of a manually annotated database as in Naturel's work may somehow overcome this problem since the database contains some of these programs.

Figure 10 shows an example of such a problem. As we can see, the "Dart d'art" program segment is not announced in the EPG. However, its accurate start and end time were correctly detected and the correct P label has been assigned to the segment. But it was finally erroneously annotated, due to the lack of the correct label in the EPG. The same phenomenon occurs with "La meteo" segment which is annotated as "Journal de nuit" since the first label is not present in the EPG.

4 CONCLUSION

In this paper, we present our proposed method for TV stream structuring. The proposed method overcomes the drawback in the initial work of Naturel et al. In a first step, the repeated content are detected automatically and then post-processed in a second step. Then, a classification step is applied to separate the programs from the breaks. In its turn, the classification is achieved at the segments level or at the level of the sets of repeated segments, using a mixture between local and global features. Once classified, the segments serve to segment and classify the remaining part of the stream. Finally, the segmented stream is aligned with the electronic program guide (EPG) in order to propagate the program labels (their title most of the time.). For evaluating our method, we used the same corpus as Naturel's and our results proved the efficiency of the proposed solution. A video that shows the execution of the tool is made available on:

www.zibrahim.info/AVSST_Video.avi.

References

[1] Rui Y., Huang T.S. and Mehrotra S. 1999. Constructing Table of Content for videos. In *Journal of ACM Multimedia Systems*, Vol. 7, No. 5, pages 359-368.

[2] Naturel X., Gravier G. and Gros P. 2006. Fast Structuring of Large Television Streams using Program Guides. In *4th International Workshop on Adaptive Multimedia Retrieval (AMR)*, Volume 4398, pages 223-232, Switzerland.

[3] Manson G. and Berrani S.A. 2008. An Inductive Logic Programming-Based Approach for TV Stream Segment Classification. In *Proceedings of the IEEE International Symposium on Multimedia*, pages 130-135, USA.

[4] J. Poli. 2008. An automatic television stream structuring system for television archives holders. In *Multimedia systems*, 14(5), pages 255-275.

[5] Berrani S.A., Lechat P. and Manson, G. 2007. TV broadcast macro-segmentation : Metadata-based vs. content-based approaches. In *Proc. Of the ACM Int. Conf. on Image and Video Retrieval*. Netherlands.

[6] Truong B. T., Dorai C. and Venkatesh, S. 2000. New enhancements to cut, fade, and dissolve detection processes in video segmentation. In *proceedings of the 8th International Conference on Multimedia*, pages 219-227, USA.