Fast Structuring of Large Television Streams Using Program Guides

Xavier Naturel, Guillaume Gravier, and Patrick Gros

IRISA - INRIA Rennes
Campus Universitaire de Beaulieu
35042 Rennes, France

Abstract. An original task of structuring and labeling large television streams is tackled in this paper. Emphasis is put on simple and efficient methods to detect precise boundaries of programs. These programs are further analysed and labeled with information coming from a standard television program guide using an improved Dynamic Time Warping algorithm (DTW) and a manually labeled reference video dataset. It is shown that the labeling process yields a very high accuracy and opens the way to many applications. We eventually indicate how the dependency to a manually labeled video dataset can be removed by providing an algorithm for a dynamic update of the reference video dataset.

1 Introduction

Television is designed to be watched live. Accessing archived television streams is problematic because no metadata describing the structure of the stream exists. It is therefore difficult to extract information from unlabeled TV archives. Television program guides provide the kind of interesting information for retrieving programs (title, genre, date, abstract) but they are far from accurate. Schedules are not respected, some programs are missing while some are inserted, and of course commercials are not indicated. Program guides still carry important information that would be very difficult to extract from the stream, like program title, genre and an approximate time of broadcast.

Our goal is to synchronize the video stream with the program guide, removing errors and adding information when posible, thus building a more accurate post-diffusion program guide which enables easy browsing and retrieval from large television archives.

This can perhaps be seen as a trivial problem, since such a synchronization could be easily done by channels, such as in the European Standard PDC [1]. However, this system cannot be used for archives, and even for live streams it is not satisfactory. Most channels are reluctant to use such a system because of its commercial skip capability, and even for the channels that use it, not every program has a PDC signal. Finally, audiovisual regulation authorities[1] cannot

[1] Such as the CSA (Conseil Supérieur de l'Audiovisuel) in France, which monitors the number and duration of commercial breaks, subjected to legal regulations.

S. Marchand-Maillet et al. (Eds.): AMR 2006, LNCS 4398, pp. 223–232, 2007.

rely on a channel-provided signal if they are to monitor the stream to detect frauds.

The method is divided into two parts: segmentation and labeling. Section 2 first gives a short overview of the method. Section 3 explains the segmentation process and shows that a combination of standard methods and detection of duplicates leads to an effective segmentation of large television streams into programs. Section 4 is dedicated to the assignement of labels coming from the program guide to the stream segments. In section 4.3 labeling results on three days of TV show that the method can handle large amounts of video. A final section acknowledges the need for a dynamic update of the reference video dataset and proposes an algorithm to solve this problem.

2 Overview of the Method

The general flow of the method is given in Figure 1. The first step is to segment the stream to find boundaries of programs. This gives a segmentation into programs/non-programs. A first labeling is then done using duplicate detection, i.e. detecting segments that have already been broadcasted and which have been then labeled and stored in a database (see section 3.2). This labeling is very precise since it is done at the shot level, but it labels only a very small part of the stream, mainly non-program segments. The most important part of the algorithm is the LDTW (Landmarked Dynamic Time warping) algorithm which finds the best alignment between the automatic segmentation and the EPG (Electronic Program Guide) segmentation, while taking into account labels given by duplicate detection.

The post-processing step is present to resolve ambiguity between possibly different labeling (coming from LDTW and duplicate detection).

3 Stream Segmentation into Programs

A lot of work has been devoted to find commercials in a video stream. One of the most used and effective technic is to take advantage of the rule that black frames are inserted between commercials [2,3]. While true for most of the channels and countries this method leads to a very high rate of false alarms and must be used together with another feature to provide acceptable results. Sadlier et al. [3] used silence detection, and popular features are shot frequency, motion activity [2], or text presence [4]. Another efficient method is to recognize previously known commercials [2,5]. However the drawback of the constant need to update the database of commercials used for recognition is not addressed.

Detecting program boundaries is however not equivalent to detecting commercials. Most previous works have regrettably been elusive about what is included in the term commercial, and few methods are tested over large and heterogeneous datasets. Some recent publications have not these limitations however [6,7]. In this paper, non-program segments are defined as segments that can be composed

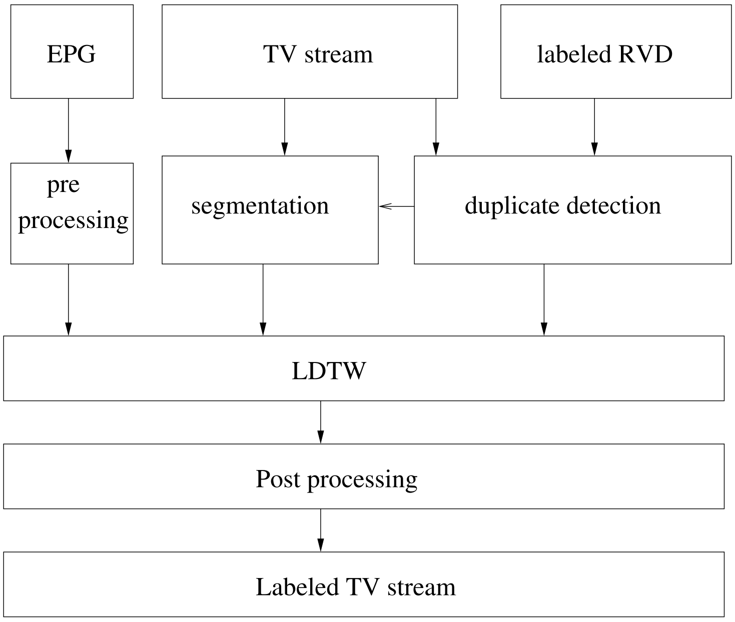


Fig. 1. Overview of the algorithm for labeling TV streams

of commercials, trailers/teasers, sponsorship or channel jingles. As its name indicates, a non-program segment is everything which is not a regular program, i.e. news, talk-show, soap opera, weather broadcast...

The proposed method for segmenting a TV stream uses two kind of independant information. The first is the classical monochromatic frame and silence indicator, explained in section 3.1. The second kind of information comes from a duplicate detection process. Non-programs are recognize as such because they have already been broadcasted and are present in a labeled reference video dataset. section 3.2 explains the process.

3.1 First Step: Using Silence and Monochromatic Frames

Silence detection emerged as a very reliable indicator of presence of commercials, at least on French television. Figure 2 shows the log-energy of the audio signals over a 1-hour duration and actually shows that energy is null between two commercials. Considering Figure 2, a simple thresholding of the log energy achieves almost flawless results. Note that the threshold does not change when considering different channels. From the image point of view, commercials are separated by black but also white or blue frames. A simple monochrome frame detector is constructed using the entropy of the luminance histogram. It unfortunately produces a very high rate of false alarms (see Table 1) since monochrome frames may appear anywhere in the stream, not only at commercials boundaries.

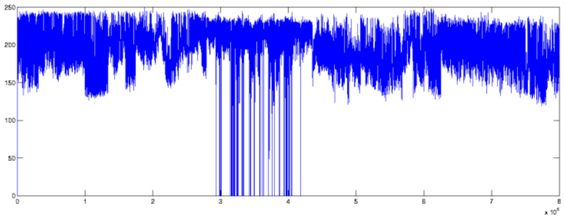


Fig. 2. Log-energy of the audio signal on a 1-hour television stream

Combining results from multiple media is usually not straightforward. Since the audio feature is far more reliable than the image one, we use a successive analysis approach, by first detecting the silence segments, then performing monochrome image detection in an enlarged window around these silence segments. The main problem is that while this method works quite well for detecting commercial breaks, it is not suited to detect others non-program segments because these are not flagged by monochrome frames and silence.

Table 1. Commercial break detection on a three hours videoset

Modality	Precision	Recall
Audio	82	90
Image	41	89
Fusion	100	90

3.2 Second Step: Improving Segmentation with Detection of Duplicates

An effective way to detect non-program segments is to look for duplicate sequences. By duplicate we mean that the content is the same, minus transmission noise and very small edition effect (modifications due to progressive transitions or small insertions). Because non-program segments are very repetitive, duplicate detections really helps to segment the stream into program/non-program segments. It is proposed in [8] to compute on each image a signature which is the concatenation of binarized low-frequency DCT coefficients. This signature is sufficiently robust to noise to be queried by exact matching, thus allowing the use of a fast retrieval structure like a hash table. The retrieved shots are further analyzed by computing a similarity function defined as the average Hamming distance between the signatures of the retrieved and query shots. More details can be found in [8].

In order for this method to be interesting, it has to have a (manually) labeled Reference Video Dataset (RVD). The RVD is a set of labeled shots. A detection of a duplicate between the query and the RVD thus results in labeling the query shot by the label of the found shot. For all our tests, 24 hours of continuous TV were labeled, indicating program names and distinguishing non-programs segments between commercials, trailers, sponsorship or other.

3.3 Information Fusion and Results

The results of the detection of duplicates and the monochromatic/silence detector are expressed as a set of images considered as non-program. Let X_1 be the set of images for silence/monochromatic detection and X_2 the set for duplicate detection, with S the entire set of images from the input stream. A pre-segmentation is then computed by $Y = S \setminus (X_1 \cup X_2)$. The resulting set Y can be seen as a set of segments, which are then classified according to their length, a small segment being a non-program segment while a large one is considered as a program segment. The threshold is set to 60 seconds. Despite this very crude definition of a program segment, results given in Table 2 are satisfactory. This table shows the results using only silence and monochromatic frames (Method 3.1) and the method using both silence/monochromatic and duplicate detection (Method 3.3).

Table 2. Program/non-program detection on three days of TV

	Program		Non-program	
Method	Precision	Recall	Precision	Recall
Method 3.1	97.9	99.7	97.2	82.8
Method 3.3	99.5	99.8	98	95.8

4 Automatic Labeling

As a pre-processing step, the EPG can be modified to become more realistic. Domain knowledge about the channel can be included in this pre-processing step. However only one simple rule is used here. It simply states that long programs (more than 1h30) should be cut into at least two parts (because they will usually be cut by commercials).

The next step is to match the stream segmentation with the program guide. A well-known method for aligning two sequences is the dynamic time warping algorithm (DTW) [9]. The DTW between 2 sequences X and Y is the minimum weight for transforming X into Y by a set of weighted edit operations. These operations are most of the time defined as substitution, insertion and deletion. The path used to reach this minimum weight provides the best alignment between X and Y, with respect to the edit operations. DTW can be efficiently computed by dynamic programming.

Given a segmentation of the stream $X_i = \{x_0 \dots x_i\}$ and the associated program guide $P_j = \{p_0 \dots p_j\}$ the DTW is given by:

$$D(X_i, P_j) = Min \begin{cases} D(X_{i-1}, P_{j-1}) + C_{sub}(x_i, p_j) \\ D(X_i, P_{j-1}) + C_{del}(p_j, i) \\ D(X_{i-1}, P_j) + C_{ins}(x_i, j) \end{cases}$$

Each element of X and P are a couple of values indicating the start and end of the program $x_i = (x_i^s, x_i^e)$. $C_{sub}, C_{del}, C_{ins}$ are respectively the costs of substitution, deletion and insertion and are application dependant. A classical definition is:

$$\begin{aligned} C_{sub}(x_i, p_j) &= \gamma d(x_i, p_j) \\ C_{del}(p_j, i) &= d(x_i, p_j) \\ C_{ins}(x_i, j) &= d(x_i, p_j) \end{aligned}$$

$1 < \gamma < 2$ to favor a substitution over a deletion+insertion; $d(x_i, p_j)$ is the local distance between the vectors coordinates of X_i and P_i and is defined as:

$$d(x_i, p_j) = \alpha|p_j^e - p_j^s - (x_i^e - x_i^s)| + \beta \left[|p_j^s - x_i^s| + |p_j^e - x_i^e|\right]$$

This local distance is designed so as to measure the similarity of length of x_i and p_j (first part, weighted by α) as well as the closeness of their broadcast time (second part, weighted by β). It has been experimentally observed that the two terms contribute in a somewhat equivalent way to the alignement, so that we have in fact $\alpha = \beta = 1$.

The DTW with this local distance is robust but it is unfortunately not error-free (see Table 4.3). Two improvements are proposed in the next sections.

4.1 Adding Landmarks in the DTW

In building the alignment method, we overlooked an important source of information: the labels attached to some program segments. These labels come from the detection of duplicates of section 3.2. While programs rarely get repeated entirely, lead-in, lead-out or special sequences specific to a program are frequently repeated. Suppose that such a detection is found in the middle of an unlabeled program segment x_i (prior to DTW). Attached to this detection is a label, indicating its type and title. If this title is found nearby in the EPG, yielding a program p_j, then one would like to force the DTW to go through the landmark (i, j).

The idea is to prune all paths that do not go through (i, j) by filling the cost matrix before computing any costs with:

$$\begin{cases} d(x_i, p_j) = 0 \\ d(x_l, p_k) = \infty \end{cases}$$

$\forall(k, l)$ such as $(k < j \ , \ l > i)$ et $(k > j \ , \ l < i)$.

Figure 3 shows this method for landmark (i, j). The same process is applied for every duplicate that match with an EPG label. The cost matrix is then computed by the (almost) standard DTW algorithm. This modified algorithm is called Landmarked DTW (LDTW).

4.2 Choosing Best Labels

The last improvement we introduce is a post-processing method, which therefore takes place after the DTW. The problem is the following. If duplicates have been

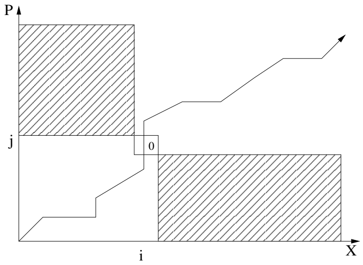


Fig. 3. Forcing local alignement in the DTW cost matrix

detected inside a program segment, it may happen that the duplicates have a different label from the one attached to the segment by the DTW. This problem arises especially when information is lacking in the EPG. Consequently the label given to the segment by the DTW cannot be right.

Two hypothesis are defined: H_0, the correct label comes from the detection of duplicates; H_1, the correct label comes from DTW. Given an observation O, the decision is made via a Bayesian hypothesis test:

$$\frac{P(O|H_1)}{P(O|H_0)} > \frac{P_0}{P_1} \text{ then } H_1 \text{ else } H_0$$

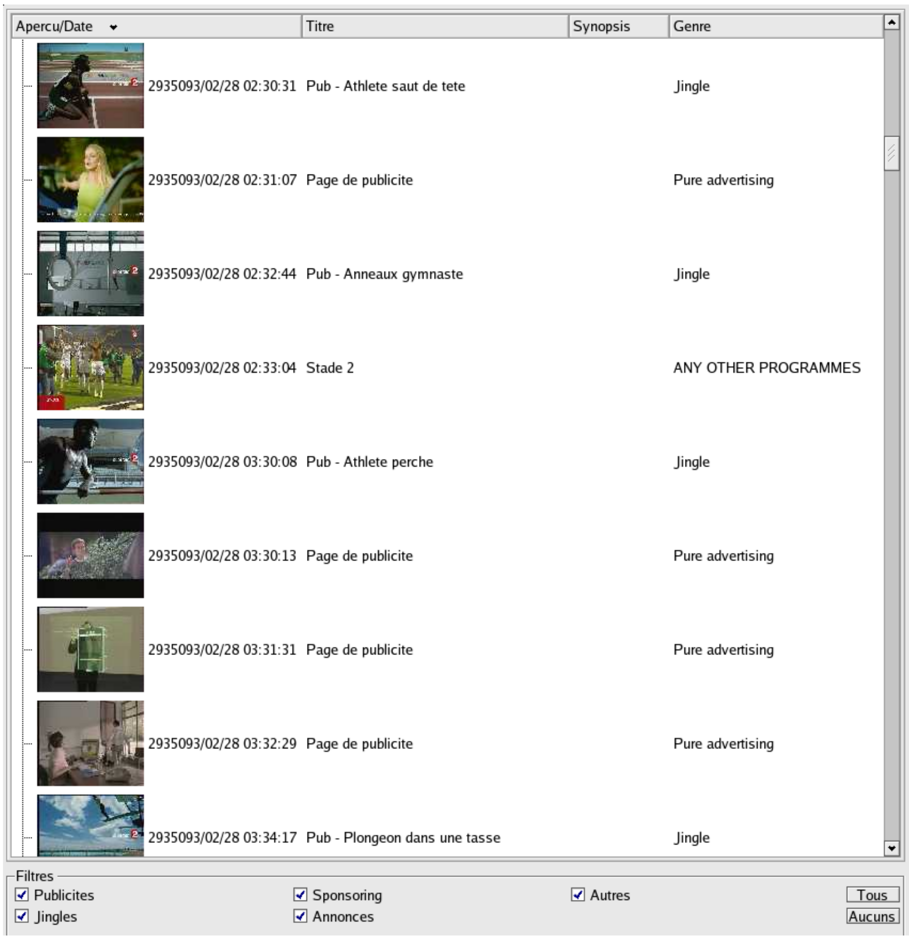
where P_i is the prior probability of hypothesis H_i. To estimate $P(O|H_i)$, the observation O is considered to be made of elementary independant observations o_k. These are then easy to estimate using a training corpus. We have:

$$P(O|H_i) = \prod_k p(o_k|H_i)$$

Three elementary observations are defined. o_1 is the length of the segment, and is considered to be gaussian. o_2 and o_3 are binary observations, which are true when a duplicate exists respectively at the beginning and at the end of the program segment. o_2 and o_3 are defined to take into account that program lead-in and lead-out are often well detected and are more likely to yield the correct label than the DTW.

4.3 Results

Evaluation of the correct labeling of program segments is not straightforward. To have a clear view of the performance of the labeling process two statistics are given in table 3: correct labeling on a frame-by-frame basis (*Image*), and correct labeling on a program basis (*Program*). The former takes into account the precision of the segmentation while the latter reflect only the quality of labeling. All methods include the pre-processing step. Only LDTW2 includes the post-processing step. As a reference, the scores obtained by the EPG are also given. Since there is no miss in the EPG, only false labeling, the recall score is always 100%.



Apercu/Date	Titre	Synopsis	Genre
2935093/02/28 02:30:31	Pub - Athlete saut de tete		Jingle
2935093/02/28 02:31:07	Page de publicite		Pure advertising
2935093/02/28 02:32:44	Pub - Anneaux gymnaste		Jingle
2935093/02/28 02:33:04	Stade 2		ANY OTHER PROGRAMMES
2935093/02/28 03:30:08	Pub - Athlete perche		Jingle
2935093/02/28 03:30:13	Page de publicite		Pure advertising
2935093/02/28 03:31:31	Page de publicite		Pure advertising
2935093/02/28 03:32:29	Page de publicite		Pure advertising
2935093/02/28 03:34:17	Pub - Plongeon dans une tasse		Jingle

Filtres

☑ Publicites ☑ Sponsoring ☑ Autres Tous

☑ Jingles ☑ Annonces Aucuns

Fig. 4. Example of TV stream labeling. Program labels, e.g. 'Stade2' on this figure, usually comes from the EPG while labels for commercials, trailers and sponsorship comes from the database.

The difference between the *image* and *program* score may be surprising. It is explained by the fact that a lot of small programs are wrongly labeled. This does not really impact the *image* score, while it dramatically affects the *program* one. Most of these errors are due to a lack of information: labels are available neither from the program guide nor from the labels of the detection of duplicates. Future works include the possibility of using text detection to retrieve the correct program label directly from the stream. Figure 4 shows an exemple of labeling on 1 hour of french television.

The labeling process only takes a few seconds once the features have been computed. The feature extraction, which includes shot segmentation, silence/

Table 3. Percentage of correct labeling on three days of TV

	Image		Program	
	Precision	Recall	Precision	Recall
EPG	77.3	100	48.4	100
DTW	88	99.9	55.6	83.5
LDTW	91.6	99.9	62.1	84.9
LDTW2	92.8	99.9	78.7	88.1

monochrome frame detection and detection of duplicates, is very fast since it runs at a frame rate of 115 frames/s on a standard 3Ghz PC.

5 Updating the Labeled Reference Video Dataset

As stated in the introduction, the major drawback of using a manually labeled reference video dataset is the need to update it as new non-program segments appear. This problem can be overcome by iterating successively duplicate detection and the proposed segmentation/labeling process. Figure 5 details the procedure.

The idea is to feed the RVD with new non-program segments found by the LDTW algorithm. The updated RVD is then used to improve the accuracy of the initial segmentation which in turn might affect the labeling. This algorithm is thus used both to update the RVD and to improve the accuracy of the labeling process by using a recently updated RVD. The convergence condition is satisfied when no new labels are found, i.e. the labeling is stable. This algorithm is currently under test.

```
Function(RVD, query : video stream)
do
list_duplicates = find_duplicates(RVD,query);
update_segmentation(list_duplicates, segmentation);
Labeling = LDTW(list_duplicates, segmentation);
update_RVD(RVD, Labeling);
until (convergence)
```

Fig. 5. RVD update algorithm

6 Conclusion

A complete method for indexing large TV streams has been presented. It builds on traditionnal commercial detection technics a more elaborate process which yields much more precise and useful information, paving the way for exact and enriched EPG. Simple and efficient methods leads to a very fast process, more than 4 times faster than real-time, and is shown to be effective on three days of digital TV. Applications are numerous, from TV archives management to intelligent digital VCR and TV monitoring. Future works include improving the dynamic update of the reference video dataset and extensive testing on three weeks of TV.

References

1. EBU: Ets 300 231, television systems; specification of the domestic video programme delivery control system (pdc) (1993)
2. Lienhart, R., Kuhmunch, C., Effelsberg, W.: On the detection and recognition of television commercials. In: International Conference on Multimedia Computing and Systems. (1997) 509–516
3. Sadlier, D., Marlow, S., OConnor, N., Murphy, N.: Automatic tv advertisement detection from mpeg bitstream. Journal of the Patt. Rec. Society **35** (2002) 2–15
4. McGee, T., Dimitrova, N.: Parsing tv program structures for identification and removal of non-story segments. In: in SPIE Conf. on Storage and Retrieval for Image and Video Databases. (1999)
5. Duygulu, P., yu Chen, M., Hauptmann, A.: Comparison and combination of two novel commercial detection methods. ICME (2004)
6. Covell, M., Baluja, S., Fink, M.: Advertisement detection and replacement using acoustic and visual repetition. In: MMSP'06, IEEE 8th workshop on Multimedia Signal Procesing. (2006)
7. Liang, L., Lu, H., Xue, X., Tan, Y.P.: Program segmentation for tv videos. In: ISCAS, IEEE International Symposium on Circuits and Systems. Volume 2. (2005) 1549–1552
8. Naturel, X., Gros, P.: A fast shot matching strategy for detecting duplicate sequences in a television stream. In: CVDB'05, Baltimore (2005)
9. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing **26** (1978) 43–49