An Unsupervised Approach for Recurrent TV Program Structuring

Alina Elma Abduraman
Orange Labs – FT R&D
35510 Cesson-Sévigné.
France.
alinaelma.abduraman@orange-ftgroup.com

Sid-Ahmed Berrani
Orange Labs – FT R&D
35510 Cesson-Sévigné.
France.
sidahmed.berrani@orange-ftgroup.com

Bernard Merialdo
EURECOM
06904 Sophia Antipolis.
France.
Bernard.Merialdo@eurecom.fr

ABSTRACT

This paper addresses the problem of unsupervised TV program structuring. Program structuring allows to improve browsing and interaction with TV programs. Our work addresses the structuring of recurrent TV programs like entertainment programs, shows, magazines... It relies on the automatic detection of separators in programs using a repeated sequence detection method that is applied over a set of episodes of the same TV program. Separators are short sequences that are inserted between different parts of a program and that are repeated between and/or within the episodes of the same program. The effectiveness of the approach has been evaluated over 54 episodes of daily TV programs.

Categories and Subject Descriptors

H.3.1 [**Information Storage and retrieval**]: Content Analysis and Indexing; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Algorithms, Experimentation

Keywords

TV Program Structuring, Non-linear browsing, TV Content Indexing, TV Services

1. INTRODUCTION

Automatic video segmentation and indexing provide access to meaningful/relevant parts of a video. The idea is to extract features that allow users to directly and quickly find interesting moments in a video. This processing step cannot be performed manually when dealing with the huge amount of available video content, in particular when dealing with broadcasted content (TV streams).

This paper focuses on TV program structuring. The objective is to recover the original structure of the program. In other terms, the aim of structuring is to detect the start and end times of each part of the program content. When watching, this enables users to directly access to the desired parts of the program or to skip the current part and directly go to the next one. It thus provides an advanced non-linear access functionality that could be an alternative to the basic fast forward/backward functions. The structure could also be used by interactive services, for instance, by providing specific informations or features depending on the part currently being watched.

Existing approaches for structuring can be classified into two classes. The first one uses a predefined model based on prior knowledge. These methods are limited to certain types of programs such as news [2] or sport [3, 4]. These have a well defined structure. For news, these specific methods analyze the temporal (the set, reportage, advertising, forecast, etc.) and spatial (images with the anchorperson, logos, etc.) structure. For the sport programs, events that are predetermined by the context are detected using knowledge from the domain. These methods are hence supervised.

The second class of methods tries to develop generic solutions without using any prior knowledge. It is based only on the content features (color, motion, audio...). The idea is to segment the video into "logical units" (frames, shots, scenes) and then, based on similarity measures, to group together those that are similar. Similar units are likely to belong to the same part of the video [8, 9]. This way, the resulting groups represent the main parts of the video. The basic unit commonly used is the *video shot*. Shots boundaries are detected based on visual features. A shot usually lasts few seconds. Thus, it cannot be used for structuring a program into its own parts. This is why the next stage is to apply a clustering method in order to group similar shots. This allows to obtain *scenes* that are constructed and interpreted according to the type of program and application. In this case, the structure of the video is represented by all of the scenes in the video. The definition of a scene for these approaches is very ambiguous. Many authors use the principle of the Scene Transition Graph in order to group similar shots into scenes, where the nodes are shots or clusters of visually similar and temporally neighboring shots, and the edges represent the temporal flow of the story [5, 6]. Unfortunately, the clustering of shots into scenes depends on subjective judgments of the scene definition. The notion of scene is based on human understanding of its meaning and it is not easy to find a general definition that covers all possible judgments. Methods that create the structure using scene segmentation are thus very subjective and their efficiency is difficult to evaluate.

In this paper, we propose a different approach that implements a completely unsupervised method for program structuring. As it is difficult to find a general method for all the types of existing TV programs, we focus in this work on *recurrent* TV programs. A recurrent TV program is a program composed of several “episodes” that are periodically broadcasted (e.g. daily, weekly,...). Examples of this type of programs are entertainments, game shows or mag-

EuroITV'11, June 29–July 1, 2011, Lisbon, Portugal.

azines. We explain our choice for this type of programs by the applicative interest this type of programs have, by the fact that they generally have a clear structure, and by their high frequency in TV streams. Indeed, we noticed that the number of recurrent programs on general-interest TV channels represents about 60% of the total number of broadcasted programs.

As for their clear structure, this is done on purpose by content producers in order to allow viewers to easily follow an episode of the program even if they do not watch it from the beginning without interruption. The different parts of these programs are hence generally delimited by short video sequences that we call "separators". Moreover, the duration of each part is often approximately the same over all the episodes of a recurrent program. Our approach makes use of these properties of recurrent TV programs in order to propose an unsupervised method. Our goal is to detect, in a completely unsupervised way, the separators by analyzing several episodes of the same recurrent TV program. Once we have these separators, the different parts of the TV program can be delimited. The method does not need any other prior knowledge on the structure of a program or on the number of parts that the program has.

The rest of the paper is organized as follows. Section 2 presents the proposed solution for recurrent TV program structuring. Section 3 presents the experiments. Section 4 concludes the paper.

2. THE PROPOSED SOLUTION

In order to avoid any confusion, we propose to use *program* to refer to a recurrent TV program and *episodes* to refer to the different broadcasts of the same recurrent TV program. One of the main properties of this kind of recurrent program is their clear and steady structure. In Figure 1, an example of separators is illustrated. The horizontal lines represent the timeline of different episodes of the same TV program. The boxes represent the appearance of the separators that delimit the main parts of the show. The 3 images below are extracted from three different separators of the same episode of a TV program and are meant to illustrate the appearance of a separator.

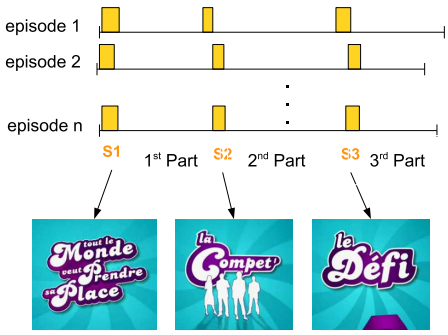

Figure 1: Example of separators.

The main idea of our approach is to detect these separators. The parts of each episode are indirectly identified using the separators boundaries, e.g. the end of a separator is the start of a new part. The resulting structure for an episode is hence composed of a set of timecodes. These timecodes refers to the start and end of each part.

To achieve this processing automatically, our solution analyzes a set of episodes of a recurrent program in order to detect the separators and to structure the episodes. It can also be used to structure a new episode by including it into a set of previously broadcasted episodes to be analyzed. The number of episodes that have to be analyzed in order to get a good structuring result ranges from two to about ten. This parameter has been experimentally studied (cf. Section 3.4).

The unsupervised detection of the separators makes use of two properties of recurrent programs:

1. Repeatability: Different episodes of the same recurrent TV program share the same structure and their separators are almost identical. This repeatability of separators can be found, as said, between episodes of the same recurrent TV program (inter-episode repeatability) but also inside a single episode (intra-episode repeatability).

2. Temporal stability: As the different parts of a program have approximatively the same duration, a separator can be found at approximately the same time-offset, for different episodes of the same recurrent TV program.

The first of these properties allows us to detect the separators as repeated sequences between the episodes of the same recurrent TV program or inside the same episode of the recurrent TV program.

The 2 main steps of our solution are described below.

2.1 Repeated sequence detection

This step performs the description of the visual content of the episodes that are going to be structured. It also proceeds with detecting the repeated sequences using the technique described in [1].

The first step is shot segmentation. For each shot, few keyframes are chosen. A two level description of the frames is used. First, a DCT-based 64-bits Basic Visual Descriptor (BVD) is calculated for each frame. Its role is to match nearly identical frames and it needs to be invariant only to small variations (due to compression for instance). The second level focuses on the keyframes and computes for each keyframe a more sophisticated and robust descriptor (KVD). It is a 30-dimensional descriptor and it is also DCT-based. KVDs are clustered using a micro-clustering technique in order to group together the near-identical shots. The number of KVDs per cluster corresponds to the number of times a sequence is repeated. A KVD is associated with a frame of the repeated sequence but does not provide information on the sequence boundary. This is where BVDs are used, in order to precisely determine the boundaries by matching corresponding frames in all occurrences of the repeated sequence. The clusters are analyzed, and based on the temporal diversity of KVDs within a cluster and on the inter-cluster relationships, the set of repeated sequences is created. The different occurrences of the repeated sequences are possible separators. For more details concerning this approach for the detection of repeated sequences, the reader can refer to [1].

2.2 Separator detection

All the occurrences of the repeated sequences, detected during the step described in Section 2.1, are not necessarily separators. It might happen that within the analyzed episodes there are sequences that are replayed or that are very similar. These of course are not separators. For instance, shots showing the moderator in the same position but at different times of the episode might be detected as occurrences of a repeated sequence. They are not separators but their content is very similar. We call these occurrences "false alarms". In order to remove them, a post-processing step is used.

The detected repeated sequences are first passed through a first filter that removes a repeated sequence that has all its occurrences only coming from the same episode. Even if a separator can be repeated within the same episode, it has to be also repeated over at least two episodes in order to be valid. It is very unlikely to have a separator created specifically for a single episode.

As explained previously, in the case of inter-episode repeated sequences, false alarms may also appear. In order to remove them, the second property of the separators (i.e. temporal stability) is exploited. To achieve that, a study of the temporal density of the occurrences of detected repeated sequences from the input episodes is performed. All the occurrences from different episodes are projected on the same temporal axis. From this projection, a histogram is computed by counting the number of occurrences during each 40ms window (each frame). A kernel-based density estimation is then performed [7]. In this study, a Gaussian kernel has been used:

$$f_i = \sum_{j=i-3\sigma}^{i+3\sigma} h_j e^{-\frac{(j-i)^2}{2\sigma^2}}, \quad (1)$$

where f_i represents the filtering result for frame i and $h(j)$ represents the number of occurrences computed from the histogram, corresponding to frame j.

The idea is to find the areas with high concentrations. These areas are likely to be times when a separator is broadcasted. Isolated occurrences are likely to be false alarms as they appear quasi-randomly without any temporal stability.

The result of the temporal density analysis is a distribution curve where a maximum represents an area of high concentration of the separators. A threshold is then defined as a fraction of the mean of all the maxima. The separators that have a density under the threshold are rejected. An illustrative example is given in Figure 2. The occurrences in dark are isolated occurrences.

After applying the filters, the occurrences of the remaining repeated sequences will be considered as the final set of separators.

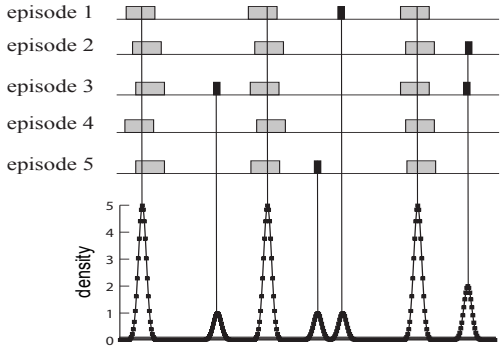

Figure 2: Temporal density of detected occurrences.

3. EXPERIMENTS

We performed experiments on real TV broadcasts. The first experiment studies the ability of detecting separators as repeated sequences in recurrent TV programs. The objective is to validate the main idea behind our approach. The second experiment evaluates the effectiveness of filtering "false alarms" from the set of detected repeated sequences. The last experiment places the method in the real context of segmenting a new episode of a recurrent TV program using a "history" of previously broadcasted episodes of the same program. Precision (P), recall (R) and the F-measure (F1) have been used for the evaluation.

3.1 Datasets

Our dataset is composed of 54 episodes of 3 daily TV shows broadcasted in 2009 on 2 French TV channels. These programs are denoted *Motus*, *Tout le monde veut prendre sa place (TLMVPSP)* and *10H Le Mag* (cf. Table 1). The first two are game shows. They have a steady structure and they have only inter-episode repeated separators, i.e. there is no separator that is repeated inside the same episode. The third program is a magazine composed of reports and anchor scenes. It has both inter and intra-episode separators. For all the episodes of the three programs, we have manually created a ground-truth. Each separator has been precisely determined.

		Dataset	# episodes	# separators
Motus	June 1-5	M1	5	4
	June 8-12	M2	5	4
	June 22-26	M3	5	4
TLMVPSP	May 18-24	T1	7	5
	May 25-31	T2	6	5
	June 1-7	T3	6	5
	June 8-14	T4	7	5
	June 17-21	T5	5	5
10H Le Mag	May 25-29	L1	4	22*
	June 1-5	L2	4	22*

Table 1: Datasets description.

3.2 Repeated sequence detection

The aim of this first experiment is to evaluate the effectiveness of the repeated sequence detection algorithm for the detection of separators. From the detected repeated sequences for each dataset, we have then evaluated precision and recall on a per-occurrence basis. A separator is considered as being correctly detected if it overlaps with a separator from the ground-truth. The obtained results are presented in Table 2.

Dataset	Precision	Recall	F1
M1	0.11	0.95	0.20
M2	0.13	0.95	0.22
M3	0.17	0.95	0.30
T1	0.56	0.86	0.68
T2	0.53	0.84	0.65
T3	0.54	0.80	0.64
T4	0.59	0.80	0.68
T5	0.39	0.72	0.50
L1	0.49	0.89	0.63
L2	0.55	0.95	0.70

Table 2: Exp1: Performance on a per occurrence basis.

The results presented in Table 2 show a good detection of the separators with very high recall values, especially for datasets M1, M2 and M3. For datasets T, the recall is relatively low compared to the recall of M and L. This can be explained by the fact that most of the separators that were not found are sequences that present an object that will be awarded. This object may be different from one show to the other. Regarding the precision, the results are low for all the datasets, which confirm that a filtering step is required.

3.3 From repeated sequences to separators

In this experiment, we focus on the filtering step and assess its ability to improve the precision of the detected separators. We first use the filter that removes the repeated sequences that have all the occurrences coming from the same episode. These occurrences are likely to be false alarms.

To further improve the precision, a second filter is applied. Based on the temporal stability of the separators, the distribution over a temporal axis is calculated. All the peaks that have their maximum

value under a certain threshold will be considered as false alarms time regions. That means that if an occurrence of a repeated sequence is located in these time regions, it is filtered out. Moreover, a sequence that has one of its occurrences that has been identified as a false alarm will be completely rejected.

Table 3 summarizes the obtained results on a per occurrence basis and also on frame basis. It is important to consider both basis because the separators do not have the same duration. Of course, from an application point of view, missing a short or a longer separator has the same impact. This double evaluation is however important to assess to which kind of separators our method is sensitive.

	P	R	F1
M1	0.92	0.95	0.93
M2	1	0.95	0.97
M3	1	0.95	0.97
T1	1	0.86	0.92
T2	1	0.84	0.91
T3	1	0.80	0.89
T4	1	0.80	0.89
T5	1	0.72	0.84
L1	0.79	0.88	0.83
L2	0.73	0.92	0.81

(per occurrence basis)

	P	R	F1
M1	0.99	0.90	0.94
M2	1	0.90	0.95
M3	0.99	0.87	0.93
T1	0.94	0.77	0.85
T2	0.97	0.73	0.83
T3	0.90	0.62	0.73
T4	0.90	0.60	0.72
T5	0.95	0.70	0.80
L1	0.86	0.81	0.83
L2	0.77	0.77	0.77

(per frame basis)

Table 3: Exp2: Results after applying the temporal filtering.

The results presented in Table 3 show a major improvement of the precision for the detection of separators. For datasets L1 and L2 the precision is relatively low compared to the other datasets. This is explained by the fact that at the beginning of each episode of *10H Le Mag*, there is a brief overview of the reports that will be presented in the episode. This overview is segmented with a lot of very short separators that were not included into the ground-truth. Due to their very short durations, these separators are not always detected by the repeated sequence detection technique. If we consider that these short separators are proper separators following our definition of separators and we add them to the ground-truth, the precision would increase but the recall might slightly decrease.

As for the recall, it remains unchanged for the first two datasets, meaning that our filters do not alter the results. The filters have wrongly filtered out few separators from L1 and L2. Nevertheless, this does not have an important impact over the results, regarding the high number of separators of this set and also the short duration of the missed separators (2 seconds).

3.4 Impact of the number of episodes

For the use-case where we have to structure each episode as soon as it arrives, the number of previously broadcasted episodes to put together with the current episode to structure, is an important parameter. The idea is to determine how many previously broadcasted episodes should be considered in order to get the best structure of the current one. Intuitively, if we take a very long history, we may have a lot of noise. Conversely, if we take a shorter history, we may miss separators that do not repeat in the considered episodes.

This experiment evaluates the performance of our solution when the number of previously broadcasted episodes varies. We consider dataset M2. For each episode of M2, we use a history of N previous episodes, with N ranging from 1 to 5. We apply the algorithm on each set of N+1 episodes in order to detect the separators for the N^{th}+1 one. Precision and recall of the obtained structuring results (only on the episode to structure) were computed on a per frame basis. The obtained results are presented in Figure 3.

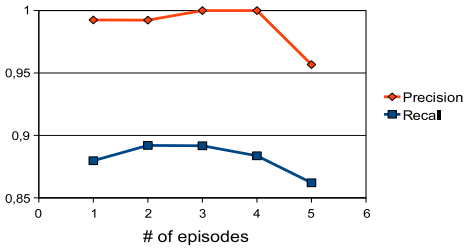

Figure 3: Exp. 3: Impact of the number of episodes.

The best trade-off seems to be around a history of 3 episodes. This is very good for real-world applications as only three previously broadcasted episodes are needed to structure the current one.

4. CONCLUSION

In this paper we have presented a method for intra program structuring. Based on a short history of previously broadcasted episodes, it detects separators as repeated sequences and, based on these separators it structures each new episode of the TV program.

Our future work will focus on the use of the audio information in order to detect the separators that were not found by the visual repetition detection approach. An important point will also be to relate the recall and precision measures to the end-user satisfaction. Certain types of errors may be of little disturbance for the user, while other types are important.

5. REFERENCES

[1] S.-A. Berrani, G. Manson, and P. Lechat. A non-supervised approach for repeated sequence detection in tv broadcast streams. *Image Communication*, 23(7):525–537, August 2008.

[2] M. Bertini, A. D. Bimbo, and P. Pala. Content-based indexing and retrieval of tv news. *Pattern Recognition Letters*, 22(5):503–516, April 2001.

[3] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos constructing plots learning a visually grounded storyline model from annotated. In *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Florida, USA, June 2009.

[4] H. Li, J. Tang, S. Wu, Y. Zhang, and S. Lin. Automatic detection and analysis of player action in moving background sports video sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(3):351 – 364, March 2010.

[5] Z. Rasheed and M. Shah. Detection and representation of scenes in videos. *IEEE transactions on Multimedia ISSN 1520-9210*, 7(6):1097–1105, December 2005.

[6] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, and I. Trancoso. Multi-modal scene segmentation using scene transition graphs. In *17th ACM International Conference on Multimedia*, pages 665–668, Beijing, China, October 2009.

[7] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[8] W. Tavanapong and J. Zhou. Shot clustering techniques for story browsing. *IEEE Transactions on Multimedia*, 6(4):517 – 527, August 2004.

[9] Y. Zhai and M. Shah. Video scene segmentation using markov chain monte carlo. *IEEE Transactions on Multimedia*, 8(4):686 – 697, August 2006.