Audio/Visual Recurrences and Decision Trees for Unsupervised TV Program Structuring

Alina Elma Abduraman[1], Sid-Ahmed Berrani[1] and Bernard Merialdo[2]
[1]*Orange Labs – France Telecom R&D, 35510 Cesson-Sévigné. France.*
[2]*Multimedia Communications Dept. – Eurecom, 06904 Sophia Antipolis. France.*
{*alinaelma.abduraman, sidahmed.berrani*}*@orange.com, Bernard.Merialdo@eurecom.fr*

Keywords: Program Structuring, Non-linear Browsing, Audio/Visual Recurrence Detection, Decision Tree Classification.

Abstract: This paper addresses the problem of unsupervised TV program structuring. Program structuring allows direct and non linear access to the desired parts of a program. Our work addresses the structuring of programs like news, entertainment, shows, magazines... It is based on the detection of audio and visual recurrences. It proposes an effective classification and selection system, based on decision trees, that allows the detection of *"separators"* among these recurrences. *Separators* are short audio/visual sequences that delimit the different parts of a program. The decision trees are built based on attributes issued from techniques like applause detection, scenes segmentation, face/speaker detection and clustering. The approach has been evaluated on a 112 hours dataset corresponding to 169 episodes of TV programs.

1 INTRODUCTION

TV programs have an underlying structure that is lost when these are broadcasted. The linear mode is the only available reading mode when viewing programs recorded using a Personal Video Recorder or through a TV-on-Demand service. The fast-forward/backward functions are the only available tools for browsing. In this context, program structuring becomes important in order to provide users with novel and useful browsing features. The idea is to recover the original structure of the program by finding the start time of each part composing it. In addition to advanced browsing features, TV program structuring can also be used for summarization, indexing and querying, archiving, etc.

Our approach relies on "separators", which are short audio/visual sequences that delimit the different parts of a program. They are recurrent, as they generally repeat within or between different episodes of a program. The idea is to detect these separators among all the recurrences that are present in a program and use them to delimit the different parts of the program. In our previous works we have proposed different approaches for the detection of recurrences. In this paper we advance an effective classification and selection system, based on decision trees, that allows us to detect as many "separators" as possible among these recurrences. The attributes used for the construction of decision trees are issued from techniques like applause detection, scenes segmentation, face/speaker detection and clustering.

We continue to focus on *"recurrent"* programs which are composed of several "episodes" that are periodically broadcasted. We explain our choice for this type of programs by their applicative interest and by their important properties that make the task feasible (meaning they generally have a clear structure with well defined main parts delimited by separators).

The rest of the paper is organized as follows. Section 2 presents existing techniques for program structuring. Section 3 presents the proposed approach. Section 4 evaluates the proposed approach. Section 5 concludes the paper.

2 RELATED WORK

TV programs structuring can be classified into two categories: **specific** and **generic**.

The specific approaches try to structure programs in a supervised manner using prior knowledge of the domain in order to extract relevant data and construct a structured model of the analyzed video. Within this category there are most of the **sport program structuring techniques**. The rules of the game provide prior knowledge that is used to provide constraints on the appearance of events or the succession of those events. In a first step the video is segmented into

narrative segments like play and break making use of production rules and different outputs of camera views (Tjondronegoro and Chen, 2010) or based on predefined models (Xie et al., 2004). Then highlights/events are detected based on object tracking (player/ball tracking, detection of goal mouth, center line) and/or specific sounds identification (exciting speech, applause, hitting ball).

News programs structuring techniques also rely on specific approaches. Most news videos exhibit a similar and well defined structure. This structure is used to segment the program and to classify its different parts into anchor person shots, report, weather forecast, interview... Template matching techniques but also face detection, statistical approaches, HMMs or SVMs are often employed ((Misra et al., 2010; Eickeler et al., 2001)). Once the shots are classified, a more challenging task is to segment the broadcast into coherent stories. To do so, temporal models, predefined rules, Finite State Automatons or HMMs (Chaisorn et al., 2003) are used.

More generic techniques tend to make use of the "**recurrence**" of certain audio/video sequences. In this sense in (Jacobs, 2006), the video self-similarity is exploited to identify anchor shots as frequent patterns occurring in news and magazines. Audio recurrences are detected in radio broadcast streams in order to identify the jingles that separate different topics or to extract meaningful information. In (Muscariello et al., 2009), a dynamic time wrapping technique is used to discover repeating words. The principle is inspired from (Herley, 2006) where repetitions in audio streams are detected by time correlating low-dimension audio representations. The video repetitions are used to validate the audio ones. Audio and video consistency of repeated segments is used in (Ben and Gravier, 2011) where an event mining technique is used to detect relevant events.

Within generic approaches, a lot of works has developed on the use of **scenes** as structuring elements. They use clustering methods, scene transition graphs, cinematic rules, HMMs or SVM classifiers (Sidiropoulos et al., 2010; Goela et al., 2007; Zhai and Shah, 2006). However, the definition of a scene is very ambiguous and depends on the subjective human understanding of its meaning. Consequently, these methods are difficult to evaluate.

For more details about the presented approaches, readers may refer to (Kompatsiaris et al., 2011).

3 THE PROPOSED SOLUTION

We focus in this work on structuring recurrent TV programs. One of the main properties of these programs is their clear and steady structure. Moreover, the different parts of a program are generally delimited by short audio/visual sequences that we named *"separators"*. In Figure 1, examples of separators are illustrated. The horizontal lines represent the timeline of different episodes of the same program. Boxes represent the separators that delimit the main parts of the program. The three images are extracted from separators of a French game show.

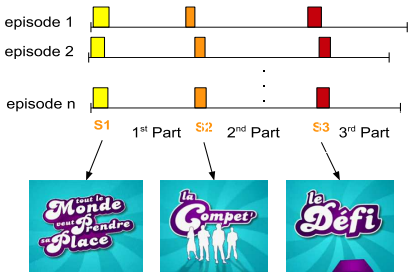

Figure 1: Example of separators.

The main idea of our approach is to automatically detect these separators. The parts of each episode are then identified using the separators boundaries: the end of a separator is the start of a new part. There is no need of any prior knowledge on the structure or on the number of parts the program might have.

3.1 Overview of the system

The proposed approach analyzes, in a first step, a set of episodes of a recurrent program in order to detect the audio and visual recurrences separately. Among these recurrences there are also the separators but not all the recurrences are necessarily separators. Consequently in a second step, a *prior-filtering* process is applied. Furthermore, in a third step, all the remaining recurrences are passed through a classification module based on decision trees, that decides whether a recurrence is a separator or not.

3.2 Visual recurrence detection

This algorithm extracts descriptors from the visual content for the episodes that are going to be structured. It also proceeds with detecting the visual recurrences using the technique described in (Berrani et al., 2008). First, shot segmentation is performed. 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 computed for each frame. Its role is to match nearly identical frames. It is used to precisely determine the boundaries of a recurrence. The second level focuses on keyframes and computes for each a more 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. The clusters are then analyzed and the set of recurrences is created.

3.3 Audio recurrence detection

To detect audio recurrences, a recurrence detection algorithm is applied over the set of episodes to be structured. The algorithm is described in details in (Abduraman et al., 2011b).

For each episode, a *descriptor* is computed using basic audio features that rely on the short time spectrum variations. When searching for recurrences, each of the episodes, in turn, becomes query while the rest become references. The query and the reference are compared by computing an average Hamming distance between their corresponding descriptors. In order to detect the intra episode recurrences as well, the query is also matched with itself. A match is confirmed if the distance between the descriptors falls below a pre-determined threshold. At the end of the processing, the set of obtained matching segment pairs are analyzed and clustered. First, each pair of segments instantiates a cluster. Second, each two overlapping clusters are merged. Two clusters overlap if a segment belongs simultaneously to both clusters.

3.4 Prior Filtering Module

Not all of the audio and visual recurrences are separators. Sequences that are replayed or that are very similar might also appear in the recurrences. We call these "*false alarms*". To remove them, a post-processing step is used. First, a prior filter is used to remove the recurrences having all their occurrences coming from the same episode. In this way, the intra-episode false alarms are filtered out. Second, to filter the inter-episode false alarms, a study of the temporal density of the occurrences of the remaining recurrences is performed. The idea is to find areas with high concentrations. These are likely to be times when a separator is broadcasted. The isolated occurrences are likely to be false alarms. For more details about this method the reader may refer to (Abduraman et al., 2011a).

The temporal filter can only be applied to programs that have a temporal stability, like TV games. For the case of news programs, where the number of reports and their time lengths are different from one episode to another, the temporal stability is not satisfied. Moreover, in the case of the audio modality, the detected recurrences are often far too many and randomly distributed. We decided thus to apply the temporal filter only for the visual recurrences that come from temporally stable programs.

3.5 Decision Trees Classification

The remaining recurrences are next passed through a classification step based on decision trees. We have chosen to work with decision trees as these are most appropriate for our goal and the type of data we use. Moreover, results can be easily analyzed and understood. In the same time it is a powerful classification tool. The idea is to build a model through a sequence of questions made on a set of several input variables. Each next question depends on the previous answer. Once the model is build it can be used afterward to predict the value of a target variable based on the input variables. In our case, we will consider all the variables as binary. We define the target variable as the category to which the recurrence belongs to: 1 for separator and 0 otherwise; and we call the input variables "attributes".

3.5.1 Attributes description

The first attribute, denoted A0, is computed from the intersection of the audio and visual recurrences. It equals 1 if for an analyzed visual recurrence there is a corresponding audio recurrence and vice versa.

Applause detection: The method used for the detection of applause, is part of an audio classification approach that allows the segmentation of an audio visual content into speech, music, applause, silence, other. It is based on the computation of low level features that are used to discriminate between different types of sound. The classifier is based on the Bayes theorem and the features are modeled using Gaussian Mixture Models as in (Scheirer and Slaney, 1997).

Three attributes are issued from this module:

- A1 = 1 if applause exist in the analyzed recurrence
- A2, A3 = 1 if applause exist in the right/left neighborhood of the analyzed recurrence

Scene segmentation: Wz used for the segmentation of videos into scenes a hierarchical classification algorithm that emphasize the similarities of shots that are temporally close. The idea is to build a similarity tree where each leaf represents a shot and each node represents a possible scene. The similarity is computed based on the colorimetric distribution.

The attribute issued from this module is:

- A4 = 1 if there is a scene cut in the close neighborhood of the analyzed recurrence

Face Detection and Clustering: The technique used for face detection is based on the convolutional face finder described in (Garcia and Delakis, 2004). It consists of a pipeline of convolution and sub-sampling modules that perform automatic features extraction and classification. A micro-clustering technique is used to cluster together the most similar faces, belonging to the same person.

Four attributes are computed:

- A5 = 1 if there is a face in the analyzed recurrence

- A6, A7 = 1 if the face in the right/left neighborhood of the analyzed recurrence is the same as the one inside the analyzed recurrence
- A8 = 1 if the same face is in the right and left neighboring of the analyzed recurrence

Speaker Diarization and Clustering: The algorithm used for the speaker diarization and clustering is similar to the one described in (Claude Barras and Gauvain, 2006) and proposes to index people independently by the audio information. The audio signal is first segmented into speech/non speech sequences. Each speech sequence is then segmented into speaker turn. The sequences corresponding to the same person are then clustered together using a *Bayesian Information Criterion* and *Cross-likelihood Ratio*.

Four attributes are computed:

- A9 = 1 if a speaker is in the analyzed recurrence
- A10, A11 = 1 if the speaker in the right/left neighborhood of the analyzed recurrence is the same as the one inside the analyzed recurrence
- A12 = 1 if the same speaker is in the right and left neighborhood of the analyzed recurrence

3.5.2 Training phase

In order to build the decision tree, a set of *n* samples (recurrences) from the training data is used. Each sample is described by the 13 binary attributes previously presented and by a target category that confirms if the recurrence is a separator or not. The tree is built by asking questions on attributes. The attribute selected for questioning is chosen by a predefined criterion. The tests on this attribute are used to split the current training data set into subsets. The algorithm is recursive and progressively splits the training data set into smaller subsets as long as the subset is not "pure"(all the samples belong to the same category) or there is at least another attribute to test. The split can be stopped beforehand by imposing a constraint on the criterion's parameters. In this case the node is transformed into a leaf and assigned a category.

We used three different criteria for the choice of the attribute to be tested in the current node:

Purity Criterion: This criterion considers the number of samples classified in each leaf. The idea is to choose the attribute that makes the immediate descendant nodes as pure as possible. Let us consider A_i, i = 0,...12, one of the defined attributes and j = 0,1 the values that attribute A_i can take. Suppose that n_{cj} is the number of samples of category c, c = 0,1, the purity function computed for each leaf is: $p_j = \max_c n_{cj}$ and the purity for A_i is: $P_{A_i} = \sum_j p_j$.

Entropy Criterion: This second criteria follows the same idea as the previous one, meaning to choose the attribute that makes the descendant nodes as pure as possible. The difference is that in this case, the purity is treated as an entropy impurity function and the chosen attribute will be the one that provides the smaller impurity. Considering the same notation as for the previous case, the purity function computed for each leaf is:

$p_j = -\sum_c \frac{n_{cj}}{n_j} * \log \frac{n_{cj}}{n_j}$, where $n_j = \sum_c n_{cj}$.

and the purity for attribute A_i is: $P_{A_i} = \sum_j \frac{n_j}{n} * p_j$.

Weighted Class (Error Minimization) Criterion

This last criterion takes in consideration the problem when the number of samples assigned to each of the two categories is very different. Moreover, the first criterion tried to obtain a good performance by maximizing the number of well classified samples (recurrences). In this case, we focus on obtaining a good performance from the point of view of minimizing the number of errors. We deal with two kinds of errors:

- the FP (false positive) $\rightarrow$ recurrences that were wrongly classified as separators
- the FN (false negative) $\rightarrow$ recurrences that were wrongly classified as non-separators

The cost of each type of error is weighted by a parameter λ. The attribute that minimizes the cost of the errors will be chosen.

4 EXPERIMENTS

In order to evaluate the proposed solutions we performed experiments using real TV broadcasts. We used a variety of TV programs, excluding non-recurrent programs like movies and programs which do not have a separator-based structure like TV series.

4.1 Datasets

The experimental dataset contains about 112 hours of videos, corresponding to 169 episodes (with 1594 separators) of 11 TV programs, broadcasted on 4 French TV channels. 5 of the programs, are TV games. The separators delimit different stages of the game and they vary in number between 4 and 8. 6 others are TV magazines. Their separators delimit the reports and the scenes where the anchor presents the next topic. In number they vary from 5 up to 23. The last program is a news program and it is similar to the magazines. It is composed of a set of reports and anchor person scenes. The number of separators can go from 20 up to 27. For all these programs the separators are repeated inter- or/and intra-episode.

The dataset has been divided in 2 groups: one containing the games and the other one containing the magazines and news programs. The objective was to group programs depending on their properties, and we

have noticed that magazines and news programs have almost the same structure (set of reports and anchor person scenes), different from that of the games.

For all of the episodes of these programs, we have manually annotated the separators by indicating their start time and end time.

4.2 Initial analysis of recurrences

The starting point of our study is the detection of recurrences. The main assumption of our approach is that separators are repeated and can be detected as recurrences. After applying our recurrence detection technique on each episode following the methods described in Sections 3.2 and 3.3, we get a set of 14,283 recurrences among which 14,130 are audio recurrences and 1,002 are visual recurrences. In order to compute the recurrences of an episode, we considered the 3 previously broadcasted episodes. Only the recurrences obtained for the analyzed episode are considered into the result set of recurrences.

For all our experiments, this dataset has been split into 2 datasets. The first one is used for training while the second one is used for evaluation (test).

We have first studied the recurrences in order to validate our main assumption, that is separators can be detected as recurrences. We have thus considered the training dataset and we have evaluated the proportion of recurrences that are separators. This is equivalent to consider a naive classifier that assigns all the recurrences to the class “separators”. Table 1 summarizes the obtained precision, recall and F-Measure for both program categories ”games” and ”magazines and news”, and also for audio and visual recurrences.

Table 1: Analysis of audio/visual recurrences.

	Audio rec.			**Visual rec.**		
	P	R	F	P	R	F
Games	0.0612	1	0.1154	0.8248	1	0.9040
Mag+News	0.4072	1	0.5787	0.7442	1	0.8533

Table 1 shows that for the TV games, among the audio recurrences, only almost 6% are separators while the rest of 94% should be filtered out as they belong to the category “non-separators”. These correspond to specific audio sequences that can be found in TV games. An example is the sound of pushing a certain button when a competitor wants to answer a question, or the audio sequence that announces if the answer was true or false.

On the contrary, for the visual recurrences, most of the recurrences (82%) are separators and only 18% are “non-separators”. This already gives a good ratio precision recall measured by the F score up to 0.9.

For the case of magazines and news, we noticed the same phenomenon even though the percentage of separators among the audio recurrences is higher (41%) than for the correspondent in TV games.

These differences between datasets and their correspondent audio and visual recurrences, encouraged us to continue the experiments considering separately games and magazines+news, and also training and performing classification separately for visual recurrences and audio ones.

4.3 Recurrence classification

In this section, the objective is to evaluate the ability of a classifier to distinguish if a recurrence in the test dataset is a separator or not.

For the moment, there is no study that could relate the type of errors to the end-user satisfaction. Consequently, we cannot decide whether when browsing inside a TV show, it is more annoying to have some missing separators (related to false negative errors) or to have to skip some recurrences that are not separators (related to false positives errors). Therefore, we use for evaluation the precision and recall for each of the two categories(P_0, R_0 for non-separators and P_1, R_1 for separators). We also compute the classification accuracy as a global measure that shows the proportion of true results in the entire test dataset.

As already stated, a classifier has been trained for each modality. TV games have also been separated from magazines+news. The obtained results are summarized in Tables 2 and 3.

Table 2: Evaluation of the classification results on audio/visual recurrences - Games.

Games		P_0	R_0	F_0	P_1	R_1	F_1	Acc
Audio	M1	0.9960	0.9701	0.9829	0.8848	0.5015	0.6401	0.9673
	M2	0.9961	0.9700	0.9829	0.8889	0.4985	0.6388	0.9673
	M3	0.9912	0.9761	0.9836	0.8095	0.6053	0.6927	0.9688
Video	M1	0.2581	0.5714	0.3555	0.8832	0.9667	0.9231	0.8625
	M2	0	0	0	0.8531	1	0.9207	0.8531
	M3	0.6774	0.5385	0.6000	0.9419	0.9000	0.9204	0.8673

Table 3: Evaluation of the classification results on audio/visual recurrences - Mag&News.

M&N		P_0	R_0	F_0	P_1	R_1	F_1	Acc
Audio	M1	0.8656	0.8054	0.8344	0.7139	0.6159	0.6613	0.7776
	M2	0.9381	0.7781	0.8506	0.8172	0.5086	0.6270	0.7867
	M3	0.9474	0.7709	0.8501	0.8333	0.4828	0.6114	0.7837
Visual	M1	0.2683	0.7333	0.3928	0.8750	0.9813	0.9251	0.8667
	M2	0	0	0	0.8392	1	0.9126	0.8392
	M3	0.1951	0.8889	0.32	0.8658	0.9953	0.9261	0.8667

The lines in the tables correspond to the results obtained when using the trees trained with each of the

3 criteria described in Section 3.5.2: M1 for the purity criterion, M2 for the entropy criterion and finally M3 for the error minimization criterion.

When comparing the different methods, in all of the cases, we can notice that the performances regarding the classification accuracy are very similar.

It is interesting to notice that for the case of visual recurrences and for both games and magazines+news, the classification step does not significantly improve the results. The naive classifier that considers all the recurrences as separators performs as good as the decision tree-based classifiers. In particular, when trying the entropy criteria the tree that is build assigns the category "separator" to all the recurrences. This means that for the case of visual recurrences, recurrences that are obtained after applying the filters are good enough and the classification step using decision trees was unable to further improve the results.

As already explained, there is no study that could relate the type of errors to the end-user satisfaction. Nevertheless, in a browsing use-case, separators can be used as anchors that allow users to skip a part and go directly to the next one, or for instance, to go back to the beginning of the current part. In this case, recall becomes more important than precision. Consequently, for visual recurrences, the entropy criteria should be preferred.

For the audio recurrences, in the case of games the error minimization criterion returns the best recall measure. This criterion allows equilibrating the importance of samples in the 2 classes. It is particularly important when the samples of the 2 classes in the training dataset are unbalanced, which is the case of audio recurrences. For the audio recurrences of magazines and news programs the best recall measure for the class of separators remains the entropy criterion.

4.4 Evaluation of the whole solution

In this section, we evaluate our solution that includes the detection of recurrences and their classification. The idea is to assess its performance in detecting separators. For this evaluation, we have used the test dataset and the separators ground-truth. This groud-truth is composed of all the separators manually annotated on all the episodes composing the test dataset. Precision, recall and F-measure have been used to measure the performance of separator detection.

Table 4: Separator detection using only recurrences, without any classification.

	Audio			**Visual**			**Audio ⋃ Visual**		
	P	R	F1	P	R	F1	P	R	F1
Games	0.07	0.95	0.13	0.75	0.74	0.73	0.08	0.97	0.14
M&N	0.42	0.92	0.55	0.71	0.48	0.55	0.43	0.93	0.56

Table 4 shows the performance of separator detection if we consider all the recurrences as separators, that is, no classification is performed. The lowest precision corresponds to the audio recurrences in the games dataset. Most of these recurrences are not separators, but false alarms. Regarding the visual recurrences, the precision is much higher meaning that most of these are separators. The best recall corresponds to the audio approach where a high number of recurrences is found. The recall for the visual recurrences is not as high due to the fact that the visual recurrence detection algorithm that we proposed detects only near-identical recurrences. Separators containing the logo of the show superposed on natural images of the set for example, can not be detected.

The last column in the table represents the union of the results obtained for the audio and visual approaches separately. We added this column as it can provide information about the contribution of each approach but also about the overall performance in detecting separators, when considering both approaches (audio and video).

Table 5: Separator detection after recurrence classification.

	Audio			**Visual**			**Audio ⋃ Visual**		
	P	R	F1	P	R	F1	P	R	F1
Games	0.84	0.76	0.79	0.78	0.65	0.71	0.85	0.78	0.82
M&N	0.84	0.52	0.65	0.74	0.47	0.57	0.85	0.62	0.72

To evaluate the recurrence detection and classification algorithm we applied on the detected recurrences, the classifiers that performed the best classification accuracy in the previous section and we evaluated the results using the separators ground-truth. The results are presented in Table 5.

The general tendency after classification is an improvement of precision at the cost of a decrease for the recall. This means that recurrences that are not separators are filtered out but in the same time some of the true separators too. This could be a consequence of the noise induced by the different technologies that we used when computing the attributes. Another reason, equally important, could be the erroneous choice of attributes that are not as relevant for our task as we might have expected.

Nevertheless, if we take a close look in Table 5 for the case of visual recurrences the classification does not influence the results significantly. The most important changes appear for the case of audio recurrences. In this manner, when considering the games, there is an important increase of the precision with 77% after classification, with a decrease in the recall of 19%. However the F1 score that combines the precision and recall measures is superior with 66%. For

the magazines and news the variation of the precision and recall is around 40%, with an increase of the F1 score of only 10%.

5 CONCLUSION

In this paper we proposed an approach for the detection and classification of audio and visual recurrences based on decision trees. The idea was to compute the performance of such system on the detection of ”separators”, that are at the root of an automatic recurrent TV program structuring system.

Experiments showed that our main assumption, that separators are repeated and could be detected as recurrences is validated. When classifying these recurrences, using decision trees trained with the 3 proposed criteria, the performances regarding the classification accuracy are very similar. Moreover, for the case of visual recurrences, these do not exceed significantly the naive classifier meaning that in this case a classification step in not really necessary.

The evaluation of the whole solution, showed that a lot of false alarms are filtered out (especially for audio recurrences) but with them a part of the separators too. This resulted in an increase of the precision but at the cost of a decrease in the recall. However, globally, the F-measure for the audio case is better after performing the classification step. For the visual-based approach, results are not significantly influenced.

In perspective, we intend to extend further the use of decision trees by trying combinations of attributes in the training stage. We would also like to consider the use of another type of classifier, such as the SVMs. This would allow us to compare the results and conclude if the actual results are more influenced by the limitations of decision trees or by the attributes we have defined.

REFERENCES

Abduraman, A. E., Berrani, S.-A., and Merialdo, B. (2011a). Audio recurrence contribution to a video-based tv program structuring approach. In *IEEE Int. Symposium on Multimedia*, Dana Point, CA, USA.

Abduraman, A. E., Berrani, S.-A., Rault, J.-B., and Blouch, O. L. (2011b). From audio recurrences to tv program structuring. In *4th Int. Workshop on Automated Media Analysis and Production for Novel TV Services*, Scottsdale, Arizona, USA.

Ben, M. and Gravier, G. (2011). Unsupervised mining of audiovisually consistent segments in videos with application to structure analysis. In *IEEE Int. Conf. on Multimedia and Exhibition*, Barcelone, Espagne.

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

Chaisorn, L., Chua, T.-S., and Lee, C.-H. (2003). A multi-modal approach to story segmentation for news video. *World Wide Web*, 6(2):187–208.

Claude Barras, Xuan Zhu, S. M. and Gauvain, J.-L. (2006). Multistage speaker diarization of broadcast news. *IEEE Trans. On Audio, Speech and Language Processing*, 14(5):1505–1512.

Eickeler, S., Wallhoff, F., Iurgel, U., and Rigoll, G. (2001). Content based indexing of images and video using face detection and recognition methods. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Salt Lake City, Utah, USA.

Garcia, C. and Delakis, M. (2004). Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423.

Goela, N., Wilson, K., Niu, F., and Divakaran, A. (2007). An svm framework for genre-independent scene change detection. In *IEEE Int. Conf. on Multimedia and Expo*, Beijing, China.

Herley, C. (2006). Argos: automatically extracting repeating objects from multimedia streams. In *IEEE Trans. on Multimedia*, vol. 8, pages 115–129.

Jacobs, A. (2006). Using self-similarity matrices for structure mining on news video. *Advances in Artificial Intelligence*, 3955:87–94.

Kompatsiaris, D. Y., Merialdo, P. B., and Lian, D. S., editors (2011). *TV Content Analysis: Techniques and Applications*. CRC Press, Taylor Francis LLC.

Misra, H., Hopfgartner, F., Goyal, A., Punitha, P., and Jose, J. M. (2010). Tv news story segmentation based on semantic coherence and content similarity. In *16th Int. Multimedia Modeling Conf.*, Chongqing, China.

Muscariello, A., Gravier, G., and Bimbot, F. (2009). Audio keyword extraction by unsupervised word discovery. In *Conf. of the Int. Speech Communication Association (Interspeech)*, Brighton UK.

Scheirer, E. and Slaney, M. (1997). Construction and evaluation of a robust multifeature speech/music discriminator. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Munich, Germany.

Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., and Trancoso, I. (2010). On the use of audio events for improving video scene segmentation. In *11th Int. Workshop on Image Analysis for Multimedia Interactive Services*, Desenzano del garda, Italy.

Tjondronegoro, D. W. and Chen, Y.-P. P. (2010). Knowledge-discounted event detection in sports video. *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 40(5):1009–1024.

Xie, L., Xu, P., Chang, S.-F., Divakaran, A., and Sun, H. (2004). Structure analysis of soccer video with domain knowledge and hidden markov models. *Pattern Recognition Letters*, 25(7):767 – 775.

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