HIERARCHICAL FILM SEGMENTATION USING AUDIO AND VISUAL SIMILARITY

Bertrand Delezoide

CNRS/Ircam, Paris; CEA/LIST/LIC2M, Fontenay-Aux-Roses; France

ABSTRACT

Video structure extraction is essential to automatic and content-based organization, retrieval and browsing of video. However, while many robust shot segmentation algorithms have been developed, it is still difficult to extract structures from a film. In this paper, we present a novel video and audio segmentation scheme, in which audio and image information is integrated in video structure extraction.

1. INTRODUCTION

Video structure parsing is the process of extracting construction units of video programs. It is essential to automatic and content-based organization and retrieval of video. There are usually two layers of construction units in video: shots and scenes (also often referred to story units). But it is possible to suggest more semantic layers. Therefore, a robust video structure parsing method should be able to segment a video program into these multiple semantic layers.

The paper is organized as follows. In the next section we present the four-layer model of video structure. Bottom layer segmentation is described in Sect.3, and the segmentation algorithm of the other three layers is presented in Sect.4. Our hierarchical film structure extraction scheme is introduced in Sect.5. Concluding remarks are given in Sect.6, and some potential application domains of the proposed strategies are outlined.

2. VIDEO AND AUDIO STRUCTURES

We begin with a few insights obtained from understanding the process of film-making and the psychology of cognition.

2.1. The video structure

We choose Zhu [8] **four-layer hierarchical representation** for video.

2.1.1. The shot

A video **shot** is a video sequence that consists of continuous video **frames** for one camera action.

2.1.2. Scene layer structures (group of shots)

It is possible to extract **structures** within a scene. Sundaram [1] postulates the existence of two broad categories of scenes: the N-type (based on the initial definition) and the M-type scene. An N-type scene has unity of location, time and sound. N-type scenes are divided in three types: dialogue, progressive and hybrid.

Dialogue: A simple repetitive visual structure can be present if the action in a scene is a dialogue.

Progressive: A linear progression of visuals without any repetitive structure.

Hybrid: A dialogue structure embedded in an otherwise progressive scene.

Within an M-type scene we assume there is no unity of visuals either in terms of location, time or lighting conditions.

2.1.3. The scene

We model the video **scene** as a collection of shots with a single, consistent, underlying semantic. We further assume that there is a consistency in the chromatic composition and the lighting in all the shots of a scene [1]. Indeed, film-makers seek to maintain continuity in lighting among shots within the same physical location. This is done even when the shots are filmed over several days. This is because viewers perceive the change in lighting as indicative of the passage of time.

We expect that the audio track will be consistent over the scene.

2.1.4. Film layer structures (group of scenes)

Within a film, it is possible to extract **groups of scenes** that have a global underlying semantic, location or time. Consecutive scenes can be filmed at the same location (an office, a forest etc…) or within one occurrence of time (night or day). In general, video can be separated into three parts: presenting subject or topic information, showing evidence and details, drawing conclusions.

It is interesting for indexing or retrieval to put the stress on these structures.

This four-layer film structure makes it possible to represent the temporal and hierarchical structures of a film on a hierarchical tree. Figure 1 show video structure of a film.

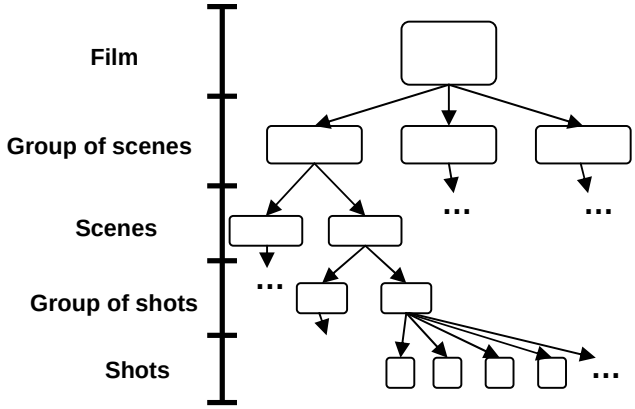

Figure 1: the four-layer hierarchical representation of a film.

2.2. The audio structure

This structure is a four-layer representation: group of scenes, scene, group of clips and clip. The bottom layer is composed of **audio clips,** as for shots in video.

3. AUDIO AND VIDEO FIRST SEGMENTATION

3.1. Audio segmentation

An **hidden Markov model** (HMM) method is used in our scheme for **audio segmentation** and **speaker change detection**.

First segmentation is an algorithm based on HMM that segments an audio signal into consecutive homogeneous segments called **clips**. Each clip contains audio that belongs to one of the semantic class: **speech / music / environmental sound / silence**. In [2] we determined the best features for this classification, and constructed a statistical model of each class. The transition probabilities are learned on a learning database. The features of an audio clip are extracted on a sliding window and the HMM decoding algorithm determines the class belonging of each window. A boundary is detected when the class of audio changes.

Then, speech segments are segmented using a HMM speaker change detection [3]. First the features of a sliding window of the audio segment are extracted (LSP/MFCC/Pitch). The HMM model is built iteratively by adding speakers one by one. The HMM transition parameters are moved to reflect the new HMM structure and an iterative adaptation process is done. During this adaptation phase the models are adapted (Baum-Welsh) corresponding to the current segmentation and a new segmentation is computed using Viterbi decoding. The last phases are repeated until no gain is observed.

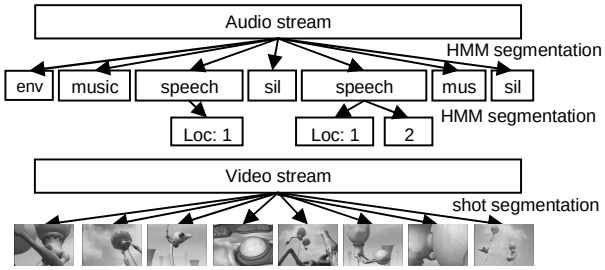

Figure 2: Audio and video stream first segmentation in audio clips and video shots.

3.2. Video

The video stream is converted into a sequence of shots using a sophisticated color based shot boundary detection algorithm [4], producing segments that have predictable consistent lighting and chromaticity. For each segment, the algorithm determines the frame that better summarizes the shot; this frame is denoted as the **key-frame** of the shot.

4. REPRESENTATION BY STATES

We assume that the film, audio and video streams exhibit instances of **similar segments**, possibly separated by other segments. For example, a common dialogue scene structure is ABABAB, where A are shots showing the first character and B the second. We aim to group the segments of such a scene into two class is corresponding to the two different characters. Once this is done, the scene could be summarized by two key-frames representing each character.

The structure extraction we consider here is based on the representation of a film as a **succession of states**. Each state represents similar information found in different parts of the film.

The information is constituted here by the **dynamic features** (possibly on different temporal scales) derived from audio or video analysis.

Human segmentation and grouping performs better when watching (listening to) something several times. A similar approach based on Cooper [5] and Peeters [6] popular music summary generation algorithm is followed here.

The first pass allows the detection of variations in the film without knowing if a specific part will be repeated later.

The second pass allows one to find the structure of the stream by using the previously created segments.

The second pass operates in three stages: 1) the segments are compared in order to reduce redundancies; 2) the reduced set of segments is used as initialization for a K-means algorithm 3) the output states of the K-means algorithm are used for the initialization of a hidden Markov model learning.

Finally, with the third pass the optimal representation of the piece as a HMM state sequence is obtained by application of the Viterbi algorithm.

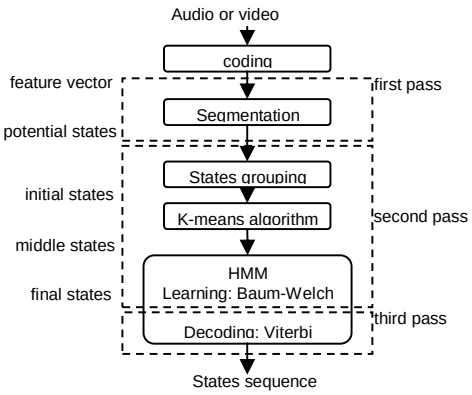

Figure 3: States representation flowchart

This multi-pass approach allows solving most of the unsupervised algorithm's problems.

4.1 First pass: Segmentation

Foote [7] showed that a **similarity matrix** (Figure 4) applied to well-chosen features allows a visual representation of the structural information of a video or audio signal.

The structure of the similarity matrix can be analyzed to find structure boundaries [5]. Generally, the boundary between two coherent segments produces a checkerboard pattern. The two segments will exhibit high within-segment similarity, producing adjacent square regions of high similarity (black regions) along the main diagonal of the matrix. The two segments will also produce rectangular regions of low between-segment similarity (white regions) off the main diagonal. The boundary is the crux of this checkerboard pattern.

To identify these patterns, we choose a matched-filter approach. A Gaussian kernel is correlated along the main diagonal of the similarity matrix. Large peaks are detected in the time-indexed correlation and labelled as segment boundaries. Throughout, we use an $l \times l$ kernel, where l depends on the hierarchical layer of the segmentation.

4.2. Second Pass: Statistical segment clustering

Above, we computed a similarity matrix to detect video (or audio) segment boundaries. In the second step, we use similarity analysis to efficiently cluster the detected segments [6]. This process both locates "repeated" segments separated in time, and corrects over-segmentation errors. Given segment boundaries, we can easily calculate a full similarity matrix of substantially lower dimension, indexed by longer segments.

To facilitate the initialization of the unsupervised learning algorithm, we need to group nearly identical states (similarity>threshold).

Then, to further cluster these segments, we use a **K-means** algorithm on the segment of the similarity matrix to find repeated or substantially similar groups of segments. The

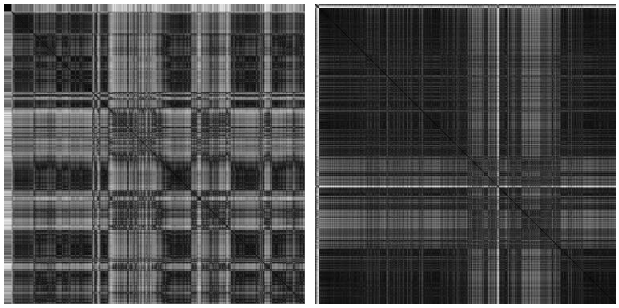

Figure 4: Similarity matrix (left: audio, right: video) of the 30 first minutes of the film "8miles" computed on audio features and video features.

inputs of the algorithm are the number of classes and states initialization, both given by the segmentation/grouping step.

4.3. Third Pass: Introducing Time constraints HMM

Film has a specific nature; it is not just a set of events but a specific temporal succession of events. K-means algorithm does not take into account the temporal nature of video and audio streams. We found appropriate to formulate this constraint using a Markov model approach. Since we only observe the features and not directly the states of the network, we are in the case of a **hidden Markov model** (HMM).

The resulting model is represented in:

- **Training:** The learning of the HMM model is initialized using the K-means states. The Baum-Welch algorithm is used in order to train the model. The outputs of the training are the state observation probabilities, the state transition probabilities and the initial state distribution.
- **Decoding**: The state sequence corresponding to the stream is obtained by decoding using Viterbi algorithm given the hidden Markov model and the signal features.

Figure 5 represent the three-pass segmentation algorithm.

5. HIERARCHICAL SEGMENTATION

5.1. The system

In this section we present the **hierarchical segmentation algorithm**. This algorithm aims at storing a film composed of a video and audio stream into the hierarchical tree presented in Sect.2. Therefore, we use a three-step algorithm.

The first step presented in Sect.3 segments the video stream in shots and the audio stream in clips. These structures correspond to the bottom layer of the tree. We can then compute two similarity matrixes of the film one for audio and one for video.

The second step iteratively segments both similarity matrixes using the segmentation algorithm presented in Sect.4, and stores the structures into the three top layers iteratively (Figure 5).

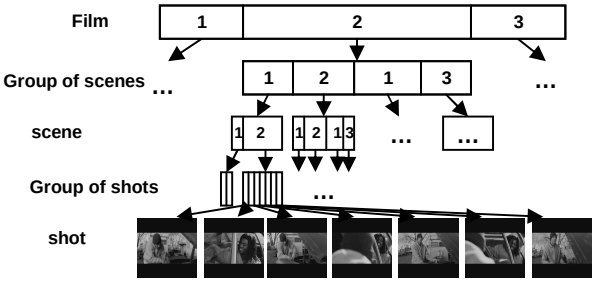

Figure 5: hierarchical segmentation results. 1, 2, 3, are the states given by the segmentation algorithm.

The third step builds a multimedia tree representation of the film using the two hierarchical trees and other knowledge from film making techniques.

5.3. Media fusion

At this point, we have two **hierarchical trees** corresponding to the structure of the two streams. The aim is to merge these trees in one multimedia tree corresponding to the underlying structure of the treated film. The trees are merged at each hierarchical state:

- **Shot**: video shots and audio clips boundaries do not correspond. This is why we take the video shot layer as the multimedia shot layer.
- **Group of shots**: Usually, boundaries of group of video shots and groups of audio clips do not correspond (except for special cases as some hybrid scenes). That is why, the video group of shots layer is taken as the multimedia group of shots layer.
- **Scenes**: Films show interesting interactions between audio and video scenes. We use the **computational scene** definition proposed by Sundaram [1]. Correspondences between the audio and the video scene boundaries are generated using a time-constrained nearest neighbor approach. We obtain computational scene containing audio and video. Singleton audio and video scene boundaries can be caused due to some particular directing effects that we do not take into account here.
- **Group of scenes**: For this layer, a model similar to that of the scenes layer is used. Most of the time, group of scenes boundaries are aligned for audio and video (no singleton are detected here). Thus, audio/video group of scenes are obtained.

Therefore a **multimedia hierarchical tree** is designed representing the underlying structure of the film.

6. CONCLUSION

Film **structural representation** is a recent topic of interest in the multimedia realm. In this paper, we investigated a multi-pass approach for the **automatic generation** of multi-layer film structure. Dynamic features seem to allow deriving powerful information from the **signal** for both detection of sequence repetition in the film and representation of a film in terms of "states". The representation in terms of "states" is obtained by means of segmentation and unsupervised learning methods (K-means and hidden Markov model) repeated iteratively on the hierarchical layers of the film for video and audio stream. The states are then used for the construction of a **multimedia hierarchical tree**.

Examples produced with this approach are available at (in construction) and will be given during the presentation of this paper.

Perspectives:

As for text, once we have a clear and fine picture of the film structure we can extrapolate any type of **summary** desired. In this perspective, further work will concentrate on the development of hierarchical summaries of the four-layer structure.

We believe that the hierarchical architecture proposed in this paper can be generalized as a toolkit for **video content** management indexing and retrieval.

7. REFERENCES

[1] H.Sundaram and S.F.Chang, "Determining computable scenes in films and their structures using audio-visual memory models," www.ctr.columbia.edu/~sundaram/pub/acmmm2k-final.pdf , 2000.

[1] B.Delezoide and X.Rodet, "Audio features selection for speech/music discrimination", Proc. JJCAAS 2002.

[3] S.Maignier, D.Moraru, C.Fredouille, L.Besacier and J.F.Bonastre, "Benefits of prior acoustic segmentation for automatic speaker segmentation", Proc. ICASSP 2004, www.lia.univ-avignon.fr/fich_art/530-ICASSP2004Revised_Fredouille.pdf

[4] P.Josserand, *Détection de transitions à l'intérieur d'une séquence vidéo en vue de son indexation*, Rapport de stage, Université du littoral de Calais, 2000.

[5] M.Cooper and J.Foote, "Summarizing popular music via structural similarity analysis ", Proc. FXPAL 2003, www.fxpal.com/publications/FXPAL-PR-03-204.pdf

[6] G.Peeters, A.LaBurthe and X.Rodet, "Toward automatic audio summary generation from signal analysis", Proc. ISMIR 2002, www.ismir2002.ismir.net/proceedings/02-FP03-3.pdf

[7] J.Foote and M.Cooper. "Media Segmentation using Self-Similarity Decomposition," *Proc. SPIE Storage and Retrieval for Multimedia Databases*, Vol. 5021, pp. 167-75, 2003, www.fxpal.com/people/cooper/Papers/SPIE02.pdf

[8] X.Zhu, J.Fan, A.Elmagarmid and X.Wu. "Hierarchical video content description and summarization using unified semantic and visual similarity", www.cs.uvm.edu/~xwu/Publication/Multimedia-03.pdf