VISUAL SEGMENT TREE CREATION FOR MPEG-7 DESCRIPTION SCHEMES

Philippe Salembier, Joan Llach, Luis Garrido

Universitat Politècnica de Catalunya, Barcelona, SPAIN
{philippe,jllach,oster}@gps.tsc.upc.es

ABSTRACT

This paper deals with the creation of visual segment trees involved in MPEG-7 Description Schemes. After a brief overview of MPEG-7 description schemes in general and of the Segment Description Scheme in particular, tools allowing the creation of segment trees are discussed. It is proposed to create a Binary Partition Tree in a first step and to restructure the tree in a second step. Several examples involving spatial and temporal partition trees are presented.

1. INTRODUCTION

The goal of the MPEG-7 standard [6] is to allow interoperability among devices that deal with audio-visual content description. The standard will involve four types of normative elements: descriptors (Ds), description schemes (DSs), description definition language (DDL), and coded descriptions. A descriptor is a representation of a feature that characterizes the audio-visual content. A description scheme specifies the structure and semantics of the relationships between its components, which may be both descriptors and description schemes. The description definition language allows specifying description schemes and descriptors. It also allows the extension and modification of existing Description Schemes. Finally, description encoding schemes are needed to satisfy requirements such as compression efficiency, error resilience, random access, etc.

Descriptors deal with low-level features such as color, texture, motion, audio energy, etc., as well as high-level features such as semantic description of objects and events, production process or information about the storage media, etc. It is expected that most descriptors corresponding to low-level features will be instantiated by automatic analysis tools whereas human interaction will be used for most high-level descriptors. DSs combine individual descriptors as well as DSs within common structures and define relationships among them. As for descriptors, the relationships instantiation can rely on automatic tools or on human interaction. The goal of this paper is to discuss hierarchical relationships that are involved in the segment DS and to propose a set of analysis tools to instantiate them.

The organization of this paper is as follows: Section 2 briefly reviews the current MPEG-7 DS and focuses in particular in the Segment DS. Automatic analysis tools are discussed in section 3 and conclusions are reported in section 4.

2. OVERVIEW OF MPEG-7 DESCRIPTION SCHEMES

The current MPEG-7 DSs [5] describe a single AV document. The extension to collection of documents will be done in the near future. MPEG-7 DSs are organized following their functionalities.

Features	Video Segment	Still Region	Moving Region	Audio Segment
Time	X		X	X
Color, Texture	X	X	X	
Spatial geometry		X	X	
Editing effect	X		X	X
Motion	X		X	
Camera motion	X		X	
Mosaic	X		X	
Audio features				X

Table 1: Features describing the various Segments

Seven main entities are defined: 1) the Structure DS, 2) the Semantic DS, 3) the structure-semantic links DS, 4) the Model DS, 5) the Summarization DS, 6) the Creation Meta information DS, 7) the Usage Meta information and 8) the Media information DS.

The structure DS is composed of Segments describing physical structures and signal properties. It may be used to define tables of contents of the documents. Physical features such as shots, regions, color, texture, motion, etc. are described under here. The semantic DS is used to specify semantic features in terms of semantic objects and events. It can be viewed primarily as a set of indexes. The structural and the semantic descriptions are related by the structure-semantic link DS. The links relate semantic notions with their instances in the structure DS.

The Usage and Creation Meta information DSs contain descriptors that carry author-generated information about an AV program or an image that cannot usually be extracted from the content itself. Example of Meta information descriptions are title, author, etc. The Media information DS contains descriptors that are specific to the storage media.

The Summarization DS is used to enable fast browsing of the content, e.g., highlights of various lengths. Finally, the Model DS provides a description of multimedia material that relates the structural information and descriptors to semantic information, synthetic models and other multimedia material. The Model DS is used in cases where the audio-visual signals are closely related to interpretation through models.

Let us detail more precisely the Structure DS. It consists of an arbitrary number of Segment DSs and of an arbitrary number of segment relation graph DSs. The segments are recursive i.e. they may be decomposed into sub-segments, and thus may form a hierarchy (tree) to define the structure of an AV program. The hierarchical decomposition is useful to design efficient search strategies (global search to local search). It also allows the description to be scalable: A segment may be described by its direct set of de-

0-7803-6536-4/00/$10.00 (c) 2000 IEEE

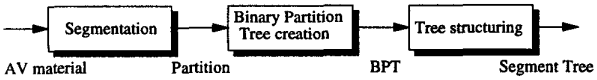

Figure 1: Outline of the Segment Tree creation.

scriptors and DSs, but it may also be described by the union of the descriptors and DSs that are related to its sub-segments. Multiple segment trees may be created and therefore multiple tables of contents are possible. The segment relation graph DS is used to describe temporal, spatial or spatio-temporal relationships that are not described by the tree structures.

There are four types of segments: VideoSegment, StillRegion, MovingRegion and AudioSegment. The VideoSegment describes a set of frames belonging to a video sequence. A single frame extracted from a video sequence is considered as a VideoSegment. The StillRegion defines a spatial area (e.g. set of pixels) belonging to a still image or a single frame of a video sequence. A still image is a particular case of StillRegion. The MovingRegion corresponds to a spatio-temporal area belonging to a video sequence. Finally, the audio information is given by the AudioSegment. The features describing each segment type are reported in Table 1. Most of the descriptors corresponding to these features can be extracted automatically from the original content. For this purpose, a large number of tools have been reported in the literature. Less attention has be devoted to the design of tools allowing the instantiation of the decomposition involved in the Segment DS. This decomposition can be viewed as a hierarchical segmentation problem where elementary entities (region, video segment, etc.) have to be defined and structured by inclusion relationship within a tree. In the following section, we propose and discuss a possible strategy to create VideoSegment and Region trees.

3. TOOLS FOR SEGMENT TREE CREATION

3.1. Visual Segment Tree creation

The extraction of individual spatial or temporal regions and their organization within a tree structure can be viewed as a hierarchical segmentation problem also known as a partition tree creation [8]. The strategy we propose here involves three steps illustrated in Fig. 1: the first step is a conventional segmentation. Its goal is to produce an initial partition with a rather high level of details. Depending on the nature of the segment tree, this initial segmentation can be a shot detection algorithm (for VideoSegment) or a spatial segmentation following a color homogeneity criterion (for Regions). The second step is the creation of a Binary Partition Tree (BPT). Combining the segments (VideoSegments or regions) created in the first step, the BPT defines the set of segments to be indexed and encodes their similarity with the help of a binary tree. Finally, the third step restructures the binary tree into an arbitrary tree. Since the first step is rather classical[1], the following sections focus on the two last steps.

3.2. Binary Partition Tree creation

The second step of the Segment tree creation is the computation of a BPT [7]. The node of the tree represents connected components in space (region) or in time (VideoSegment). The leaves of the tree represent the segments defined by the initial segmentation step. The remaining nodes represent segments that are obtained by merging segments represented by the children. This representation should be considered as a compromise between representation accuracy and processing efficiency. Indeed, all possible merging of initial segments are not represented in the tree. Only the most "useful" merging steps are represented. However, the main advantage of the binary tree representation is that efficient techniques are available for its creation (see [2] for example) and it convey enough information about segment similarity to construct the final tree.

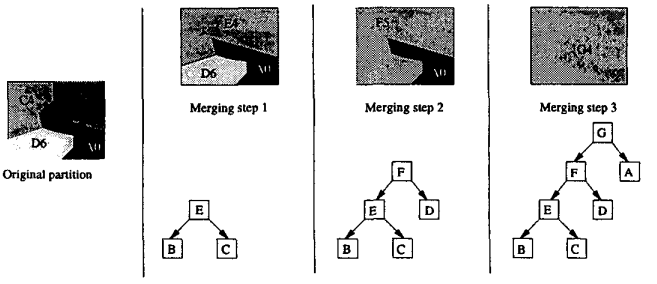

Figure 2: Example of Binary Partition Tree creation with a region merging algorithm.

The Binary Partition Tree should be created in such a way that the most "useful" segments are represented. This issue can be application dependent. However, a possible solution, suitable for a large number of cases, is to create the tree by keeping track of the merging steps performed by a segmentation algorithm based on merging (see [4, 2]). In the following, this information is called the *merging sequence*. Let us consider a simple case with StillRegions: Starting from the initial partition produced in the first step, the algorithm merges neighboring regions following a homogeneity criterion until a single region is obtained. An example is shown in Fig. 2. The original partition involves four regions. The regions are indicated by a letter and the number indicates the mean grey level value. The algorithm merges the four regions in three steps. In the first step, the pair of most similar regions, B and C, are merged to create region E. Then, region E is merged with region D to create region F. Finally, region F is merged with region A and this creates region G corresponding to the region of support of the whole image. In this example, the merging sequence is: $(B, C)|(E, D)|(F, A)$. This merging sequence progressively defines the Binary Partition Tree as shown in Fig. 2.

To create the BPT, the merging algorithm may use several homogeneity criteria (or distance d) based on low-level features. For example, if the image for which we create the BPT belongs to a sequence of images, motion information can be used to generate the tree: in a first stage, regions are merged using a color homogeneity criterion, whereas a motion homogeneity criterion is used in the second stage. Fig. 3 presents an example of BPT created with color and motion criteria on the *Foreman* sequence. The nodes appearing in the lower part of the tree as white circles correspond to the color criterion, whereas the dark squares correspond to the motion criterion. As can be seen, the process starts with a color criterion and then, when a given Peak Signal to Noise Ratio (PSNR) is reached, it changes to the motion criterion. As can

[1] An overview can be found, for example, in [8] and the references contained herein.

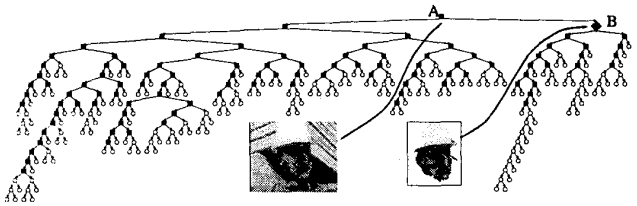

Figure 3: Examples of creation of Binary Partition Tree with color and motion homogeneity criteria

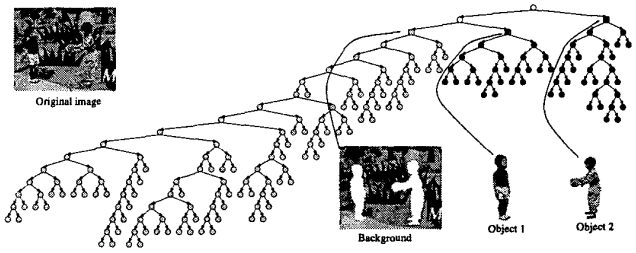

Figure 4: Example of partition tree creation with restriction imposed with object masks

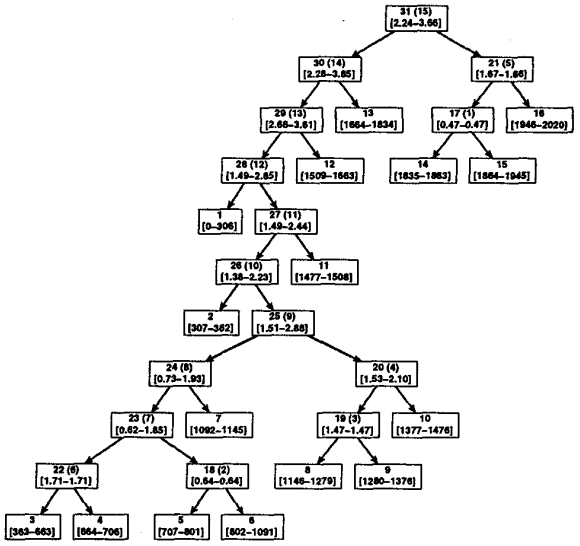

Figure 6: Binary tree created by the shot merging algorithm. The initial shots are shown in Fig. 5. The top node represents the whole sequence, while the leaf nodes represent the initial shots.

be seen, region that are homogeneous in motion as the face and helmet are represented by a single node (B) in the tree.

Furthermore, additional information about previous processing or detection algorithms can also be used to generate the tree in a more robust way. For instance, an object mask can be used to impose constraints on the merging algorithm in such a way that the object itself is represented as a single node in the tree. Typical examples of such algorithms are face, skin, character or foreground object detection. An example is illustrated in Fig. 4. Assume for example that the original *Children* image sequence has been analyzed so that masks of the two foreground objects are available. If the merging algorithm is constrained to merge regions within each mask before dealing with remaining regions, the region of support of each mask will be represented as a single node in the resulting BPT. In Fig. 4, the nodes corresponding to the background and the two foreground objects are represented by squares. The three sub-trees further decompose each object into elementary regions.

The same approach can be used for VideoSegments [3]. Assume for example that the initial segmentation of the *Drama* sequence has produced the set of elementary shots shown in Fig. 5. The BPT of Fig. 6 can be constructed by keeping track of the merging sequence of a shot merging algorithm. Here, the merging criterion is defined by a distance between shots. The features used to compute the shot similarity or distance can involve average images, histograms of luminance/chrominance information (YUV components) and histograms of motion information (x and y components of motion vectors). In order to compute the distance between a pair of VideoSegments, two different cases must be considered depending on the type of nodes. At the beginning of the merging process, nodes represent a single shot and a single shot distance can be used. In the following merging steps, the distance must be computed between nodes that may represent a large number of shots. Since the merging process may combine shots with quite different luminance/chrominance or motion information, the minimum and maximum distances (d_{min}, d_{max}) between all pairs of shots included in each VideoSegment is computed [1]. The merging order is defined by the minimum distance but the merging is actually performed if the maximum distance is below a given threshold. An example of BPT is shown in Fig. 6. Inside the leaf nodes, we have indicated the shot number, and between brackets its starting and ending frame number. Inside the remaining nodes, we have indicated a label, between parenthesis the merging order and between brackets the minimum and maximum distance between its two siblings.

3.3. Restructuring of the Binary Partition Tree

The purpose of the this algorithm is to restructure the BPT into an arbitrary tree that should reflect more clearly the video or image structure. To this end, nodes that have been created by the binary merging process but that do not convey any relevant information should be removed. The criterion used to decide if a node must appear in the final tree is based on the variation of distance between the segments. Note that leaves node are not allowed to disappear. Otherwise, a node is kept in the final tree if the following conditions are satisfied:

$$|d(\text{analyzed node}) - d(\text{parent node})| < \Theta \qquad (1)$$

In the case of a more complex model as the one used for shots merging where the minimum and maximum distances are used, the condition becomes:

$$\begin{cases} |d_{min}(\text{analyzed node}) - d_{min}(\text{parent node})| < \Theta \\ |d_{max}(\text{analyzed node}) - d_{max}(\text{parent node})| < \Theta \end{cases} \qquad (2)$$

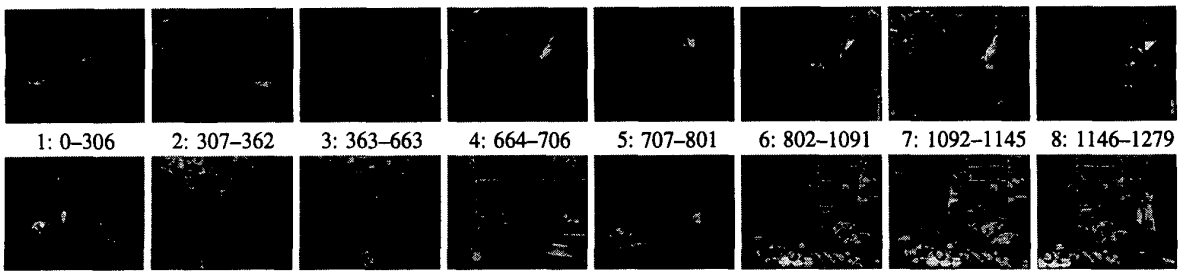

Figure 5: Initial temporal segmentation of the *Drama* sequence. Each image represents the central frame of the shot, the numbers show the starting and ending frame of the shot.

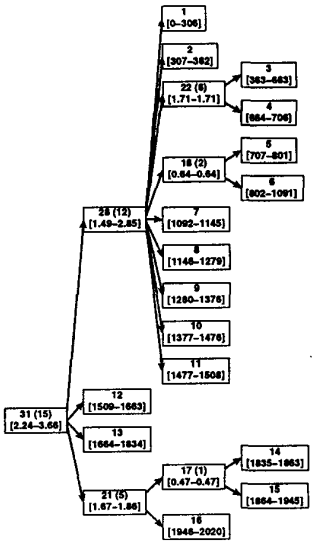

Figure 7: Final VideoSegment tree resulting from the tree structuring algorithm.

The restructuring of the BPT presented in Fig. 6 is shown in Fig. 7. this final tree represents more clearly the structure of the video sequence. Nodes in the first level of the hierarchy (28, 12, 13, 21) represent the four scenes of the sequence, while nodes in the third or fourth level represent the initial shots.

4. CONCLUSION

This paper has reviewed the Segment Description Scheme (DS) involved in MPEG-7 Description Schemes. The Segment DS is used to specify physical structures and signal properties. It may be used to define tables of contents of the documents. Low-level features such as shots, regions, color, texture, motion, etc. are described in the Segment DS. The review has highlighted the necessity to use automatic analysis tools not only for low-level descriptors but also for some of the relationships defined in the Segment DS. In particular, each segment may be decomposed to form a tree. This tree computation goes beyond the classical segmentation problem and new tools are needed.

In this paper, the strategy proposed to compute segment trees involves three steps. The first one is a classical segmentation which creates elementary segments. These elementary segments form the leaves of a Binary Partition Tree. The BPT itself is computed in a second step by keeping track of the merging steps resulting from a segment merging algorithm. Finally, the BPT is restructured to create an arbitrary segment tree reflecting the signal properties. As can be seen, the BPT is used as an intermediate representation. This approach is proposed because efficient algorithms exist for the computation of BPTs and moreover, the restructuring of the BPT can be performed in a robust way.

5. REFERENCES

[1] D.A. Adjeroh, M.C. Lee, and I. King. A distance measure for video sequence similarity matching. In *Proceedings of the 1998 International Workshop on Multimedia Database Management Systems*, pages 72–79, June 1998.

[2] L. Garrido, P. Salembier, and D. Garcia. Extensive operators in partition lattices for image sequence analysis. *EURASIP Signal Processing*, 66(2):157–180, April 1998.

[3] J. Llach and P. Salembier. Analysis of video sequences: Table of contents and index creation. In *Proceedings of the International Workshop on Very Low Bit-rate Video, VLBV'99*, Kyoto, Japan, October 1999.

[4] O. Morris, M. Lee, and A. Constantinidies. Graph theory for image analysis: an approach based on the shortest spanning tree. *IEE Proceedings, F*, 133(2):146–152, April 1986.

[5] MPEG. MPEG-7 generic audio-visual description schemes. Technical Report ISO/IEC JTC1/SC29/WG11/M5380, MPEG, Maui, USA, December 1999.

[6] MPEG. MPEG-7: Requirements document. Technical Report ISO/IEC JTC1/SC29/WG11/w2859, MPEG, Vancouver, Canada, July 1999.

[7] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. *IEEE Transactions on Image Processing*, 9(4):561–576, April, 2000.

[8] P. Salembier and F. Marqués. Region-based representations of image and video: segmentation tools for multimedia services. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8):1147–1169, December 1999.