HIERARCHICAL VISUAL DESCRIPTION SCHEMES FOR STILL IMAGES AND VIDEO SEQUENCES

P. Salembier[1], N. O'Connor[2], P. Correia[3] and F. Pereira[3]

[1] Universitat Politècnica de Catalunya
Barcelona, SPAIN
philippe@gps.tsc.upc.es

[2] Dublin City University
Dublin, IRELAND
Noel.OConnor@Teltec.DCU.IE

[3] Instituto Superior Técnico
Lisboa, PORTUGAL
{paulo.correia,fernando.pereira}@lx.it.pt

ABSTRACT

This paper proposes two DSs to describe the visual information of an AV document. The first one, is devoted to still images. It describes the image visual appearance and its structure with regions as well as its semantic content in terms of objects. The second DS is devoted to video sequences. It describes the sequence structure as well as its semantic content in terms of events. Features such as motion, camera activity, etc. are included in this DS. Moreover, it involves static visual representations such as key-frames, background mosaics and key-regions. These elements are considered as still images and are described by the first DS.

1. INTRODUCTION

Describing Audio-Visual (AV) content is currently becoming an active field of research [1]. This activity is motivated by the increasing availability of AV documents as well as standardization efforts such as MPEG-7 [2]. Using MPEG-7 terminology, an AV document description involves Descriptors (termed Ds: syntax & semantic of a representation entity for a feature) and Description Schemes (termed DSs: set of Ds and DSs plus the structure and semantics of their relationships). Most of the work on AV indexing has focussed on Ds and very few contributions are available for DSs.

This paper proposes two DSs to describe the visual information of an AV document. The first DS, called *Still Image DS*, is devoted to still images. It describes the visual appearance and structure of the image as well as its semantic content in terms of objects. The visual appearance is described with regions and their associated color and geometrical features. The semantic description is based on objects and defines their type, identity and possible activity.

The second DS, called *Video DS*, is devoted to video sequences. It describes the sequence structure as well as its semantic content in terms of events. Features related to sequence properties such as motion, camera activity, etc. are included in this DS. Moreover, it involves visual representations such as key-frames, background mosaics and key-regions. These elements are considered as still images and are described by the *Still Image DS*. The proposed DSs are not intended to address audio & production features.

2. TOOLS FOR DOCUMENT DESCRIPTION

The proposed DSs are inspired from the classical way of describing books. Let us briefly comment on the two basic tools used in this context: the *Table of Contents* and the *Index*.

2.1. The Table of Contents

The *Table of Contents* is a hierarchical representation that splits the document into elementary pieces (chapters, sections, subsections, etc). The order in which the items are presented follows the 1-D structure of the book. Although the section titles may carry semantic information, the main goal of the *Table of Contents* is not to describe the content itself but to define the document structure. The corresponding descriptors are the page numbers. The role and structure of the *Table of Contents* are described in the left side of Fig. 1.

2.2 The Index

The goal of the *Index* is not to define the book structure, but to define a set of potentially interesting items and to provide references to the sections where these items are discussed. The items have been selected because of their semantic value. Obviously, a given item may appear in several sections of the book. This situation is described by multiple references as shown in the right side of Fig. 1 (these multiple references are not present in the case of a *Table of Contents*). In many cases, the *index* is also presented in a hierarchical fashion to allow fast access to the item of interest. Finally, the references involved in the *Index* are generally defined in terms of page number. However, if *the Table of Contents* has a fine granularity, the index could directly refer to subsections of the *Table of contents*.

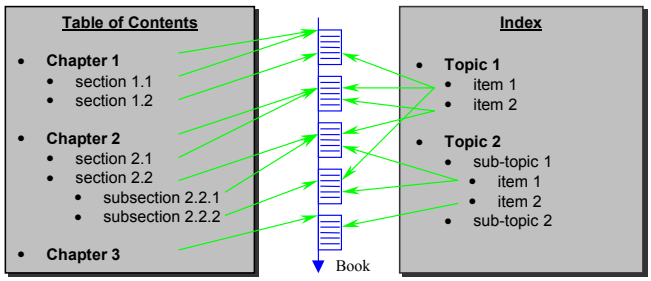

Figure 1: Description of a book content by "Table of Contents" & "Index"

As can be seen, the classical approach for book content description relies on a dual strategy: 1) define the document structure (the *Table of Contents*) and 2) define the locations where items carrying a specific semantic meaning appear (the *Index*). In the following, this dual strategy will be used for both the *Still Image* and the *Video DSs*.

3. STILL IMAGE DS

To describe images, an approach similar to that of section II may be used. It involves two hierarchical structures represented by trees: *Region* and *Object Trees*:

3.1. The Region Tree

The *Region Tree* plays the role of the *Table of Contents*. Its goal is to describe the spatial organization of the image. Note that the

structure of the document is now 2-D and not only 1-D as for books. The nodes of the tree represent connected components of the space called "regions". We use the term "region", and not "object", because regions themselves need not have a clear semantic meaning (e.g. region R_9 in Fig. 2). The structure of the *Region Tree* defines the inclusion relationship between elementary items as in the *Table of Contents*. Instead of having a section, which can be further decomposed into subsections, we have now regions that can be further decomposed into sub-regions. Note that the tree describes the entire image (similarly, the *Table of Contents* describes the entire book without leaving any holes).

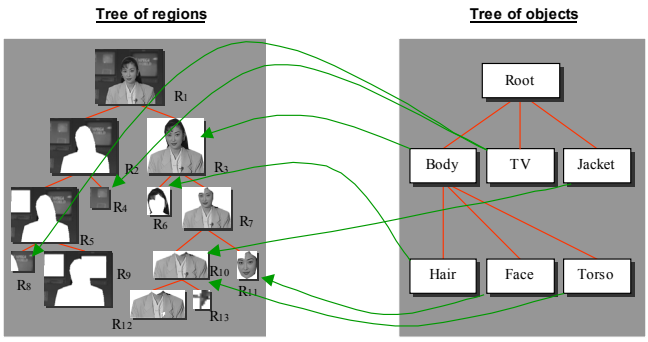

Figure 2: Example of *Region* and *Object Tree*.

A very simple example is illustrated in the left side of Fig. 2. The entire original Akiyo image can be seen as the root of the tree. The tree defines how this global region can be subdivided into elementary regions. Finally, the descriptors attached to the *Region Tree* describe visual properties of the regions: color, spatial and geometrical characteristics. The *Region Tree* could be an arbitrary tree, that is a tree where each node can have an arbitrary number of children. However, in the sequel, we will only use the inclusion relationship between regions. This relation can be represented by a simple binary tree. As a result, the tree will describe how a region can be decomposed into two components. It is known as a "Binary Partition Tree" [3]. The binary partition tree can be viewed as a region-based multiscale representation of the visual content. Large regions are represented on higher levels of the tree and fine details can be obtained from lower levels.

The *Region Tree* involves descriptors related to the signal properties. It may be created automatically or at least semi-automatically. In particular, regions appearing in the lower levels of the tree should be defined by their homogeneity in terms of signal properties (color for example).

3.2. The Object Tree

The *Object Tree* is the image *Index*. It is composed of a list of objects that were judged as being of potential interest during the indexing. This tree is composed of objects with a semantic meaning. The corresponding descriptors define the type, identity and activity of objects. Moreover, as an *Index*, an important functionality of this tree is to relate its objects to regions of the *Region Tree*. Note that the *Object Tree* refers to the *Region Tree* and not to the original image itself. This choice has been made because it is assumed that the granularity of the *Region Tree* is sufficiently fine.

An example of an *Object Tree* is shown in the right side of Fig. 2. The hierarchical relation is of the type "is-made-of". For example, the object "Body" is made of sub-entities called "Hair", "Face" and "Torso". Each object in the *Object Tree* has to refer to one or several regions in the *Region Tree*. For example in Fig. 2, the object "Body" refers to region R_3. The object "TV" refer to regions R_4 and R_8 since two TV screens are visible in the image. Conversely, one region may be referred to by several objects. This is the case for regions involving various semantic meanings. For example in Fig. 2, region R_{10} is referred to by both the "Jacket" and the "Torso" objects. Finally, note that all objects should refer to at least one region but not all regions have to be referred by an object. In this case, we have an unidentified region, that is a region described only by its visual appearance (e.g., R_9 in Fig. 2).

The descriptors involved in the *Object Tree* have semantic meaning. Even if some of these descriptors may be instantiated automatically (presence of face for example), the structure of the tree itself as well as a large number of descriptors imply (at least today) human assistance.

3.3. Global view of the Still Image DS

The global view of the *Still Image DS* is represented in Fig. 3. In this diagram, a DS is represented by a gray rectangle and a descriptor by a white rectangle. The symbol ♦ ( ) represents the aggregation of elements of different (same) types. The numbers close to the rectangles indicate the cardinality with which the respective elements may be present in the DS.

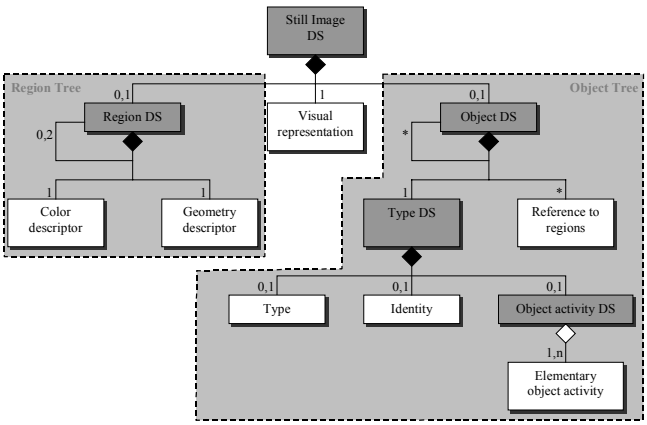

Figure 3: Global view of the Still image DS. White (gray) rectangles represent descriptors (description schemes)

At the higher level, the DS is decomposed in a visual representation and two sub-DSs representing the *Region* and the *Object Trees* respectively. The visual representation is a descriptor used for browsing (an icon image, for example).

The *Region DS* represents the *Region Tree*. The first node represents the region of support of the entire image which can be of arbitrary shape as in the case of MPEG-4 VOP. It is described by a color descriptor, a geometry descriptor and two (sub-)region DSs. The geometry descriptor deals with features such as position, size, orientation and shape. The two region DSs describe how the image can be split into two components (the *Region Tree* is binary). If the region is a leaf node of the tree, it is only described by color and geometry descriptors. The tree structure introduces a

notion of scalability in the description itself. A given region in the *Region Tree* can be described by its color and geometry descriptors. This gives a first description level. If necessary, this first description can be improved by accessing the descriptors of the children or even the descriptors of the leaves of the sub-tree starting from the region of interest.

The *Object DS* represents the *Object Tree*. It follows the same hierarchical approach as the *Region DS* except that the tree is not binary. Each object is described by a Type DS and an arbitrary number of references to regions and of (sub-)object DSs. The Type DS provides semantic information about the object: type, identity and activity. The exact definition of these descriptors may involve various thesauruses. In this context, at least three kind of thesaurus seem to be interesting:

1. Thesaurus of types: Typical examples include face, body, specific props, etc.
2. Thesaurus of identity: For some object types, the description can go into more details by defining the identity of the object. A typical example consists of being able to associate an instance of the object type "face" with a specific person.
3. Thesaurus of activity: Depending on the object type, the activity is an important descriptor. The term activity should be understood in a broad sense. Static (such as standing, sitting, etc) as well as dynamic activities (such as entering, taking something, etc) may be included.

4. VIDEO DS

The *Video DS* also relies on the dual strategy discussed in section II. It involves two trees. The first one is devoted to the description of the video structure and is called the *Sequence Tree* whereas the second one, termed the *Event Tree*, describes what is happening. The similarity between these trees and the *Table of contents* and the *Index* used to describe a book is very high since the documents they describe are both 1-D in essence.

4.1. The Sequence Tree

The *Sequence Tree* is the *Table of Contents* of the video sequence [4]. It defines the structure of the sequence and its visual properties. Each node of the tree represents a connected component in time called "sequence". The tree describes how each sequence can be divided in shorter (sub-)sequences. The leaves of the tree are assumed to be shots (or at least part of shots). In particular, they do not involve any editing effect. A very simple example is illustrated in the left side of Fig. 4. The time line of the video sequence is segmented in a hierarchical fashion. Each tree node points to a time segment of the video.

The *Sequence Tree* involves descriptors related to the visual properties of the video signal. The descriptor associated to non-leaf nodes of the *Sequence Tree* (sequences that are not shots) is simply a time reference indicating the beginning and the end of the corresponding sequence. The leaves (the shots) are more precisely described. Their description involves a time reference, a classification of the type of transition (e.g. cut, fade, wipe), a characterization of the camera activity and several visual representations: key-frames, a background mosaic and key-regions. The *Sequence Tree* can be constructed using tools such as shot transition detection algorithms and shot clustering algorithms. A large part of the process can be automated. Of course, user interaction may be necessary to correct mistakes.

4.2. The Event Tree

The *Event Tree* is the *Index* of the Video sequence. It defines in a hierarchical fashion a set of events and sub-events and characterizes and relates them with sequences belonging to the *Sequence Tree*. A simple example is illustrated in the right side of Fig. 4. If we assume for instance that the video sequence corresponds to the beginning of a TV news program, the main events can be classified in categories such as: credits, studio scenes, speaker views, news reports (politic, economy, social life, sport), etc. Each element of this tree points to the occurrences of the corresponding shots or sequences in the *Sequence Tree*.

The *Event Tree* concentrates the descriptors related to the semantic in the video. Some of these descriptors can be pre-defined in a thesaurus of events and simply consists of an index defining the event type. Annotation is another way to characterize the semantic value. Finally, to relate events with temporal video segments, a descriptor called "reference to sequence" is assigned to each event. The construction of the *Event Tree* may very likely rely on supervised techniques and human interaction.

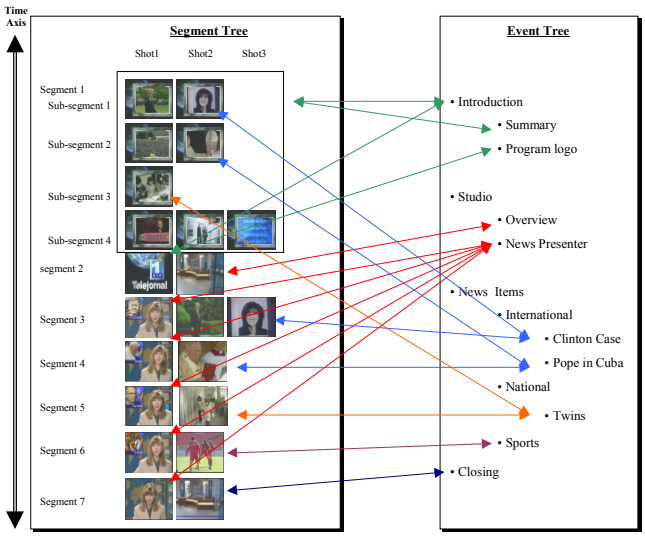

Figure 4: example of *Sequence* and *Event Trees*.

4.3. Global view of the Video DS

The global view of the Video DS is presented in Fig. 5. The higher level of the Video DS is decomposed into three major elements: The *Sequence DS*, *the Event DS* and the Visual summary. The **visual summary** is a short and sub-sampled version of the sequence used for browsing. Both the *Sequence* and the *Event DS* are arbitrary trees. A **sequence** is described by two time references (start & end), by an arbitrary number of (sub-) sequences and by shots. The shot itself is a rather complex DS and will be more precisely described later on. An **event** is described by an arbitrary number of (sub-)events and three descriptors: an index defining the type of event, an annotation and references to sequences. As in the case of the Still Image DS, the "reference" descriptor plays a major role in relating the semantic information contained in the *Event Tree* with the signal-based characterization of the video contained in the *Sequence Tree*.

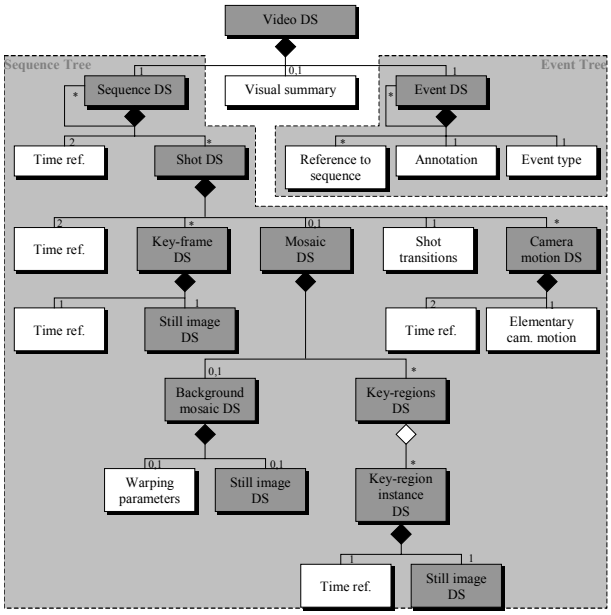

Figure 5: Global view of the video DS

Let us recall that a shot is a continuous set of frames without editing effects. The goal of the **Shot DS** is to characterize the signal properties of this set of frames. The set of descriptors includes two time references (start & end), a characterization of the transition effects at both sides of the shots, a DS devoted to the camera motion (which is assumed to be continuous) and some visual information. The camera motion can be characterized by parameters such as the camera translation, rotation, focal distance, etc and by typical activity classes such as "static", "pan", "zoom", "tilt", etc. As the camera activity may not be constant during the entire shot, homogeneous activities may be segmented in time and are represented by the Elementary camera motion descriptor.

The visual information can be characterized, as a first example, by a set of key-frames. Each key-frame is a still image extracted from the shot. As a result, the key-frame DS is composed of the Still Image DS presented in section III plus a time reference. If the visual content remains stable during the shot, one (or a few) key-frame(s) is(are) sufficient to characterize the entire shot. In the presence of significant changes due to camera motion, an alternative representation consists of a background mosaic [5] plus a set of foreground regions (called "key-regions"). In practice, key-regions have been identified as not belonging to the background because they have a different motion or are not in the same depth plane. This information is gathered in the mosaic DS which is decomposed into one background mosaic DS and several Key-region DSs. The **background mosaic DS** is also a static representation that is conveniently described by the Still Image DS. This description can be enhanced by making available the set of warping parameters used to create the mosaic. The **key-regions DS** is decomposed into several key-region instances. This decomposition is included here in order to be able to represent various visual instances of the same region within the shot. Typical examples include various instances of regions representing a face or a human body. Finally, each key-region instance is described by a Still Image DS and a time reference.

The global view of the Video DS is summarized in Fig. 5. Finally, note that even if the Still Image DS is primarily used to characterize static information such as key-frames, or key-objects, its description accuracy can be improved by describing the time evolution of its composing regions.

5. SUPPORTED SEARCH FUNCTIONALITIES

These DS support a large number of search functionalities related to the visual properties of the image or the sequences. In the sequel, the most important type of queries are discussed.

5.1. Semantic queries

The *Object* and *Event Trees* concentrate the semantic information extracted during the indexing process. The semantic value of possible object types, identities, activities and event types is assumed to be precisely defined in thesauruses. As a result, if the corresponding semantic notion has been recognized during the indexing process, the description is able to answer to semantic queries. The search process is straightforward, since the query has simply to be translated into a set of thesaurus indexes that are matched with the indexes involved in the description. Furthermore, part of the semantic description is also included as free text in the Annotation descriptors. In this case, classical text-based search and retrieval is assumed to be used.

5.2. Signal & Similarity-based queries

The *Region* and *Event Tree* define the visual appearance and the structure of the signal. As a result, this part of the description supports signal & similarity-based queries. Signal-based queries refer to all queries related to region color or geometry, to time references, to shot transitions, to camera activity. Typical examples include finding regions with a specific color, size, orientation, shape characteristics or sequence with a specific camera motion or any combination of these properties.

5.3. Combined semantic and signal-based queries

A typical example of collaboration between the semantic and the signal-based description deals with queries referring to composition of objects or sequences. For instance, it is not realistic to describe in the *Object Tree* all possible spatial relations between objects. Queries of the type: "find images where object *A* is placed just below object B which is in turn on the right side of object *C*" have to be solved by a search involving both the *Object* and the *Region Tree*. Indeed, the *Region Tree* describes all possible spatial relationships between objects that refer to regions.

5.4. Flexibility to deal with unexpected queries

The main interest of the *Object* and *Event Trees* is their efficiency to deal with high level queries. However, they have limited flexibility. Indeed, during the indexing process, one has defined all the objects and events of interest and their corresponding descriptors. In practice, the user has made a selection of the objects and events he/she thinks might be of interest for the retrieval phase. In the example of Fig. 2, the user has assumed that the most important objects are instances of "body", "face", "hair", "torso", "TV" and "jacket". Objects such as "microphone", "eye", "mouth", "shirt" were discarded and no information about them is available in the *Object Tree*. What can be done if the query is unexpected, that is if it addresses an object, an event or a property that has not been considered as being relevant during the indexing process? A partial answer is given by the *Region* and *Segment*

Trees and their relation with the *Object* and *Event Trees*. Let us consider two examples dealing with still images:

1) Unexpected query involving an unknown property of a known object: Considering the example of Fig. 6, assume the following query: "find images of a woman with dark hair". The search process may involve 3 steps and would naturally start with the *Object Tree* in order to select images with a woman:

1. Analyze the *Object Trees* to use as much as possible the semantic descriptors and efficiently pre-select candidates. Find images with a descriptor of type "woman".
2. Look for the "hair" descriptor that is related to the "woman" descriptor. At this stage, we assume that the "hair" has been considered as an object of interest during the indexing process.
3. Since no information about the hair color is available in the *Object Tree* (this object property has not been considered as being of interest during the indexing process), the search has to look for information in the *Region Tree*. The region describing the visual properties of the "hair" object is found by following the reference from the "hair" object to the *Region Tree*. The color descriptor of the region (here R_6) gives the final answer.

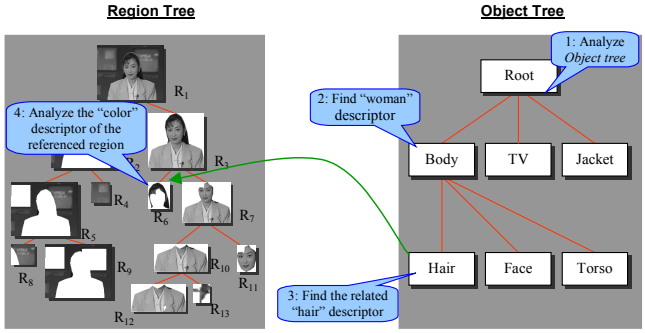

Figure 6: example of the search process for a query involving semantic and signal properties: "find images of a woman with dark hair".

2) Unexpected query involving an unknown object: Consider the example of Fig. 7 and assume that the query is: "find images of a woman speaker with a microphone". The problem with this query is that the object "microphone" has not been considered as being of interest during the indexing process. In this case, two solutions are possible:

- either to say that the second part of the query cannot be processed and that only the first part will be dealt with, or
- to open the door to intelligent search engines that could try to figure out whether the unknown object is present or not (Note that the following example implies a fair amount of intelligence in the search engine side. The goal is not to claim that this solution is easily implementable with today's technology, but to show the DS flexibility and how it could cope with future progresses. In particular, this solution allows using more intelligent search engines as technology evolves).

Let us describe a possible search scenario involving 4 steps as described by Fig. 7:

1. Analyze the *Object Trees* to use as much as possible the semantic descriptors and pre-select candidates. Find images with a descriptor of type "woman" with activity "speaking".
2. Look for one (or several) sub-component(s) of the body where the microphone might be located, for example the "Torso".
3. Go to the *Region Tree* following the reference from the "Torso" object. Analyze the corresponding region R_{10} and its sub-regions.
4. The search engine finds region R_{13} that is small, located in the center of region R_{10} and that is dark. Taking into account all these pieces of information, the search engine may come to the conclusion that the probability of R_{13} being a microphone is high. As a result, the image matches quite well the query.

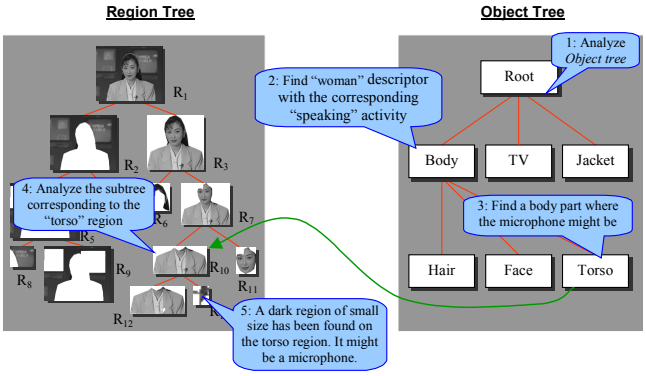

Figure7: example of search process for a query involving an unknown object: "find images of a woman speaker with a microphone"

6. CONCLUSIONS

This paper proposes two DSs to describe the visual information of an AV document. The *Still Image DS* describes the visual appearance and structure of an image as well as its semantic content in terms of objects. The *Video DS* describes the sequence structure as well as its semantic content in terms of events. Features related to sequence properties such as motion, camera activity, etc. are included in this DS.

ACKNOWLEDGEMENT

This work was carried out within the ACTS project DICEMAN in collaboration with France-Telecom CCETT. The work was part-funded by the European Commission. The views expressed are those of the authors and should not be taken to represent the views of the European Commission or its services.

REFERENCES

[1] Y. Rui, T. Huang, S.F. Chang, Image retrieval: Past, Present and Future, accepted for publication in Jour. of Visual Communication and Image Representation, 1999

[2] MPEG-7: context and objectives, Tech. report ISO/IEC JTC1/SC29/WG11/w2460, Oct. 98, Atlantic City.

[3] P. Salembier, L. Garrido, Binary partition tree as an efficient representation for filtering, segmentation and information retrieval, IEEE ICIP'98, Chicago (IL), USA, Oct. 4-7, 98.

[4] Y. Rui, T. Huang, S. Mehrotra, Exploring video structure beyond the shots, Proc. of IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS), pp237-240, June 28-July 1, 1998, Austin, Texas USA

[5] H. Sawhney, S. Ayer. Compact representations of videos through dominant and multiple motion estimation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 18:814-830, August 1996.