ELSEVIER

Signal Processing: *Image Communication* 16 (2000) 211–234

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.nl/locate/image

Description schemes for video programs, users and devices

P. Salembier[a,*], R. Qian[b,1], N. O'Connor[c], P. Correia[d], I. Sezan[b], P. van Beek[b]

[a]*Universitat Politècnica de Catalunya, Barcelona, Spain*
[b]*Sharp Laboratories of America, Camas (WA), USA*
[c]*Dublin City University, Dublin, Ireland*
[d]*Instituto Superior Técnico, Lisboa, Portugal*

Abstract

This paper presents a set of description schemes (DS) dealing with video programs, users and devices. Following MPEG-7 terminology, a description of an AV document includes descriptors (termed Ds), which specify the syntax and semantics of a representation entity for a feature of the AV data, and description schemes (termed DSs) which specify the structure and semantics of a set of Ds and DSs. The Program DS is used to describe the physical structure as well as the semantic content of a video program. It focuses on the visual information only. The physical structure is described by the temporal organization of the sequence (segments), the spatial organization of images (regions) as well as the spatio-temporal structure of the video (regions with motion). The semantic description is built around objects and events. Finally, the physical and semantic descriptions are related by a set of links defining where or when instances of specific semantic notions can be found. The User DS is used to describe the personal preferences and usage patterns of a user. It facilitates a smart personalizable device that records and presents to the user audio and video information based upon the user's preferences, prior viewing and listening habits, as well as personal characteristics. Finally, the Device DS keeps a record of the users of the device, available programs, and a description of device capabilities. It allows a device to prepare itself based on the existing users, profiles and available programs. These three types of DSs and the common set of descriptors that they share are designed to support personalization, efficient management of AV information and the expected variability in the capabilities of AV information access devices.

Keywords: MPEG-7; Description schemes; Indexing; Retrieval; Multimedia content; Signal description; Semantic description

1. Introduction

Multimedia content provides information and entertainment to a wide range of people. Furthermore, the amount of multimedia content that one can access is increasing at a rapid speed. Besides professional users, also in many households today audiovisual information can be obtained from multiple sources such as cable television, satellite dish, radio, world-wide-web, CD/DVD/tapes, etc. In addition, users can create multimedia content using their personal cameras and computers. To help users find and retrieve relevant information effectively, and to facilitate new and better ways of

* Corresponding author.
[1] Now with Intel Architecture Labs in Hillsboro, Oregon, USA.
E-mail addresses: philippe@gps.tsc.upc.es (P. Salembier), richard.j.qian@intel.com (R. Qian), oconnorn@teltec.dcu.ie (N. O'Connor), paulo. correia@lx.it.pt (P. Correia), sezan@sharp-labs.com (I. Sezan), pvanbeek@sharplabs.com (P. van Beek).

0923-5965/00/$ - see front matter
PII: S0923-5965(00)00026-6

entertainment, advanced technologies need to be developed for browsing, filtering, and searching the vast amount of multimedia content available. There are ongoing efforts towards new advances in hardware and software technologies and the communication infrastructure. In addition, there are efforts aimed at developing exchangeable formats capable of hosting rich descriptions of (i) the multimedia content, (ii) the users of the content, and (iii) the multimedia devices that access and consume the content, so that effective browsing, filtering, and search may be performed on the basis of this description data. This paper is concerned with such descriptions. We focus on video programs and consider descriptions of their visual content as well as users and devices that access and consume these programs.

To define exchangeable formats, MPEG has initiated a new work item, formally called "Multimedia Content Description Interface", better known as MPEG-7 [11]. In the context of MPEG-7, a description of an AV document includes descriptors (termed Ds), which specify the syntax and semantics of a representation entity for a feature of the AV data, and description schemes (termed DSs) which specify the structure and semantics of a set of Ds and DSs. Descriptions are expressed in a common description definition language (DDL) to allow their exchange and access. In this paper, we present (1) a DS for describing the visual content of a video program – Program DS; (2) a DS for describing a user of audiovisual content – User DS; and (3) a DS for describing an audiovisual device – Device DS. The proposed description schemes support the following functionalities:

- Effective content-based filtering and searching of audiovisual information;
- Efficient interactive browsing of audiovisual information;
- Personalizable browsing, filtering and searching of audiovisual information and the ability to personalize audiovisual systems regardless of their brand name and physical location;
- Integrated representation of still images and video.

The following is a short overview of the description schemes proposed.

A Program DS is used to describe both the physical structure and semantic content of a video program. The physical structure involves the description of the temporal organization of the sequence (segments), the spatial organization of images (regions) as well as the spatio-temporal structure of the video (regions with motion). The semantic description is built around objects and events. Finally, the physical and semantic descriptions are related by a set of links defining where or when instances of specific semantic notions can be found. The Program DS allows filtering and search to be performed based on the content of a video program. It also enables a user to access only a portion of a particular video program that the user is interested in, while skipping the remainder of the program.

A User DS is used to describe the personal preferences and usage patterns of a user. It facilitates a smart personalizable device that records and presents to the user audio and video information based upon the user's preferences, prior viewing and listening habits, as well as personal characteristics. It permits the device to automatically discover and record desirable information and to automatically customize itself to the user. The user information contained in the User DS should be portable and usable by different devices so that other devices may likewise be configured automatically to the particular user's preferences upon receiving the user information regardless of their brand name or physical location.

A Device DS keeps a record of the users of the device, available programs, and a description of device capabilities. It allows a device to prepare itself based on the existing users' profiles and available programs. It also allows efficient communication between different devices. For example, a content provider may supply a customized version of the content to a particular device based on a description of its capabilities.

There is a synergistic interrelation amongst the three types of DSs in the following sense. The Program DS and the User DS use a common vocabulary of descriptors, at least partially, so that the potential desirability of a program can be determined by comparing descriptors representative of

the same information. For example, a Program DS and a User DS may include the same set of program categories and actors. A Program DS and a device DS should also include partially overlapping descriptors. With the overlapping descriptors, a device DS will be capable of storing the information contained within a Program DS, e.g., program category, so that the content-related information is properly indexed. With proper indexing, a device is capable of matching such content related information with the user information, if available, for instance for obtaining and recording suitable programs. A User DS and a Device DS should also include partially overlapping descriptors. With these overlapping descriptors, a device can capture the desired device-related information, which would otherwise not be recognized as desirable. A Device DS preferably includes a list of users and available programs. Based on the master list of available programs, and associated Program DS, a device can determine the desired programs for each one of its users.

In the following section, we present a Program DS for describing the visual content of a video program. In Section 3, we present a User DS for describing the personal preferences and usage history of a user. In Section 4, we present a Device DS for describing the capabilities of a device and keeping a record of its existing users and available programs. Finally in Section 5, we summarize our work to date.

2. Program DS

2.1. Table of contents and index

The Program DS is largely inspired from the classical way of describing the content of written documents such as books: the *Table of Contents* and the *Index* [18,21]. 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 linear structure of the book itself. As a result, the *Table of Contents* is a representation of the linear, one-dimensional structure of the book. Although the titles of the various sections may carry semantic information, the main goal of the *Table of Contents* is not to describe the content itself but to define the structure of the document. It can be considered as a DS and the corresponding descriptors are the page numbers defining the beginning of each section. Furthermore, the *Table of Contents* describes the entire document and leaves no "holes" ("holes" would be pages not assigned to any sections). The role and the structure of the *Table of Contents* are described in the left side of Fig. 1.

The goal of the *Index* is not to define the linear structure of the book, but to define a set of potentially interesting items and to provide references to the book sections where these items are discussed. In order to be of practical interest to human users,

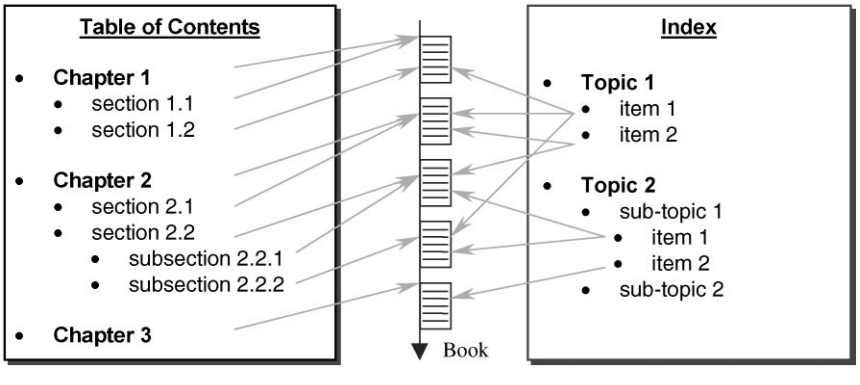

Fig. 1. Description of the content of a book by its "Table of Contents" and its "Index".

the items are selected based on their semantic value. 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 (Note that 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 for the user. Finally, let us mention that for practical reasons, the references involved in the *Index* are generally defined in terms of page number. However, if the *Table of Contents* has a sufficiently fine granularity, the index could directly refer to subsections of the *Table of Contents*.

As can be seen, the classical approach for book content description relies on a dual strategy: (1) definition of the physical or syntactic structure of the document (the *Table of Contents*) and (2) definition of the semantic structure and the locations where semantic notions appear (the *Index*). In the case of visual information of AV material, the description has to deal with temporal (1D), spatial (2D) as well as spatio-temporal (3D) information. Let us illustrate the concept of syntactic structure and semantic structure for spatial and temporal information of AV material.

The first example deals with 2D images and is illustrated in Fig. 2. The description involves two hierarchical structures termed a *Region Tree* and an *Object Tree*. The *Region Tree* defines the syntactic structure. Its main purpose is to describe the spatial organization of the image. Note that the structure of the document is now 2D and not only 1D. 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 (see for example, region R_9 in Fig. 2). The structure of the *Region Tree* defines the inclusion relationship between elementary regions. Note that the tree describes the entire image (similarly, the *Table of Contents* describes the entire book without leaving any holes).

The *Object Tree* is devoted to the semantic structure. It is composed of a list of objects that were judged as being of potential interest during the indexing process. Objects have a semantic meaning. Moreover, one of the important functionalities 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 relationship represented in this hierarchy is of the type "is-made-of". For example, the object "Body" is made of sub-entities

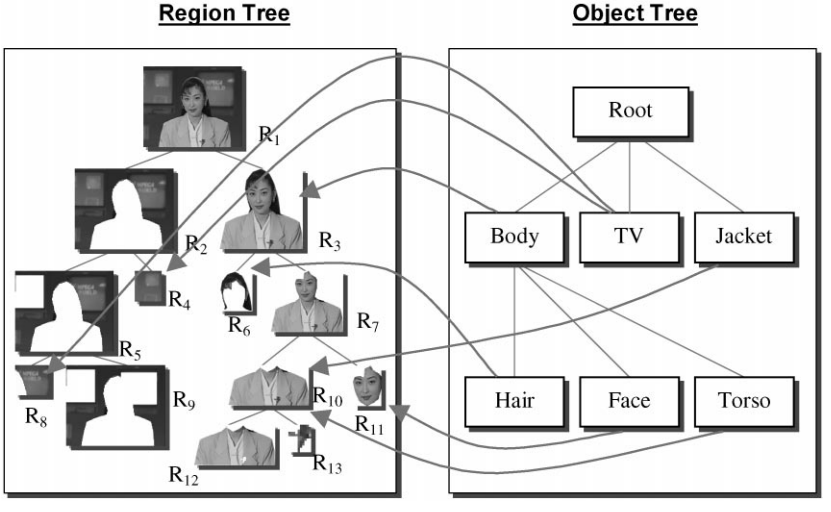

Fig. 2. Simple example of a *Region Tree* (syntactic structure) and an *Object Tree* (semantic structure) describing a still image. Links relate a semantic notion (object) with its occurrences in the image (regions).

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" refers to regions R_4 and R_8 since two *different* 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 at the same time by the "Jacket" and by the "Torso" objects. Finally, note that all objects should refer to at least one region but not all regions have to be referred to by an object. In this case, we have an unidentified region, that is a region described only by its visual appearance (color, geometry, etc.) and no semantic value has been associated to it. Examples of unidentified regions in Fig. 2 are R_1, R_2, R_5, R_9, R_7, R_{12}, and R_{13}.

Fig. 3 shows an example of a description dealing with the temporal information of a video. The description also involves two hierarchical structures represented by trees. The first one is devoted to the syntactic structure and is called the *Segment Tree* whereas the second one describes what is happening, i.e. the semantics, and is termed the *Event Tree*. A segment is a group of contiguous frames in a video program. A segment may contain an arbitrary number of (sub-)segments and/or shots. This creates a Segment Tree where shots are the leaves of the tree. An event may contain an arbitrary number

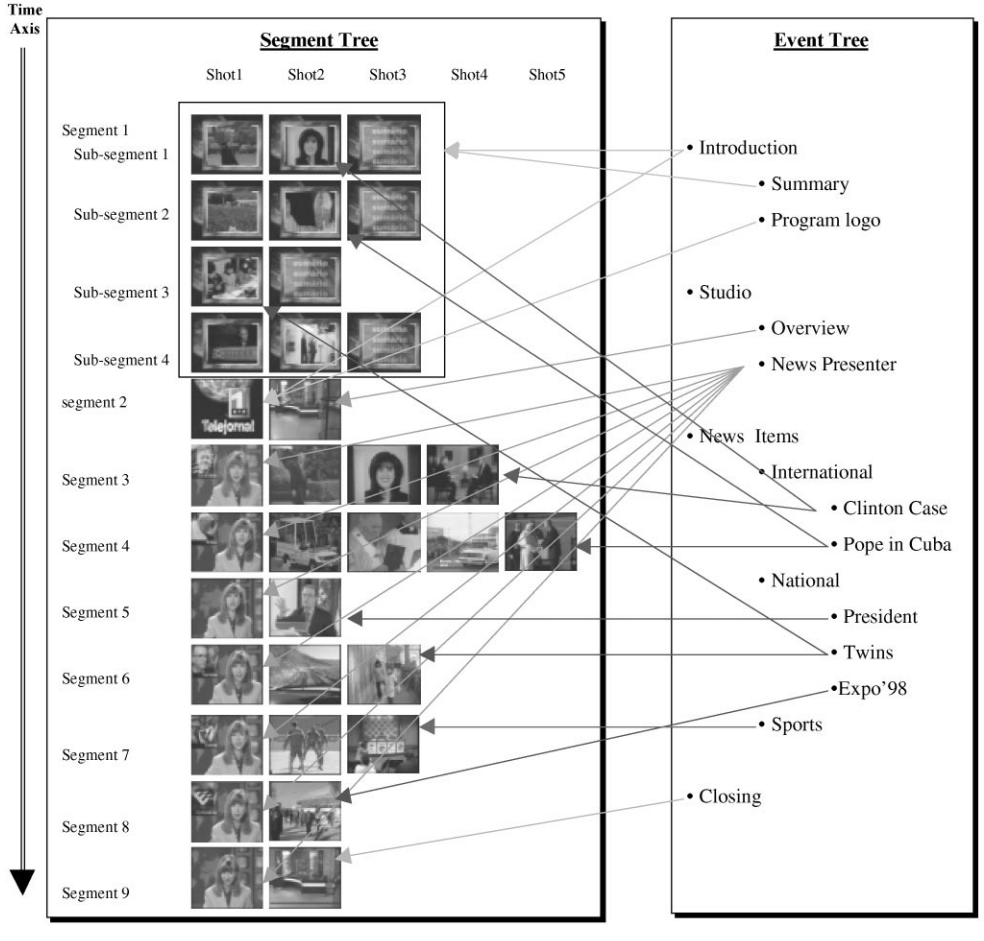

Fig. 3. Example of *Segment Tree* (syntactic structure) and *Event Tree* (semantic structure). Shots are described with one keyframe.

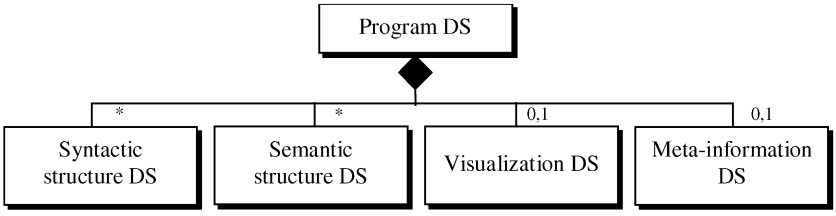

Fig. 4. Program DS.

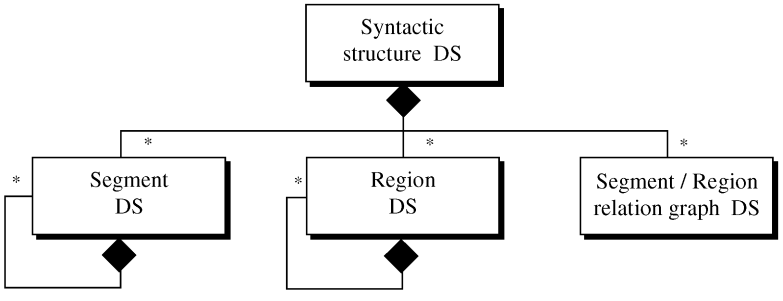

Fig. 5. Syntactic structure DS.

of (sub-)events and therefore form an Event Tree. The links from the Event Tree to the Segment Tree relate semantic notions (events) with one or several occurrences of these notions in time.

This strategy based on syntactic and semantic structures forms the basis of the Program DS presented in Fig. 4. It is composed of a Syntactic structure DS that describes the physical and signal-based properties of the video and a Semantic structure DS that deals with semantic notions. Moreover, the description is enhanced by a Visualization DS and a Meta information DS. In this diagram, the symbol ◆ defines an aggregation of sub-DSs (a symbol ◇ would represent the aggregation of elements of the same type). The numbers close to the rectangles indicate the cardinality with which the respective elements may be present in the DS (a "*" denotes an arbitrary number).

2.2. Syntactic structure DS

The syntactic structure DS describes the physical organization of the video program and its signal-based properties. The temporal structure of the signal is mainly described by Segment DSs whereas the spatial or spatio-temporal structure is defined by Region DSs. The Region DS is able to describe regions within a single image as well as regions across multiple frames (that is a region with its temporal trajectory). Segment and Region DSs are hierarchical and describe the physical structure of the Content for 1D (time), 2D (space) or 3D (space-time) signals. The tree decomposition represents a partitioning relationship: the set of region (or segment) of the child nodes is a partition of the region (or segment) of the father node. Note that the description may involve several Segment and Region DSs to reflect several ways of analyzing the signal. In this way, segment trees and region trees allow the creation of various Tables of Contents. However, trees, by contrast to graphs, include restrictions in the relationships among nodes they can express. To improve the description flexibility, the Syntactic DS also involves a Segment/Region relation graph DS. Fig. 5 describes the organization of the Syntactic structure DS.

2.2.1. Segment DS

The major functionality of the *Segment Tree* is to define the temporal structure of the video program

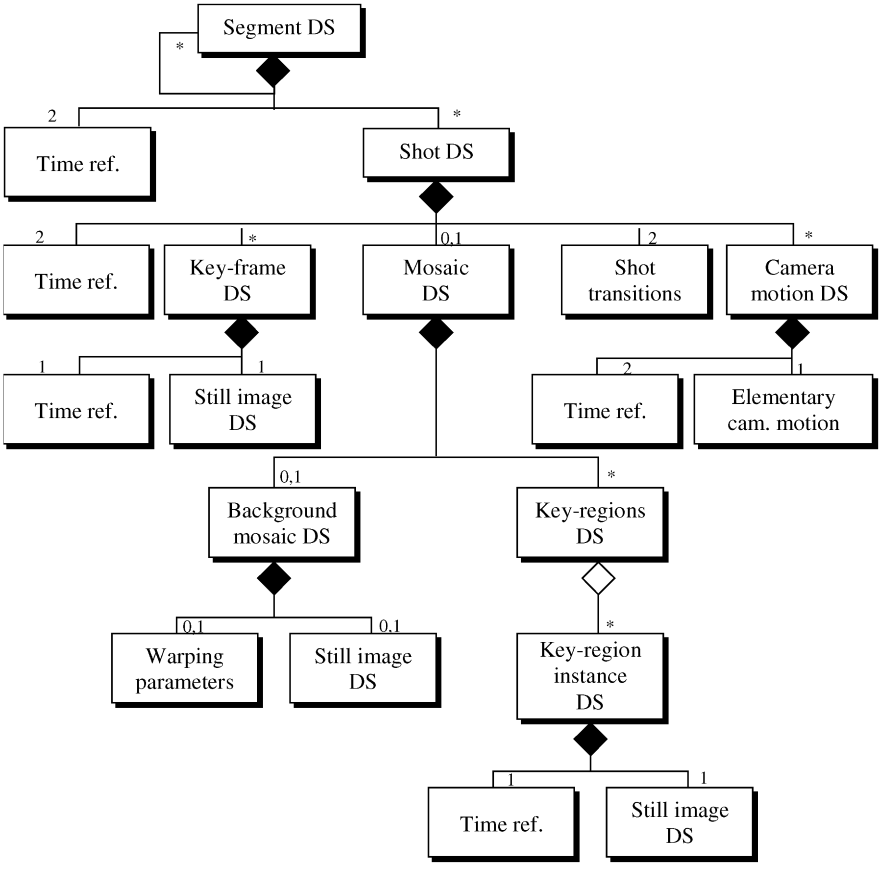

Fig. 6. Description of the Segment and Shot DSs.

and to describe its visual properties. Each node of the tree represents a connected component in time called "segment". The tree describes how each segment can be divided into shorter (sub-)segments (see Fig. 6). The leaves of the tree are assumed to be shots or at least parts of shots.

The *Segment Tree* involves descriptors related to the visual properties of the video signal. The descriptor associated to non-leaf nodes of the *Segment Tree* (segments that are not shots) is simply a time reference indicating the beginning and the end of the corresponding segment. The leaves (the shots) are more precisely described. Let us recall that a shot is a continuous set of frames without editing effects. Most of the time, the shot boundaries coincide with two editing effects such as "cut", "fade", "dissolve", etc. The main goal of the Shot DS is to characterize the signal properties of the corresponding 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: Keyframe and Mosaic DSs. 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 keyframes. Each keyframe is a still image extracted from the shot. The **Keyframe DS** is composed of a time reference and a Still Image DS. The Still Image DS is a particular case of the Program DS (it corresponds to the Program DS without temporal description). If the visual content remains stable during the shot, one (or a few) keyframe(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 (a panoramic image constructed by accumulating all the background components that appear during the shot [19,1]) plus a set of foreground regions (in the following, we call these regions "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 mainly for browsing functionality by making available the set of warping parameters used to create the mosaic (the warping parameters are the parameters defining the geometrical transformation applied to each image in order to represent it within a common reference system). 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. Note that, with the Still Image DS, the background mosaic and the key-regions can be decomposed into elementary regions which may be described by their geometrical, color, and motion properties.

An illustration of the difference between Keyframes and Mosaic DSs is presented in Figs. 7 and 8. Fig. 7 shows the sequence timeline and three keyframes. The three keyframes describe the sequence content at three time instant t_0, t_1 and t_2. An alternative description relying on a Mosaic DS is given in Fig. 8. The representation involves a background mosaic plus three key-regions. Key-region 1 corresponds to a man walking. During the analysis/indexing process, it has been judged that one visual instance is enough to characterize its appearance. Key-regions 2 and 3 are represented by two instances because their visual appearance exhibits significant variation during the shot. The man represented by Key-region 2 is crouching at time t_1 and walking at time t_2. Key-region 3 is

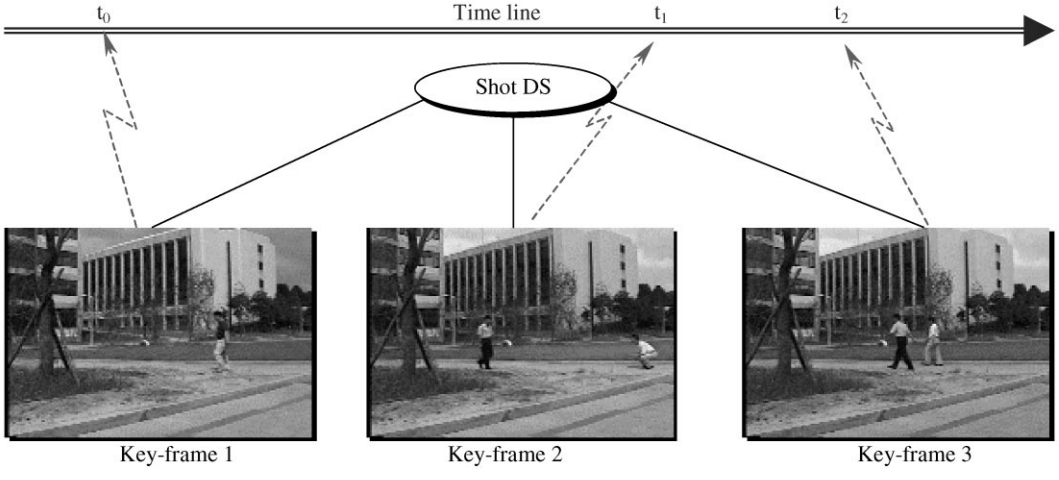

Fig. 7. Shot representation with the keyframe DS.

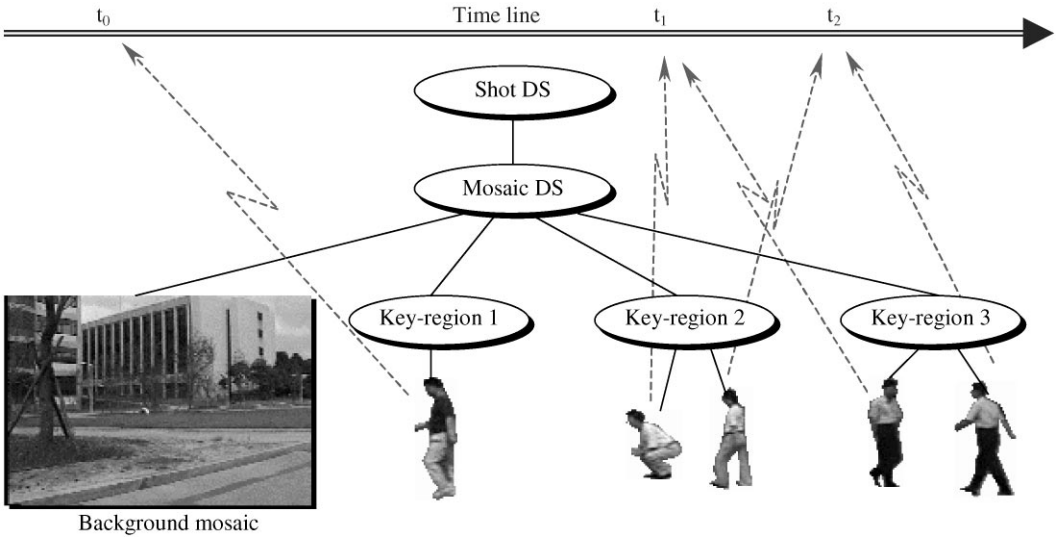

Fig. 8. Shot representation with the Mosaic DS.

describing a man walking towards the camera at time t_1 and then walking away at time t_2. Note that the example shows a description of a shot based on keyframes, and also a description of a shot based on mosaic. In practice, these two descriptions are not exclusive and they can be used jointly to describe a shot.

Although it is not the intention of this paper to propose specific descriptors, it is believed that the precise definition of potential descriptors related to the DS may provide some clarification and allow a better understanding of the approach. Beside the Still image DS, the list of descriptors involved in the Segment DS addresses the following features: time reference, shot transitions, elementary camera motion, and warping parameters. The time reference can be expressed in terms of frame number, second or the SMPTE timecode. To define a time segment, two descriptors are necessary (start-end). This is why the segment, shot and Elementary camera motion DSs involve two time reference descriptors. A simple solution for the shot transitions descriptor consists of using a thesaurus of shot transitions ("cut", "fade", "dissolve", "wipe", etc.). Two index values should be provided to characterize transitions at the beginning and the end of the shot [14,27].

An attractive solution for elementary camera motion descriptor consists of describing quantitatively as well as qualitatively the evolution of the camera parameters. Concerning the quantitative characterization, the most important parameters include [22]: the focal distance, the translation in the 3D space, the rotation with respect to the 3 axes and the zoom factor. The qualitative description of the camera activity can involve an index referring to a thesaurus of typical activities: "Static", "Pan", "Tilt", "Zoom in", "Zoom out", etc.

As shown in Fig. 6, the background mosaic DS involves the warping parameters. These parameters define the set of geometrical transformations that has been applied to each individual frame in order to create the mosaic representation [19]. Most of the time, the transformation corresponds to the global apparent motion of the scene and may be closely related to the camera motion parameters.

The *Segment Tree* creation can be viewed as a hierarchical segmentation problem [18,25,27]. In the case of the *Segment Tree*, the leaves of the tree structure are shots. Therefore, a possible bottom-up solution could consist of first detecting the shots and then merging or clustering them by similarity. A large number of such algorithms have been published in the literature. Most of the process can be

automatic, however, human supervision should not be excluded to correct possible mistakes.

The estimation of the number of elementary camera motions, the number of key-frames, the number of key-objects and the number of the instances of each key-region can also be considered as a segmentation problem. In all cases, several instances (of camera motions, key-frames, key-regions, etc.) are included in the description because of the possible lack of homogeneity either in the extracted parameters or in the visual representation. The segmentation processes are based on low-level features such as:

- similarity between the camera parameters at different time instants, or
- visual similarity between key-frames or key-regions, or
- motion and shape similarity between various temporal instances of a key-region.

As a result, large parts of the various segmentation processes can be automated.

2.2.2. Region DS

The *Region Tree* defines the structure of spatial or spatio-temporal information. Its main purpose is to describe the spatial organization of an entire image or of a region within an image. The region can be considered as being static (pure spatial description) or as evolving in time (spatio-temporal description). It involves descriptors related to the signal properties. It may be created automatically or at least semi-automatically. In particular, regions appearing on the lower level of the tree should be defined by their homogeneity in terms of signal properties (color for example).

The *Region DS* is shown in Fig. 9. The first node represents the support of the entire image or region. It is described by a color descriptor, a geometry descriptor, a motion trajectory descriptor and an arbitrary number of (sub-)region DSs. The geometry descriptor deals with features such as position, size, orientation and shape. The Region DSs describe how the image can be split into various components. If the region is a leaf node of the tree (i.e. it is not further sub-divided), it is described using only color, geometry and motion trajectory descriptors.

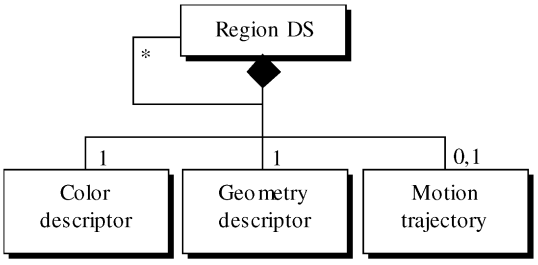

Fig. 9. Region DS.

Note that the tree structure introduces a notion of scalability in the description itself. A given region in the *Region Tree* can be described by its own descriptors. This can be considered 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. Suppose, for example, that the color descriptor is a set of three values describing the mean color of the region in the YUV space. The first level of description will only provide a rough approximation of the color characteristic of the region. However, if the mean color values of all the leaves related to the region are taken into account, together with the size and shape information, a very precise description may be obtained. The description scalability is illustrated in Fig. 10. Assume that we are interested in a region A. The first row of the figure shows the sub-tree hanging from A and the information that can be extracted from the descriptors attached to A. The color and geometry description is presented here by an image. It gives a rough approximation of the shape and of the luminance value. The corresponding luminance histogram is also given. At this level, it is a single value histogram. The second row of the figure shows the information extracted if the descriptors that are three levels below region A are used. Note that since the shape descriptor considered in this illustrative example is highly lossy, the regions do not perfectly fit together and some holes and overlapping may appear. Finally, the finest description is obtained if the leaf descriptors are used. This is shown in the last row of the figure. This description scalability is an attractive feature provided by the tree structure itself. It has been

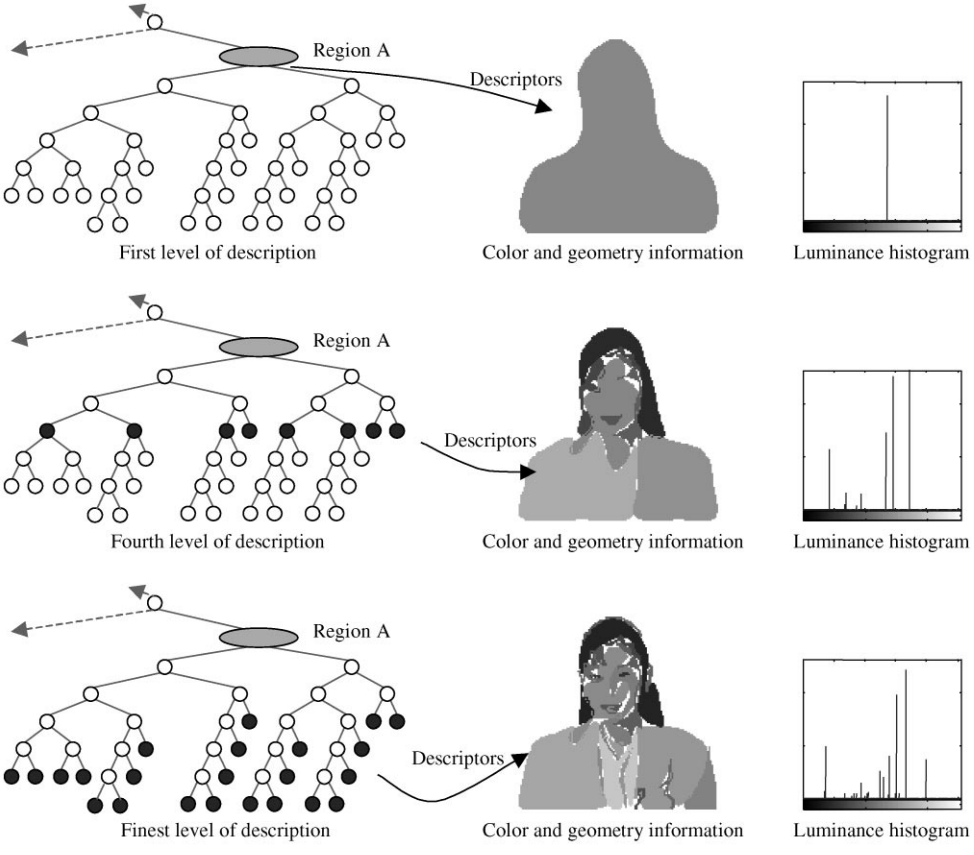

Fig. 10. Illustration of the description scalability.

illustrated here on the Region DS but the same functionality is also available for the Segment DS.

The list of descriptors involved in the Region DS addresses the following features: color, geometry and motion trajectory. Any color descriptor can be used. However, taking into account the multiscale representation ability of the *Region Tree*, a simple and compact descriptor seems attractive. Typically, a low-order polynomial approximation of the color in the YUV or HLS spaces may be used. If the polynomial approximation order is equal to zero, the descriptor is a set of three constant values. Although this is a poor approximation at the level of the region itself, as explained above, it can be improved by accessing the set of descriptors corresponding to the region sub-components. This elementary color description can be enhanced by adding a parameter characterizing the texture of the region. One of the simplest texture descriptors is the variance.

The geometry descriptor encapsulates the information about the spatial properties of the region. It should allow the creation of an approximation of its region of support. To this end, the following features should be addressed: position, size normalization, rotation normalization, reflection normalization and normalized shape. The normalization parameters are introduced in order to be able to derive a normalized shape representation. Descriptors of position, size and rotation are easily extracted using a principal component analysis [9,10]. Consider the population of vectors $x = [x_1, x_2]^T$, where T stands for vector transposition, and x_1, x_2 are the coordinates of the pixels

that belong to a region R. The mean vector is defined as $m_x = E\{x\}$ (where E denotes the mathematical expectation), and the covariance matrix of the population of vectors as $C_x = E\{(x - m_x)(x - m_x)^T\}$ (whose size is 2×2). Since C_x is real and symmetric, finding the two orthonormal eigenvalues is always possible. Let e_i and λ_i, $i = 1, 2$, be the eigenvectors and corresponding eigenvalues of C_x, respectively. The next step is to construct the matrix A, whose rows are formed by the eigenvectors of C_x, ordered so that the first row of A is the eigenvector corresponding to the largest eigenvalue. A is a transformation matrix that maps x into vectors denoted by y by the transformation $y = A(x - m_x)$. The population of vectors y corresponds to the rotation and position invariant representation of the region R. Within this framework, m_x corresponds to the position descriptor, whereas A is a rotation matrix. It is therefore easy to extract the angle α that generates the matrix. Size invariance is accomplished by normalizing by $\lambda_1 + \lambda_2$, which is equivalent to normalization by the mean-squared energy of the population of vectors of y.

The reflection axis with respect to the horizontal and vertical axis (in the rotation invariant space) is computed by estimating the sign of $E\{y_1^3\}$ and $E\{y_2^3\}$, where y_1 and y_2 are the horizontal and vertical coordinates of vectors y, respectively. These values can be estimated directly on R computing $E\{(A_x)^3\}$. The reflection factors make the transformed region invariant to reflection with respect to the vertical and horizontal axis. As a particular case, it solves the π radian uncertainty when computing transformation matrix A.

The external boundary of the region can be coded using techniques like chain code [7], spline approximations, wavelet [3], Fourier descriptors [26,9] or even the shape coding technique used in the MPEG-4 standard. As an example, let us assume that Fourier descriptors are used to code the contour. The contour of R is sampled at equally spaced samples in a clock-wise manner. After applying the position, rotation, size and reflection transformation, the contour is subsampled in order to have a constant length contour. A reduced number of Fourier descriptors are then used to represent the resulting contour. These are the shape descriptors. They are invariant to size, rotation, position and reflection.

Finally, the last descriptor deals with the region motion trajectory. To instantiate this descriptor, the first step is to model the time evolution of a region between two successive frames. This can be achieved by estimating the parameters of a transformation, which describes the apparent motion of the region under study in terms of its image coordinates. The classical choices include projective, affine and constant models [19]. The region trajectory is then given by the set of transformations within the shot. Note that instead of listing the set of transformation parameters at each time instant, it may be more useful to model their evolution. Most of the time, the evolution of the transformation parameters is slow and smooth. As a result, a low-order polynomial approximation may be sufficient to model these parameters.

The *Region Tree* (also called partition tree [20]) is a structured and compact representation of the most "meaningful" regions that can be extracted from an image. Several approaches can be taken to create this tree. An attractive solution consists in using a segmentation algorithm that follows a bottom-up approach [8]. Starting from an initial fine partition, the algorithm recursively merges neighboring regions based on a homogeneity criterion until one region is obtained. The homogeneity criterion used to merge regions can rely on low-level features such as color and texture. However, additional information of previous processing or detection algorithms can be used to generate the tree in a more robust way. For instance, a mask of an object included in the image can be used to constrain the merging so that the object itself is represented with a node in the tree. Typical examples of such algorithms are face, skin or character detection [2,15,24]. As can be seen, most of the process can be done automatically. However, user interaction could also be included either as an initial step, which derives the constraints used to control region merging (e.g. mark with a scribble the semantic objects present in the scene) or as a final step of checking and correction [4]. In this latter case, user interaction can be used to modify the *Region Tree* structure so that more meaningful regions are

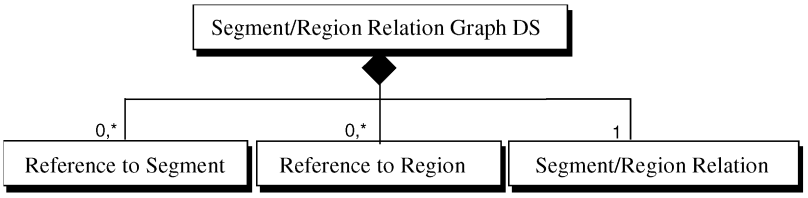

Fig. 11. Segment/Region relation graph DS.

represented in the tree nodes. With respect to current tools and practice, the automatic process is adequate for the lower part of the tree but manual interaction is likely to be necessary to correct the upper part of the tree.

2.2.3. Segment/region relation graph DS

A segment/region relation graph is a conceptual graph in terms of segments, regions, and their relations. Although hierarchical structures such as trees are adequate for efficient access and retrieval, some relationships cannot be expressed using such structures. The relation graph is provided to add some flexibility in the relationships that can be described. For example, it can be used to group together segments or regions that are not neighbors in their respective tree structures. The Relation graph DS is similar to the entity relation graph proposed in [12,13]. It allows expressing both directional and topological spatial and temporal relations (e.g. "before of", "sequential", "top of", "nearby"), as well as semantic relations (e.g. "belongs to", "related to", "part of").

Fig. 11 shows a diagram of the proposed Segment/Region relation graph DS. It includes a relation descriptor and an arbitrary number of references to segments or regions.

2.3. Semantic structure DS

The Semantic structure DS addresses the high-level description of the video program. It involves Event and Object DSs. Both types of DS are hierarchical and the tree decomposition defines an "is-made-of" relationship. The Events and Objects appearing in the semantic DS are assumed to be types of semantic notions and not their instances. The specific semantic instances in the image or the video are described through references with the syntactic description. Note that the description may involve several Event and Object DSs to reflect several ways of interpreting the program. As in the case of the Syntactic structure, the description flexibility is improved by a graph: the Event/Object relation graph DS. Fig. 12 describes the organization of the Semantic structure DS.

2.3.1. Event DS

The *Event Tree* describes semantic notions related to time intervals in the video sequence. It

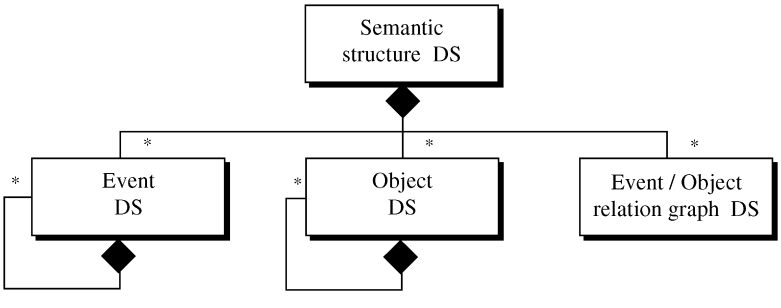

Fig. 12. Semantic structure DS.

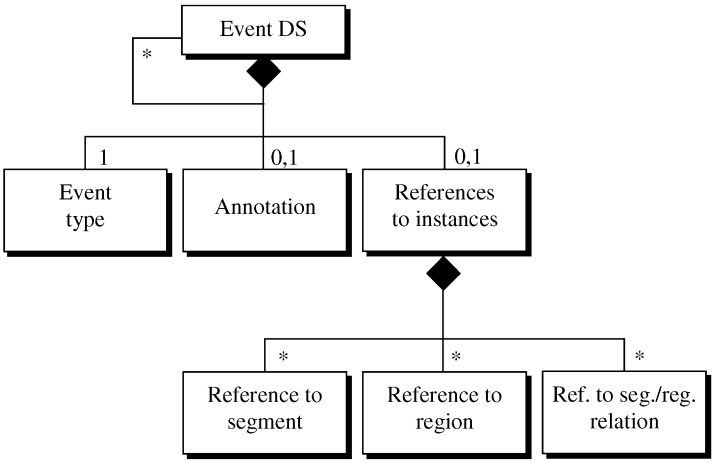

Fig. 13. Event DS.

defines in a hierarchical fashion a set of events and sub-events, characterizes them and relates them with segments, regions or segment/region relations defined in the syntactic structure. The Event Tree concentrates all the descriptors related to the semantic of what is happening during a time interval of the video. Some of these descriptors can be predefined in a thesaurus of events and thus simply consist of indexes defining the type of events. Annotation (free text) is another way to characterize the semantic value of the events. Finally, in order to be able to relate events with the Syntactic structure DS, a set of descriptors called "reference to segment", "reference to region", "reference to segment/region relation" is assigned to each event. Fig. 13 presents the structure of the Event DS.

With today's technology, the construction of the *Event Tree* very much relies on supervised techniques and human interaction. For example, the definition of the tree structure is closely related to the definition of an ontology.[2] It encodes high-level knowledge and is dependent on a semantic interpretation of the content. Most of the time, the automatic recognition of event type is not feasible. However, in some cases where the environment is controlled, the instantiation of the event type as well as the references to segment or region may be done automatically. A typical example is surveillance applications [5], where most of the time the visual context is known (e.g. static camera, known background, etc.) and the events of interests are simple: person entering/leaving a room, car exiting a parking, action [23], etc. Finally, the annotation descriptors must be instantiated manually.

2.3.2. *Object DS*

The *Object Tree* defines semantic notions related to spatial or spatio-temporal information. It is composed of a list of objects that were judged as being of potential interest during the indexing process. This tree is composed of objects with a semantic meaning.

As shown in Fig. 14, each object is described by an Object Type DS, an optional annotation descriptor (free text), an arbitrary number of references to segments, regions, segment/region relations and an arbitrary number of (sub-) Object DSs. The Type DS provides semantic information about the object: type, identity and activity. Note that the identity and the activity of an object depend on its type. For example, the object "face" may have an identity; the object "movie character" may have an activity; etc. Object type and identity are described using a single descriptor for each. Object activity is described using the Object

[2] That is, a (hierarchically) structured specification of the sum total of semantic knowledge to be used in the description.

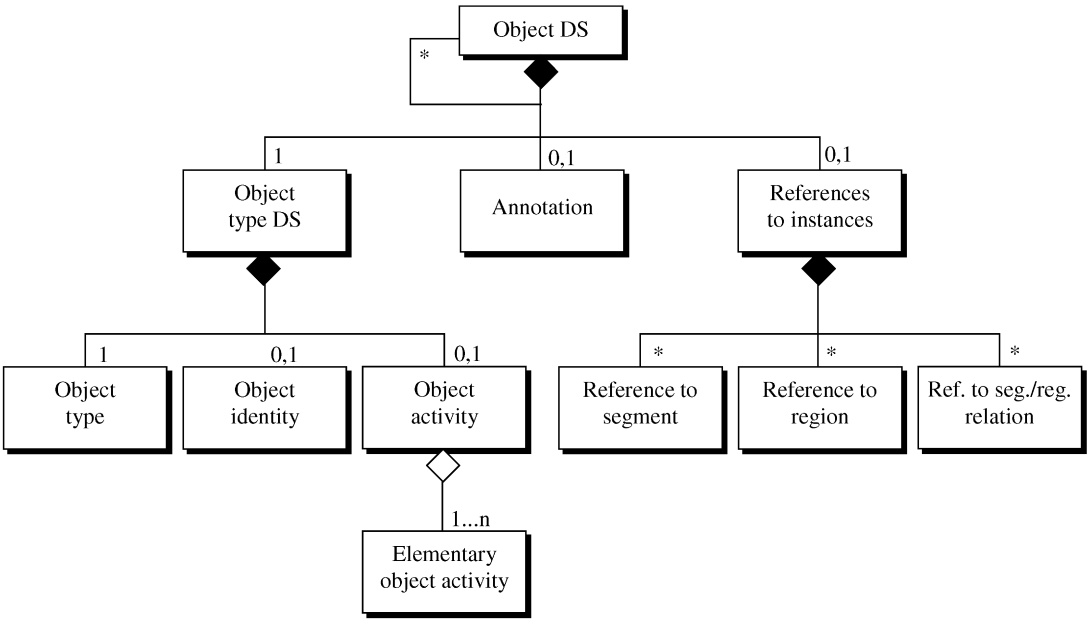

Fig. 14. Object DS.

activity DS. This simply consists of an arbitrary number of elementary object activity descriptors in order to account for the fact that an object can exhibit multiple activities (e.g. 'sitting' and 'speaking'). Note that an object activity could be considered as an event. However, it is described here as part of an object description when it only involves the object. If several objects have to be considered at the same time, then the semantic notion should be described as an event. The exact definition of these type, identity and activity descriptors may involve various thesauruses. In this context, at least three kinds of thesaurus seem to be interesting:

1. Thesaurus of types. This declares the types of objects of interest. Typical examples of objects of interest include face, body, specific accessories, etc.
2. Thesaurus of identity. For some (not all) 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 the object type "face" with a specific person.
3. Thesaurus of activity. Depending on the object type, the activity is an important descriptor describing the image content. The term activity should be understood in a broad sense. The thesaurus may define static activities (such as standing, sitting, etc.) as well as dynamic activities (such as entering, taking something, etc.).

Finally, one of the most important functionality of this tree is to relate its objects to elements of the Syntactic structure DS. This is done through the Reference to instances as in the case of the Segment DS.

As in the case of the Event Tree, the structure of the *Object Tree* encodes high-level knowledge about the world and about the interesting semantic properties of the objects present in the image. With today's technology, it seems rather difficult to automatically create the *Object Tree* structure. Alternative solutions include either to use an already existing tree structure or to create it manually. In the general case, instantiation of type, identity, activity descriptors as well as of the references has to be performed manually. However, for a limited set of objects, the process can be automatic or at least semi-automatic.

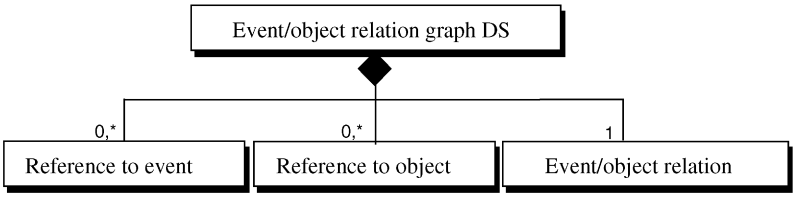

Fig. 15. Event/object relation graph DS.

2.3.3. Event/object relation graph DS

An event/object relation graph is a conceptual graph in terms of events, objects, and their relations. As before, this graph is introduced to add some flexibility in the relationships that can be described. The definition of the semantic structure of the content may very likely require non-tree structures. The Relation graph DS is also similar to the entity relation graph proposed in [12,13]. Fig. 15 shows a diagram of the proposed event/object relation graph DS.

2.4. Visualization DS

The Visualization DS enables fast and effective browsing of video programs by allowing access to the necessary data in a one-step process. It contains a number of view-specific DSs to support customizable representation of a video program and audiovisual summaries. Views including thumbnail view, keyframe view, highlight view, event view, close-up view, and alternate view, are commonly used in a wide range of video applications. For performance reasons, it is advantageous to specify the summary data which are needed to render such views in a centralized and straightforward manner. By doing so, it is then possible to access the data in a simple one-step process without complex parsing in other parts of the Program DS. Fig. 16 shows a diagram of the proposed Visualization DS and the six-view DSs under it. In the following, we explain each view and its associated view DS. They are expressed in [16] using the Extensible Markup Language (XML) notation.

2.4.1. Thumbnail view

The purpose of the thumbnail view is to enable visualization of a video program by a representative still image. The Thumbnail View DS specifies an image as the thumbnail representation of a video program. The thumbnail image may be a particular video frame of the program. In that case it is specified by using the time reference to the program. Alternatively, the thumbnail may be any still image, not necessarily extracted from the video program itself. For example, one can use a still image shot by a still camera during a certain event as a thumbnail for the home video shot during the same event. In that case, the thumbnail image may be specified via an image or a simple link. In any case, at most one thumbnail image can be used to represent a program in a thumbnail view.

2.4.2. Slide view

The purpose of the slide view is to provide a set of images that provide a time-sequential representation of a video program, or presentation of a set of still images that may be grouped, for instance in one particular album. The Slide View DS specifies references to an arbitrary but predetermined set of still images, or frames in a video program, so that they can be viewed as snapshots or in a slide show manner. For instance, still images from an electronic album can be viewed in a slide show. Similarly, a selected set of frames of a video can be viewed in a time-sequential manner at a particular rate.

2.4.3. Keyframe view

The purpose of the keyframe view is to access keyframes of a video program. Keyframes are those frames of a video program that are most representative of its content, and thus serve as a visual summary composed of a subset of its frames. Various keyframe extraction algorithms exist in the literature where different criteria are utilized for

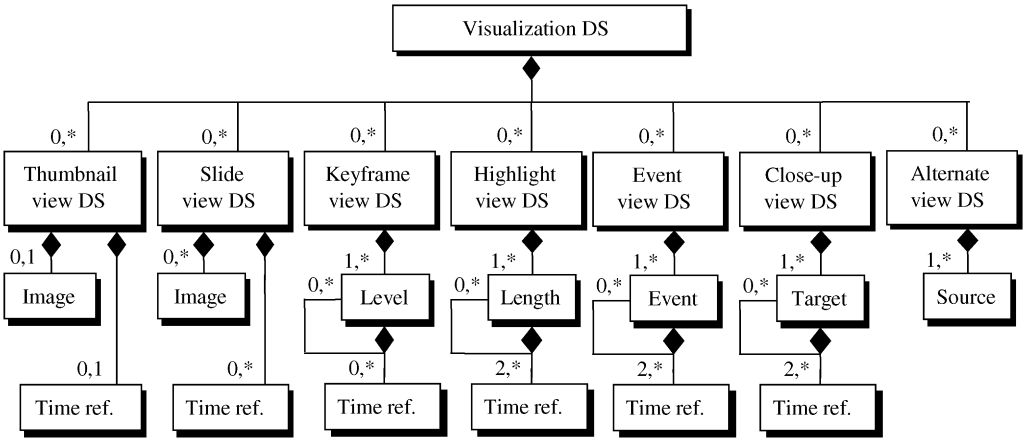

Fig. 16. Visualization DS.

defining representative frames. The keyframes may be organized in a hierarchical manner from coarse to fine temporal resolution, where the number of keyframes, i.e., their temporal sampling frequency, increases at finer levels [17]. The hierarchy is captured by a level attribute. The keyframe view provides a summary of the video in terms of its keyframes at varying levels of detail. Each keyframe represents a clip, for instance the frames of the video that are between that and the next keyframe, where the union of all clips is equal to the entire video. The clips that are associated with each keyframe are referenced by a time reference.

2.4.4. Highlight view

The purpose of the highlight view is to provide a digest of the video formed by a temporal concatenation of its selected sub segments (clips). In particular, the Highlight View DS specifies clips that are concatenated to form a video highlight of a program. A program may have different versions of highlights which are of different time durations. The clips are grouped into each version of the highlight which is specified by a length attribute. For example, 5 and 15 minute highlights of a basketball game may be specified. The clips that are associated with each highlight are referenced by a time reference.

2.4.5. Event view

The event view is similar in nature to highlight view. Clips that contain a particular event, e.g., a goal in a soccer game, are combined together to form an event view, e.g., "Goals View" of a soccer game video. The Event View DS specifies clips that are associated with certain events in a video program. The clips are grouped into the corresponding events that are specified by an event name attribute. The clips that are associated with a particular event are referenced by a time reference.

2.4.6. Close-up view

The close-up view is similar in nature to highlight and event views. The Close-up View DS specifies clips which may be zoomed in to certain targets in a program such as a favorite actor or sports player. The clips are grouped into the corresponding targets which are specified by a target name attribute. The clips that are associated with a particular target are referenced by a time reference.

2.4.7. Alternate view

The purpose of the alternate view is to facilitate the visualization of an alternate view of a particular video program. For instance, an alternate view may be a video clip of a particular scene taken at a different camera angle. The Alternate View DS specifies those sources which may be shown as alternate

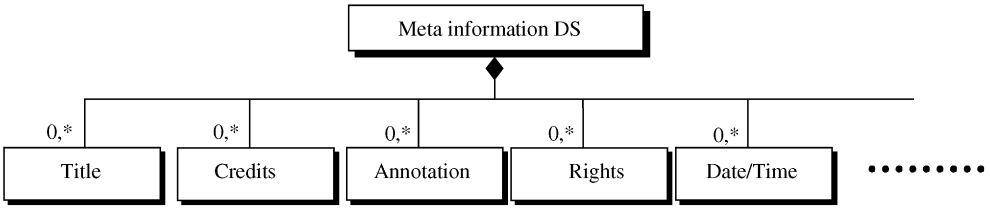

Fig. 17. Meta information DS.

views of a program. Each alternate view is specified by a source ID attribute. The location of the source may be specified in URL format.

2.5. Meta information DS

Fig. 17 shows a diagram of the proposed Meta information DS. The Program DS contains a Meta information DS. The Meta information DS contains descriptors that carry author-generated information about a video program or an image that cannot usually be extracted from the content itself. Example of meta information descriptions are title, author, etc. The Meta information DS is similar to the metadata set defined by the Dublin Core Metadata Initiative (Dublin Core [6]), for the description of electronic resources. Some descriptors that can be included within the Meta information DS are:

Title (the name of the program),
Credits (director, producer, script's writer, music writer, etc.),
Rights (copyrights ownership),
Description (textual abstract),
Subject or Annotations (textual keywords),
Date (date the program was made available in its current form).

Of course, this list can be extended to include other information such as program category, characters, etc.

3. User DS

A User DS facilitates personalized access and consumption of audiovisual information. A User DS contains descriptors that describe a user's preferences, usage history and demographics pertaining to audiovisual content. The User DS can be used to filter programs according to user preferences, make suggestions to the user on the availability of content that fit the user's preference, and take actions on behalf of the user according to user preferences, usage history, and demographics. A User DS enables, for instance, personalized TV viewing, where filtering of programs are performed according to the user profile. Another example for an application that is enabled by the User DS is a smart recording device that programs itself according to users' profiles, and discovers and records broadcast programs accordingly.

A standardized User DS allows users to transport their User DSs from one device to another regardless of their brand name and location, over a network or via removable smart cards. One can for example personalize a hotel room receiver via a smart card. As expected, the User DS contains descriptors that are common to those that describe video programs, such as descriptors of program categories, titles, etc. In other words, the User DS shares a common vocabulary with the Program DS as well as the Device DS that is discussed in Section 4.

The proposed User DS includes three sub-DSs, (1) User preferences DS, (2) Usage history DS and (3) User demographics DS, for describing a user. The User preferences DS contains a number of settings that may be preferable by the user. The Usage history DS contains some statistics which may reflect certain usage patterns of the user. The User demographics DS contains some demographic information about the user. Such information can be used to determine programs that are targeted to users belonging to a particular demographics group. Fig. 18 shows the overall structure of the proposed User DS, which is expressed in [16]

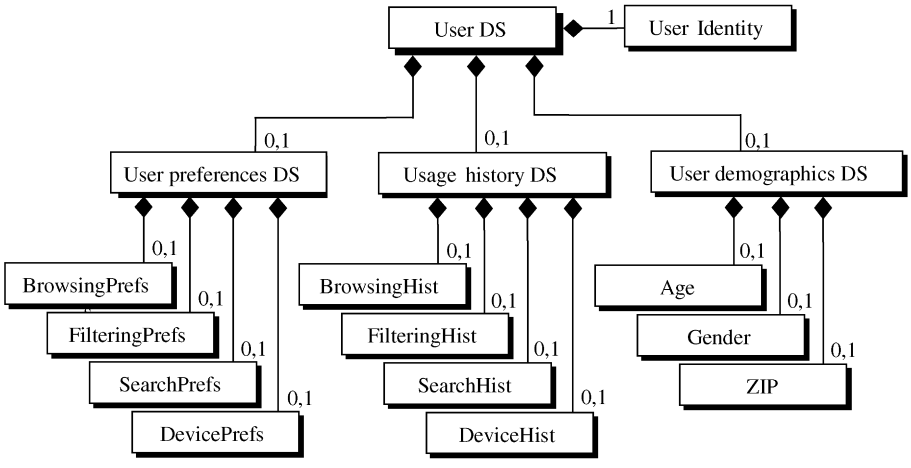

Fig. 18. User DS.

using the Extensible Markup Language (XML) notation.

3.1. User identity

User identity identifies a particular user.

- User ID. The identity of a particular user is determined by a number or a string that identifies the user.
- User name. The name of a particular user is specified by this descriptor.

3.2. User preferences

User preferences are expressed using descriptors of browsing, filtering, search and device-setting preferences.

Browsing preferences. This descriptor specifies the browsing preferences of a user. The user's preferred views are specified by view types discussed in Section 2.4. For example, one user may prefer 10-minutes highlights of basketball games, described by the Highlight view DS. Another user may prefer an event view of the basketball game, such as the "slam–dunk events", described by the Event view DS. The same user's preferred browsing mode for home videos may be a coarse-level keyframe summary, described by the Keyframe view DS.

Filtering preferences. This descriptor specifies the filtering related preferences of a user. Filtering related preferences include descriptors that are used in the Program DS, especially in the Meta-information DS, describing for example the title, date, and category of the program.

Search preferences. This descriptor specifies the search related preferences of a user. Search-related preferences include descriptors that are used in the Program DS, especially in the Meta-information DS, describing for example the title, date, and category of the program.

Device preferences. This descriptor specifies the device-setting preferences of a user. Device settings refer to, for example, brightness and contrast setting of a display device, or volume setting of an audio amplifier.

3.3. Usage history

Usage history of a user is expressed using descriptors of browsing, filtering, search and device-setting related usage patterns of the user.

Browsing history. This descriptor captures the history of a user's browsing-related activities expressed using the descriptors used in the Program DS.

Filtering history. This descriptor captures the history of a user's filtering-related activities expressed using the descriptors used in the Program DS.

Search history. This descriptor captures the history of a user's search-related activities expressed using the descriptors used in the Program DS.

Device history. This descriptor captures the history of a user's device-setting related choices. On the basis of this history, it is possible to customize device settings for different program categories, such as action movies versus drama or opera versus rock music, according to the user's personal taste. For example, the user may prefer a volume setting that is higher for an action movie than a drama.

3.4. User demographics

The User demographics DS contains information that describes some demographic information about the user, like its social group or its residence area.

Age. This descriptor specifies the age of a user.

Gender. This descriptor specifies the gender of a user.

ZIP code. This descriptor specifies the ZIP code of the location where a user lives. (The ZIP code is specific to USA; it may be replaced by some other mail code in other countries.)

4. Device DS

The purpose of a Device DS is to describe a device identified by a device ID. By a device we mean, for example, an audiovisual information appliance with browsing, filtering and search capabilities. By description of a device we mean information about the users of the device, audiovisual content known to the device (including programs that are stored in the device and programs that will be broadcast in the future), and the capabilities of the device. The proposed Device DS includes three sub-DSs, (1) Device users DS, (2) Device programs DS and (3) Device capabilities DS. The Device users DS keeps a list of all known users of the device. The Device programs DS keeps lists of available programs. The Device capabilities DS describes the capabilities of the device, such as its visualization and display capabilities. A particular device may only support some of the views defined in Section 2.4. For instance, a particular device may not be capable of displaying motion video. Such a device may therefore be limited only to the thumbnail view of a video program. Fig. 19 shows the overall structure of the proposed Device DS.

4.1. Device identity

A description of the device itself is given by the descriptors:

Device ID. This descriptor contains a number or a string to identify a video device or device.

Device name. This descriptor specifies the name of a video device or device.

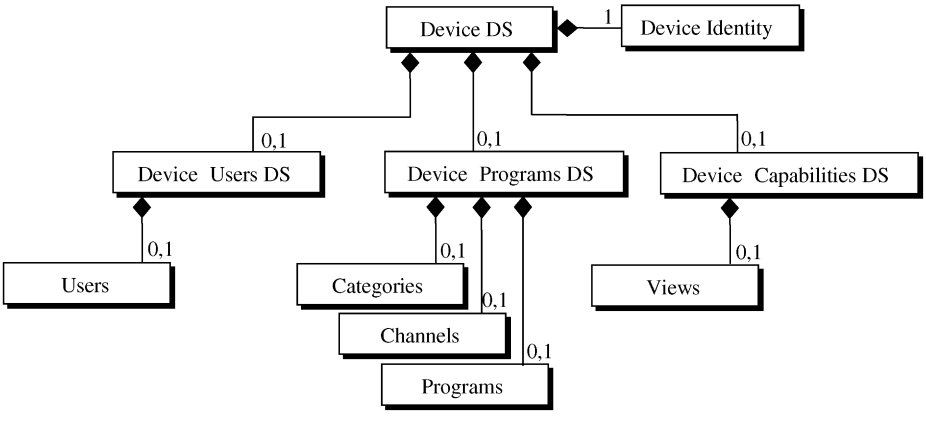

Fig. 19. Device DS.

Device serial number. This descriptor specifies the serial number of a video device or device.

4.2. Device users

Users. This descriptor lists a number of users, who have registered to the device. Each user is identified by user name and user ID which should match with the values specified in one of the user DSs discussed in Section 3.1.

4.3. Programs available to the device

Categories. This descriptor lists a number of program categories which have been registered on the device. Each category is specified by a category descriptor, which should be common to the Program DS. The major subrelationship between categories is also captured by a subcategory descriptor.

Channels. This descriptor lists a number of channels which have been registered on the device. Each channel is specified by a channel descriptor. The relationship between a major channel and its (virtual) subchannels is also captured by a subchannel descriptor.

Programs. This descriptor lists programs that are known by the device. Programs themselves may be stored in the device, or they may be future programs known by the device. The programs are grouped under corresponding categories or channels. Each group of programs are specified by a category-programs or channel-programs descriptor. The program ID contained in the descriptor should match with the number or string specified in one of the Program DSs.

4.4. Device capabilities

Views. This descriptor lists views which are supported by a video device or device. Each view is identified by a string which should match with one of the views defined in the Visualization DS defined in Section 2.4, such as Thumbnail View, Slide View, Shot View, Keyframe View, Highlight View, Event View, and Close Up View.

The set of descriptors defined here is by no means exhaustive. For example, it should be possible to add descriptors describing capabilities pertaining to other characteristics of the device, such as the bit-depth, spatial, and temporal resolution of its display.

5. Conclusions

We have proposed description schemes describing video programs, users of devices that access, store and consume these programs, and the devices themselves. These three types of description schemes do indeed share a common set of descriptors to facilitate a complete solution that simultaneously accounts for the desirability of personalization, efficient management of programs and users that are known to devices, and variations in the capabilities of multimedia access devices.

The proposed Program DS focuses on visual characteristics only. It is organized into Syntactic Structure DS, Semantic Structure DS, Meta-information DS and Visualization DS, respectively, to describe the physical structure, semantic content, meta-information and the information that is needed for fast rendering of a set of views of the program. The physical structure involves the description of the temporal organization of the sequence (segments), the spatial organization of images (regions) as well as the spatio-temporal structure of the video (regions with motion). The semantic description is built around objects and events. Finally, the physical and semantic descriptions are related by a set of links defining where or when instances of a specific semantic notions can be found. The Program DS facilitates content-based search and filtering as well as fast visual browsing and navigation.

The proposed User DS describes preferences and usage patterns of users. Personalized search, filtering and browsing is enabled in a system where Program DSs and User DSs share a common set of descriptors. Once standardized, a User DS can be transported by the user from one device to another for immediate personalization regardless of the brand name and physical location of the new device. The proposed Device DS maintains a list of programs and users known to a device as well as the computational and display resources available

to the device. When a User DS and the Device DS share a common set of descriptors, the device settings for consuming particular types of multimedia information can be personalized. Further, when a Program DS and a Device DS share a common set of descriptors, devices access to those views of the programs for which they have sufficient presentation resources.

Standardization of such description schemes will enable exchangeable formats that host rich descriptions that will in turn enable personalized universal access to multimedia information. The ongoing MPEG-7 standardization effort is aimed at defining exchangeable description formats.

As we write this paper, the MPEG-7 group on Description Schemes is in the collaboration mode and converging on a generic Audiovisual Program DS, integrating visual and audio programs. Recently, the MPEG-7 requirements group has adopted a set of requirements for describing user preferences and usage history pertaining to audiovisual information while the privacy of users is respected. It is expected that collaborative work on defining a User DS will start in the near future. The Device DS related matters, or the Multiprogram DS, has not yet been worked out by the committee and are expected to be addressed as the Program DS specification becomes stable. In parallel activities, various groups are working on descriptors, detailed definition of description schemes (e.g., Meta DS), and the specification of the Description Definition Language.

References

[1] P. Bouthemy, F. Ganansia, Video partitioning and camera motion characterization for content-based video indexing, in: IEEE International Conference on Image Processing, ICIP'96, September 1996, Vol. I, pp. 905–909.

[2] R. Chellappa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces: a survey, Proc. IEEE 83 (5) (May 1995) 705–740.

[3] G.C.H. Chuang, C.C. Jay Kuo, Wavelet descriptor of planar curves: Theory and applications. IEEE Trans. Image Process. 5 (1) (January 1996) 56–70.

[4] P. Correia, F. Pereira, The role of analysis in content-based video coding and indexing, Signal Processing 66 (2) (April 1998) 125–1142.

[5] J.D. Courtney, Automatic video indexing via object motion analysis, Pattern Recognition 30 (4) (April 1997) 607–625.

[6] Dublin Core Metadata Initiative, see http://purl.org/DC/.

[7] H. Freeman, On the coding of arbitrary geometric configurations, IRE Trans. Electron. Comput. EC-10 (June 1961) 260–268.

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

[9] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1992.

[10] I.T. Jolliffe, Principal Component Analysis, Springer, New York, 1986.

[11] MPEG Requirements Group, MPEG-7 context, objectives and technical roadmap, Doc. ISO/IEC JTC1/SC29/WG11 N2729, Seoul meeting, March 1999.

[12] S. Paek, C.-S. Li, A. Puri et al., Image description scheme, proposals P480, Document ISO/IEC JTC1/SC29/WG11, P480, Lancaster, UK, February, 1999.

[13] S. Paek, A. Puri C.-S. Li et al., Image description scheme, proposals P481, Document ISO/IEC JTC1/SC29/WG11, P481, Lancaster, UK, February, 1999.

[14] N.V. Patel, I.K. Sethi, Video shot detection and characterization for video data-base, Pattern Recognition 30 (4) (April 1997) 607–625.

[15] R.J. Qian, M.I. Sezan, K.E. Matthews, A robust real-time face tracking algorithm, in: IEEE International Conference on Image Processing, ICIP'98, Vol. MA05.03, Chicago, USA, October 1998.

[16] R.J. Qian, M.I. Sezan, P.J.L. van Beek, Description schemes for consumer video applications, Document ISO/IEC JTC1/SC29/WG11, P429, Lancaster, UK, February 1999.

[17] K. Ratakonda, M.I. Sezan, R. Crinon, Hierarchical video summarization, in: IS & T/SPIE Conference on Visual Communications and Image Processing, VCIP99, San Jose, USA, January 1999, pp. 1531–1541.

[18] Y. Rui, T.S. Huang, S. Mehrotra, Exploring video structure beyond the shots, in: Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS), Austin, Texas USA, 28 June–1 July 1998, pp. 237–240.

[19] H. Sawhney, S. Ayer, Compact representations of video through dominant and multiple motion estimation, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8) (August 1996) 814–830.

[20] P. Salembier, L. Garrido, Binary partition tree as an efficient representation for filtering, segmentation and information retrieval, in: IEEE International Conference on Image Processing, ICIP'98, Vol. TU, Chicago, USA, October 98.

[21] P. Salembier, N. O'Connor, P. Correia et al., Visual Description Scheme, proposals P185 & P186, Document ISO/IEC JTC1/SC29/WG11, P185, Lancaster, UK, February 1999.

[22] M.V. Srinivasan, S. Venkatesh, R. Hosie, Qualitative estimation of camera motion parameters from video sequences, Pattern Recognition 30 (4) (April 1997) 593–606.
[23] N. Vasconcelos, A. Lippman, Towards semantically meaningful feature spaces for the characterization of video content, in: IEEE International Conference on Image Processing ICIP'97, Paper number 825, Santa Barbara, October 1997, Vol. 1, p. 25.
[24] G. Yang, T.S. Huang, Human face detection in complex background, Pattern Recognition 27 (1) (1994) 53–63.
[25] M.M. Yeung, B. Lui, Efficient matching and clustering of video shots, in: IEEE International Conference on Image Processing, ICIP'95, Washington, USA, October 1995, pp. 338–341.
[26] C.T. Zahn, R.Z. Roskies, Fourier descriptors for plane closed curves, IEEE Trans. Comput. (1972).
[27] H.J. Zhang, A. Kankanhalli, S.W. Smoliar, Automatic partitioning of full motion video, Multimedia Systems 1 (1993) 10–28.

Philippe Salembier received a degree from the Ecole Polytechnique, Paris, France, in 1983 and a degree from the Ecole Nationale Superieure des Telecommunications, Paris, France, in 1985. He received the Ph.D. from the Swiss Federal Institute of Technology (EPFL) in 1991. He was a Postdoctoral Fellow at the Harvard Robotics Laboratory, Cambridge, MA, in 1991. From 1985 to 1989 he worked at Laboratoires d'Electronique Philips, Limeil-Brevannes, France, in the fields of digital communications and signal processing for HDTV. In 1989, he joined the Swiss Federal Institute of Technology in Lausanne, Switzerland, to work on image processing. At the end of 1991, after a stay at the Harvard Robotics Lab., he joined the Polytechnic University of Catalonia, Barcelona, Spain, where he is currently associate professor. He is lecturing on the area of digital signal and image processing. His current research interests include image and sequence analysis, compression and indexing, image modeling, segmentation problems, texture analysis, mathematical morphology and nonlinear filtering. In terms of current applications, he is particularly interested in video indexing and the MPEG-7 standardization process. He has served as an Area Editor of the Journal of Visual Communication and Image representation (Academic Press) from 1995 until 1998 and is currently an AdCom officer of the European Association for Signal Processing (EURASIP) in charge of the edition of the Newsletter. He has edited (as guest editor) two special issues of Signal Processing on "mathematical morphology" (1994) and on "video sequence analysis" (1998). He is currently co-editing a special issue of "Signal processing: Image communication" on the MPEG-7 proposals which were recently submitted for evaluation. Finally, he is Deputy-Editor of Signal Processing.

Richard Qian received the B.S. degree in computer science from Tsinghua University, Beijing, China in 1986. He received the M.S. and Ph.D. degrees in electrical engineering from University of Illinois at Urbana-Champaign in 1992 and 1996, respectively. From 1996 to 1999, he was a researcher at Sharp Labs of America in Camas, Washington, and worked on video analysis and description. He received a Sharp outstanding R&D award in 1998. Since 1999, he has been a senior researcher at lntel Architecture Labs in Hillsboro, Oregon, and leading research in the field of multimedia content modeling, analysis and description. His present research interests also include human-computer interface, mobile computing and digital infotainment.

Noel E. O'Connor received his primary degree from Dublin City University, Dublin, Ireland, in October 1992. He received his Ph.D. also from Dublin City University in October 1998. From September 1992 to November 1998 he was a Research Assistant in the Video Coding Group of Teltec Ireland. He is currently a lecturer of digital signal processing in the School of Electronic Engineering of Dublin City University. His current research interests include image and

sequence compression, region-based and object-based segmentation, and video sequence analysis for indexing applications. He is one of two Irish representatives to the ISO/IEC MPEG standards body.

Paulo Lobato Correia graduated as an Engineer and obtained an M.Sc. in electrical and computers engineering from Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Portugal, in 1989 and 1993, respectively. He is currently working towards a Ph.D. in the area of image analysis for coding and indexing. Since 1990 he is a Teaching Assistant at the electrical and computers department of IST, and since 1994 he is a researcher at the Image Communication Group of IST. His current research interests are in the area of video analysis and processing, including content-based video description and representation.

M. Ibrahim Sezan received the B.S. degrees in Electrical Engineering and Mathematics from Bogazici University, Istanbul, Turkey in 1980. He received the M.S degree in Physics from Stevens Institute of Technology, Hoboken, New Jersey, and the Ph.D. degree in Electrical Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, New York in 1982 and 1984, respectively. He is currently a Senior Manager at the Digital Video Department at Sharp Laboratories of America, Camas, Washington, where he is heading a group focusing on algorithm and system development for video resolution enhancement, visual quality optimization, and smart appliances for audiovisual information access, management, and consumption. From 1984 to 1996, he was with Eastman Kodak Company, Rochester, New York, where he headed the Video and Motion Technology Area in the Imaging Research and Advanced Development Laboratories from 1992 to 1996. Dr. Sezan contributed to a number of books on image recovery, image restoration, medical imaging, and video compression. He edited Selected Papers in Digital Image Restoration (SPIE Milestone Series, 1992), and co-edited Motion Analysis and Image Sequence Processing (Kluwer Academic Publishers, 1993). Dr. Sezan is a senior member of IEEE.

Peter van Beek was born in Amsterdam, the Netherlands, in 1967. He received the M.Sc. Eng. and Ph.D. degrees in Electrical Engineering from the Delft University of Technology, Delft, the Netherlands, in 1990 and 1995, respectively. From March 1996 to September 1998, he was a Research Associate with the Department of Electrical Engineering and Center for Electronic Imaging Systems, University of Rochester, Rochester, New York. In October 1998, he joined Sharp Laboratories of America, Camas, Washington. His research interests include digital image and video analysis, video storage and retrieval and hybrid natural/synthetic media coding.