TR-ITS-2004.09, Swiss Federal Institute of Technology (EPFL), Lausanne, March 2004.

MPEG-7 Description for Scalable Video Reconstruction

Olivier Steiger, Andrea Cavallaro and Touradj Ebrahimi

Signal Processing Institute, Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland

ABSTRACT

We present an MPEG–7 compliant description of video sequences for scalable transmission and reconstruction. The proposed object-based method permits efficient and flexible video coding while keeping the benefits of textual descriptions in database applications. Video objects are described in terms of shape, color, texture and motion. These features are extracted automatically and are sufficient in a wide range of applications like smart video cameras and Universal Multimedia Access (UMA). Scalable sequence reconstruction is obtained by providing at least one quantitative as well as a qualitative descriptor for each feature; each video object is described by a subset of the proposed descriptors. The scalability and the compactness of the description are demonstrated aid of an outdoor and a sport sequence.

Keywords: MPEG–7, XML, content-based coding, object-based coding, scalable video reconstruction.

1. INTRODUCTION

For the past few years, new algorithms and standards such as H.26x and MPEG–1, 2 and 4 have been developed for the compression of digital video. Along with them, the need for efficient multimedia database management has brought up new search, indexing and content-based retrieval techniques. To fulfill these functions, extensive research has been carried out into finding adequate *metadata*, i. e. information about information, for multimedia content. The traditional approach of keyword annotation has the drawback that it requires huge manual efforts and cannot characterize the rich visual content efficiently; therefore, more effective representation schemes had to be developed. Features like content summaries, object names or copyright information are usually represented in text form and often have to be edited manually. Shape, texture or color on the other hand can be extracted automatically and described based on histograms, filter bank coefficients, polygon vertices or other representations.

To insure the interoperability and continuation of different data sets, the need for a uniform description framework for multimedia data arose. To meet this challenge, ISO's Moving Pictures Experts Group (MPEG) has undertaken the standardization activity of a "Multimedia Content Description Interface" called MPEG–7,[1] which is based on the Extensible Markup Language XML.[2] The main elements of the MPEG–7 standard are Descriptors (D) describing features, attributes or groups of attributes of multimedia content, and Description Schemes (DS) that specify the structure and semantics of their components, which may be description schemes, descriptors or datatypes. In addition, MPEG–7 provides a Description Definition Language (DDL)[3] to define the syntax of description tools and to permit the creation of new description schemes and, possibly, descriptors, as well as the modification and extension of existing ones.

Since the release of the first MPEG–7 working drafts in 1999, significant efforts have been deployed to study potential applications of the new standard. In their paper on object-based multimedia content description,[4] the authors propose diverse image, video and multimedia description schemes for image classification (*The Visual Apprentice*), object-based video indexing and searching (*AMOS–search*) and multimedia meta-search engines (*MetaSEEK*). The matching of individual user profiles with MPEG–7 content descriptions for digest video delivery has been studied by Echigo *et al.*,[5] and a low-level description scheme based on color, texture, shape

E-mail addresses of the authors:
{olivier.steiger, andrea.cavallaro, touradj.ebrahimi}@epfl.ch

and motion for image retrieval has also been proposed.[6] Various other papers[7–10] also introduce description schemes for video indexing and content-based retrieval, some of which stem from the standardization process itself. But while the use of video descriptors in databases has been widely studied, they have not been used for sequence reconstruction so far. We believe however that the human readable MPEG–7 descriptors and the corresponding compact binary format BiM can also favorably be used for scalable and storage-efficient, self-describing video coding.

In Section 2, we propose an original description method of video sequences for reconstruction. Its components are a MPEG–7 descriptor set for the shape, color, texture and motion of video objects, and description schemes for the structuring of the descriptors. We then state the main attributes of our method along with some applications. The experiments in Section 3 show the scalable reconstruction of a highway surveillance and a soccer sequence and opposes the size of the MPEG–7 description to MPEG–4. Discussions and conclusions are given in Section 4.

2. MPEG-7 DESCRIPTION OF VIDEO SEQUENCES

Present video coding systems[11] operate along two main lines: first generation coding (e.g., MPEG–1 and 2, H.261/263) uses *block-based* transform coding to achieve compression and does not take into account the semantic construct of the video image. Second generation coding standards like MPEG–4 on the other hand process video objects individually, thereby adding compression efficiency and new editing possibilities to the former approach. But even though MPEG–4 is *object-based*, the objects are still transform coded. To avoid decoding, it is desirable to join a textual *content description* to the video for multimedia database browsing and retrieval. The description can be organized so as to contain visual information in addition to the metadata needed for content management. If this information is properly reconstructed, there is no need for further video codes like MPEG–4. We next propose a MPEG–7 description for scalable video reconstruction that can easily be inserted within existing descriptions. In our method, a sequence comprises the textual description of video objects along with a background shot (Figure 1). Object features are specified using a subset of the MPEG–7 descriptors we propose in Section 2.1.

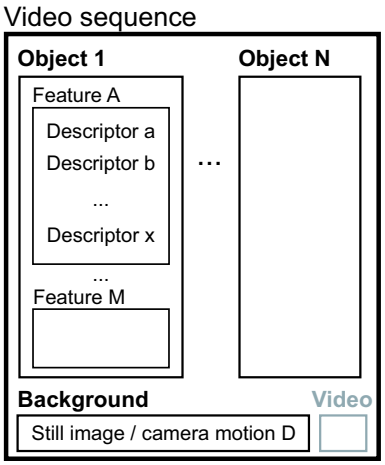


Figure 1. In the proposed description, each feature of video objects is described by at least one MPEG–7 descriptor. A video sequence comprises the description of all the objects along with a background shot.

2.1. Descriptor set for video objects

The visual features of video objects we describe are shape, color, texture and motion. These features can be extracted from video signals automatically using common machine vision techniques.[12] Also, they are directly related to physical object properties such as form, color, surface and trajectory. In order to support *scalable*

descriptions, that is, descriptions permitting a coarse-to-fine reconstruction of the original sequence, we provide at least one qualitative and one quantitative descriptor for each feature (Table 1).

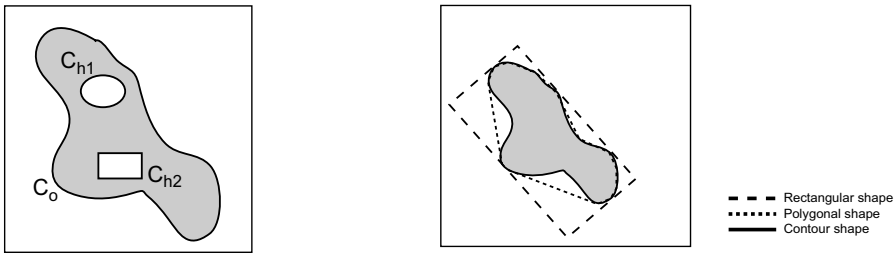
Feature	descriptor	Purpose
Shape	`Region locator`	Box or polygon shape
	`Contour shape`	Closed contour shape
Color	`Dominant color`	Set of up to 8 dominant colors
	`Color layout`	Spatial distribution of colors
Texture	`Texture browsing`	Perceptual texture description
	`Homogeneous texture`	Structural texture description
Motion	`Motion activity`	Perceptual motion descriptor
	`Motion trajectory`	Single-point motion
	`Parametric motion`	Describe moving regions
	`Camera motion`	Specifies 3–D camera motion parameters

Table 1: MPEG-7 descriptor set for visual object features.

The syntax of these descriptors is specified by *MPEG–7 Part 3: Visual*[13]; an overview of their structure is given in Appendix B. Non-normative extraction examples as well as conditions of usage of the descriptors are given in the MPEG–7 *Visual Experimentation Model*.[14]

2.1.1. Shape

Shape is the the most important visual feature for object recognition and classification both by humans and machines. But while a coarse shape description often suffices for classification, the original shape has to be matched closely for unambiguous object recognition. `Region locator`, which specifies regions within images with a brief and scalable representation of a box or polygon, approximates the size, orientation and geometry of objects as closely as desired. `Contour shape` describes the closed contour of a 2–D object in the Curvature Scale Space (CSS).[15] None of them explicitly supports the description of holes. These should therefore be encoded as inner contours, as shown in Figure 2(a): each contour C surrounded with an outer contour C_o is considered a hole C_h of the object delimited by C_o. Similarly, objects made of multiple distinct regions can be encoded using multiple contours. The combination of `Region locator` and `Contour shape` permits scalable representations ranging from rectangles over x-sided polygons down to the original form (Figure 2(b)).



(a) (b)

Figure 2. Description of shapes. (a) Holes are encoded as inner contours: the contours C_{h1} and C_{h2} define holes of the object delimited by the contour C_o. (b) Scalable shape representation: shapes can be represented as rectangles, x-sided polygons or closed contours.

2.1.2. Color

Color provides valuable information about the nature and illumination of objects. To distinguish different identically shaped objects one from another, it is in general sufficient to know some of the dominant colors of an object. On the other hand, the spatial distribution of color must be given to represent illumination shades such as shadows. `Dominant color` specifies a set of up to 8 dominant colors in an arbitrarily-shaped region. `Color layout` gives the spatial distribution of colors with the DCT coefficients of an 8x8 array of local representative colors. Scalability is achieved by augmenting the number of specified dominant colors and by adding structural information using the latter descriptor.

2.1.3. Texture

The material an object is made of is visually translated into a texture. The material characteristics are specified by perceptual texture descriptions. The actual object material is given by its texture image. `Texture browsing` characterizes textures perceptually in terms of regularity, coarseness and directionality. At the time of reconstruction, predefined textures with similar perceptual properties are displayed. `Homogeneous texture` specifies a region texture using its energy and energy deviation in a set of frequency channels. This permits a reasonable approximation of the original texture image. Scalability comes from the description ranging from a perceptual characterization of the texture down to its actual representation.

2.1.4. Motion

Motion represents the trajectory of objects. The 3D trajectory of a real objects can be calculated from its motion and shape when perspective information and object geometry are known. The action of a video scene is given by its perceptual motion description. To move rigid objects, it suffices to know the trajectory of one representative point. Deforming objects on the other hand ask for more complex motion models. `Motion activity` captures the notion of "intensity of action" or "pace of action" in a video scene. At the time of reconstruction, this is typically displayed using texts or arrows whose length corresponds to the action intensity and orientation to the motion direction. `Motion trajectory` is the spatio-temporal localization of one of the representative points (e.g., gravity center) of a moving region. This is displayed by moving the object shape, or some symbol when the shape is unknown, along the trajectory. `Parametric motion` characterizes the evolution of arbitrarily shaped regions over time in terms of a 2–D geometric transform (translation, rotation/scaling, affine transformation, perspective models or quadratic models). `Camera Motion` should be used to reconstruct moving background from a panoramic background shot (Section 2.2.2). As shown in Figure 3, these descriptors permit to scale from perceptual motion descriptions over rigid object movement to shape deformations.

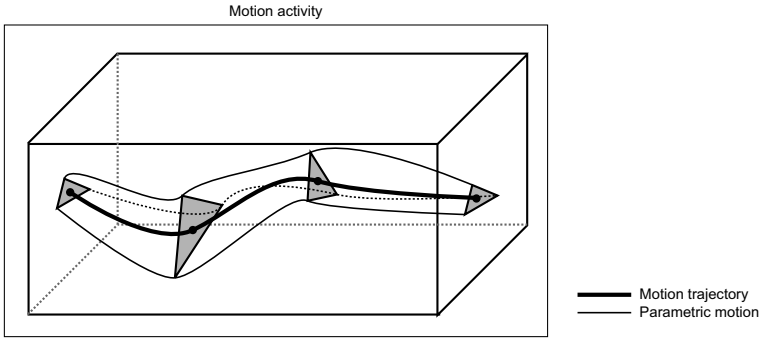


Figure 3. Scalable motion representation. `Motion trajectory` gives the spatio-temporal localization of a representative object point; `parametric motion` characterizes the evolution of arbitrarily-shaped objects over time.

2.1.5. Disregarded descriptors

MPEG–7 provides additional descriptors for most of the above features. For reasons we explain next, we did not keep them within our method. `Region shape` has been disregarded despite its support for holes and disjoint regions. In fact, the Angular Radial Transform (ART) used to describe pixel distributions does not permit shape reconstruction.[14] As our method applies to 2D video, `Shape 3D` has been discarded as well. `Scalable color` uses a Hue-Saturation Value (HSV) histogram to specify colors in an arbitrarily-shaped region. It has been discarded for the histogram can not be translated into visually interesting information at reconstruction*. `Color structure` characterizes the relative frequency of structuring elements that contain an image sample with a particular color. This is useful for image-to-image matching but does not permit reconstruction. `GoF/GoP color` specifies a structure for representing the color features of a collection of video frames by means of the `Scalable color` descriptor. In our description, this information is given by the *intra-frame* color value (Section 2.2.1). `Edge histogram` specifies the spatial distribution of five edge types in local image regions. This representation does not lead to satisfying visual reconstruction as it does not provide any information about pixel intensity distributions.

2.2. Organization of the description

The organization of our description matches the semantic construct of video closely (Figure 1). All descriptors of a video object are grouped together in an entity called *object*. All objects belonging to a scene in turn form a *video sequence*. An example of an outdoor video sequence description is given in Appendix A.

2.2.1. Grouping descriptors to form objects

The `Object DS`[16] describes an object, which is a semantic entity having spatial and temporal extent. It can therefore be used to "wrap up" visual descriptors. To differentiate objects, each one of them must get a unique identifier. In principle, any alphanumerical character chain can be used. However, it often proves useful to store information about past object interactions (e.g., merging with other objects) textually. This eliminates the need for sequence reconstruction to gain knowledge about object history. Therefore, we include this information in the identifier whenever it is available.

Inside an object, there must be at least one visual descriptor. However, more than one descriptor can be used simultaneously for each feature. This adds *scalability* to the transmission and decoding process, as compact descriptors (*base layer*) can be sent first, and bigger ones (*enhancement layer*) later. The decoder exploits the same property for partial reconstruction. To help setting up processing priorities, a *reliability* measure is associated with each parameter. More reliable parameters are then sent and/or decoded first. Finally, as video is a process taking place in time, shape, color or texture may have to be updated within the video sequence. To reflect this in the description, we use *intra-frames* which specify the new parameters of the feature(s) to be updated. The corresponding new descriptor is inserted right after the last valid motion parameter.

2.2.2. Grouping objects to form a video sequence

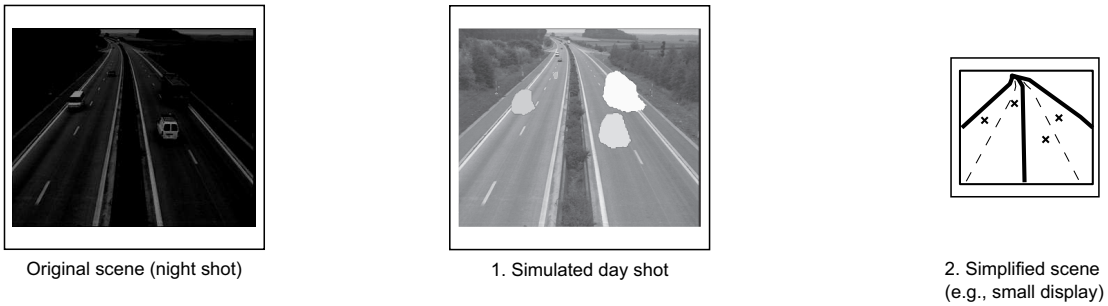
The top-level entity of our description is the video sequence. Each sequence results in one MPEG–7 description which is wrapped up in the `Semantic DS`.[16] When a sequence needs to be linked to a bigger entity, such as a movie, MPEG–7 linking and localization tools are used.

The background of the video sequence must also be encoded. Static backgrounds are easily stored as still images, possibly using some compression algorithm (e.g., JPEG, JPEG–2000, ...). Moving backgrounds can be formed from a panoramic background shot, which is displayed according to the `camera motion` descriptor. Finally, the background is linked to the description by the `MediaLocator` datatype.[16]

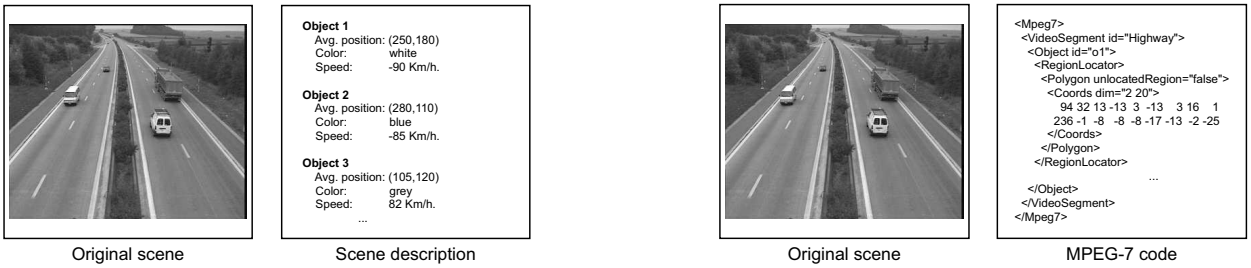
2.3. Description attributes

The proposed description shows a number of attractive attributes, namely *video manipulation*, *scene description* and *compactness* (Figure 4).

*We believe that showing all colors of an object without any information about their spatial distribution would overload the representation.



(a)



(b) (c)

Figure 4. Description attributes. (a) Video Manipulation. 1: *Scene Visualization* aims at making video shots more understandable, for instance by background substitution; 2: *Video Simplification* eliminates semantically irrelevant details. (b) *Scene Description*: video object parameters are available textually. (c) *Compactness*: the video content is summarized in a compact MPEG–7 description.

2.3.1. Video Manipulation

Video manipulation refers to the modification of the original visual content. Video objects can be modified individually in any object-based code. However, the modification of individual parameters like shape, color, texture or trajectory requires at least partial image reconstruction when pixel-based codes such as MPEG–4 are used. With MPEG–7, it is sufficient to edit the textual description.

Compositing, where different video objects and backgrounds are blended to form a new scene, is particularly easy to perform on object-based video because objects are available separately.

Scene Visualization is a specific kind of video manipulation aiming at representing video so as to make it more understandable for a human observer. The simplest example is background substitution. Sometimes, video is hard to interpret because of the background being either not visible (e.g., night shot), or hard to discern from foreground objects (complex background). A more appropriate image (daylight shot, simplified background) can then be used instead of the original background. Another visualization possibility is to change feature values by editing the description parameters. Typically, objects can be given *pseudo-colors*.

Video Simplification takes advantage of the scalability of the description to eliminate semantically irrelevant details. This is done by leaving out certain features (e.g., display position but not shape of objects, leave out color, ...), by ignoring certain descriptors for a given feature (box instead of contour shape) and by using a subset of parameters *inside* a descriptor (1 instead of 8 dominant colors). As we explain in Section 2.4.2, this helps to adapt content to receivers and substantially reduces the description size.

2.3.2. Scene Description

A possible way to describe video scenes is to give the feature parameters of video objects. In XML-based descriptions, these parameters are available textually and can often be translated into a physical description of the real object using simple calculations. Object position, together with some a priory knowledge on perspective, provide information about the localization and speed of moving objects. By further taking into account the shape, object deformations and interactions (occlusion, merging, etc.) are given as well. Texture and color descriptors finally permit to gain insight on the appearance of objects.

2.3.3. Compactness

Compactness has a somewhat different meaning here than with traditional coding techniques. Coding schemes like MPEG–1, 2 and 4 achieve data compression by removing redundancies from the encoded material while minimizing perceptual quality losses. The MPEG–7 description on the other hand *summarizes* the scene content, thus ignoring semantically unimportant elements. This does not in general permit an exact reconstruction of the original scene but does, as we show in Section 3, vehicle enough visual information for its interpretation. Another substantial size reduction can be obtained by using the MPEG–7 Binary Format (BiM) instead of XML. This however involves the (straightforward) transcoding between both formats.

2.4. Applications

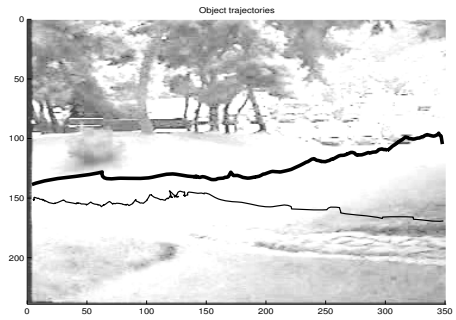
Thanks to its attributes (Section 2.3), our description notably supports certain applications which are difficult to implement using non content-based coding approaches. We next present two such applications.

2.4.1. Smart video cameras

As opposed to conventional video cameras which "blindly" transform images into an electrical signal, *smart cameras* analyze the filmed material to gain insight into its semantic construct. This knowledge can be employed to highlight valuable visual information at reconstruction and facilitates automatic event detection, efficient coding (e.g., object-based MPEG–4) and video editing. A smart *MPEG–7 camera*[17, 18] is obtained by combining the proposed description with automatic video object detection and tracking. This device describes filmed scenes without any user intervention. An example of such a description is given in Appendix A. Figure 5 shows an example of *scene visualization*. The path of two walking people has been reconstructed from their MPEG–7 description. This result is obtained by using the *scene description* feature (Section 2.3.2). Scene description is exploited to superimpose the gravity center trajectories of both people on the background. This kind of

reconstruction is particularly useful for video surveillance, as the focus of an observer is immediately caught by essential image parameters such as object trajectory. Surveillance tasks can also be automated by comparing the description parameters with some reference ones representing normal conditions in the monitored scene.

(a)



(b)

Figure 5. Reconstruction of object trajectories from their MPEG–7 camera description. (a) Original sequence (frames 10, 110 and 160). (b) Path of the two objects reconstructed from their motion trajectory (the original background has been modified to improve readability).

2.4.2. Universal Multimedia Access

Universal Multimedia Access (UMA) refers to the delivery of rich multimedia content to diverse client devices over various channels.[19] The "info-pyramid" approach to this end is to store distinct *variations* of the content for diverse client devices. The most adequate variation is then sent on demand. However, it is more storage efficient and flexible to only store the content in a single modality, which is then adapted to the device "on-the-fly". This can easily be achieved with the proposed description, thanks to the *Video Simplification* feature, and because of its scalability. In the UMA setup shown in Figure 6, the MPEG–7 description of each video is stored in a *content database*. On content request, a *user profile database* specifies *which* features, descriptors and parameters must be streamed to the client. For instance, a simple pocket device might only display crosses rather than object shapes.

3. EXPERIMENTS

The MPEG–7 description for scalable video reconstruction proposed in Section 2 is here tested. An outdoor and a sport sequence are described using the proposed method and reconstructed at five different scales. In the description, shapes are represented as rectangles and 20-sided polygons. Both are specified by the `region locator` descriptor. The `dominant color` of an object is given the maximum value in its color histogram. `Motion trajectory` describes the path of object gravity centers. For reconstruction, five scales were chosen

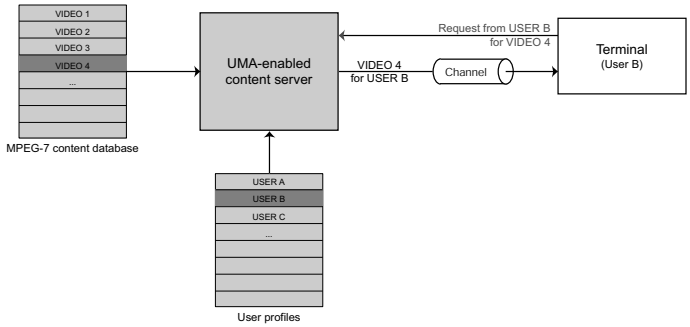


Figure 6. Usage of MPEG–7 description to provide Universal Multimedia Access: only features, descriptors and parameters that can be displayed by the user device are streamed.

so as to approximate the original video increasingly closely. The tests have been performed using an outdoor and a sport sequence from the MPEG–7 Video Content Set.[20] The outdoor sequence[†] (Section 3.1) stems from a highway surveillance video. Video objects were extracted and tracked automatically using an MPEG–7 camera.[17] The sport sequence[‡] (Section 3.2) shows an extract of a soccer game. Due to its high complexity, this sequence was segmented by hand.

Scalable sequence reconstruction is shown in Figures 7 and 8. Part (a) of the Figures gives three frames of the original video sequences. Parts (b)–(f) show the *position*, *position & color*, *rectangular shape*, *20-sided polygon shape* and *polygon shape & color* of the video objects, as reconstructed from the corresponding descriptions. Additionally, the description size at each scale is given in Figures 9(a) and 9(b). Here, *XML* refers to the textual MPEG–7 format, while *BiM* is the binary format. XML to BiM transcoding is performed using the MPEG–7 eXperimentation Model (XM) software Version 5.5. Backgrounds are encoded as still images using the *JJ2000* Java implementation of JPEG-2000; the target bit rate is set to 1.25 bits per pixel. We also compare the MPEG–7 description size to "equivalent", that is object-based, MPEG–4. These streams are generated using the MoMuSys-FDIS Version 1.0 encoder comprised in the MPEG–4 reference software package; all encoding parameters are set to the values given by the enclosed *foreman* demo configuration file.

3.1. Highway surveillance sequence

The *highway surveillance* sequence shows cars driving down a highway filmed by a static monocular color camera. For an except of the description of this sequence, we refer the reader to Appendix A. On Figure 9(a), the *position* description is particularly compact. The size of both MPEG–7 BiM and the JPEG–2000 background is 27 KBytes, which is about 1/3 of the MPEG–4 size (77 KBytes). This description nevertheless carries valuable information about the scene (Figure 7(b)). Thanks to the *scene description* attribute, speed and trajectory of real objects are easily deduced from the description when a calibrated camera is used. Automatic speeding detection is then achieved by comparing the calculated object speed to some preset threshold. Incidents such as objects leaving their path are detected as well by monitoring the trajectory direction. *Color* (Figure 7(c)) is added to the description at virtually no extra cost (2 KBytes). This additional information facilitates the identification of individual objects. *Rectangular shape* (Figure 7(d)) provides further indications about the nature of objects, for instance their size. The extra cost for this is 11 KBytes. The finer description given by *polygon shape* (Figure 7(e)) is used to gain detailed knowledge about the filmed scene. In surveillance, this permits (automatic) collision detection. The size increase of 41 KBytes however is more substantial. This is due to the higher number of parameters (20 X-Y coordinates for each polygon) of the shape description. The size of the *polygon shape & color* description (83 KBytes) finally is almost similar to MPEG–4. But while the MPEG–7 reconstruction in Figure 7(f) is not as close visually to the original video as MPEG–4, it shows all the attributes discussed in Section 2.3.

[†]Item V29, *speedwa2.mpg*, frames 11670-11970.
[‡]Item V4, *news2.mpg*, frames 14834-14874.

3.2. Soccer sequence

The *soccer* sequence shows a goal scene. This complex sequence involves partial occlusions, deforming objects, and a camera zoom. The *position* reconstruction shown in Figure 8(b) locates the players relatively to the ball (in red), the referee (in black) and the goal. However, one must know the pseudo color given to each player in order to identify them. The size of this description with background amounts to 27 KBytes (Figure 9(b)). Comparatively, the MPEG–4 size is 70 KBytes. *Color*, when properly extracted, may help to determine each player's team (Figure 8(c)). The extra cost for color is 4 KBytes. *Rectangular shapes* (Figure 8(d)) are not of much use in this specific case, as large rectangles often occlude important parts of the game. Moreover, the object size indication given by the rectangles does not contribute to the understanding of this sequence. Rectangular shapes need an extra space of 13 KBytes. Unlike rectangles, many-sided *polygons* reflect object deformations (Figure 8(e)). This for instance indicates each player's current state. The cost is an extra size of 42 KBytes. The *polygon shape & color* description in Figure 8(f) finally is about 20% bigger than MPEG–4. This is due to the presence of many objects in this very short sequence. In such cases, the MPEG–7 overhead predominates the visual information itself. This is usually not the case with longer sequences, like the highway surveillance video.

4. DISCUSSION AND CONCLUSIONS

We have presented a new MPEG–7 description for video reconstruction. Particular attention is given to the scalability of the description. This is achieved by providing a visual descriptor set together with an organization permitting reconstruction while keeping the advantages of MPEG–7 in databases. Three main description attributes, namely video manipulation, scene description and compactness, have been identified. Also, the usage of our description for smart cameras and Universal Multimedia Access have been investigated. Scalability and compactness have then been demonstrated experimentally using an outdoor and a sport sequence.

The experiments confirm the scalability of the proposed method. By leaving out individual features, descriptors or parameters, description accuracy and size can be easily adapted to the needs and resources of specific applications. Also, the *scene description* attribute proves very valuable for automatic event detection, as the description parameters directly relate to physical object properties. An issue for further investigation is the inclusion of the proposed descriptions within databases. For instance, the visual descriptors given in Section 2.1 can be used for sketch- or image-based query. Further evaluation of this topic is expected to be conducted in the future.

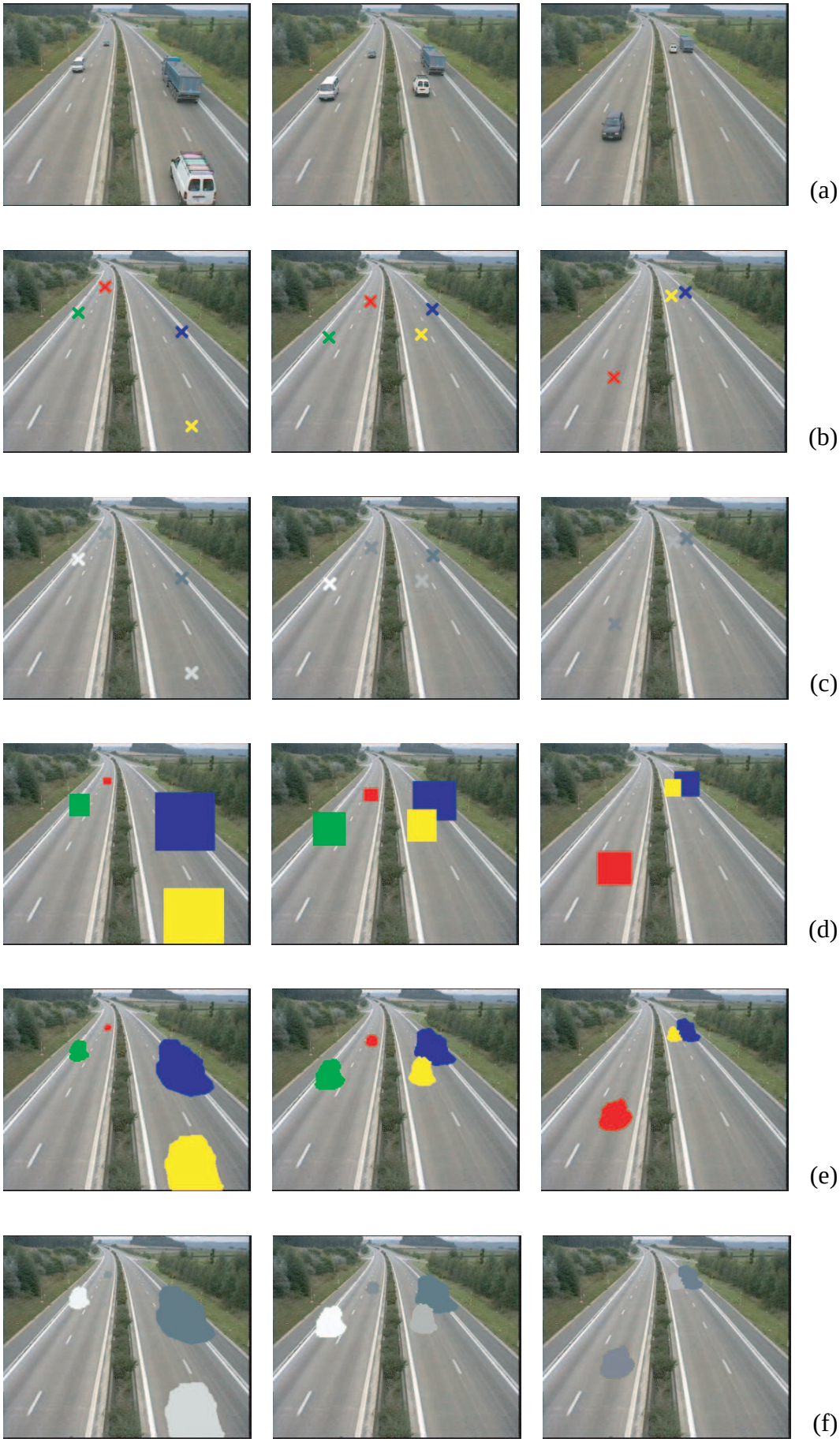


Figure 7. Example of scalable reconstruction of the *highway surveillance* sequence (frames 59, 75 and 104). (a) Original sequence; (b) object position; (c) position & color; (d) rectangular shape; (e) polygon shape (20 sides); (f) polygon shape & color.

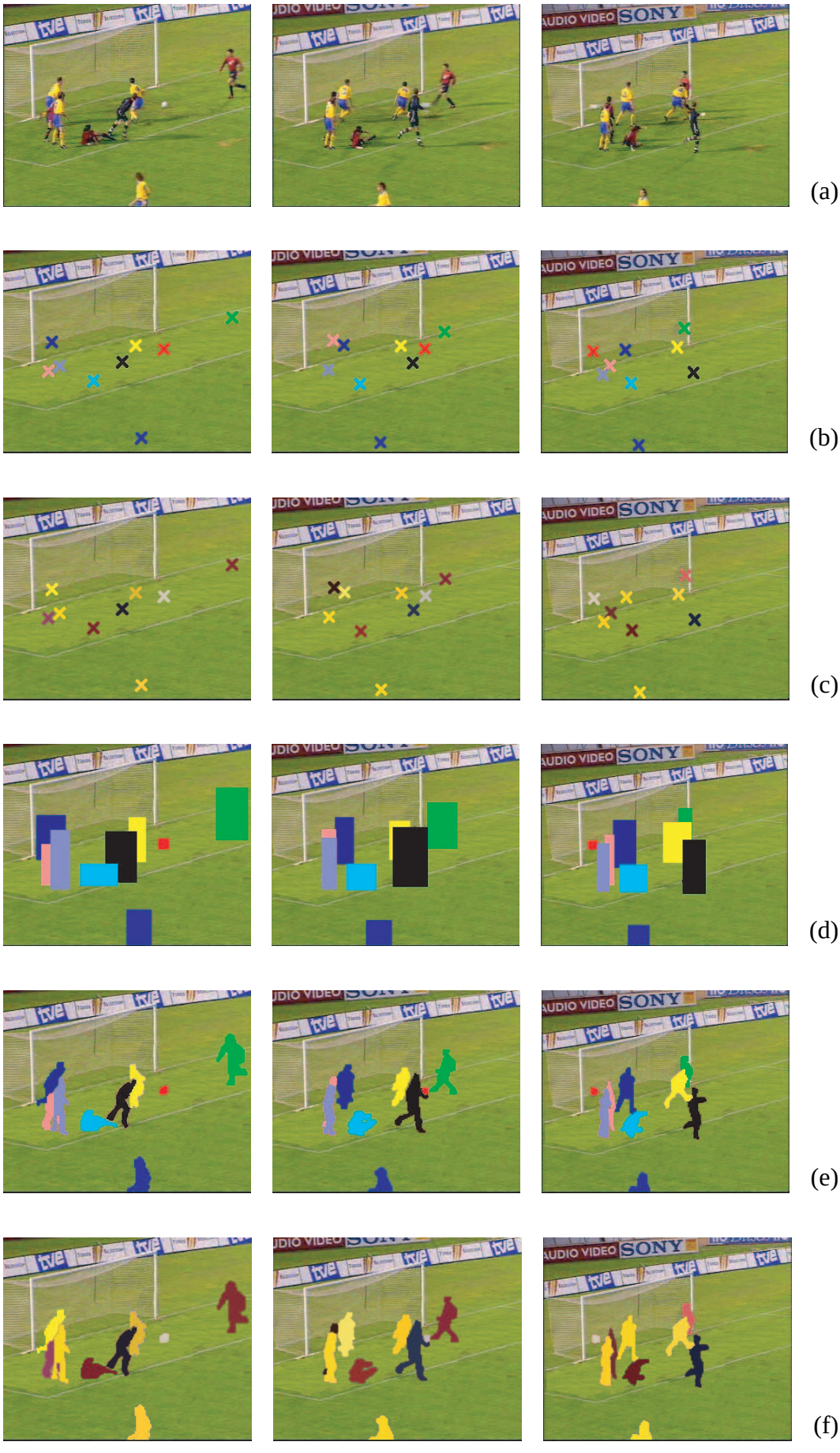
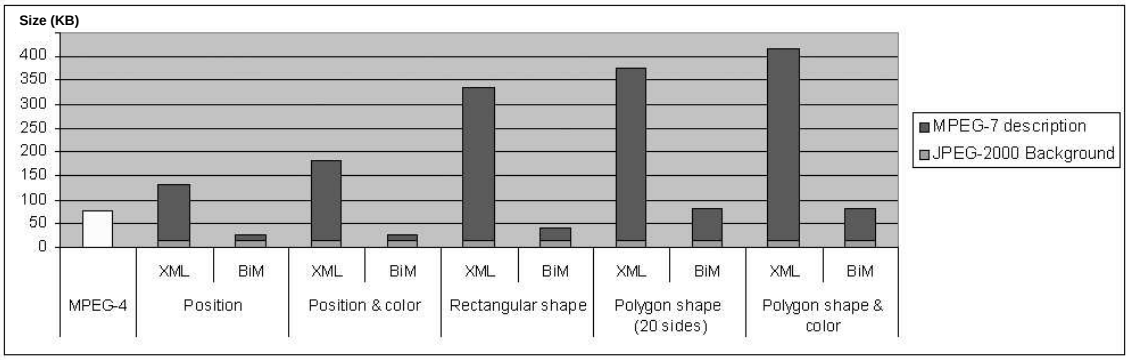
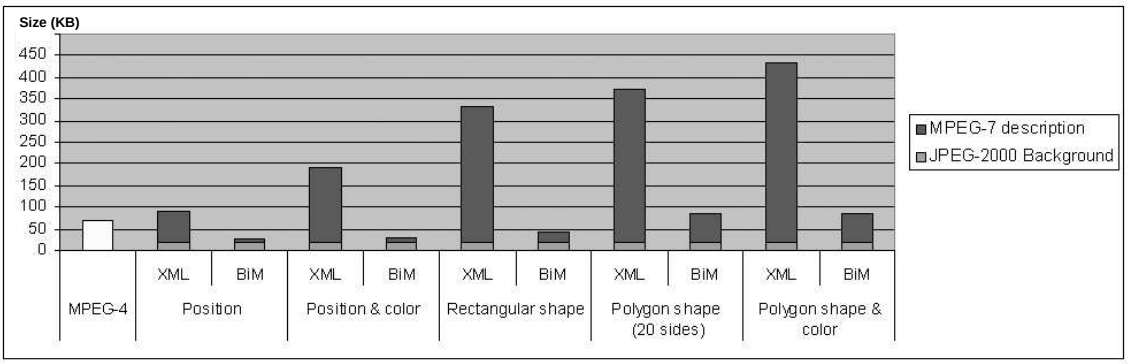


Figure 8. Example of scalable reconstruction of the *soccer* sequence (frames 20, 30 and 36). (a) Original sequence; (b) object position; (c) position & color; (d) rectangular shape; (e) polygon shape (20 sides); (f) polygon shape & color.



(a)



(b)

Figure 9. MPEG–7 file size at different scales versus "equivalent" MPEG–4. (a) Highway sequence; (b) soccer sequence.

APPENDIX A. DESCRIPTION EXAMPLE

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!-- ############################################ -->
<!-- MPEG-7 description of the "Highway" sequence  -->
<!-- ############################################ -->
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
       xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 .\Mpeg7-2001.xsd">
  <Description xsi:type=SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Description of the highway sequence </Name>
      </Label>
      <SemanticBase xsi:type="ObjectType">
        <Label>
          <Name> Full Sequence </Name>
        </Label>
        <!-- BACKGROUND -->
        <Object id="0">
          <Label>
            <Name> Static Background </Name>
          </Label>
          <MediaOccurrence>
            <MediaUri>highway_background.jpg</MediaUri>
          </MediaOccurrence>
        </Object>
        <!-- FOREGROUND OBJECTS -->
        <Object id="1">
          <Label>
            <Name> Description of object 1 </Name>
          </Label>
          <MediaOccurrence>
            <VisualDescriptor xsi:type="RegionLocatorType">
              <Polygon unlocatedRegion="false">
                <Coords dim="2 6">
                  133   1   2   3   0   1
                   77   0   3   0   1  -1
                </Coords>
              </Polygon>
            </VisualDescriptor>
```

```
            <VisualDescriptor xsi:type="DominantColorType">
              <ColorSpace colorReferenceFlag="false" type="LinearMatrix">
                <ColorTransMat>
                   76   150   29
                  -37   -74  111
                  157  -131  -25
                </ColorTransMat>
              </ColorSpace>
              <SpatialCoherency> 0 </SpatialCoherency>
              <Value>
                <Percentage> 100 </Percentage>
                <Index> 14 15 16 </Index>
              </Value>
            </VisualDescriptor>
            <VisualDescriptor xsi:type="MotionTrajectoryType" cameraFollows="false">
              <CoordDef units="pictureWidthAndHeight" />
              <Params>
                <KeyTimePoint>
                  <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F">  116  </MediaRelIncrTimePoint>
                  <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F">  117  </MediaRelIncrTimePoint>
                  <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F">  118  </MediaRelIncrTimePoint>
                  <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F">  119  </MediaRelIncrTimePoint>
                </KeyTimePoint>
                <InterpolationFunctions> <!-- X-values -->
                  <KeyValue type="notDetermined"> 141 </KeyValue>
                  <KeyValue type="notDetermined"> 141 </KeyValue>
                  <KeyValue type="notDetermined"> 140 </KeyValue>
                  <KeyValue type="notDetermined"> 139 </KeyValue>
                </InterpolationFunctions>
                <InterpolationFunctions> <!-- Y-values -->
                  <KeyValue type="notDetermined"> 73 </KeyValue>
                  <KeyValue type="notDetermined"> 75 </KeyValue>
                  <KeyValue type="notDetermined"> 77 </KeyValue>
                  <KeyValue type="notDetermined"> 80 </KeyValue>
                </InterpolationFunctions>
              </Params>
            </VisualDescriptor>
            <!-- UPDATE SHAPE/COLOR/TEXTURE -->
            <VisualDescriptor xsi:type="...">
              ...
            </VisualDescriptor>
            <!-- Update more features -->
            ...
            <!-- CONTINUE MOTION TRAJECTORY -->
            <VisualDescriptor xsi:type="MotionTrajectoryType" cameraFollows="false">
              ...
            </VisualDescriptor>
              ...
          </MediaOccurrence>
        </Object>
        <!-- MORE OBJECTS -->
        <Object id="2">
          ...
        </Object>
        ...
        <Object id="N">
          ...
        </Object>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>
```

APPENDIX B. STRUCTURE OF THE MPEG–7 VISUAL DESCRIPTORS

(Optional parameters are typeset in *italic*).

RegionLocator

```
{  BoxPoly;                              % Select box/polygon representation
   Coords[numOfVertices];                % Coordinates of the box/polygon vertices
   UnlocatedRegion; };                   % Specify if inner/outer shape region is located
```

ContourShape

```
{  GlobalCurvatureVector;                % Eccentricity/circularity of original contour
   PrototypeCurvatureVector;             % Eccentricity/circularity of prototype contour
   HighestPeakY;                         % Highest CSS peak
   Peak[numOfPeaks-1]; };                % Remaining peaks
```

DominantColor

```
{  ColorSpace;
   ColorQuantization;                    % Uniform quantization of the color space
   ColorValueIndex[numOfDomCols];        % Index of each dominant color
      Percentage;                        % Percentage of pixels with associated dominant color
      Variance;                          % Variance of dominant color
   SpatialCoherency; };                  % Dominant color pixels coherency
```

ColorLayout

```
{  DCCoeff[3];                           % First quantized DCT coefficient of Y/Cb/Cr component
   ACCoeff[3][numOfCoeffs-1]; };         % Successive DCT coefficients of each color component
```

TextureBrowsing

```
{  Regularity;                           % Periodicity of underlying basic texture elements
   Direction;                            % Dominant direction of the texture
   Scale; };                             % Coarseness of the texture
```

HomogeneousTexture

```
{  Average;                              % Average of image pixel intensity
   StandardDeviation;                    % Standard deviation of image pixel intensity
   Energy[30];                           % Energies from each frequency channels
      EnergyDeviation[30]; };            % Corresponding energy deviations
```

MotionActivity

```
{  Intensity;                            % Motion intensity
   DominantDirection;                    % Motion direction
   SpatialDistributionParameters[3];     % Number and size of active objects in the scene
   SpaLocNumber;                         % Subdivide frame into SpaLocNumber rectangles
      SpatialLocParameters[SpaLocNumber];% Relative activity of each rectangular region
   TemporalParameters[5]; };             % Activity histogram
```

MotionTrajectory

```
{  CameraFollows;                        % Specifies whether camera follows object
   TrajParams[numOfKeyPoints]; };        % Specifies key-points and interpolation of moving point
```

ParametricMotion

```
{  MotionModel;                          % Translational/rotation/affine/perspective/quadratic
   CoordRef                              % 2D coordinate system
   Duration                              % Length of described motion time interval
   Parameters[numKeyPoints]; };          % Motion parameters for each key point
```

CameraMotion

```
{  MediaTime;                            % Time interval to which camera motion applies
   Horizontal/verticalPosition           % Coordinates of the focus of expansion / contraction
   FractionalPresence                    % Amount of motion time / total video time
   AmountOfMotion; };                    % Amount of track, boom, dolly, pan, tilt, roll and zoom
```

REFERENCES

1. J. M. Martinez, "Overview of the MPEG–7 Standard," Tech. Rep. N4509, ISO/IEC JTC1/SC29/WG11, Pattaya, TH, December 2001.
2. E. T. Ray, *Learning XML*, O'Reilly, Sebastopol, CA, jan. 2001.
3. J. Hunter, "Text of 15938-2/FDIS Information Technology – Multimedia Content Description Interface – Part 2 Description Definition Language," Tech. Rep. N4288, ISO/IEC JTC1/SC29/WG11, Sydney, AU, July 2001.
4. A. B. Benitez, S. Paek, S.-F. Chang, A. Puri, Q. Huang, J. R. Smith, C.-S. Li, L. D. Bergman and C. N. Judice, "Object-Based Multimedia Content Description Schemes and Applications for MPEG–7," *Signal Processing: Image Communications*, Vol. **16**, 2000, pp. 235–269.
5. T. Echigo, K. Masumitsu, M. Teraguchi, M. Etoh and S.-I. Sekiguchi, "Personalized Delivery of Digest Video Managed on MPEG–7," *Proc. Int. Conf. on Information Technology, Coding and Computing*, 2001, pp. 216–219.
6. J.-R. Ohm, F. Bunjamin, W. Liebsch, B. Makai, K. Mller, A. Smolic and D.Zier, "A Set of Visual Feature descriptors and their Combination in a Low-level Description Scheme," *Signal Processing: Image Communications*, Vol. **16**, 2000, pp. 157–179.
7. A. M. Ferman, A. M. Tekalp and R. Mehrotra, "Effective Content Representation for Video," *Proc. Int. Conf. on Image Processing*, Vol. **3**, Chicago, IL, Oct. 1998, pp. 521–524.
8. P. Salembier, R. Qian, N. O'Connor, P. Correia, I. Sezan and P. van Beek, "Description Schemes for Video Programs, Users and Devices," *Signal Processing: Image Communications*, Vol. **16**, 2000, pp. 211–234.
9. J. M. Martinez, J. Cabrera, J. Bescós, J. M. Menéndez and G. Cisneros, "Description Schemes for Retrieval Applications Targeted to the Audiovisual Market," *Int. Conf. on Multimedia and Expo*, Vol. **2**, 2000, pp. 793–796.
10. A. D. Doulamis, N. D. Doulamis and S. D. Kollias, "A Fuzzy Video Content Representation for Video Summarization and Content-based Retrival," *Signal Processing*, Vol. **80**, 2000, pp. 1049–1067.
11. Y. Wang, J. Ostermann and Y.-Q. Zhang, *Video Processing and Communications*, Prentice Hall Signal Processing Series, Upper Saddle River, NJ, 2002.
12. M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis, and Machine Vision*, PWS Publishing, Pacific Grove, CA, 1999.
13. A. Yamada, M. Pickering, S. Jeannin, L. Cieplinski, J.-R. Ohm and M. Kim, "Text of 15938-3/FDIS Information Technology – Multimedia Content Description Interface – Part 3 Visual," Tech. Rep. N4358, ISO/IEC JTC1/SC29/WG11, Sydney, AU, July 2001.
14. A. Yamada, M. Pickering, S. Jeannin, L. Cieplinski, J.-R. Ohm and M. Kim, "MPEG–7 Visual part of eXperimentation Model Version 12.0," Tech. Rep. N4548, ISO/IEC JTC1/SC29/WG11, Pattaya, TH, December 2001.
15. S. Abbasi, F. Mokhtarian and J. Kittler, "Curvature Scale Space Image in Shape Similarity Retrieval," *Springer Journal of MultiMedia Systems*, 1999.
16. P. van Beek, A. B. Benitez, J. Heuer, J. Martinez, P. Salembier, Y. Shibata, J. R. Smith and T. Walker, "Text of 15938-5/FDIS Information Technology – Multimedia Content Description Interface – Part 5 Multimedia Description Schemes," Tech. rep. N4242, ISO/IEC JTC1/SC29/WG11, Sydney, AU, July 2001.
17. O. Steiger, "Smart Camera for MPEG–7", Tech. Rep., Swiss Federal Institue of Technology, Lausanne, 2001.
18. T. Ebrahimi, Y. Abdeljaoued, R. M. Figueras i Ventura and O. Divorra Escoda, "MPEG–7 Camera," *Proc. Int. Conf. Image Processing*, IEEE, Vol. **3**, 2001, pp. 600–603.
19. A. Perkis, Y. Abdeljaoued, C. Christopoulos, T. Ebrahimi and J. F. Chicharo, "Universal Multimedia Access from Wired and Wireless Systems," *Circuits and Systems for Signal Processing*, Vol. **30**, No. 3, 2001, pp. 387–402.
20. S. Paek, "Description of MPEG–7 Content Set," Tech. Rep. N2467, ISO/IEC JTC1/SC29/WG11, Atlantic City, USA, October 1998.