Object-Based Multimedia Content Description Schemes and Applications for MPEG-7

Ana B. Benitez [(a)] [*], Seungyup Paek [(b)], Shih-Fu Chang [(b)]
Atul Puri [(c)], Qian Huang [(c)]
John R. Smith [(d)], Chung-Sheng Li [(d)], Lawrence D. Bergman [(d)]
Charles N. Judice [(e)]

(a)
Image and Advanced Television Laboratory (ADVENT)
Department of Electrical Engineering
Columbia University
1312 S. W. Mudd, 500 West 120th Street, M.C. 4712
New York, NY 10027, USA

(c)
AT&T Labs
100 Schultz Drive – Middletown
Red Bank, NJ 07701, USA

(d)
IBM T.J Watson Research Center
30 Saw Mill River Rd
Hawthorne, NY 10532, USA

(e)
Eastman Kodak
1447 St Paul Street
Rochester, NY 14653, USA

** Corresponding Author: Phone: +1 212-854-7473; Fax: +1 212-932-9421; E-mail: ana@ee.columbia.edu.*

Abstract

In this paper, we describe description schemes (DSs) for image, video, multimedia, home media, and archive content proposed to the MPEG-7 standard. MPEG-7 aims to create a multimedia content description standard in order to facilitate various multimedia searching and filtering applications. During the design process, special care was taken to provide simple but powerful structures that represent generic multimedia data. We use the eXtensible Markup Language (XML) to illustrate and exemplify the proposed DSs because of its interoperability and flexibility advantages. The main components of the image, video, and multimedia description schemes are object, feature classification, object hierarchy, entity-relation graph, code downloading, multi-abstraction levels, and modality transcoding. The home media description instantiates the former DSs proposing the 6-W semantic features for objects, and 1-P physical and 6-W semantic object hierarchies. The archive description scheme aims to describe collections of multimedia documents, whereas the former DSs only aim at individual multimedia documents. In the archive description scheme, the content of an archive is represented using multiple hierarchies of clusters, which may be related by entity-relation graphs. The hierarchy is a specific case of entity-relation graph using a containment relation. We explicitly include the hierarchy structure in our DSs because it is a

natural way of defining composite objects, a more efficient structure for retrieval, and the representation structure used in MPEG-4. We demonstrate the feasibility and the efficiency of our description schemes by presenting applications that already use the proposed structures or will greatly benefit from their use. These applications are the Visual Apprentice, the AMOS-Search system, a multimedia broadcast news browser, a storytelling system, and an image meta search engine, MetaSEEk.

Keywords: MPEG-7, multimedia description scheme, multimedia representation, object-based description, multimedia, image, video, home media, archive.

---

1. Introduction

During the past few years, multimedia content has become available at an increasing rate, especially in digital format. Therefore, it has become increasingly important to develop systems that process, filter, search, summarize, store, exchange, and organize this information so that useful knowledge can be derived from this mass of information. To improve current applications and enable new exciting ones that efficiently manage this multimedia information, an efficient, compact, flexible, and interoperable representation of the multimedia content is necessary.

The MPEG-7 standard [18] aims to satisfy the need by standardizing a framework for compact, efficient, and interoperable descriptions of audio-visual content. The standardization effort has been divided into four steps where work is being done concurrently: descriptors (Ds), description schemes (DSs), description definition language (DDL), and coded descriptions. Descriptors represent the features that characterize the data of the audio-visual content. Description schemes specify structures and semantics grouping other description schemes and descriptors. The description definition language allows specifying new description schemes and descriptors. Finally, schemes to code the descriptions are needed to satisfy the compression and the transmission requirements. MPEG-7 will improve current multimedia applications and enable new exciting ones; some examples are distributed processing, content exchange, personalized representation, and efficient storage/retrieval of multimedia content.

This paper presents the description schemes for image [13] [15], video [14] [15], multimedia [6], home media [3], and archive content [4] proposed to MPEG-7, which meet the requirements outlined by MPEG-7. The image, video, and multimedia description schemes are generic in the sense that they do not target any specific applications; the home media and archive DSs are more application oriented because the deal with more specialized content and functionality. Our description schemes are independent of any description definition language (DDL); however, we use eXtensible Markup Language (XML) [25] in the implementation. The use of XML is motivated by its recognized advantages: self-describing capability, intuitive readable format, simplicity, and extensibility, which support and add to the powerful features of our description schemes. During the presentation of our description schemes, the use of XML is intended to illustrate the use of the proposed description schemes.

The goal of the image, the video, and the multimedia description schemes is to describe single content documents, i.e., an image, a video sequence, and a multimedia stream, respectively. They share the same fundamental components: object, feature classification, object hierarchy, entity-relation graph, multiple levels of abstraction, modality transcoding, and code downloading. In these DSs, a content document is represented by a set of objects and relations among objects. Each object may have one or more associated features, which are grouped in the following categories: media features, visual features, temporal features, and semantic features. Each feature includes descriptors that can point to external extraction and similarity matching code. Relations among objects can be described by object hierarchies and entity-relation graphs. Object hierarchies can also incarnate the concept of multiple levels of abstraction. Modality transcoding [12] allows terminals with different capabilities (e.g. palmpilot and PCs) to receive the same content in different resolutions and/or modalities.

The image, the video, and the multimedia description schemes were developed for general applications. The home media description scheme is an example of how these DSs can be instantiated for use in home media applications. It proposes the 6-W semantic object features – who, what object, what action, why, where, when –, and 1-P physical and 6-W semantic object hierarchies to satisfy the requirements of home media applications.

Contrary to the former DSs, the archive description scheme provides structures to describe collections of content documents: images, video sequences, and/or multimedia streams, which are already described using the previous DSs. A multimedia archive is represented as multiple hierarchies of clusters that group objects of the individual content documents in the archive based on the semantic, the visual, the temporal, and/or the media features associated with those objects. Clusters, as objects, can be characterized by features in different categories (e.g. media and statistical). General relationships among clusters can be expressed using multiple entity-relation graphs in the archive DS.

All our description schemes include hierarchies and entity-relation graphs. In fact, a hierarchy is a specific case of entity-relation graph. We will show that a hierarchy is an instantiation of an entity-relation graph where the relations among the entities are of containment. Nevertheless, we decided to explicitly include the hierarchical structure in our DSs because the hierarchy is a natural way of defining composite objects, a more efficient structure for retrieval, and the representation structure used in MPEG-4.

We will demonstrate the utility and the use of each one of our description schemes with respect to how they are used and/or impact some application systems that we have developed. Some of these application systems have capabilities that critically depend on interoperable multimedia descriptions. These applications are the Visual Apprentice [10], a model-based image classification system; the AMOS-Search system [27], an object-based video search system; a multimedia broadcast news browser [9]; a storytelling system; and MetaSEEK [2], a meta search engine for mediation among multiple search engines for audio-visual information.

This paper is organized as follows. In section 2, we briefly discuss how our description schemes uniquely address the requirements of MPEG-7 and the specific role that XML performs on them. We describe the components, provide examples, and present an application scenario of the proposed image, video, multimedia, home media, and archive description schemes in section 3, section 4, section 5, section 6, and section 7, respectively. The basic components of the proposed DSs (e.g. hierarchies and entity-relation graphs) are described in depth for the simplest DS, the image DS. The explanation is extended for the other DSs referencing back to the image DS. Finally, we summarize the key points of our description schemes and the open issues for future research in section 8.

2. Design Principles and Goals

The proposed description schemes are designed to address the MPEG-7 requirements for the description of visual content [18]. We first explain how our design satisfies most of MPEG-7 requirements, then focus on specific requirements of flexibility and extensibility that are addressed by the use of XML.

2.1 Satisfying General MPEG-7 Requirements

The proposed description schemes satisfy the criteria stated by MPEG-7 as follows:

- *Object-Based Multi-Level Abstraction:* The DSs use “object” as the fundamental entity in describing multimedia content at various levels, which can be defined along different dimensions. For instance, objects can be used to describe image regions, groups of regions, video segments, and groups of segments among others. High-level objects can be used to describe groups of primitive objects based on semantics or visual features. Different types of features can be used for different levels of objects. As an example, visual features are adequate for objects corresponding to physical components in the content, while semantic features can be applied to objects at any level.

- *Effectiveness:* Our object-based DSs provide an efficient framework for accommodating various types of multimedia content and their features in different domains. For each DS, we present an example application that uses the proposed structures or will greatly benefit from their use. Some of these systems are large-scale visual content search and filtering systems.

- *Application Domain:* The proposed DSs are generic and support a very broad range of applications.

- *Comprehensiveness:* The proposed DSs covers a variety of multimedia content types: images, video sequences, multimedia documents, home media content, and multimedia collections.

- *Flexibility:* The flexibility of the proposed DSs is achieved by (1) allowing parts of the DSs to be instantiated; (2) using efficient categorization of features and clustering of objects (the indexing hierarchy); and (3) supporting efficient linking, embedding, or downloading of external feature descriptors and execution code.

- *Extensibility:* Elements defined in the DSs can be used to derive new elements for different domains. An example is the home media description scheme, which simply extends the feature set and the type of object hierarchies of the multimedia, the image, and the video description scheme.

- *Scalability:* One unique aspect of our DSs is the capability to define multiple abstraction levels based on any arbitrary set of criteria using the object hierarchies. The criteria can be specified in terms of visual features (e.g., size and color), semantic relevance (e.g., relevance to user interest profile), service quality (e.g., media features), and/or temporal features.

- *Simplicity:* These DSs specify a minimal set of components: objects, features classes, object hierarchies, and entity-relation graphs. Additional objects and features can be easily added in a modular and flexible way. Different types of object hierarchies and entity-relation graphs can be defined in a similar fashion.

2.2 Extensible Markup Language (XML)

From among the various MPEG-7 requirements, the flexibility and extensibility requirements are critical for widespread acceptance of MPEG-7 in many application domains. We propose the use of XML [25] to satisfy these requirements; a brief historical overview of XML leading to the justification of its use in our proposal is presented next.

SGML (Standard Generalized Markup Language, ISO 8879) is a standard language for defining and using document formats. SGML allows documents to be self-describing, i.e., they describe their own grammar by specifying the tag set used in the document and the structural relationships that those tags represent. However, full SGML contains many optional features that are not needed for Web applications and has proven to be too complex to current vendors of Web browsers. The World Wide Web Consortium (W3C) has created an SGML Working Group to build a set of specifications to make it easy and straightforward to use the beneficial features of SGML on the Web [25]. This subset, called XML (eXtensible Markup Language), retains the key SGML advantages in a language that is designed to be vastly easier to learn, use, and implement than full SGML.

A major advantage of using XML is that it allows the descriptions to be self-describing, in the sense that they combine the description and the structure of the description in the same format and document. XML also provides the capability to import external structural descriptions (e.g. for feature descriptors) into our multimedia description schemes in a highly modular and extensible way. We will see an example in the next section.

3. Image Description Scheme

We present the image description scheme [13] [15] for interoperable image descriptions in this section. To clarify the explanation, we use the example shown in Figure 1. Figure 1 shows the description of an image by the proposed image DS: image objects, object features, object hierarchy, and entity-relation (E-R) graph.

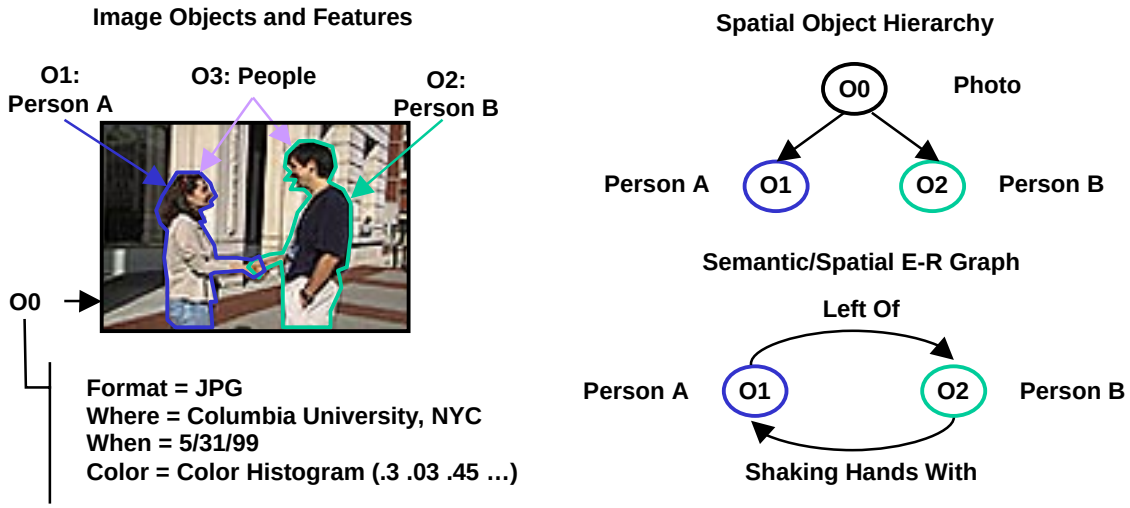


Figure 1: Description of an image by proposed description scheme.

Figure 2 shows the graphical representation of the proposed image description scheme following the UML notation[1] [22]. Under the proposed image description scheme, an image is represented as a set of image objects, which are related by object hierarchies and entity-relation graphs. Objects can have multiple features that can link to external extraction and similarity matching code. Features are categorized into media, visual, and semantic features. Objects can be organized in multiple object hierarchies. Non-hierarchical relationships among two or more objects can be described using multiple entity-relation graphs. Multiple levels of abstraction in indexing and viewing the objects in a large image based on media, visual, and/or semantic features can be implemented using object hierarchies. A special media feature that we propose is modality transcoding that allows users with different terminal specifications (e.g. cellular phones and computers connected to high-speed networks) to access the same image content in adequate modalities and resolutions. We explain these components in the remainder of this section. We will also describe how the Visual Apprentice [10] uses the same structures for image classification.

In our image DS (see Figure 2), the image element (<image>), which represents the image description, includes an image object set element (<image_object_set>), one or more object hierarchy elements (<object_hierarchy>), and one or more entity-relation graphs (<entity_relation_graph>). The hierarchy is a special case of entity-relation graph whose entities are related by containment relationships. We explicitly include the hierarchy in our DS for several reasons: (1) the hierarchy is a more efficient structure for retrieval than a graph; (2) a hierarchy is the most natural way of defining composite objects; and (3) MPEG-4 objects are built following a hierarchical structure.

We separate the definition of the objects from the structures that describe relationships among these objects for flexibility and generality. First, the same object may appear in different hierarchies and entity-relation graphs. If we included the object features at each node in the object hierarchy and the entity-relation graph, we would be duplicating the features for objects that appear in more than one hierarchies and/or graphs. Besides, a mechanism would be needed to identify the nodes of the hierarchies and the graphs that represent, in fact, the same object. Second, an object can be defined without the need for it to be included in any relational structures, a

[1] The diamond symbol represents the composition relationship. The range associated to each element represents frequency in the composition relationship (e.g. "0..*" means zero or more and "1..*" means one or more").

hierarchy or a graph, so the extraction of objects and relations among objects could be done at different stages permitting the distributed processing of the image content.

3.1 Image Object and Object Set

An image object refers to one or more arbitrary regions of an image, so it can be continuous or discontinuous in space. In Figure 1, O1 ("Person A"), O2 ("Person B"), and O0 ("Photograph") are objects with only one associated continuous region. O3 ("People") is an example of an object composed of multiple regions separated in space. We distinguish between local objects and global objects; a global object contains the features that are common to an entire image; a local object only contains the features of a section of the image. O0 ("Photograph") is the global object representing the entire image. We treat the image as another object in the object set because all share the same features, descriptors, and relations.

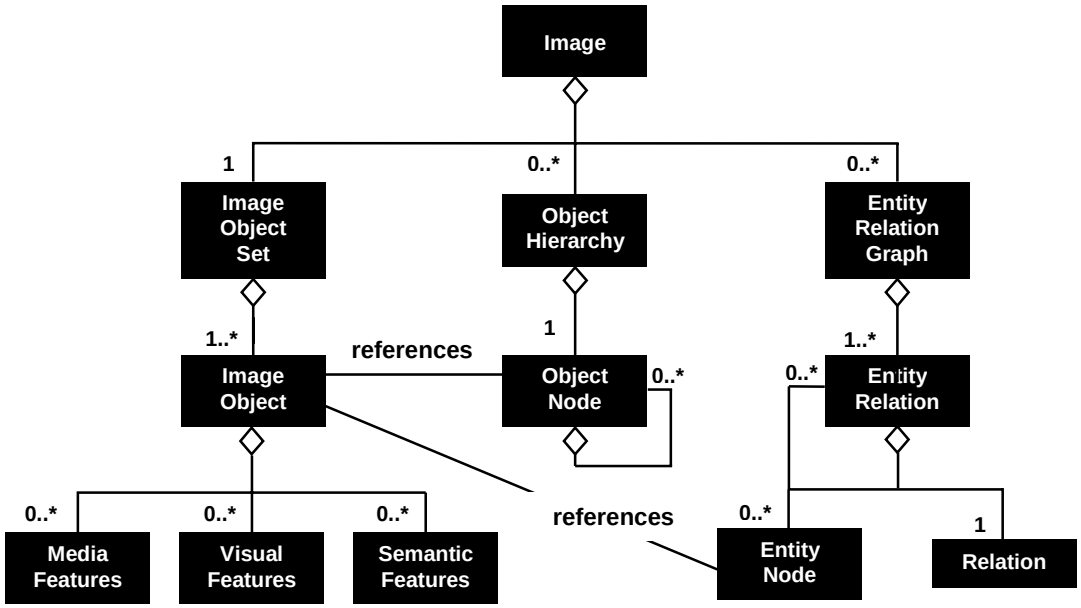


Figure 2: UML representation of the image description scheme.

In an early stage of the development of the image DS [15], we proposed two different types of objects: physical objects (continuous regions) and logical objects (semantic objects). However, we decided to remove this categorical distinction for generality. We identify different types of objects like visual objects, i.e., objects that are defined by visual features (color, texture, etc.; e.g. red color object), media objects, semantic objects, and objects defined by the combination of semantic, visual, and media features. In other words, we believe the type of an object is determined by the features used to describe that object. Furthermore, new types of objects can be added as needed. For each application, different types of objects can be derived from the proposed generic object by making use of inheritance relationships. The current specification of the MPEG-7 DDL [17] (MPEG Meeting at Melbourne) supports inheritance.

As shown in Figure 2, the set of all image object elements (<image_object>) that are described in an image is included within the object set element (<image_object_set>). Each object element can have a unique identifier within an image description. The identifier and the type of an object (local or global) are expressed as attributes of the object element, id and type, respectively. The objects that we have chosen to describe for the image in Figure 1 are listed in XML below. The text between "<!--" and "-->" are comments as in HTML.

```
<image_object_set>
        <image_object id="O0" type="GLOBAL"> </image_object> <!-- Photograph -->
        <image_object id="O1" type="LOCAL"> </image_object> <!-- Person A -->
        <image_object id="O2" type="LOCAL"> </image_object> <!-- Person B -->
        <image_object id="O3" type="LOCAL"> </image_object> <!-- People -->
</image_object_set>
```

3.2 Object Features

Image objects can contain three feature elements that group features based on the information they convey (see Figure 2): media (<img_obj_media_features>), visual (<img_obj_visual_features>), and semantic (<img_obj_media_ features>) features [11]. Table 1 compiles a list of example features for each feature class. Note that we propose six semantic features (who, what object, what action, why, when, and where) plus another text annotation feature for annotations. We will talk about the 6-Ws in the home DS section (section 6).

Table 1: Feature classes and features.

Feature Class	Features
Semantic	Text Annotation, Who, What Object, What Action, Why, When, Where
Visual	Color, Texture, Position, Size, Shape, Orientation
Media	File Format, File Size, Color Representation, Resolution, Data File Location, Modality Transcoding, Author, Date of Creation

Each feature element in an image object will include the descriptors selected by MPEG-7. Descriptors that may be associated with some visual features are shown in Table 2. Specific descriptors can include links to external extraction and similarity matching code. It is important to emphasize that our image DS can include any number of features for each object in an extensible and modular way.

Table 2: Examples of visual features and associated descriptors.

Feature	Descriptors
Color	Color Histogram, Dominant Color, Color Coherence Vector, Visual Sprite Color
Texture	Tamura, MSAR, Edge Direction Histogram, DCT Coefficient Energies, Visual Sprite Texture
Shape	Bounding Box, Binary Mask, Chroma Key, Polygon Shape, Fourier Shape, Boundary, Size, Symmetry, Orientation

The XML example below shows how features and descriptors are included in the image object. In particular, the example expresses the features associated with the global object O0 in Figure 1: two semantic features (where and when), one media feature (file format), and one visual feature (color with color histogram descriptor). An object can be described by different concepts (<concept>) in each of the 6-W semantic categories as shown in the example below.

```
<image_object id="O0" type="GLOBAL"> <!-- Global object: Photograph -->
        <img_obj_semantic_features>
                <where>
                        <concept> Columbia University, NYC </concept>
                        <concept> Outdoors </concept>
                </where>
                <when> <concept> 5/31/99 </concept> </when>
        </img_obj_semantic_features>
        <img_obj_media_features>
                <file_format> JPG </file_format>
        </img_obj_media_features>
        <img_obj_visual_features>
```

```
                    <color>
                        <color_histogram>
                            <value format="float[166]"> .3 .03 .45 … </value>
                        </color_histogram>
                    </color>
                </img_obj_visual_features>
            </image_global_object>
```

3.3 Object Hierarchy

In the image DS, the object hierarchy can be used to organize the image objects in the object set based on different criteria: media features, visual features, semantic features, or combinations. Each object hierarchy is a tree of object nodes which reference image objects in the object set. We shall describe the object hierarchy in this section. We also include an example that illustrates the use of object hierarchies to build indexing hierarchies and to generate multi-abstraction level descriptions.

A hierarchy implies a containment relation from the child nodes to the parent node. This relation may be of different types depending on the object features selected: semantic, visual, and/or media. For example, the spatial object hierarchy in Figure 1 describes a visual containment because it was created considering a visual feature (spatial location). Figure 3 shows two more examples of object hierarchies. The first hierarchy is based on the "who" semantic feature and the second hierarchy is based on color and shape features (O7 can be defined as the region of the object satisfying some color and shape constraints). Hierarchies combining different features can also be built satisfying the requirements of a broader range of application systems. In an earlier phase of the development process of the image DS, we distinguish between physical and logical hierarchies, as for objects. However, we decided against it for similar reasons: generality, need for indexing hierarchies based on any combination of visual, media, and semantic features (see section 3.2), and capacity for generating specific types of hierarchies using inheritance relationships.

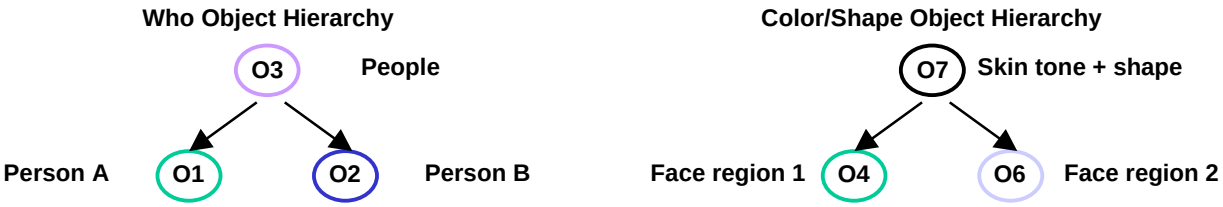


Figure 3: Examples of different types of hierarchies.

As shown in Figure 2, each object hierarchy (<object_hierarchy>) is a tree of object nodes. The object hierarchy includes an optional string attribute, type. A thesaurus could provide the values of the attribute type so that applications know exactly what type of hierarchies they are dealing with. However, it has been left as open text. Every object node (<object_node>) references an object in the object set. Objects can actually point back to the object nodes referring them. This bi-directional linking mechanism allows efficient transversal from objects in the object set to corresponding object nodes in the object hierarchy, and vice versa. Each object node references an object through an attribute, object_ref, by using the latter's unique identifier. Each object node element can also include a unique identifier in the form of an attribute. By including the object nodes's unique identifiers in their object_node_ref, objects can point back to object nodes referencing them. The spatial object hierarchy in Figure 1 is expressed in XML below.

```
<object_hierarchy type="SPATIAL"> <!-- Object hierarchy: spatial hierarchy -->
        <object_node id="ON0" object_ref="O0"> <!— Photograph -->
                <object_node id="ON1" object_ref="O1"> </object_node> <!-- Person A -->
                <object_node id="ON2" object_ref="O2"> </object_node> <!-- Person B -->
```

```
        </object_node>
</object_hierarchy>
```

In the reminder of the section, we will describe how object hierarchies can be used to build indexing hierarchies and to generate multiple-abstraction levels. In describing large images (e.g. satellite images), the main problem that usually arises is how to describe and retrieve many objects contained in the images in an efficient and scalable way. The most frequently used structures to solve this issue are *indexing hierarchies.* Figure 4 shows an image whose objects we want to index hierarchically based on their size. Instead of using only one feature to create an indexing hierarchy, we may have wanted to create multiple indexing hierarchies using different criteria involving multiple features.

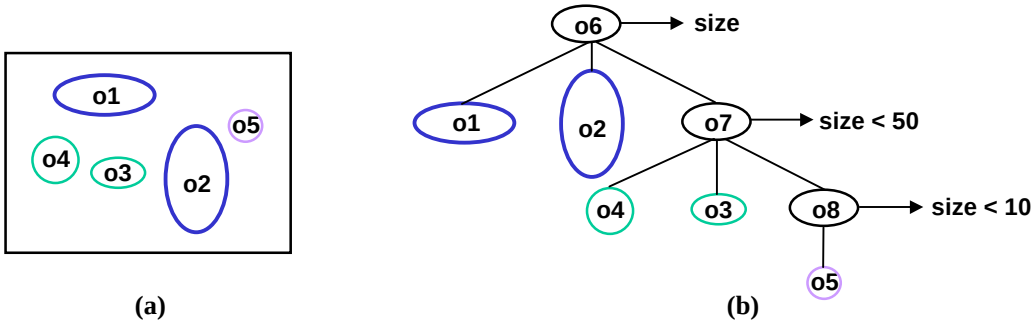


Figure 4: (a) Example of objects in a large image (e.g. lakes in a satellite image). (b) Indexing hierarchy of objects based on their size.

Using the proposed object hierarchy, we could describe such indexing hierarchies that cluster objects based on some selected media, visual, and semantic features. The procedure will be similar to the one used to cluster images in visual information retrieval engines. Each object in the large image would be assigned an image object in the object set and some associated features. The intermediate nodes of the object hierarchy will also be represented as image objects but will include the criteria, conditions and constraints on one or more features, used for grouping the objects at that level. The image description could include as many indexing hierarchies as required. See part of the description (object definition and indexing hierarchy) of the image in Figure 4 expressed in XML below. In this example, the features associated with the intermediate nodes of an object hierarchy provide an indication of the type of object hierarchy in the description (feature <size> in object o7).

```
<image>
        <image_object_set>
                <image_object type="LOCAL" id="o1"> <!-- Real objects of the image -->
                        <size> <num_pixels> 120 </num_pixels> </size>
                </image_object> <!-- Others objects -->
                <image_object type="LOCAL" id="o7"> <!-- Intermediate nodes in the hierarchy -->
                        <size> <num_pixels> <less_than> 50 </less_than> </num_pixels> </size>
                </image_object> <!-- Others objects -->
        </image_object_set>
        <object_hierarchy>
                <object_node id="on1" object_ref="o6">
                        <object_node id="on2" object_ref="o1" />
                        <object_node id="on3" object_ref="o2" />
                        <object_node id="on4" object_ref="o7">
                                <object_node id="on5" object_ref="o3" />
                                <object_node id="on6" object_ref="o4" />
```

```
                                        <object_node id="on7" object_ref="o8">
                                                <object_node id="on8" object_ref="o5" />
                                        </object_node>
                                </object_node>
                        </object_node>
                </object_hierarchy>
        </image>
```

Continuing the example in Figure 4, we have actually defined three levels of abstraction based on the size of the objects (see Figure 4 and Table 3). This multi-level abstraction scheme provides a scalable method for retrieving and viewing objects in the image. This approach can also be used to represent multiple abstraction levels based on other features such as semantic classes (e.g. forest, lake, and house).

Table 3: Objects in each resolution layer.

Resolution Layer	Objects
1	o1, o2
2	o1, o2, o3, o4
3	o1, o2, o3, o4, o5

3.4 Entity-Relation Graph

Although a hierarchical structure is adequate for retrieval purposes, some relationships among objects can not be expressed using such a tree structure. Our description scheme also allows the specification of more complex relationships among objects using entity-relation graphs. An entity-relation graph is a graph of entity nodes and relations among them. Table 4 lists several types of relationships with examples. We do not define specific types of graphs; we provide a general structure that could be customized for each application using inheritance relationships. For example, the entity-relation graph in Figure 1 describes a spatial relationship, “Left Of”, and a semantic relationship, “Shaking Hands With”, between the object O1 and the object O2 of the image.

Table 4: Examples of relation types and relations.

Relation Type		Relations
Spatial	Directional	Top Of, Bottom Of, Right Of, Left Of, Upper Left Of, Upper Right Of, Lower Left Of, Lower Right Of
	Topological	Adjacent To, Neighboring To, Nearby, Within, Contain
Semantic		Relative Of, Belongs To, Part Of, Related To, Same As, Is A, Consist Of

Figure 2 shows that the image DS allows for the specification of one or more entity-relation graphs (<entity_relation_ graph>). An entity-relation graph includes a set of entity-relation elements (<entity_relation>) and has two optional attributes, a unique identifier, id, and a string, type, to describe the binding expressed by the graph. Values for types could be given by thesaurus. An entity relation (<entity_node>) includes one relation element (<relation>), zero or more entity node elements (<entity_node>), and zero or more entity-relation elements. The relation element contains the specific relationship being described. Each entity node element references an object in the object set using an attribute, object_ref. Objects can point back to entity nodes referring them. Consider the entity-relation graph in Figure 1, it includes two entity relations between the object O1 (“Person A”) and the object O2 (“Person B”). The first entity relation describes how object O1 is positioned to “Left Of” (spatial relation) object O2. The second entity relation describes how object O1 is “Shaking Hand With” (semantic relation) object O2. The XML implementation of the example follows.

```
<entity_relation_graph>
        <entity_relation> <!— Spatial, directional entity relation -->
                <relation type="SPATIAL.DIRECTIONAL"> Left Of </relation>
```

```
            <entity_node id="ETN1" object_ref="O1"/> <entity_node id="ETN2" object_ref="O2"/>
        </entity_relation>
        <entity_relation> <!— Semantic entity relation -->
            <relation type="SEMANTIC"> Shaking hands with </relation>
            <entity_node id="ETN3" object_ref="O2"/> <entity_node id="ETN4" object_ref="O1"/>
        </entity_relation>
</entity_relation_graph>
```

Entity-relation elements can include other entity-relation elements for efficiency. It allows creating efficient nested graphs of entity relationships as the ones in SMIL [26]. SMIL synchronizes different media documents by using nested "parallel" (<par>) and "sequential" (<seq>) relationships. In the SMIL example below, an image is displayed in parallel with an audio and a video sequence that are displayed sequentially.

```
<par>
        <img />
        <seq> <audio /> <video /> </seq>
</par>
```

In a similar way, nested relationships are used to efficiently describe temporal relationships among objects in example below. In the example below, object O1, object O2, and the sequence of O3 and O4 are parallel in time.

```
<entity_relation_graph>
        <entity_relation>
            <relation type="TEMPORAL.TOPOLOGICAL"> Parallel </relation>
            <entity_node id="ETN1" object_ref="O1"/>
            <entity_node id="ETN1" object_ref="O1"/>
            <relation type="TEMPORAL.TOPOLOGICAL"> Sequential </relation>
                <entity_node id="ETN3" object_ref="O3"/>
                <entity_node id="ETN4" object_ref="O4"/>
            </entity_relation>
        </entity_relation>
</entity_relation_graph>
```

A hierarchy can be implemented using an entity-relation graph whose entities are related by containment relationships - the "contain" relationship is a spatial, topological relationship (see Table 4). To show that a hierarchy is a particular case of entity-relation graph, we express the object hierarchy in Figure 1 using an entity-relation graph in XML below. The hierarchy shown in Figure 1 describes how object O0 ("Photograph") spatially contains objects O1 ("Person A") and O2 ("Person B"). Applications can decide between the convenience of using a coherence structure, an entity-relation graph, to implement hierarchies and graphs or the efficiency of using object hierarchies to implement hierarchies. Therefore, applications can decide to use entity-relation graphs or hierarchies based on specific trade-off considerations.

```
<entity_relation_graph>
        <entity_relation>
            <relation type="SPATIAL"> Contain </relation>
            <entity_node object_ref="O0"/> <entity_node object_ref="O1"/>
        </entity_relation>
        <entity_relation>
            <relation type="SPATIAL"> Contain </relation>
            <entity_node object_ref="O0"/> <entity_node object_ref="O2"/>
        </entity_relation>
</entity_relation_graph>
```

3.5 Code Downloading

For descriptors associated with any type of features, the image DS includes links to extraction and similarity matching code, as shown in the following XML example. These links provide a mechanism for content from different sources using proprietary descriptors to be searched and filtered efficiently. In the archive description scheme, we will motivate the code downloading mechanism by its intended use in MetaSEEk (see section 7.3).

Each descriptor in our description schemes can include the descriptor value and a code element, which contain information regarding extraction and similarity matching code for that descriptor. The code elements (<code>) can include pointers to the executable files (<location>), and the description of the input (<input_parameters>) and output (<output_parameters>) parameters to execute the code. Information about the type (extraction or similarity), the language (e.g. Java or C), and the version of the code are defined as attributes of the code element.

The example below includes the description of a Tamura texture [21] feature that provides the specific feature values (coarseness, contrast, and directionality) and also links to external code for feature extraction and similarity matching. In the feature extraction example, information about input and output parameters is also provided. This description could have been generated by a search engine as a response to a texture query from a meta search engine. The meta search engine could then use the code to extract the same feature descriptor from the results received from other search engines in order to generate a homogeneous list of results for the user (see section 7.3). In other cases, only the extraction and similarity matching code is included, but not the specific feature values. Then filtering agents would extract the feature values if necessary for their processing.

The example below also shows the way in which XML enables externally defined description schemes for descriptors to be imported and combined into the image DS. In the example, an external descriptor for the Croma Key shape feature is imported into the description by using XML namespaces. In this framework, new features, types of features, and descriptors can be included in an extensible and modular way.

```
<texture> <tamura>
        <tamura_value coarseness="0.01" contrast="0.39" directionality="0.7"/>
        <code type="EXTRACTION" language="JAVA" version="1.1"> <!-- Link extraction code -->
                <location> <location_site href="ftp://extract.tamura.java"/> </location>
                <input_parameters> <parameter name=”image” type=”PPM”/> </input_parameters>
                <output_parameters>
                        <parameter name=”tamura texture” type=”double[3]”/>
                </output_parameters>
        </code>
        <code type="DISTANCE" language="JAVA" version="4.2"> <!-- Link similarity code -->
                <location> <location_site href="ftp://distance.tamura.java"/> </location>
        </code>
</tamura> </texture>

<shape> <!-- Import external shape descriptor DTD -->
        <chromaKeyShape xmlns:extShape "http://www.other.ds/chromaKeyShape.dtd">
                <extShape:HueRange>
                        <extShape:start> 40 </extShape:start> <extShape:end> 40 </extShape:end>
                </extShape:HueRange>
        </chromaKeyShape>
</shape>
```

3.6 Modality Transcoding

Consider a content broadcaster that needs to transmit image content to their users. Due to the difference of terminals and bandwidth that each user has, the content broadcaster will need to transcode the image content into

different media modalities and resolutions as needed for each specific terminal [12]. A clear example is an image being transcoded into an audio sequence in order to be received by a cellular phone.

An important media feature that we include in our image DS is modality transcoding. Local objects as well as global objects can include the modality transcoding media feature. This media feature contains the media modality, the resolution, and the location of transcoded versions of the image object, or links to external transcoding code. The descriptor can point to code for transcoding the image object into different modalities and resolutions that would satisfy the requirements of the each user terminal. See the example in XML that follows where a audio transcoded version is available for an image object.

```
<image_object type="GLOBAL" id="O0">
        <img_obj_media_features>
                <location> <location_site href="Hi.gif"/> </location>
                <modality_transcoding>
                        <modality_object_set>
                                <modality_object id="mo2" type="AUDIO" resolution="1">
                                        <location><location_site href="Hi.au.xml"?o1/></location>
                                </modality_object>
                        <modality_object_set>
                </modality_transcoding>
        <img_obj_media_features>
</image_object>
```

3.7 Application: The Visual Apprentice

In the section, we shall present the Visual Apprentice [10], a model-based image classification system. The work on the Visual Apprentice is prior to the work on description schemes for MPEG-7; however the system uses the elements and structures proposed for the MPEG-7 description schemes: objects, object features, object hierarchies, and entity-relation graphs.

Many automatic image classification systems are based on a pre-defined set of classes in which class-specific algorithms are used to perform classification. The Visual Apprentice [10] allows users to define their own classes and build Visual Object Detectors for each one of the classes. The learning framework of the Visual Apprentice is based on an Object-Definition Hierarchy consisting of the following levels (Figure 5 (a)): region, perceptual, object-part, object, and scene.

To construct a Visual Object Detector (VISOD) for a class, the user begins by selecting the training images and building an object definition hierarchy according to his interests. The recognition strategy is based on automatic segmentation of training images - during training, each image is automatically segmented and example regions are manually labeled by the user according to the hierarchy he/she defined (see Figure 5 (b) and (c)). The labeled regions from all of the training examples are then used to compute the training set. Features (color, texture, shape, etc.) for each node of the hierarchy and spatial relationships (above/below, far/near, left/right, etc.) for the perceptual areas and the object-parts are automatically extracted and stored in a database. This data is then used for training by multiple learning algorithms that perform feature selection and yield a set of fuzzy-classifiers for each node (e.g., region, object-part, etc.).

The Visual Object Detector performs automatic classification of a new image by first applying automatic segmentation, and then combining classifiers and grouping strategies at the levels of Figure 5 (a): regions are classified first and combined to obtain perceptual-areas which are used by object-part classifiers. Object-parts, in turn, are combined and passed to object classifiers.

We will explain now how the visual model of the Visual Apprentice could be mapped into the proposed image DS. First, the objects, the object-parts, the perceptual areas, and the regions in the Visual Apprentice would be described as image objects. Each one of the four levels of the object-definition hierarchy is associated

with a specific set of features (low-level features and semantic labels), so four specialized types of objects could be derived from the generic object proposed in the image DS, one for each level. Second, the Object-Definition Hierarchy could be represented with the object hierarchy in the image DS. Third, entity-relation graphs in the image DS could be used to describe the spatial relationships among perceptual areas and object-parts. Finally, the semantic labels assigned by the user and generated by the automatic classification could be associated with a descriptor of a semantic annotation feature.

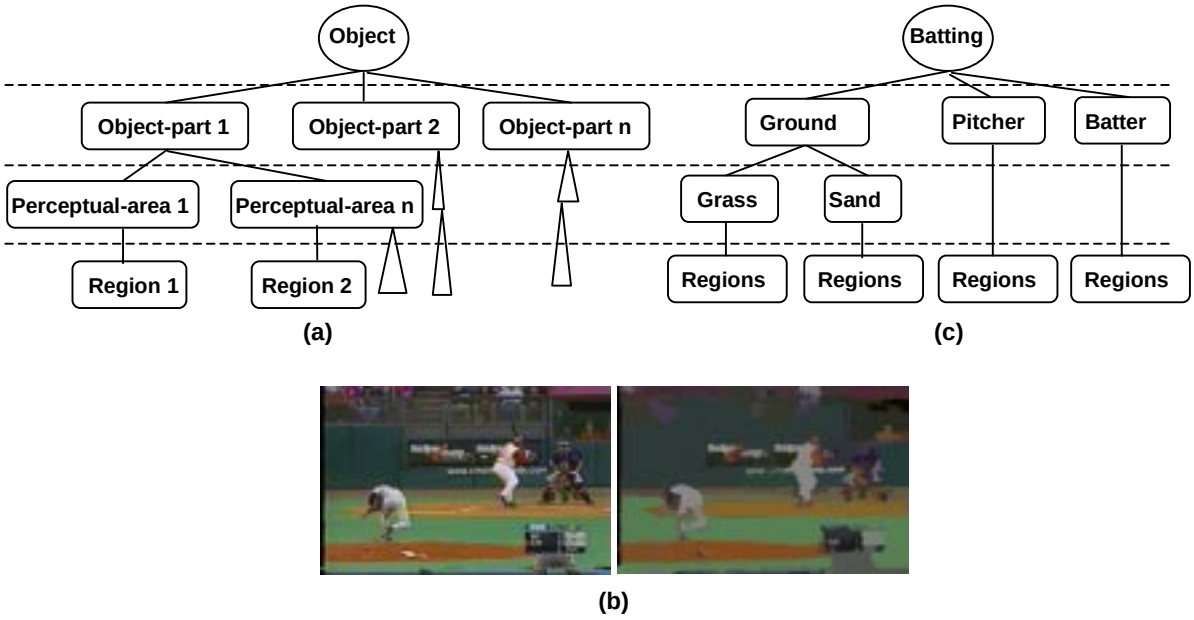


Figure 5: (a) Object-Definition Hierarchy in the Visual Apprentice. (b) Original and automatically segmented baseball image. (c) Instance of the Object- Definition Hierarchy for the batting class shown in (b).

As the entire visual model of the Visual Apprentice can be represented using the image DS, the system could be adapted to function on the proposed description scheme. It could accept descriptions of the user classes and the training examples as instances of the proposed image description scheme in order to generate the visual model for the class. The Visual Apprentice could also be used to generate descriptions of images following its Object-Definition Hierarchy compliant with the image DS.

4. Video Description Scheme

We present the video DS [14] [15] in this section. The video DS is the natural extension of the image DS adding the temporal dimension. To clarify the explanation, we use the example shown in Figure 6. Figure 6 presents a simple example of video description following the proposed DS: video objects, object features, object hierarchy, and entity-relation (E-R) graph. The UML graphical representation of the video DS is the video equivalent of the image DS in Figure 2 with the addition of temporal features to the video object. We will discuss the use of the structures proposed in the video DS in the AMOS-Search system [27].

4.1 Video Object and Object Set

The basic description element of our video DS is the video object. The video object extends the concept of image object adding the temporal dimension. In other words, a video object refers to one or more arbitrary regions in

one or more frames of a video sequence. We distinguish video objects that represent a region within the video (local object), entire frames of the video (segment object), or the entire video (global object). Objects need not be continuous in time or in space.

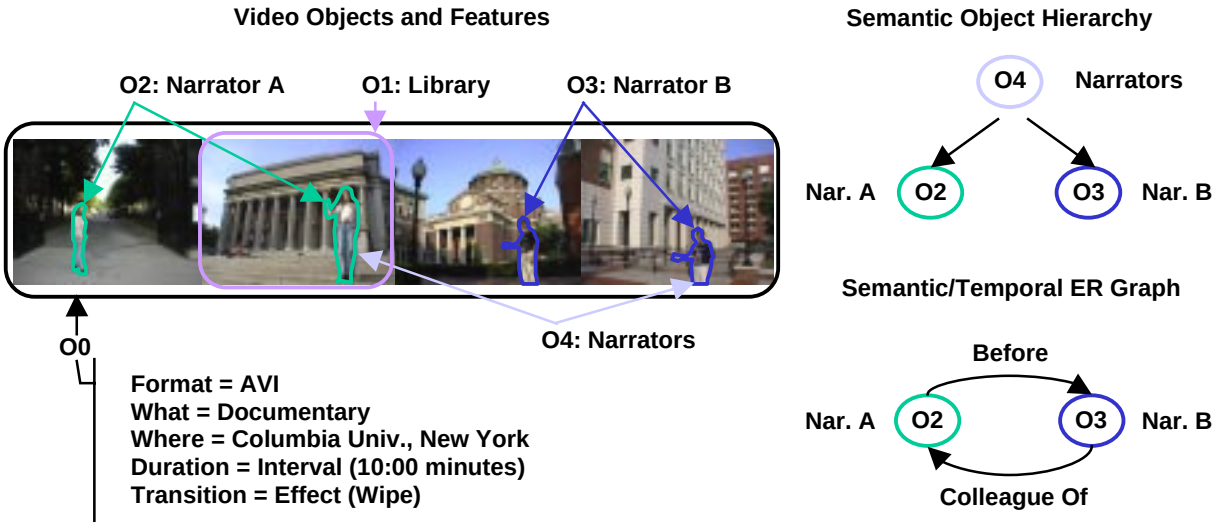


Figure 6: Description of a video by proposed description scheme.

Examples of video objects for the documentary video are shown in Figure 6. O0 ("Documentary") is the global object representing the entire video sequence. Object O1 ("Library") is a segment object continuous in space and time. Objects O2 ("Narrator A"), O3 ("Narrator B"), and O4 ("Narrators") are local video objects, the two former objects are continuous in time and space while the latter is discontinuous in space. The objects that we have chosen to describe for the video sequence in Figure 6 are listed below in XML.

```
<video_object_set>
        <video_object type="GLOBAL" id="O0"> </video_object> <!-- Documentary -->
        <video_object type="SEGMENT" id="O1"> </video_object> <!-- Library -->
        <video_object type="LOCAL" id="O2"> </video_object> <!-- Narrator A -->
        <video_object type="LOCAL" id="O3"> </video_object> <!-- Narrator B -->
        <video_object type="LOCAL" id="O4"> </video_object>  <!-- Narrators -->
</video_object_set>
```

4.2 Object Features, Object Hierarchy, Entity-relation graph, and Code Downloading

In the video DS, the object features, the object hierarchy, the entity-relation graph, and the code downloading are elements inherited directly from the image DS. The only extension regarding the image DS is the inclusion of temporal features, temporal relations, and the visual and media features related specifically to video. See Table 5 and
Table 6 for examples of the new relations, relation types, features, and feature classes for the video DS. Figure 6 shows an example of a semantic object hierarchy and a spatial/temporal entity-relation graph.

Table 5: Examples of new relation types and relations.

Relation Type		Relations
Temporal	Directional	Before, After, Immediately Before, Immediately After
	Topological	Co-Begin, Co-End, Parallel, Sequential, Overlap, Within, Contain, Nearby

Table 6: New feature classes and features.

Feature Class	Features
Visual	Motion, Editing Effect, Camera Motion
Temporal	Start Time, End Time, Duration

4.3 Application: AMOS-Search

AMOS-Search [27] is an object-based video indexing and search system. It uses a semi-automatic segmentation tool to extract and segment semantic objects (e.g. people, car, and animal) from a video sequence. The extracted video objects are represented using a hierarchical structure (see Figure 7 (a)) similar to the one proposed above. Each video object is characterized by the underlying regions, their features, and spatial-temporal relationships.

Visual features and spatial-temporal relation graphs at the semantic object and the region levels are computed to build a visual feature library. Semantic objects and regions include color, texture, shape, motion, and time features. For each object, structural features describe the spatial and temporal positions and boundaries of its regions. Also at the object level, there are the 6-Ws semantic annotations. The video description generated by this system is shown in Figure 7. AMOS-Search accepts queries in the form of sketches or examples and returns similar semantic objects based on different features. Figure 7 (b) shows the query interface of AMOS-Search: the query results, the query canvas, and the feature weights.

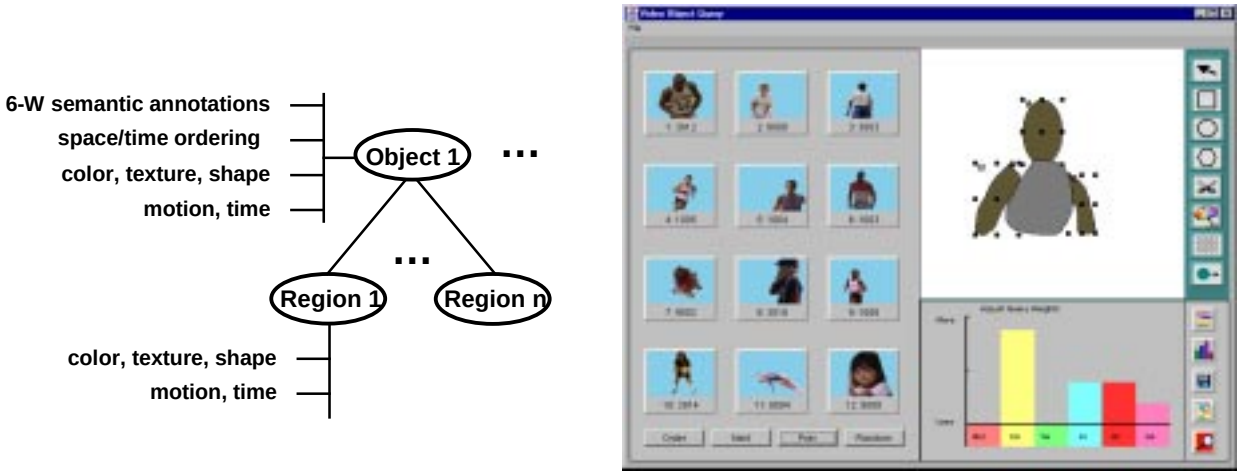


Figure 7: (a) Video object representation in AMOS. (b) Query interface of AMOS-Search: query results, query canvas, and feature weights.

We can map the object-based video model of the AMOS system (Figure 7 (a)) to the proposed video DS. First, the objects and the regions in the AMOS system would be described as video objects in the video DS. Two specialized classes of objects could be derived from the generic video object. The region object would contain only low-level features like color, texture, shape, motion, and time. The semantic object would contain the visual features and the semantic annotations. An object hierarchy would be used to express the spatial containment relationships among the objects and their regions. Entity-relation graphs would be used to represent the structural features at the object level, i.e. the spatial-temporal relations among the object's regions.

Using the proposed DS, the AMOS segmentation system could output descriptions of segmented video objects compliant with the proposed video DS. The AMOS-Search system could also accept video descriptions compliant with the video DS as query inputs to the database.

5. Multimedia Description Scheme

The multimedia description scheme [6] aims to describe multimedia content resulting from the integration of multiple media streams. Examples of individual media streams are images, audio sequences, natural video sequences, synthetic video sequences, and text captions. An example of such an integrated multimedia stream is a documentary program that includes a video, an audio, and a text stream; see Figure 8. We use the name multimedia stream to refer to any multi-modal document. Other examples of multimedia streams are slide shows composed of a set of images, video sequences, and text; albums composed of images and graphics; and web pages including text, video sequences, and images.

The proposed multimedia DS (MMDS) builds on top of the individual single-media description schemes, including the image DS and the video DS. One of our goals was to achieve the maximum synergy among the single-media DSs and the multimedia DS; all elements and structures used in the multimedia DS are intuitive extensions of those used in the image and the video DS for multiple single-media streams. Figure 8 shows the high-level description of the MMDS: multimedia objects, single-media objects, object features, object hierarchy, and entity-relation graph. Figure 9 presents the UML representation of the MMDS showing the relations with the single-media DSs.

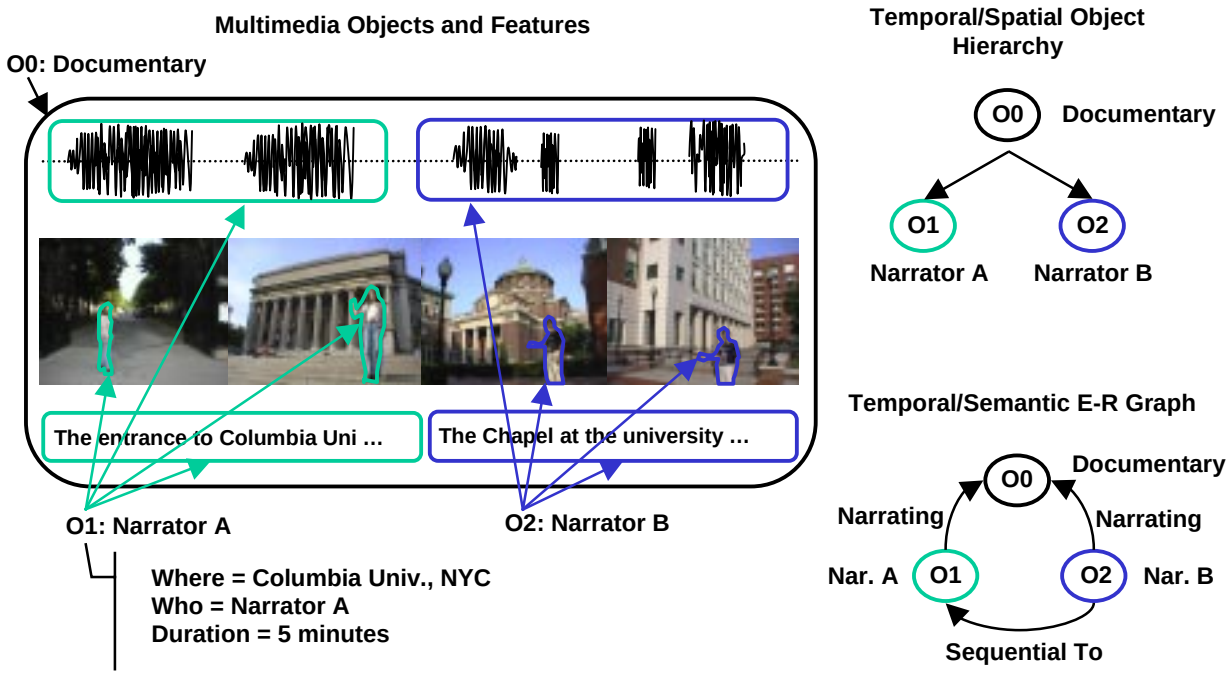


Figure 8: Description of a multimedia stream by proposed description scheme.

As shown in Figure 9, a multimedia stream is represented as a set of multimedia and single-media objects. A multimedia object refers to a set of single-media and other multimedia objects. Examples of single-media objects are image, video, and audio objects defined in the respective single-media DSs. The object hierarchy, the entity-relation graph, the object features, and the code downloading are elements inherited directly

from single-media DSs (image and video). Figure 8 includes examples of a temporal/spatial object hierarchy and temporal/semantic entity-relation graph. In the rest of the section, we will explain the components of the MMDS extended from the image and the video DSs'. We will also discuss how a multimedia broadcast news browser [9] already uses the structures in the MMDS.

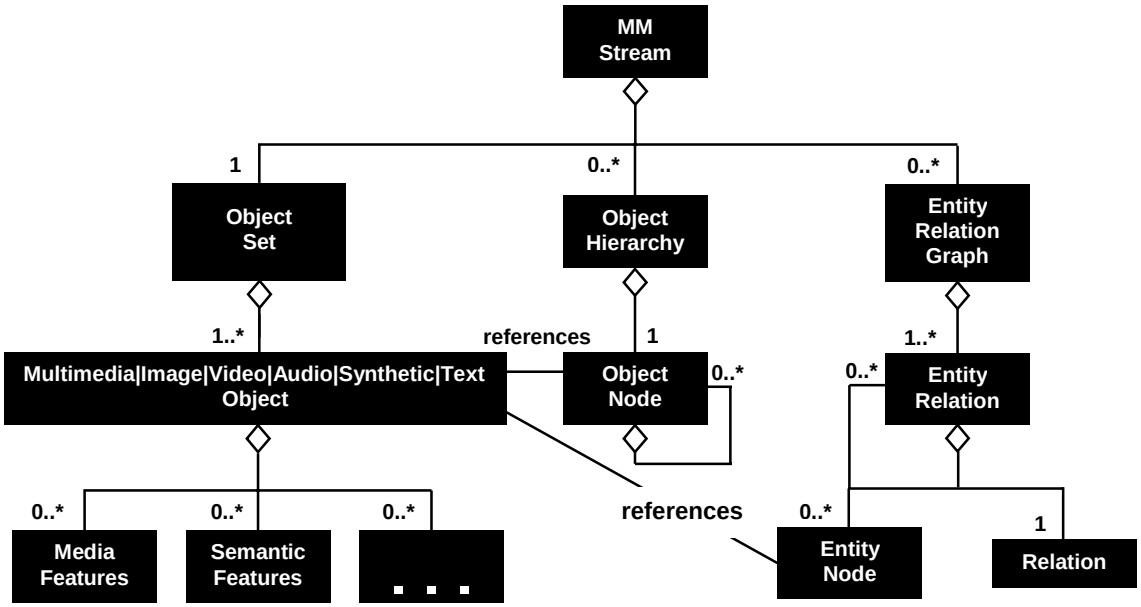


Figure 9: UML representation of the multimedia DS.

5.1 *Multimedia Object*

The basic description elements of the multimedia DS are the multimedia and the single-media objects. The set of all the multimedia and the single-media objects in a multimedia stream is included within the object set (see Figure 9). We will introduce the multimedia object in this section. A multimedia object represents a collection of single-media objects from one or more single-media streams, e.g. image, video, text, and audio objects, and other multimedia objects. The single-media objects may belong to the same single-media stream or different single-media streams, and may not bet synchronized in time or space. The object hierarchy is used to express the single-media and multimedia objects composing a multimedia object.

We distinguish two types of multimedia objects: local and global objects. A global multimedia object element represents the entire multimedia stream. On the other hand, a local multimedia object has a limited scope within the multimedia stream. In the example shown in Figure 8, O0 ("Documentary") is the global object representing all three single-media streams of the documentary (audio sequence, video sequence, and text captions); object O1, and O2 are local objects grouping all the single-media objects related to the "Narrator A" and "Narrator B", respectively. Each multimedia object can have associated multiple features and corresponding feature descriptors. A multimedia object can include semantic information, temporal information, and media specific information. The objects for the multimedia stream in Figure 8 are listed in XML below.

```
<Object_Set> <!-- A object set element -->
        <MM_Object type="GLOBAL" id="O0" …> </MM_Object> <!-- Documentary -->
        <MM_Object type="LOCAL" id="O1" …> </MM_Object> <!-- Narrator A -->
        <MM_Object type="LOCAL" id="O2" …> </MM_Object> <!-- Narrator B -->
</Object_Set>
```

5.2 Single-Media Object

Each single-media object refers to one type of media such as image, video, audio, synthetic video, and text. These single-media objects are defined in the description scheme of the corresponding type of media and may have associated object hierarchies and entity-relation graphs. It is important to point out that the single-media and the multimedia objects share the same structures except for different types of feature objects. See below how the single-media objects corresponding to the object O1 ("Narrator A") in Figure 8 are included in the object set element and are defined as its composing objects using an object hierarchy.

```
<object_set>
        <mm_object id="O1" … > </mm_object> <!-- Narrator A -->
        <!-- Single-media objects composing the "Narrator A" multimedia object -->
        <audio_object id="AO1"> </audio_object> <!--Audio segment -->
        <video_object id="VO1"> </video_object> <!-- Video object  -->
        <text_object id="TO1"> </text_object> <!-- Text caption -->
</object_set>
<object_hierarchy>
        <object_node object_ref="O1">
                <object_node object_ref="AO1"/>
                <object_node object_ref="VO1"/>
                <object_node object_ref="TO1"/>
        </object_node>
</object_hierarchy>
```

5.3 Application: Multimedia Broadcast News System

We discuss the application of the proposed MMDS to the broadcast news domain. This domain is interesting because (1) there are distinct practical application scenarios for this; (2) the multimedia content in this domain contains interesting structures; and (3) it is closely related to the mission of MPEG-7 due to the fact that different broadcasters currently use different protocols to describe their content. At AT&T Labs-Research, a multimedia broadcast news browser addressing this domain has been implemented in Java. This browser is based on two unique features, first, use of integrated multimedia processing that extracts layers of content [19][7][8], and second, the use of integrated data description to represent the multimedia content [9].

We briefly discuss the description scheme of a multimedia news program in this system. A multimedia news program usually includes a video sequence, an audio sequence, and some text captions associated text information (from either closed caption or an automatic speech recognizer). A news program on a particular day is represented as a global multimedia (MM) object. Figure 10 shows the relationships among all the components, i.e. local MM objects, of this MM object. These relations are implemented using object hierarchies or entity-relation graphs, as needed. Each such MM object has two main components, the "ToC" (Table-Of-Content) object and the "Dialog" object. The "Dialog" object is in fact related to the "ToC" object. "Dialog" is an object that is not extracted from a news program but an object that acts like an intelligent and interactive agent based on its linguistic knowledge and the content in the "ToC". The "ToC" object is constructed through automated multi-modal processing by understanding the content in a news program [19][7] and consists of three sub-objects: "Summary-of-the-Day", "Headline Stories", and "Commercials". Further, both the "Headline stories" and "Commercials" sub-objects consist of a set of sub-components of the same type. For example, the "Headline Stories" represents a set of news stories.

In Figure 11 and Figure 12, we show how the MMDS interacts with individual single-media DSs by presenting the relationships among the components of MM objects "Story i" and "Dialog", respectively. As shown in Figure 11, "Story i" is composed of two parts, a "Story Summary", and a "Story Body". Further, each of these two parts may include one or more speakers with associated content embedded in different modalities. To capture this paradigm, we define "Speaker" as MM object because it is composed of several single-media

objects such as video, audio, and text. The single-media objects are shown shaded in Figure 11. Note that the single-media objects may also interact with each other.

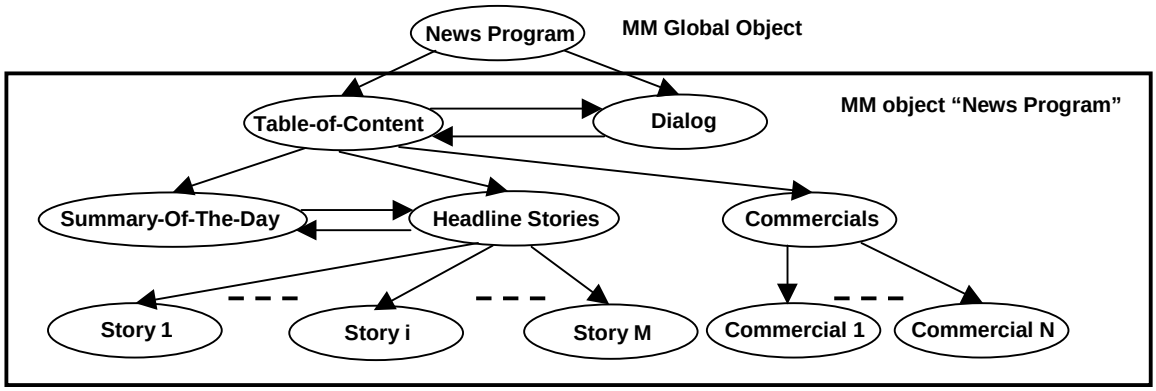


Figure 10: Relationships among components of the multimedia object "New Program".

The "Dialog" object is related not only to MM object "ToC" but also to several single-media objects. Figure 12 shows how object "Dialog" interacts with the single-media objects. Since object "Dialog" reacts to user's input (via audio or text) it has links to both audio and text objects. Similarly, since its direct output is text, it is directly related to object "Synthetic Text". Further, object "Synthetic Text" may be used to create object "Synthetic Audio" (e.g., text-to-speech) and object "Synthetic Video" (visual text-to-speech); these objects may link back to object "Dialog" to make up the integrated acoustic and visual interface during the dialog.

It can thus be seen that the MMDS can be used to generate a description of a complex MM object. This description basically provides the glue information to pull together a set of related multimedia and single-media objects, as well as their precise spatial and temporal relationships, among others.

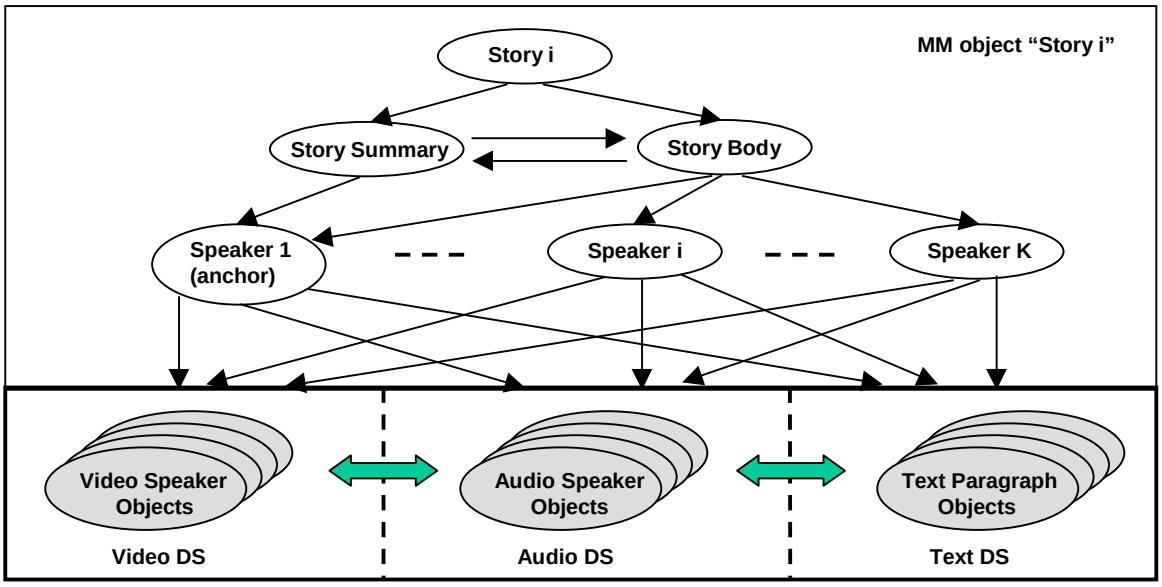


Figure 11: Relationships among the components of multimedia object "Story i".

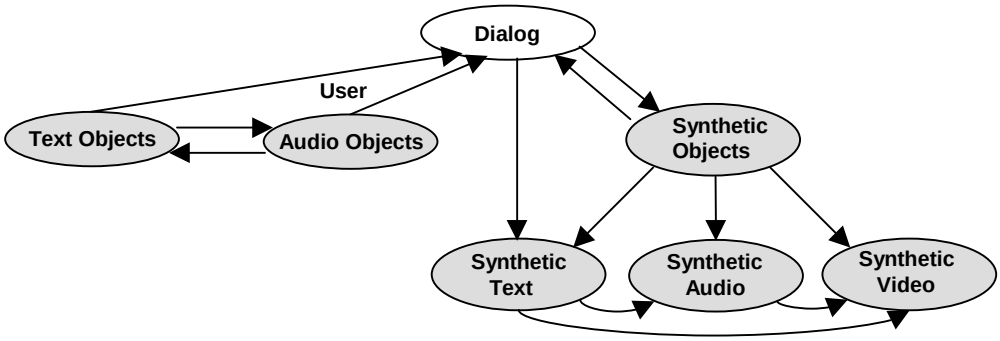


Figure 12: Relationships among the components of multimedia object "Dialog".

6. Home Media Description Scheme

In this section, we present the home media description scheme [3] developed for home media content. We considered home media content as an interesting scenario due to the vast amount of home media content generated in many households all around the world. To clarify the explanation, we use the example of home media content in the form of a video sequence[2] in Figure 13 (a). Figure 13 (b) shows the description of the home video sequence by the proposed home media DS.

The home media DS is derived from the image, the video, and the multimedia DSs. It includes the 6-Ws semantic features for the objects and 7 types of object hierarchies (1-P+6-W): 1-P physical and 6-W semantic category object hierarchies. Figure 14 summarizes the proposed scheme for the home media DS in UML notation. This is an example of how the generic description schemes for image, video, and multimedia content can be instantiated and furnished for a specific application and scenario. In remainder of this section, we will describe the 6-W semantic features and the 1-P+6-W object hierarchies. We will also discuss how a possible storytelling system could benefit from the use of the home media DS. The diagram in Figure 14 does not show the entity-relation graph element as one of the components of the Home Stream due to space limitation.

6.1 6-W Semantic Features

The home media DS describes home media content following the natural way in which humans tend to comment their home multimedia content (e.g. photographs and videos). Users often described specific sections of the content at a time (e.g. regions in an image) focussing in some more than others. In describing home content to others, authors talk about the people or animals ("who"), the objects ("what object"), the actions ("what action"), the location ("where"), the event ("why"), and/or the date ("when") that are relevant. We call these 6 semantic categories, the 6-Ws. In the home media DS, sections of the contents are represented as objects and the annotations in the 6-W categories as semantic features.

The home media content would be segmented into objects, which will be described at each of the 6-W semantic categories. For home video sequences, objects could be of three types, as described in section 4.1: GLOBAL, SEGMENT, or LOCAL. The annotations or relevant concepts at each of the 6-W semantic categories would be expressed as corresponding semantic features. "What action" annotations are special because they usually refer to the "who" or the "what object" entities involved in the action. For this reason, entity relations could be used to describe actions. We saw some examples of the 6-W semantic features in section 3.2.

As an example, the home media sequence in Figure 13 (a) has been segmented into temporal objects (TO). The first two temporal objects of the video sequence where captured during a trip to the lake on the boat.

[2] The example home video used was captured by one of the authors, Charlie Judice.

The following two temporal objects were shot during a trip to Arizona in a balloon fair. Finally, the last two temporal segments were recorded during an anniversary celebration with some relatives in New Jersey. In the video description, each temporal object is represented as a video object of type SEGMENT. The entire video sequence is assigned a video object of type GLOBAL. The temporal objects have been annotated in two of the 6-W categories: the "who" and the "what object" describing their relevant concepts. Below, we include the XML description of the TO 6 as annotated in Figure 13 (a).

```
<video_object type="SEGMENT" id="e6"…>
        <vid_obj_semantic_features>
                <who> <concept> Relatives </concept> </who>
                <where> <concept> House </concept> </where>
        </vid_obj_semantic_features>
</video_object> <!-- TO 6, Relatives in House -->
```

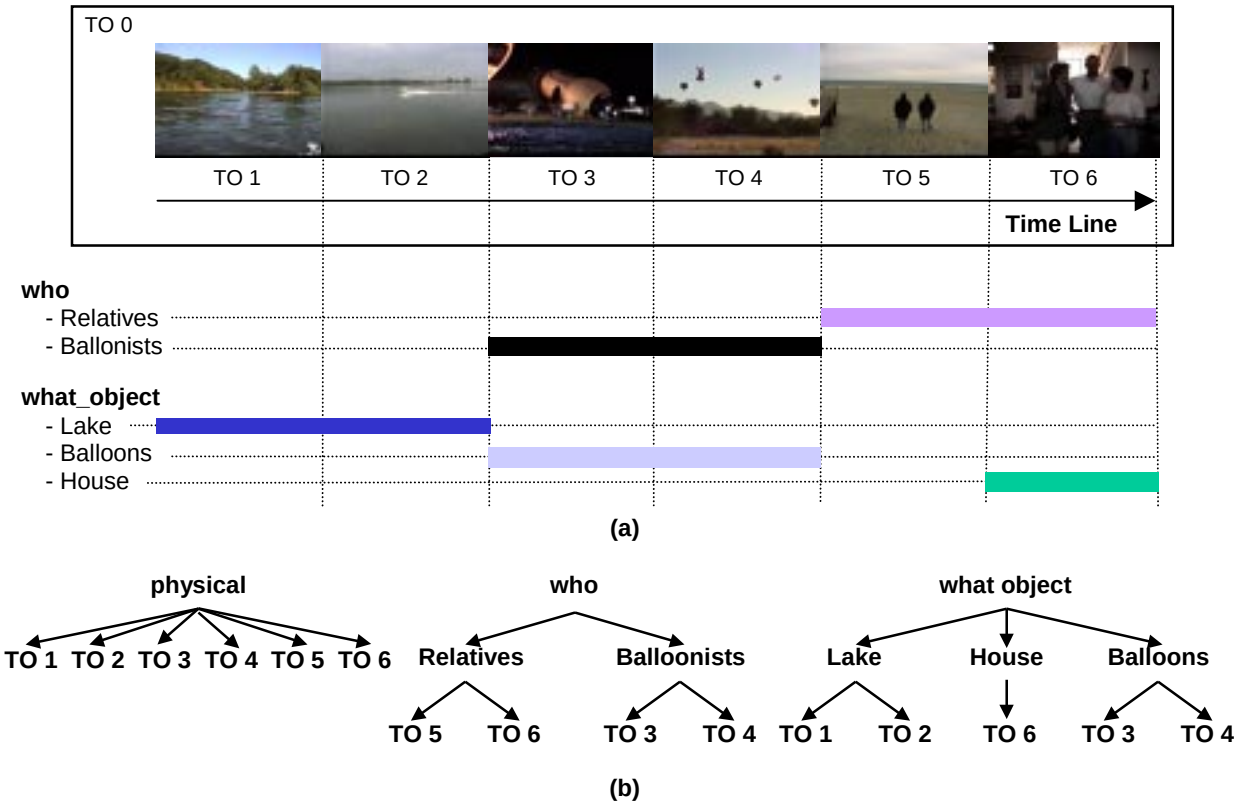


Figure 13: (a) Example of annotated home video sequence. (b) Examples of 1-P physical and 6-W semantic hierarchies.

6.2 1-P + 6-W Object Hierarchies

When describing home media content, humans describe the content in the 6-W categories. However, they also reuse previous concepts (e.g. same person in another shot) and relate concepts among each other (hierarchically, e.g. one person is another person's father; or non-hierarchically, e.g. one person is older than another is). We describe the relationships among concepts using 1-P + 6-W object hierarchies, and entity-relation graphs. We focus on the 1P + 6W object hierarchies in this section.

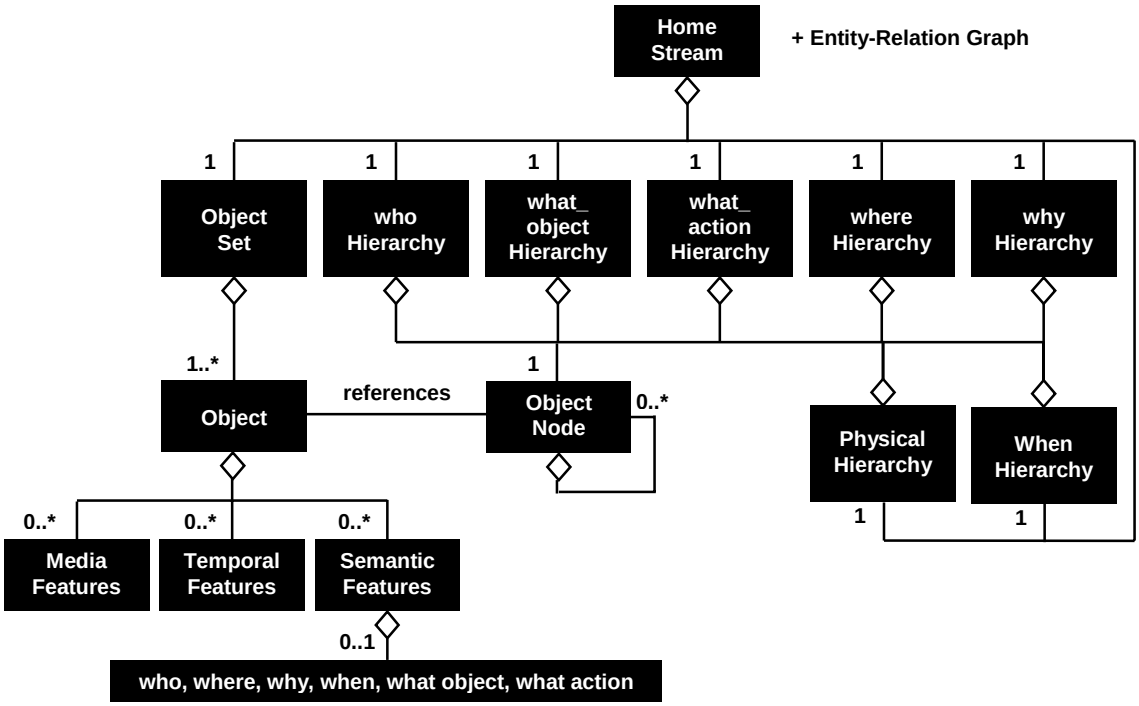


Figure 14: UML representation of home media description scheme.

Each home media description includes seven object hierarchies (1P+6W): a physical and six semantic hierarchies, one for each 6-W category; Figure 13 (b) shows the physical hierarchy and two semantic hierarchies, the “who” and the “what object”. The physical hierarchy describes the temporal and/or spatial organization of objects. The semantic hierarchy associated to each category represents the hierarchical relationships among the concepts and the objects in that category. The terminal nodes of all seven hierarchies are the objects resulting from the segmentation of the home media content (global, segment, and local object for video). Each intermediate node of the hierarchies is assigned a semantic concept object in the description (e.g. "Relatives"). Each root nodes of the hierarchies is assigned a semantic category object in the description (e.g. who semantic category).

The object set and the "who" semantic object hierarchy in Figure 13 are expressed in XML below. In this example, the terminal nodes of the object hierarchies are temporal objects. The objects corresponding to semantic concepts and semantic categories are assigned special types, SEMANTIC.CONCEPT and SEMANTIC.CATEGORY, respectively.

```
<video_object_set>
        <video_object type="GLOBAL" id="e0"…> </video_object> <!-- Home video sequence -->
        <video_object type="SEGMENT" id="e1"…> </video_object> <!-- TO1, Lake -->
        <video_object type="SEGMENT" id="e2"…> </video_object> <!-- TO2, Boat on lake -->
        <video_object type="SEGMENT" id="e3"…> </video_object> <!-- TO3, Inflating balloons-->
        <video_object type="SEGMENT" id="e4"…> </video_object> <!-- TO 4, Flying balloons-->
        <video_object type="SEGMENT" id="e5"…> </video_object> <!-- TO 5, Relatives on beach -->
        <video_object type="SEGMENT" id="e6"…> </video_object> <!-- TO 6, Relatives in House -->
        <video_object type="SEMANTIC.CATEGORY" id="e7"…> <!-- who semantic category -->
                <vid_obj_semantic_features> <who /> </vid_obj_semantic_features>
        </video_object>
        <video_object type="SEMANTIC.CONCEPT" id="e8"…> <!-- "Relatives" concept -->
```

```
                <vid_obj_semantic_features>
                        <who> <concept> Relatives </concept> </who>
                </vid_obj_semantic_features>
        </video_object>
        <video_object type="SEMANTIC.CONCEPT" id="e8"…> <!-- "Balloonists" concept -->
                <vid_obj_semantic_features>
                        <who> <concept> Balloonists </concept> </who>
                </vid_obj_semantic_features>
        </video_object>
        <!-- Other video objects would follow -->
</video_object_set>

<who_hierarchy>
        <object_node id="on1" object_ref="e7">
                <object_node id="on2" object_ref="e8">
                        <object_node id="on3" object_ref="e5" />
                        <object_node id="on4" object_ref="e6" />
                </object_node>
                <object_node id="on5" object_ref="e9">
                        <object_node id="on6" object_ref="e3" />
                        <object_node id="on7" object_ref="e4" />
                </object_node>
        </object_node>
</who_hierarchy>
```

Rather than being applied to single-media content, i.e. video sequences, the presented home media DS can also be applied to home multimedia streams composed of multiple media streams recorded during an event (e.g., video sequences, audio recordings, and still images taken during a wedding). Although developed for home media content, this description scheme could also be used for news, movies, and documentaries because of its generality.

6.3 *Application: Storytelling System*

In this section, we present a possible storytelling system using the above home media DS. The system would allow users to create stories as a collection of acts that, in turn, are a collection of shots. During the process of creation of a story, the user could assign brief descriptions to stories, acts, and shots. The media content, the descriptions, and the information recorded by the camera itself would be automatically processed and analyzed by the multimedia classification modules of the storytelling system. These modules would extract relevant information in the 6-W categories (who, what object, what action, where, when, why) and relate all the annotations of a story (e.g. identify the same concepts and organize concepts hierarchically).

We will review several multimedia classification systems that could be used in processing the user descriptions and media content to provide useful information in the 6-W dimensions. The Visual Apprentice [10], a model-based classification system, could be used to detect “what object” and “what action” (e.g. handshake, baseball hit scene, and the statue of liberty). In Lumine [16], a scene classification system that combines visual and textual features, could provide the “where” and the “what object” (e.g. indoor/outdoor, nature landscape, and city/suburb). An automatic face detection system [24] could generate useful information in the “who” dimension. Smart cameras with a GPS receiver and an internal clock would be able to provide the “where” (e.g. city and street) and the “when” (e.g. date and time) information.

This storytelling system would have the following representation for each story: a physical hierarchy describing the story as a collection of acts, and the acts as collections of shots; and 6-W semantic category hierarchies indexing the story, the acts, and the shots in the 6-W semantic dimensions. General relations among stories, acts, and shots could be described using entity-relation graphs. The storytelling system could provide the

user with retrieval capabilities of the created stories by typing a text description as a query. The natural language processing module of the system would be used to extract relevant information in the 6-W categories to match them against the 6-W semantic hierarchies and the entity-relation graphs of the stories. The 6-W indexing hierarchies and the entity-relation graphs used for the retrieval and search would actually be completely transparent to the user for simplicity and clarity.

The system design using the proposed home media DS is straightforward. It uses the physical object hierarchy, the 6-W semantic object hierarchies, and the entity-relation graphs to relate the video objects in a story, i.e., the story itself, the acts, and the shots. It is important to point out the flexibility and interoperability that our home DS provides. The content description of home video sequences is contained in a single text file. Therefore, editing the current descriptors, adding new ones, and searching for descriptors is as simple as modifying and matching words in a text file.

7. Archive Description Scheme

In this section, we present the archive description scheme [4]. The archive DS reuses elements defined in the image, the video, and the multimedia DSs. We consider a multimedia archive as a collection of multimedia documents (e.g. images and video sequences) that have already been described by the image, the video, or the multimedia DSs. Figure 15 (a) represents the video objects of all the video sequences in an archive in the feature space of a semantic subject feature (e.g. science, art, and history). Any semantic features (e.g. 6-Ws), visual features (e.g. color and texture), temporal features, media features, or combinations could have been selected.

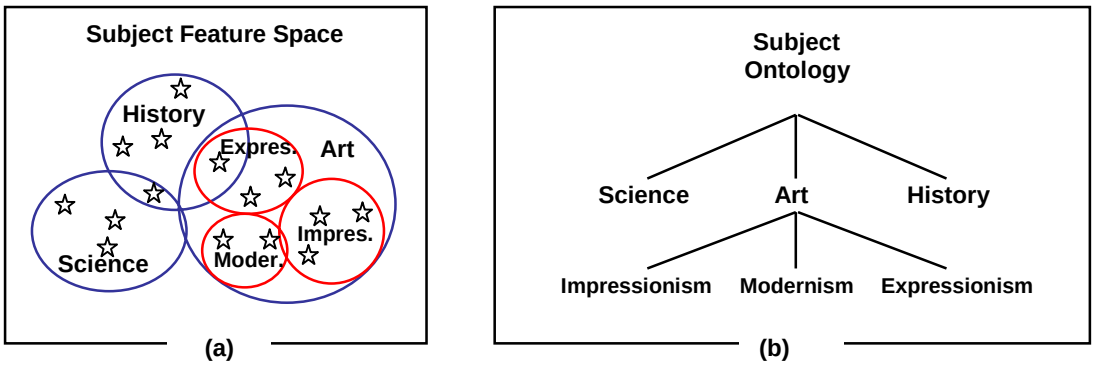


Figure 15: (a) Video objects in an archive on a subject-feature space. (b) Example of a subject ontology.

The UML representation of the archive DS is shown in Figure 16. The archive (<archive>) description is composed of one or more ontology elements (<ontology>). An ontology is a hierarchy of clusters (<cluster>) generated by constraints of one or more object features, see Figure 15 (b). General relations among clusters could be represented using entity-relation graphs. In the remaining part of this section we will focus on each the basic components of the archive DS: the cluster, the way clusters are related to individual multimedia documents, and the entity-relation graph. We will also describe the impact that such a standard archive DS could have in MetaSEEk [2], a meta search engine for audio-visual content.

7.1 Cluster

The basic element of the archive description scheme is the cluster. A cluster represents a collection of similar multimedia, image, and/or video objects that may belong to different multimedia documents. Clustering has been widely applied to efficiently search, retrieve, browse, and visualize collections of images, video sequences, and textural documents, among others. As an example, in the WebSEEk system [20], the feature clusters are based on semantic features. The statistical features of each cluster have been proved very useful for browsing. Also, in

MetaSEEk [2], feature clustering has been used to help selecting optimal image databases matching user interests. For example, image databases with matched color clusters will be searched first in response to a query with certain color histogram.

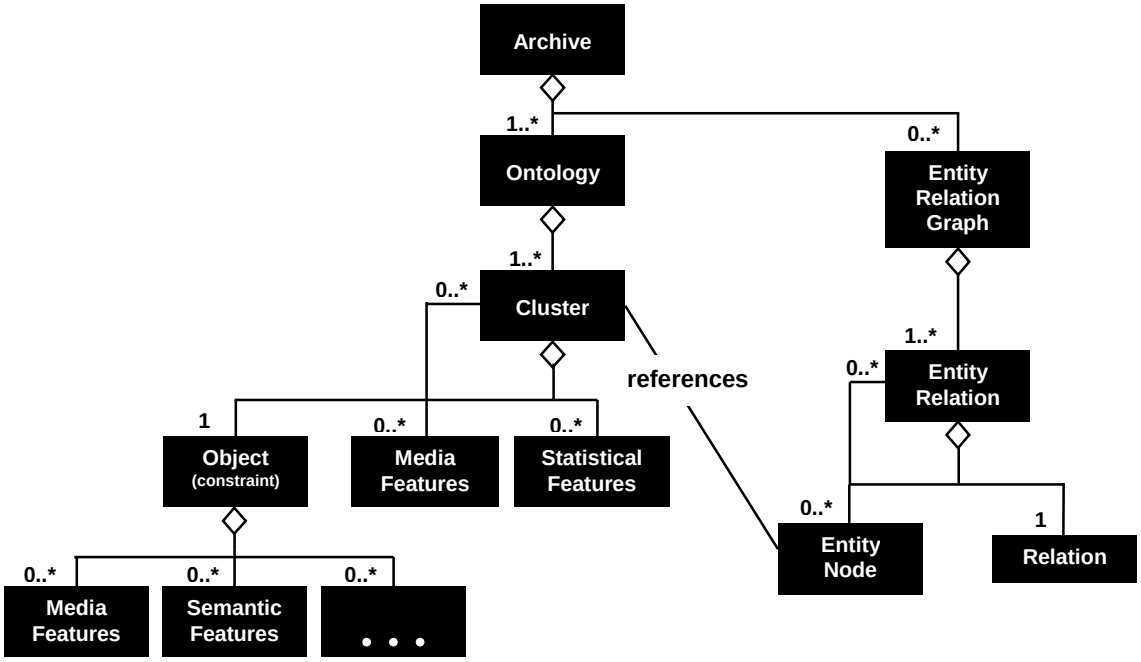


Figure 16: UML representation of proposed archive description scheme.

A cluster summarizes a set of multimedia, image, and/or video objects in a multimedia archive based on any combination of object features in the image, the video, and the multimedia DSs: media features (e.g. file format), visual features (e.g., color and texture), temporal features (e.g. duration or start time), and semantic features (e.g. subject and "who"). A cluster can include a unique identifier and a type attribute to describe the type of objects included in the cluster (e.g. image objects, video object, or both). Each cluster contains information about the requirements or constraints satisfied by its components (<object>), the cluster features (e.g. statistical and media features), and its internal clusters (see Figure 16). Examples of cluster feature classes and features are included in Table 7.

Table 7: Cluster feature classes and features.

Feature Class	Features
Media	Links to objects in the cluster, Representative icons, Size
Statistical	Distribution, N-Order Moments

Below we detail the XML description for the "Art" cluster in the ontology of Figure 15. The "Art" cluster groups video objects in the archive whose level 1 of a subject feature has an "Art" value. The subject feature could have different levels to describe the subject of the video object from less concrete to more concrete, e.g. art and impressionism, or history, American, XX Century, 90s, and 1999. The features associated with the cluster are (1) media features: links to objects in cluster, number of elements in the cluster, and representative icons; and (2) statistical features: distribution and N-order moments. The "Art" cluster also contains three internal clusters: "Impressionism", "Expressionism", and "Modernism".

```
<cluster_node id="C3" type="VIDEO_OBJECT"> <!-- Cluster: Art -->
        <object> <!-- Constraints on features of clustered elements -->
                <obj_semantic_features> <!--Weights can be associated with each feature -->
                        <subject>
                                <level1 weight="0.2"> Art </level1>
                        </subject>
                </obj_semantic_features>
        </object>
        <cluster_media_features> <!-- Cluster media features -->
                <!-- Pointers to files containing links to object in cluster and archive -->
                <links> <element_links href="http://links.to.objects.in.cluster_art.xml" /> </links>
                <size> <num_elements>45</num_elements> </size>
                <icons>
                        <location> <location_site href="http://icon.for.cluster3.gif" /> </location>
                </icons>
        </cluster_media_features>
        </cluster_media_features> <!-- Cluster media features -->
                <N-order_moments> … </ N-order_moments>          <distribution> </distribution>
        </cluster_media_features>
        <!-- Internal clusters -->
        <cluster_node id="C3.1" type="VIDEO_OBJECT"> </cluster_node> <!-- Cluster: Impress. -->
        <cluster_node id="C3.2" type="VIDEO_OBJECT"> </cluster_node> <!-- Cluster: Express. -->
        <cluster_node id="C3.3" type="VIDEO_OBJECT"> </cluster_node>  <!-- Cluster: Moder. -->
</cluster_node>
```

In this DS, the object element (<object>) describes the constraints on object descriptors satisfied by the cluster components. Constraints could be imposed in any descriptors included in the media, the visual, the temporal, and the semantic features of an object, and combinations. When more than one descriptor in combined to generate a cluster, weights can be associated with one of them. Below we give an example of a constraint over a Tamura texture descriptor.

$$(0.00 < \text{coarseness} < 0.55) \text{ AND } (0.2 < \text{contrast} < 0.34) \text{ AND } (0.67 < \text{directionality} < 0.76)$$

For efficient use of the content descriptions of the archive and the documents in the archive, links are provided to traverse from the objects in the image, the video, or the multimedia DSs to corresponding clusters in the archive DS, and vice versa. We could even assign probabilities to these links. When high-level semantic descriptors of audio-visual content are generated automatically, they may have confidence factors associated with them. This is the probability that one would assign to the links between the archive and the document descriptions.

7.2 Entity-Relation Graph

The hierarchical structure of clusters provided by the ontology in the archive DS is too restricted to describe the rich audio-visual content of an archive. Clusters can be related in non-hierarchical ways not supported by the ontology element. For example, consider the subject ontology introduced in Figure 15, under the "History" cluster there could be a "Historical Statues" cluster of "Modernism" style. We introduce the entity-relation graph in the archive DS to be able to implement such general relationships among clusters.

7.3 Application: MetaSEEk

We describe how multimedia meta search engines could benefit from our archive DS, our code downloading mechanism, and, in general, of MPEG-7. We are implementing our novel ideas in a new version of MetaSEEk [2], a meta search engine for mediation among multiple search engines for audio-visual information. MetaSEEk is designed to intelligently select and interface with multiple on-line image search engines by ranking their performance for different classes of user queries. The overall architecture of MetaSEEk is shown in Figure 17.

The three main components of the system are quite standard for meta search engines; they include the query dispatcher, the query translator, and the display interface.

Upon receiving a query, the dispatcher selects the target search engines to be queried by consulting the performance database at the MetaSEEk site. This database contains performance scores of past query successes and failures for each supported search engine. The query dispatcher only selects search engines that provide compatible capabilities with the user's query (e.g., color and keywords). The query translators, then, translate the user query to suitable scripts conforming to the interfaces of the selected search engines. Finally, the display component uses the performance scores to merge the results from each search engine, and displays them to the user. MetaSEEk evaluates the quality of the results returned by each search engine based on the user's feedback. This information is used to update the performance database.

The operation of MetaSEEk is currently very restrained to the interface limitations of current on-line search engines: (1) only support for query by example, by sketch, or by keyword, (2) results as a flat list of images (with similarity scores, in some cases), and (3) no available information about the features used in the retrieval or the content of the databases of the target search engines.

In the future, we envision multimedia search engines accepting not only queries by example, by sketch, or by keyword but also queries by MPEG-7 multimedia descriptions. Advantages of querying search engines by MPEG-7 descriptions are more efficient, more secure, and richer queries combining different types of features (e.g. semantic, visual, temporal, and media features). Queries could involve single multimedia documents or collections of multimedia documents, which would be described by the multimedia DS or the archive DS, respectively. Queries by archive description would allow efficient matching of the clusters and the statistical distribution of a multimedia collection in a selected feature space. The user interests are also more likely to be described as a collection of multimedia documents rather than one multimedia document.

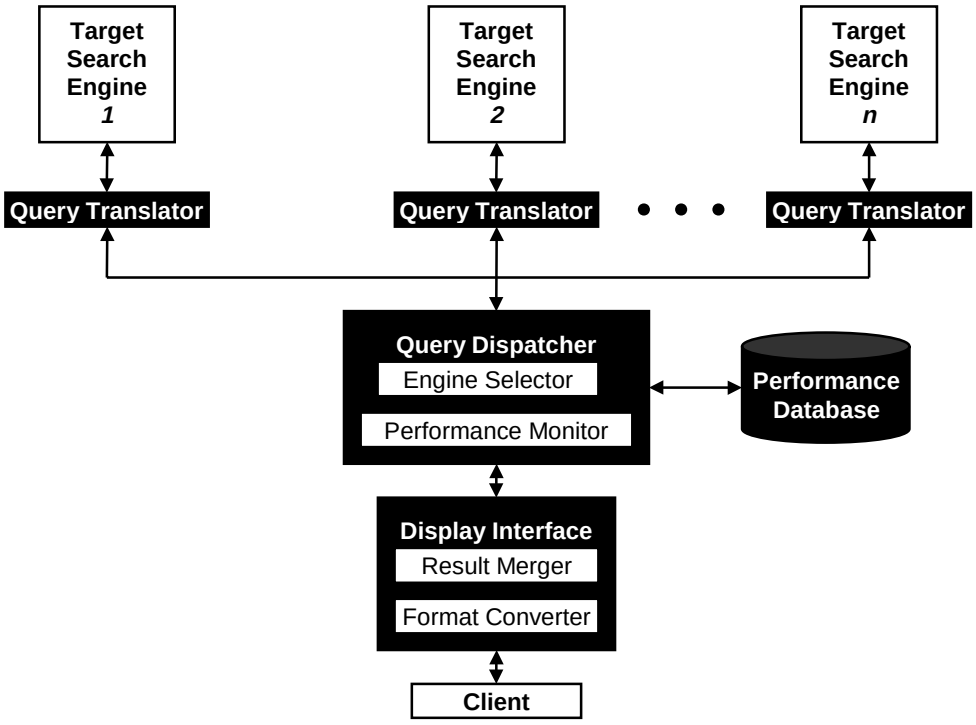


Figure 17: Overall architecture of MetaSEEk.

Furthermore, queries would result in a list of matched multimedia documents as well as their MPEG-7 descriptions (partial or complete) and, possible, their description as a collection using the archive DS. The organization and the statistics of clusters associated with the results would facilitate advanced display, indexing, and visualization of the results. This may be a very powerful feature when the number of results retrieved from a retrieval engine is very large.

Each search engine may also make available the archive description of its content and some proprietary code using our archive DS and code downloading mechanism, respectively. Meta search engines will use the archive descriptions and descriptor downloading capabilities of each target search engine to learn dynamically about the distribution of the content (hierarchical clusters and relationships among them) and the capabilities of each search engine. This knowledge will enable (1) advanced mapping of the user preferences to queries for each repository, (2) the generation of meaningful queries for each individual repository, (3) improved decisions to query search engines instead of others, (4) efficient ways of merging results from different repositories, and (5) intelligent visualization of the search results from heterogeneous sources. For example, a search engine would not be queried if the statistics of elements in the desired section of a feature space were not satisfactory (e.g. percentage of elements greater than some threshold). This information would be provided by the organization and the statistics associated with the clusters in the archive description. On the other hand, if the code to extract a specific feature was made available by a search engine, the meta search engine could run experiments with it to determine how it would map to other features or the user interests.

8. Conclusion and Future Work

We have presented our image, video, multimedia, home media, and archive description schemes. They have been designed based on an object-based multi-level framework. We use XML to describe our DSs and provide actual examples. However, it should be noted that our DSs could be generated by any DDL chosen by MPEG-7.

The presented description schemes are designed to address the requirements stated by MPEG-7 and have several unique aspects. Their structures and elements are currently being used to effectively support several large-scale multimedia search engines we have developed. They allow flexible inclusion of external feature descriptors and their proprietary code. We separate the object definitions and the object relational structures (i.e., the object hierarchy and the entity-relation graph) for clarity, flexibility, and generality allowing for distributed processing of the content. Finally, they provide effective methods for describing multiple abstraction levels and allowing modality transcoding.

The image, video, multimedia, and home media description schemes described in this paper were proposed to MPEG-7 in the Lancaster Meeting and received favorably by the designated evaluators [1]. Most basic components of our description schemes have been incorporated in the current draft of the Generic Visual DS [23] currently in development in MPEG-7: hierarchies, entity-relation graphs, feature categorization, and the 6-W semantic features. Furthermore, the archive DS was recommended to be directly included in the experimental model (XM) of the standard.

We are currently testing the proposed description schemes in our MPEG-7 testbed. We will incorporate several in-house prototypes for feature extraction and audio-visual applications, e.g., object-based video searching and content filtering. We are also developing the XML instantiation of the next-generation MetaSEEk search engine to test interoperability issues among different MPEG-7 applications.

References

[1] AHG on MPEG-7 Evaluation Logistics, “Report of the Ad-hoc Group on MPEG-7 Evaluation Logistics”, ISO/IEC JTC1/SC29/WG11 MPEG99/N4524, Seoul, Korea, March 1999.

[2] A. B. Benitez, M. Beigi, and S.-F. Chang, “A content-based image meta search engine using relevance feedback”, *IEEE Internet Computing*, Vol. 2, No. 4, pp. 59-69, Jul./Aug. 1998.

[3] A. B. Benitez, S. Paek, S.-F. Chang, C. Judice, and A. Puri, “Proposal for MPEG-7 home media description scheme”, Proposal to ISO/IEC JTC1/SC29/WG11 MPEG99/P479, Lancaster, U.K., Feb 1999.

[4] A. B. Benitez, S. Paek, S.-F. Chang, and C.-S. Li, “Proposal for MPEG-7 archive description scheme”, Proposal to ISO/IEC JTC1/SC29/WG11 MPEG99/P482, Lancaster, U.K., Feb 1999.
[5] S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatio-temporal queries", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 602-615, Sep. 1998; web site http://www.ctr.columbia.edu/videoq.
[6] Q. Huang, A. Puri, A. B. Benitez, S. Paek, and S.-F. Chang, “Proposal for MPEG-7 integration description scheme for multimedia content”, Proposal to ISO/IEC JTC1/SC29/WG11 MPEG99/P477, Lancaster, U.K., Feb 1999.
[7] Q. Huang, Z. Liu, and A. Rosenberg, “Automated Semantic Structure Reconstruction and Representation Generation for Broadcast News”, *Proc. SPIE Conference on Electronic Imaging: Storage and Retrieval of Images And Video Databases*, pp. 50-62, San Jose, CA, USA, Jan. 1999
[8] Q. Huang, Z. Liu, A. Rosenberg, D. Gibbon, and B. Shahraray, “Automated Generation of News Content Hierarchy by Integrating Audio, Video, and Text Information”, *Proc. Of International Conference on Acoustic, Speech, and Signal Processing*, Phoenix, March, 1999
[9] Q. Huang, A. Puri, and Z.Liu, “Multimedia Search and Retrieval: New Concepts, System Implementation, and Application”, to appear on *IEEE Transaction on Circuit Systems and Video Technology*, special issue on Object Based Coding and Description, 1999.
[10] A. Jaimes and S.-F. Chang, "Model-based classification of visual information for content-based retrieval", *Symposium on Electronic Imaging: Multimedia Processing and Applications - Storage and Retrieval for Image and Video Databases VII*, IS&T/SPIE'99, San Jose, CA, Jan. 1999.
[11] A. Lindsay, "Descriptor and description scheme classes", ISO/IEC JTC1/SC29/WG11 MPEG98/M4015 MPEG document, Atlantic City, NJ, Oct. 1998.
[12] R. Mohan, J. R. Smith, and C.-S. Li, “Adapting multimedia internet content for universal access”, *IEEE Transaction on Multimedia*, Vol. 1, No. 1, pp. 104-114, March 1999.
[13] S. Paek, A. B. Benitez, S.-F. Chang, C.-S. Li, J. R. Smith, L. D. Bergman, A. Puri, C. Swain, and J. Ostermann, “Proposal for MPEG-7 image description scheme”, Proposal to ISO/IEC JTC1/SC29/WG11 MPEG99/P480, Lancaster, U.K., Feb 1999.
[14] S. Paek, A. B. Benitez, S.-F. Chang, A. Eleftheriadis, A. Puri, Q. Huang , C.-S. Li, J. R. Smith, and L. D. Bergman, “Proposal for MPEG-7 video description scheme”, Proposal to ISO/IEC JTC1/SC29/WG11 MPEG99/P481, Lancaster, U.K., Feb 1999.
[15] S. Paek, A. B. Benitez, and S.-F. Chang, “Self-describing schemes for interoperable MPEG-7 multimedia content descriptions”, *Symposium on Electronic Imaging: Visual Communications and Image Processing*, IST/SPIE’99, San Jose, CA, Jan. 1999.
[16] S. Paek and S.-F. Chang, "In Lumine: A scene classification system for images and videos", ADVENT Project Technical Report #1998-03, Columbia University, Nov. 1998.
[17] Requirements Group, “MPEG-7 DDL Development Document V.2”, ISO/IEC JTC1/SC29/WG11 MPEG99/N2997, Melbourne, Australia, Oct. 1999.
[18] Requirements Group, “MPEG-7 Requirements Document”, ISO/IEC JTC1/SC29/WG11 MPEG99/N2727, Seoul, Korea, March 1999.
[19] B. Shahraray and D. Gibbon, “Automatic Generation of Pictorial Transcripts”, *Proc. SPIE Conference on Multimedia Computing and Networking*, SPIE 2417, pp. 512-518, San Jose, CA, USA, Feb. 1995.
[20] J. R. Smith and S.-F. Chang. “Searching for Images and Videos on the World-Wide Web”, Columbia Univeristy CTR Technical Report 459-96-25, August, 1996.
[21] H. Tamura, S. Mori, and T. Yamawaki, “Textural Features Corresponding to Visual Perception”, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 8, No. 6, Jun. 1978.
[22] UML Notation Guide Version 1.1, web site http://www.rational.com/uml.
[23] Video Group, “Generic visual description scheme for MPEG-7”, ISO/IEC JTC1/SC29/WG11 MPEG99/N2694, Seoul, Korea, March 1999.
[24] H. Wang and S.-F. Chang, "A highly efficient system for automatic face region detection in MPEG video sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, special issue on Multimedia Systems and Technologies, Vol. 7, No. 4, pp. 615-628, Aug. 1997.
[25] World Wide Web Consortium's (W3C) XML web site http://www.w3.org/XML.
[26] World Wide Web Consortium's (W3C) SMIL web site http://www.w3.org/AudioVideo/#SMIL.

[27] D. Zhong and S.-F. Chang, "Region feature based similarity matching of semantic video objects", to appear in ICIP'99, Kobe, Japan, Oct. 1999.

Authors' Vitae

Ana B. Benitez:

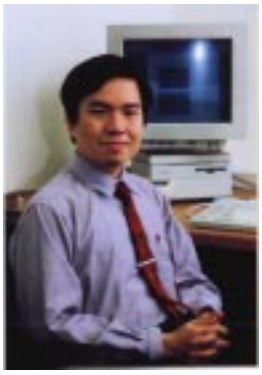
Ana B, Benitez is a Ph.D. candidate in the Department of Electrical Engineering at Columbia University, New York, USA, since 1996. She received her Telecommunications Engineer degree from the Polytechnic University of Catalonia (UPC) in Barcelona, Spain, in 1996. In 1996, she was awarded a full scholarship for graduate studies in the United States by the first Spanish financial institution, "la Caixa". She received her M. Phil. degree from Columbia University in 1996. At Columbia University, she has developed a meta search engine for mediation among multiple image search engines. Her current research interests include integration of large distributed audio-visual information retrieval systems and multimedia content representation. She is also an active participant in the MPEG-7 multimedia description standard. She is a student member of IEEE and ACM.

Seungyup Paek:

Seungyup Paek is a Ph.D. candidate in the Department of Electrical Engineering at Columbia University. His current research interests include image/video processing, artificial intelligence, and machine learning. He is currently involved in projects to develop on-line prototypes of visual information systems. He is also actively involved in the MPEG-7 international standardization activities.

Shih-Fu Chang:

Shih-Fu Chang is an Associate Professor of Electrical Engineering at Columbia University. His research focuses on multimedia content search and retrieval, digital image watermarking, digital libraries, and MPEG-7 applications. He is a co-PI of Columbia's ADVENT industrial consortium. He leads digital image research in several cross-disciplinary projects at Columbia, including Columbia's Health Care Digital Library, Multimedia Education Project supported by AT&T Foundation, and Columbia's Digital News System Project supported by NSF. He actively participates in international conferences and standardization efforts, such as MPEG-7. He has also served as a consultant in several companies. Prof. Chang has been awarded a Young Investigator Award from Office of Naval Research in 1998, a Faculty Development Award from IBM in 1995, and a CAREER Award from the National Science Foundation in 1995.



Atul Puri:

Aul Puri received his B.S. in Electrical Engineering from India in 1980, His M.S. in Electrical Engineering from the City College of New York in 1982, and his Ph.D., also in Electrical Engineering, from the City University of New York in 1988. While working on his dissertation, he was a consultant in Visual Communications Research Department of Bell labs and gained experience in developing algorithms, software and hardware for video communications. In 1988 he joined the same department at Bell labs as a Member of Technical staff. Since 1996, Dr. Puri has been a Principal Member of Technical Staff in Image Processing Research Department of AT&T Labs and is presently located at Red Bank, N.J..

Dr. Puri has represented AT&T at the Moving Pictures Experts Group Standard for past 10 years and has actively contributed towards development of the MPEG-1, the MPEG-2 and the MPEG-4 audio-visual coding standards. Currently he is participating in Video and Systems part of the MPEG-4 standard and is one of its technical editors. He has been involved in research in video coding algorithms for a number of diverse applications such as videoconferencing, video on Digital Storage Media, HDTV, and 3D-TV. His current research interests are in the area of flexible multimedia systems and services for web/internet. He is also participating in the MPEG-7 multimedia description standard.

Dr. Puri holds over 14 patents and has applied for another 8 patents. He has published over 30 technical papers in conferences and journals, including several invited papers. He is a co-author of a book entitled "Digital Video: An Introduction to MPEG-2." He is currently co-editing a book on Multimedia Systems. He has been the recipient of exceptional contribution and individual performance merit awards of AT&T. Furthermore, he has also received awards from the AT&T Communications Services and AT&T Technical Journal. He has taught graduate courses on Image and Video coding at Columbia University. Dr. Puri is a member of IEEE, its Communication, and Signal Processing societies, and is currently an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology.

Qian Huang:

Qian Huang received the B.S. degree in computer engineering in 1984 from Jiao Tong University in Shanghai, China, M.S. degree in computer science in 1989 from Kansas State University, and Ph.D. degree in computer science from Michigan State University in 1994, respectively. From 1994 to 1995, she was a postdoctoral fellow at Almaden Research Center of IBM Research Division, working on QBIC (Query By Image Example) project, particularly on object segmentation using minimum description length principle. In 1995, she joined the Imaging Department of Siemens Corporate Research where she worked on medical imaging, motion compensated OCR for text on planar objects, and automated cameraman for content based video data acquisition. Since October 1997, she has been with the Multimedia Processing Department of AT&T Labs – Research. Her current research interests include integrated audio/video/text processing for multimedia content extraction, automated semantics detection from multimedia data, and intelligent man machine interfaces.
She has published over 40 technical papers in conferences and journals, including several invited book chapters and papers. She has 12 patents pending.

Dr. Huang is a recipient of NSF Award for Women Professional in Federated Computing Research and a member of Phi Beta Delta, Phi Kappa Phi, and Upsilon Pi Epsilon.

John R. Smith:

John R. Smith is a research staff member at the IBM T. J. Watson Research Center, Hawthorne, New York, USA. He received his M. Phil and PhD. degrees in Electrical Engineering from Columbia University in 1994 and 1997, respectively. At Columbia, he developed several image and video search and retrieval systems, including the WebSEEk image and video search engine, the VisualSEEk content-based image retrieval system and the SaFe integrated spatial and feature image system. More recently at IBM, he has developed a progressive video retrieval system called VideoZoom, and the SFGraph framework for adaptive compression, access and retrieval of large images, high-resolutions documents and maps.

His research interests include multimedia and multi-dimensional data management, content-based visual query, image and video coding, on-line analytic processing, universal multimedia access and Web search engines. He is currently participating in the MPEG-7 standardization effort in the area of visual content description.

In 1997, he received the Eliahu I. Jury award from Columbia University for outstanding achievement as a graduate student in the areas of systems communication or signal processing. Dr. Smith is a member of IEEE.

Chung-Sheng Li:

Chung-Sheng Li received his B.S.E.E. degree from National Taiwan University, Taiwan, R.O.C. in 1984, and the M.S. and Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley in 1989 and 1991, respectively. He has joined the computer science division of IBM T. J. Watson Research Center as a research staff member since Sept., 1991, and manages the Image Information System Department since 1996.

His research interests include (1) Broadband applications, which include digital library, knowledge discovery and data mining; (2) Broadband network and switching, which includes all-optical networks, storage area networks, and fiber channel; (3) Broadband technologies, which include optical chip interconnects, optoelectronics, and high-speed analog/digital VLSI circuit design. He has co-initiated several research activities in IBM on fast tunable receiver for all-optical networks and content-based retrieval in the compressed domain for large image/video databases. He is currently the principle investigator of a satellite image database project funded by NASA.

Dr. Li has received a Research Division award from IBM in 1995 for his major contribution to the tunable receiver design for WDMA, and numerous invention and patent application awards. He is serving as the technical editor and feature editor for the IEEE Communication Magazine. He has authored or coauthored more than 100 journal and conference papers and received one of the best paper awards from the IEEE International Conference on Computer Design in 1992. He is a senior member of the IEEE Laser Electro-Optic Society, the Communication Society, the Computer Society, and the Circuit and System Society.

Lawrence D. Bergman:

Dr. Bergman is a graduate of the University of North Carolina, Chapel Hill, where he received his M.S and PhD in Computer Science. He is currently a Research Staff Member at the IBM T.J. Watson Research Center, where is working on developing content-based search technology for remote-sensing and other applications. Other research interests include computer graphics, programming languages, and human-computer interfaces.

Charlie Judice:

Dr. Judice conducted and led research in the areas of multimedia communications, digital photography, and image processing. He was one of the early pioneers of digital half-toning techniques, multimedia networked applications, and office automation. More recently helped create the MPEG standards initiative, championed ADSL as a telco solution to the last mile, and defined Kodak's next big opportunity in digital storytelling. He is currently a Research Fellow at Eastman Kodak's lab in Rochester, NY where he is developing the architecture for Kodak's next generation imaging backbone network. Prior to joining Kodak he held numerous research management positions at Telcordia (formerly Bellcore), Bell Atlantic, and Bell Labs.

Dr. Judice is a Fellow of the IEEE, COMSOC editor to Multimedia Magazine, chairman of Globecom 99 Multimedia Symposium. Dr. Judice is the founder of several workshops and centers including: Packet Video Workshop, Workshop on Speech Application over the Telephone Network, and Center for Networked Multimedia. He has over 50 publications and holds 10 patents including a recent one integrating web browsing with real time telephony.

Figure Legends

Table Legends