Self-Describing Schemes for Interoperable MPEG-7 Multimedia Content Descriptions

Seungyup Paek, Ana B. Benitez, and Shih-Fu Chang[1]

Image & Advanced TV Lab, Department of Electrical Engineering
Columbia University, 1312 S.W. Mudd, Mail code 4712 Box F-4
New York, NY 10027, USA

ABSTRACT

In this paper, we present the self-describing schemes for interoperable image/video content descriptions, which are being developed as part of our proposal to the MPEG-7 standard. MPEG-7 aims to standardize content descriptions for multimedia data. The objective of this standard is to facilitate content-focused applications like multimedia searching, filtering, browsing, and summarization. To ensure maximum interoperability and flexibility, our descriptions are defined using the eXtensible Markup Language (XML), developed by the World Wide Web Consortium. We demonstrate the feasibility and efficiency of our self-describing schemes in our MPEG-7 testbed. First, we show how our scheme can accommodate image and video descriptions that are generated by a wide variety of systems. Then, we present two systems being developed that are enabled and enhanced by the proposed approach for multimedia content descriptions. The first system is an intelligent search engine with an associated expressive query interface. The second system is a new version of MetaSEEk, a metasearch system for mediation among multiple search engines for audio-visual information.

Keywords: MPEG-7, self-describing scheme, interoperability, audio-visual content description, visual information system, metasearch, XML.

1. INTRODUCTION

It is increasingly easier to access digital multimedia information. Correspondingly, it has become increasingly important to develop systems that process, filter, search and organize this information, so that useful knowledge can be derived from the exploding mass of information that is becoming accessible. To enable exciting new systems for processing, searching, filtering and organizing multimedia information, it has become clear that an interoperable method of describing multimedia content is necessary. This is the objective of the emerging MPEG-7 standardization effort.

In this paper, we first give a brief overview of the objectives of the MPEG-7 standard. MPEG-7 aims at the standardization of content descriptions of multimedia data. The objectives of this standard are to facilitate content-focused applications like multimedia searching, filtering, browsing, and summarization.

Then, we present self-describing schemes for interoperable image/video content descriptions, which are being developed as part of our proposal to MPEG-7. To ensure maximum interoperability and flexibility, our descriptions use the eXtensible Markup Language (XML), developed by the World Wide Web Consortium. Under the proposed self-describing schemes, an image is represented as a set of relevant objects that are organized in one or more object hierarchies. Similarly, a video is viewed as a set of relevant events that can be combined hierarchically in one ore more event hierarchies. Both, objects and events, are described by some feature descriptors that can link to external extraction and similarity code.

Finally, we demonstrate the feasibility and efficiency of our self-describing schemes in our MPEG-7 testbed. In our testbed, we will show how our scheme can accommodate image and video descriptions that are generated by a wide variety of systems. In addition, we introduce two systems being developed, which are enabled and enhanced by our approach for multimedia content descriptions. The first system is an intelligent search engine with an associated expressive query interface. The second system is a metasearch system for mediation among multiple search engines for audio-visual information.

1. Email: {syp, ana, sfchang}@ee.columbia.edu; WWW: http://www.ee.columbia.edu/~{syp, ana, sfchang}/

2. MPEG-7 STANDARD AND SCENARIOS

2.1. MPEG-7 standard

The MPEG-7 standard [14] has the objective of specifying a standard set of descriptors to describe various types of multimedia information. MPEG-7 will also standardize ways to define other descriptors as well as Description Schemes (DSs) for the structure of descriptors and their relationships. This description (i.e. the combination of descriptors and description schemes) will be associated with the content itself to allow fast and efficient searching for material of a user's interest. MPEG-7 will also standardize a language to specify description schemes, i.e. a Description Definition Language (DDL), and the schemes for encoding the descriptions of multimedia content.

2.2. MPEG-7 scenarios

MPEG-7 will improve existing applications and enable completely new ones. We will review three of the most relevantly impacted application scenarios [3]: distributed processing, exchange, and personalized viewing of multimedia content.

- **Distributed processing**

MPEG-7 will provide the ability to interchange descriptions of audio-visual material independently of any platform, any vendor, and any application, which will enable the distributed processing of multimedia content. This standard for interoperable content descriptions will mean that data from a variety of sources can be plugged into a variety of distributed applications such as multimedia processors, editors, retrieval systems, filtering agents, etc. Some of these applications may be provided by third parties, generating a sub-industry of providers of multimedia tools that can work with the standard descriptions of the multimedia data.

The vision of the near future is one in which a user can access various content providers' web sites to download content and associated indexing data, obtained by some low-level or high-level processing. The user can then proceed to access several tool providers' web sites to download tools (e.g. Java applets) to manipulate the heterogeneous data descriptions in particular ways, according to the user's personal interests. An example of such a multimedia tool will be a video editor. A MPEG-7 compliant video editor will be able to manipulate and process video content from a variety of sources if the description associated with each video is MPEG-7 compliant. Each video may come with varying degrees of description detail such as camera motion, scene cuts, annotations, and object segmentations.

- **Content exchange**

A second scenario that will greatly benefit from an interoperable content-description standard is the exchange of multimedia content among heterogeneous audio-visual databases. MPEG-7 will provide the means to express, exchange, translate, and reuse existing descriptions of audio-visual material.

Currently, TV broadcasters, radio broadcasters, and other content providers manage and store an enormous amount of audio-visual material. This material is currently described manually using textual information and proprietary databases. Describing audio-visual material is an expensive and time-consuming task, so it is desirable to minimize the re-indexing of data that has been processed before.

Consider a media company that purchases videos from a TV broadcaster. The TV broadcaster has already described and indexed the content in their proprietary description scheme. Without an interoperable content description, the purchasing company will have to invest manpower to translate manually the description of the broadcaster into their proprietary scheme. Interchange of multimedia content descriptions would be possible if all the content providers embraced the same scheme and system. As this is unlikely to happen, MPEG-7 proposes to adopt a single industry-wide interoperable interchange format that is system and vendor independent.

- **Customized views**

Finally, multimedia players and viewers compliant with the multimedia description standard will provide the users with innovative capabilities such as multiple views of the data configured by the user. The user could change the display's configuration without requiring the data to be downloaded again in a different format from the content broadcaster.

The ability to capture and transmit semantic and structural annotations of the audio-visual data, made possible by MPEG-7, greatly expands the range of possibilities for client-side manipulation of the data for displaying purposes. For example, a browsing system can allow users to quickly browse through videos if they receive information about their corresponding semantic structure. For example, when modeling a tennis match video, the viewer can choose to view only the third game of the second set, all the overhead smashes made by one player, etc.

These examples only hint at the possible uses that creative multimedia-application designers will find for richly structured data delivered in a standardized way based on MPEG-7.

3. SELF-DESCRIBING SCHEMES

In this section, we present description schemes for interoperable image/video content descriptions. The proposed description schemes are self-describing in the sense that they combine the data and the structure of the data in the same format. The advantages of such a type of descriptions are flexibility, easy validation, and efficient exchange.

3.1. eXtensible Markup Language (XML)

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. SGML makes it possible to define your own formats for your own documents, to handle large and complex documents, and to manage large information repositories. 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 [21]. The goal of the W3C SGML activity is to enable the delivery of self-describing data structures of arbitrary depth and complexity to applications that require such structures. The first phase of this effort is the specification of a simplified subset of SGML specially designed for Web applications. 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.

Before describing the image and video DSs, we present some of the core features of XML. Let's start with a simple XML element:

```
<image> Hello MPEG-7 world! </image>
```

<image> is the start tag and </image> is the end tag; Hello MPEG-7 world! is the content of the element. What does the image tag mean? In short, it means anything you want it to mean. XML predefines no tags at all. Rather than relying on a few hundred predefined tags, XML lets you create the tags you need to describe your data. Users define what is allowed in each document by providing rules, collectively known as the Document Type Definition (DTD). The DTD states the element types with their characteristics, the notations, and the entities allowed in the document. Apart from the DTD, XML documents must follow some basic well-form rules. This is the minimum criterion for XML parsers and processors.

Text in XML documents consists of characters. A document's text is divided into character data and markup. In a first approximation, markup describes a document's logical structure while character data is the basic content of the document. Generally, anything inside a pair of <> angle brackets is markup and anything that is not inside these brackets is character data. Start tags and empty tags may optionally contain attributes. An attribute is a name-value pair separated by an equal sign. Work is in progress to include binary data in XML tags. Currently, XML allows defining binary entities pointing to binary data (e.g. images). They require an associated notation describing the type of resource (e.g. GIF and JPG).

3.2. Image description scheme

In this section, we present the proposed description scheme for images. To clarify the explanation, we will use the example shown in Figure 1. Using this example, we will walk through the image DS expressed in XML. Along the way, we will explain the use of various XML elements that are defined for the proposed image DS. The complete set of rules of the tags in the image and video description schemes is defined in our document type definitions [15]. Another advantage of using XML as the DLL is that it provides the capability to import external description schemes' DTDs to incorporate them in one description by using namespaces. We will see an example later in this section.

The basic description element of our image description scheme is the object element (<object>). An object element represents a region of the image for which some features are available. There are two different types of objects: physical and logical objects. Physical objects usually correspond to continuous regions of the image with some descriptors in common (semantics, features, etc.) - in other words, real objects in the image. Logical objects are groupings of objects based on some high-level semantic relationships (e.g. faces). The object element comprises the concepts of group of objects, objects, and regions in the visual literature. The set of all objects identified in an image is included within the object set element (<object_set>).

For the image example of Figure 1.a, we have chosen to describe the objects listed below. Each object element has a unique identifier within an image description. The identifier is expressed as an attribute of the object element (id). Another attribute of the object element (type) distinguishes between physical and logical objects. We have left the content of each object element empty to show clearly the overall structure of the image description. Later in the section, we will describe the features that can be included within the object element.

```
<object_set>
    <object id="0" type="PHYSICAL" > </object> <!-- Family portrait -->
    <object id="1" type="PHYSICAL" > </object> <!-- Father -->
    <object id="2" type="PHYSICAL" > </object> <!-- Mother -->
    <object id="3" type="LOGICAL" > </object> <!-- Faces -->
    <object id="4" type="PHYSICAL" > </object> <!-- Father's face -->
    <object id="5" type="PHYSICAL" > </object> <!-- Mother's face -->
</object_set>
```

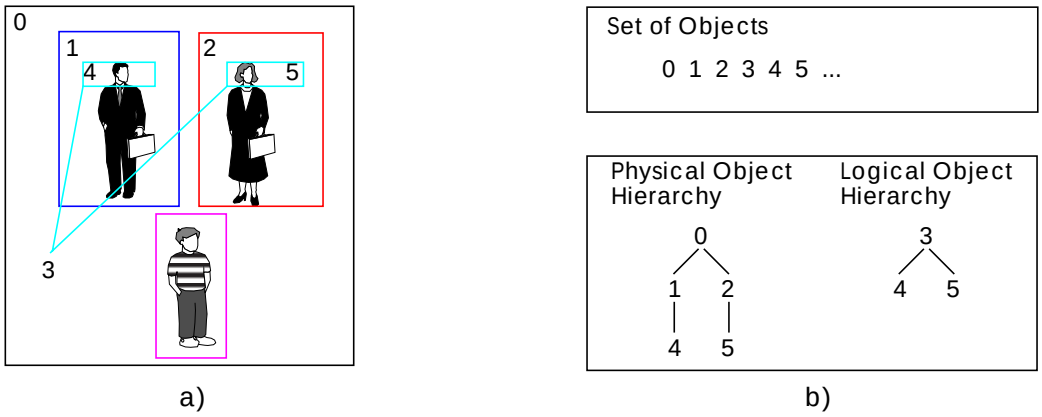


Figure 1: a) Image example. b) High-level description of the image by proposed image description scheme.

The image description scheme is comprised of object elements that are combined hierarchically in one or more object hierarchy elements (<object_hierarchy>). The hierarchy is a way to organize the object elements in the object set element. Each object hierarchy consists of a tree of object node elements (<object_node>). Each object node points to an object. The objects in an image can be organized by their location in the image or by their semantic relationships. These two ways to group objects generate two types of hierarchies: physical and logical hierarchies. A physical hierarchy describes the physical location of the objects in the image. On the other hand, a logical hierarchy organizes the objects based on a higher level understanding of their semantics, similar to semantic clustering.

Continuing with the image example in Figure 1.a, two possible hierarchies are shown in Figure 1.b. These hierarchies are expressed in XML below. The type of hierarchy is included in the object hierarchy element as an attribute (type). The object node element has associated a unique identifier in the form of an attribute (id). The object node element references an object element by using the latter's unique identifier. The reference to the object element is included as an attribute (object_ref). An object element can include links back to nodes in the object hierarchy as an attribute too (object_node_ref).

```
<object_hierarchy type="PHYSICAL"> <!-- Physical hierarchy -->
    <object_node id="10" object_ref="0"> <!-- Portrait -->
        <object_node id="11" object_ref="1"> <!-- Father -->
            <object_node id="12" object_ref="4"/> <!-- Father's face -->
        </object_node>
        <object_node id="13" object_ref="2"> <!-- Mother -->
            <object_node id="14" object_ref="5"/> <!-- Mother's face -->
        </object_node>
    </object_node>
</object_hierarchy>
<object_hierarchy type="LOGICAL"> <!-- Logical hierarchy: faces in the image -->
    <object_node id="15" object_ref="3"> <!-- Faces -->
```

```
        <object_node id="16" object_ref="4"/> <!-- Father's face -->
        <object_node id="17" object_ref="5"/> <!-- Mother's face -->
    </object_node>
</object_hierarchy>
```

An object set element and one or more object hierarchy elements form the image element (<image>). The image element symbolizes the image or picture being described.

In our image description scheme, the object element contains the feature elements; they include location, color, texture, shape, size, motion, time, and annotation elements, among others. Time and motion descriptors with have sense when the object belongs to a video sequence. The location element contains pointers to the locations of the image. Note that annotations can be textual, visual or audio. These features can be extracted or assigned automatically or manually. For those features extracted automatically, the feature descriptors can include links to external extraction and similarity matching code. An example is included below. This example also shows how external DSs can be imported and combined with ours.

```
<object id="4" type="PHYSICAL" object_node_ref="12 16"> <!-- Father's face -->
    <color> </color>
    <texture>
        <tamura>
            <tamura_value coarseness="0.01" contrast="0.39" orientation="0.7"/>
            <code type="EXTRACTION" language="JAVA" version="1.2"> <!-- Link to extraction code -->
                <location> <location_site href="ftp://extraction.tamura.java"/> </location>
            </code>
        </tamura>
    </texture>
    <shape> </shape>
    <position> </position>
    <!-- Import and use of external annotation DS's DTD -->
    <text_annotation xmlns:extAnDS="http://www.other.ds/annotations.dtd">
        <extAnDS:Class>Face</extAnDS:Class>
     </text_annotation>
</object>
```

In summary, both, the object hierarchy and object set elements, are part of the image element (<image>). The objects in the object set are combined hierarchically in one or more object hierarchy elements. For efficient transversal of the image description, links are provided to traverse from objects in the object set to corresponding object nodes in the object hierarchy and viceversa. The objects include various feature descriptors that can link to external extraction and similarity matching code.

3.3. Video description scheme

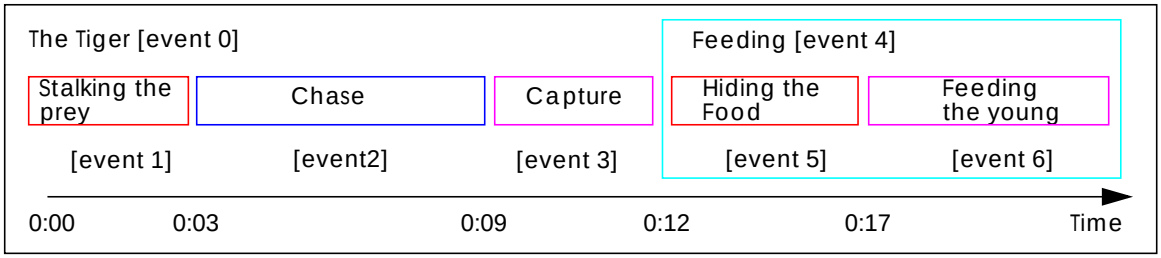
In this section, we present the proposed description scheme (DS) for videos. To clarify the explanation, we will use the example shown in Figure 2. Using this example, we will walk through the video DS expressed in XML. Along the way, we will explain the use of various XML elements that are defined for the proposed MPEG-7 video DS. The structure of the image description scheme and the video description scheme are very similar.

The basic description element of our video description scheme is the event element (<event>). An event represents one or more shots of the video for which some features are available. We distinguish three different types of events: a shot, a continuous group of shots, and a discontinuous group of shots. Discontinuous group of shots will usually be associated together based on common features (e.g. background color) or high-level semantic relationships (e.g. actor on screen). The event element comprises the concepts of story, scene, and shot in the visual literature. The set of all events identified in a video is included within the event set element (<event_set>).

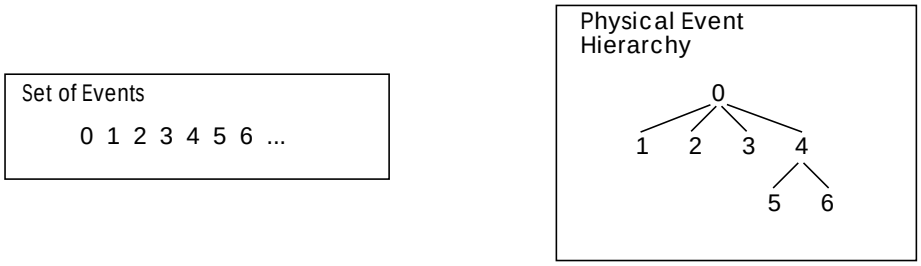
For the video example of Figure 2.b, we have chosen to describe the events listed below. Each event element has a unique identifier within a video description. The identifier is expressed as an attribute of the event element (id). Another attribute of the event element (type) distinguishes between the three different types of events. We have left each event element empty to show clearly the overall structure of the video description. Later in the section, we will describe the features that can be included within the event element.

```
<event_set>
    <event id="0" type="SHOT" > </event> <!-- The tiger -->
    <event id="1" type="SHOT" > </event> <!-- Stalking the prey -->
```

```
    <event id="2" type="SHOT"  >  </event> <!-- Chase -->
    <event id="3" type="SHOT"  >  </event> <!-- Capture -->
    <event id="4" type="CONTINUOUS_GROUP_SHOTS"  >  </event> <!-- Feeding -->
    <event id="5" type="SHOT"  >  </event> <!-- Hiding the food -->
    <event id="6" type="SHOT"  >  </event> <!-- Feeding the young -->
</event_set>
```



a)



b)

Figure 2: a) Video example. b) High-level description of the video by proposed video description scheme.

The video description scheme is comprised of event elements that are combined hierarchically in one or more event hierarchy elements (<event_hierarchy>). The hierarchy is a way to organize the event elements in the event set element. Each event hierarchy consists of a tree of event node elements (<event_node>). Each event node points to an event. The event in a video can be organized by their location in the video or by their semantic relationships. These two ways to group events generate two types of hierarchies: physical and logical hierarchies. A physical hierarchy describes the time composition of the events in the video. On the other hand, a logical hierarchy organizes the events based on a higher level understanding of their semantics, similar to semantic clustering.

Continuing with the video example in Figure 2.a, one possible hierarchy is shown in Figure 2.b. The corresponding XML is below. The type of hierarchy is included in the event hierarchy element as an attribute (type). The event element has associated a unique identifier as an attribute (id). The event node element references an event element by using the latter's unique identifier. The reference to the event element is included as an attribute (event_ref). An event element can include links back to nodes in the event hierarchy to jump between events and event nodes in both directions (event_node_ref).

```
<event_hierarchy type="PHYSICAL">
    <event_node id="10" event_ref="0"> <!-- The Tiger -->
        <event_node id="11" event_ref="1"/> <!-- Stalking the prey -->
        <event_node id="12" event_ref="2"/> <!-- Chase -->
        <event_node id="13" event_ref="3"/> <!-- Capture -->
        <event_node id="14" event_ref="4"> <!-- Feeding -->
            <event_node id="15" event_ref="5"/> <!-- Hiding the food -->
            <event_node id="16" event_ref="6"/> <!-- Feeding the young -->
        </event_node>
    </event_node>
</event_hierarchy>
```

An event set element and one or more even hierarchy elements form the video element (<video>). The video element symbolizes the video sequence being described.

In our video description scheme, the event element contains the feature elements; they include location, shot transition (i.e. various within shot or across shot special effects), camera motion, time, key frame, annotation and object set elements, among others. The object element is defined in the image description scheme; it represents the relevant objects in the event. As in the image DS, these features can be extracted or assigned automatically or manually. For those features extracted automatically, the feature descriptors can include links to extraction and similarity matching code. For example,

```
<event id="3" type="PHYSICAL" event_node_ref="10"> <!-- Capture -->
    <object_set> </object_set>
    <camera_motion>
        <backgroun_affine_model>
            <background_affine_motion_value>
                <panning direction="NE"/>
                <zoom direction="IN"/>
            </background_affine_motion_value>
            <code type="DISTANCE" language="JAVA" version="1.0"> <!-- Link to similarity matching code -->
                <location> <location_site href="ftp://dist.bacground.affine"/> </location>
            </code>
        </background_affine_model>
    </camera_motion>
    <time> </time>
</event>
```

In summary, both, the event hierarchy and event set elements, are part of the video element (<video>). The event elements in the event set element are combined hierarchically in one or more event hierarchy elements. For efficient transversal of the video description, links are provided to traverse from events in the event set to corresponding event nodes in the event hierarchy and viceversa. The events include various feature descriptors that can link to extraction and similarity matching code.

4. MPEG-7 TESTBED

The proposed self-describing schemes are intuitive, flexible, and efficient. We demonstrate the feasibility of our self-describing schemes in our MPEG-7 testbed. In our test bed, we are using the self-describing schemes for descriptions of images and videos that are generated by a wide variety of systems we have developed. We are developing two systems that are enabled and enhanced by our approach for multimedia content descriptions. The first system is an intelligent search engine with an associated expressive query interface. The second system is a new version of MetaSEEk, a metasearch system for mediation among multiple search engines for audio-visual information.

4.1. Description generator

In our MPEG-7 testbed, we are using various image/video processing, analysis, and annotation systems to generate a rich variety of descriptions for a collection of image/video items, as shown in Figure 3. The descriptions that we generate for visual content include low-level visual features of automatically segmented regions, user defined semantic objects, high-level scene properties, classifications, and associated textual information. We are also including descriptions that are generated by our collaborators. As described in section 3.1., we are using XML as the DDL for the descriptions. The descriptions have the structure of the image/video description scheme (DS) described in section 3.2. and 3.3. The DS and DDL are designed to accommodate descriptions generated by a wide variety of heterogeneous systems.

Once all the descriptions for an image/video item are generated, the descriptions are inputted into a database, which the search engine accesses. We shall describe now the systems used to generate the descriptions.

- **VideoQ: Region-based indexing and searching system.**

  This system extracts visual features such as color, texture, motion, shape, and size for automatically segmented regions of a video sequence [5]. The system first decomposes a video into separate shots. This is performed by scene change detection [13]. Scene changes may be either abrupt or transitional (e.g. dissolve, fade in/out, and wipe). For each shot, the system estimates the global (i.e. the motion of dominant background) and the camera motion. Then, it segments, detects, and tracks regions across the frames in the shot computing different visual features for each region. For each shot, the descrip-

tion generated by this system is a set of regions with visual and motion features, and the camera motion. Some keywords, assigned manually, are also available for each shot.

● **AMOS: Video object segmentation system.**

Currently, fully automatic segmentation of semantic objects is only successful in constrained visual domains. The AMOS system [23] takes on a powerful approach in which automatic segmentation is integrated with user input to track semantic objects in video sequences. For general video sources, the system allows users to define an approximate object boundary by using a tracing interface. Given the approximate object boundary, the system automatically refines the boundary and tracks the movement of the object in subsequent frames of the video. The system is robust enough to handle many real-world situations that are hard to model in existing approaches, including complex objects, fast and intermittent motion, complicated backgrounds, multiple moving objects, and partial occlusion. The description generated by this system is a set of semantic objects with the associated regions and features that can be manually annotated with text.

● **MPEG domain face detection system.**

This system efficiently and automatically detects faces directly in the MPEG compressed domain [20]. The human face is an important subject in video. It is ubiquitous in news, documentaries, movies, etc., providing key information to the viewer for the understanding of the video content. This system provides a set of regions with face labels.

● **WebClip: Hierarchical video browsing system.**

This system parsers compressed MPEG video streams to extract shot boundaries, moving objects, object features, and camera motion [12]. It also generates a hierarchical shot-based browsing interface for intuitive visualization and editing of videos.

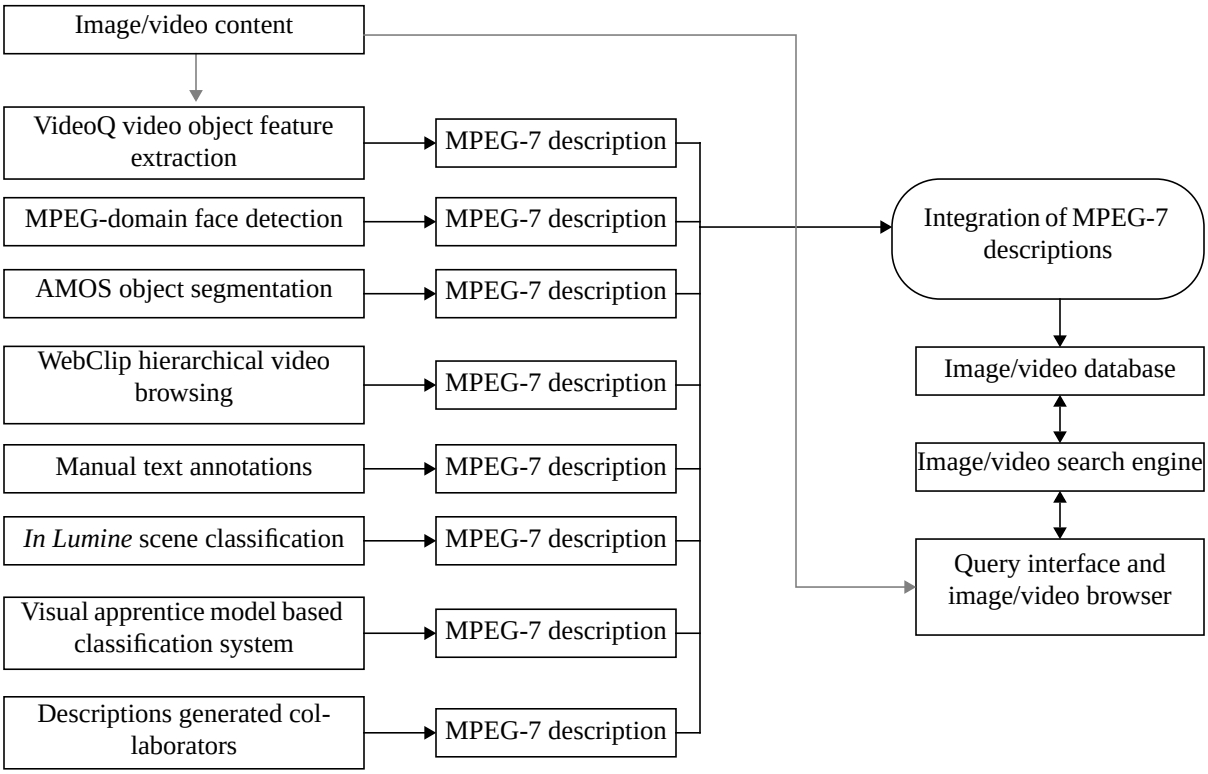


Figure 3: Architecture for combining descriptions generated by heterogeneous systems.

● **Visual Apprentice: Model based image classification system.**

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 provide exam-

ples that are used to automatically learn visual models. The visual models are based on automatically segmented regions, their associated visual features, and their spatial relationships. For example, the user may build a visual model of a portrait in which one person wearing a blue suit is seated on a brown sofa, and a second person is standing to the right of the seated person. The system uses a combination of lazy-learning, decision trees, and evolution programs during classification. The description generated by this system is a set of text annotations, i.e. the user defined classes, for each image.

● ***In Lumine:*** **Scene classification system.**

The *In Lumine* system [16] is a method for high-level semantic classification of images and video shots based on low-level visual features. The core of the system consists of various machine learning techniques such as rule induction, clustering, and nearest neighbor classification. The system is being used to classify images and video scenes into high-level semantic scene classes such as {nature landscape}, {city/suburb}, {indoor}, and {outdoor}. The system focuses on machine learning techniques because we have found that the fixed set of rules that might work well with one corpus may not work well with another corpus, even for the same set of semantic scene classes. Since the core of the system is based on machine learning techniques, the system can be adapted to achieve high performance for different corpora by training the system with examples from each corpus. The description generated by this system is a set of text annotations to indicate the scene class for each image or each keyframe associated with the shots of a video sequence.

Currently, we are exploring the integration of visual features and textual features for scene classification. Images from on-line news sources (e.g. Clarinet) have associated rich textual information (e.g. caption or articles). This textual information can be included in an expanded text-based DS in MPEG-7 as well.

4.2. Intelligent search engine

The image and video descriptions generated in our MPEG-7 testbed can be broadly categorized into three classes of descriptions [11], as follows:

● Semantic descriptions

These descriptions give the position and characteristics of visible objects in images and videos. The face detection system gives the position and characteristics of face objects in images and videos. The *In Lumine* scene classification system views an entire image or video key frame as an object and associates semantic scene labels such as indoor, city, etc. The Visual Apprentice system gives the position and characteristics of user defined objects in images and videos. Images from on-line sources include rich textual information as well. Finally, the AMOS system allows users to efficiently outline and track semantic objects in video sequences.

● Feature descriptions

These descriptions are generated by segmenting images and video frames into regions, either automatically or manually, and extracting characteristics such as color, texture, size, shape, motion, and shapes for each regions. The VideoQ system generates descriptions of this type.

● Media based descriptions

These descriptions are based on the medium the data is expressed in. What occurred in the translation from scene to video? What is the sampling rate of the digital file? Is it an analogue source? Where are the shot cuts? What is the camera's focal length? When applicable, we will rely on the program edit data about the image/video we get from the content provider (e.g. stock video companies).

The descriptions that are used by current state of the art image/video search engines can be viewed primarily as falling into the class of visual feature description [6][8] and semantic description [9][17][18][22]. The visual feature-based approach has been to obtain and utilize discriminants (features) that are useful in conducting similarity queries for visual information. Recent efforts have focused on a few specific visual dimensions such as color, texture, shape, motion and spatial information. The QBIC system [7] enables querying whole images and manually extracted image regions by color, texture, and shape. The Virage system [1] provides for querying global images by features such as color, composition, texture, and structure. The SaFE system [19] allows users to query images by regions and their spatial and feature attributes. The system enables the user to find the images that contain arrangements of regions similar to those diagrammed in a sketch. The VideoQ system [5] allows users to search for videos based on a set of visual features and spatio-temporal relationships of regions.

The semantic-level video indexing primarily uses the textual information (e.g. speech and transcript) to index the segmented video units (e.g. shot or story). Some approaches use high-level summaries (e.g. scene graphs or table of contents) to

provide efficient visualization of video content. These high-level descriptions can be implemented using the proposed self-describing description scheme for video.

As part of the MPEG-7 testbed, we are developing new expressive query interfaces and intelligent search engines that allow users to express a rich set of queries for image/video information by combining query elements that are in different the description classes, not only in a single description class.

As an example, consider the VideoQ query interface and search engine. The query is formulated exclusively in terms of elements in the visual feature description class. In VideoQ, a sketch based query interface allows users to assign motion, color, shape, and texture attributes to multiple regions. If the user is looking for a person that is walking in a particular direction in an outdoor scene, the user may try to draw a region with the color of a human face and assign a motion trajectory to the region. The user may also try to draw a blue sky to indicate that it is an outdoor scene.

However, in the new expressive query interface that we are building for the MPEG-7 testbed, users will be able to draw a region with an associated motion trajectory, indicate that the region is a human face, and also restrict the search to outdoor scenes. This will be possible because we will process all images and videos in our testbed with the face detection system and the scene classification system to generate semantic descriptions, along with the VideoQ system to extract feature descriptions. In other words, the search engine will have access to a database of descriptions of the images and videos that contains not only feature descriptions, but also semantic and media based descriptions.

Another example is the use of the video event hierarchy to build a video browsing interface based on table of contents for videos [17]. We show an example XML file for this interface.

```
<event_hierarchy type="LOGICAL">
    <event_node id ="100" event_ref="1"> <!-- In Front of the house -->
        <event_node id ="101" event_ref="10">
            <!-- Shots with actor A talking -->
        </event_node>
        <event_node id ="105" event_ref="20">
            <!-- Shots with actor B talking -->
        </event_node>
    </event_node>
    <event_node id ="200" event_ref="2"> <!-- Approaching the bridge -->
        <event_node id ="201" event_ref="30">
            <!-- Shots with actor A talking -->
        </event_node>
        <event_node id ="205" event_ref="40">
            <!-- Shots with actor B talking -->
        </event_node>
    </event_node>
</event_hierarchy>
<event_hierarchy type="PHYSICAL">
    <event_node id ="5000" event_ref="0"> <!-- Movie -->
        <event_node id ="5001" event_ref="1"/> <!-- In Front of the house -->
        <event_node id ="6001" event_ref="2"/> <!-- Approaching to bridge -->
    </event_node>
</event_hierarchy>
```

4.3. Metasearch engine

Metasearch engines act as gateways linking users automatically and transparently to multiple search engines. Most of the current metasearch engines work with text. Our metasearch engine, MetaSEEk [2], explores the issues involved in querying large, distributed, on-line visual information systems [4]. In this section, we will describe the impact that an interoperable content description for multimedia data such as MPEG-7 can have in metasearch engines.

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 4. The three main components of the system are quite standard for metasearch engines; they include the query dispatcher, the query translator, and the display interface. The procedure for each search is as follows:

- 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, 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 very restrained to the interface limitations of current search engines: solely support for query by example and by sketch, and results as a flat list of images (with similarity scores, in some cases), among others.

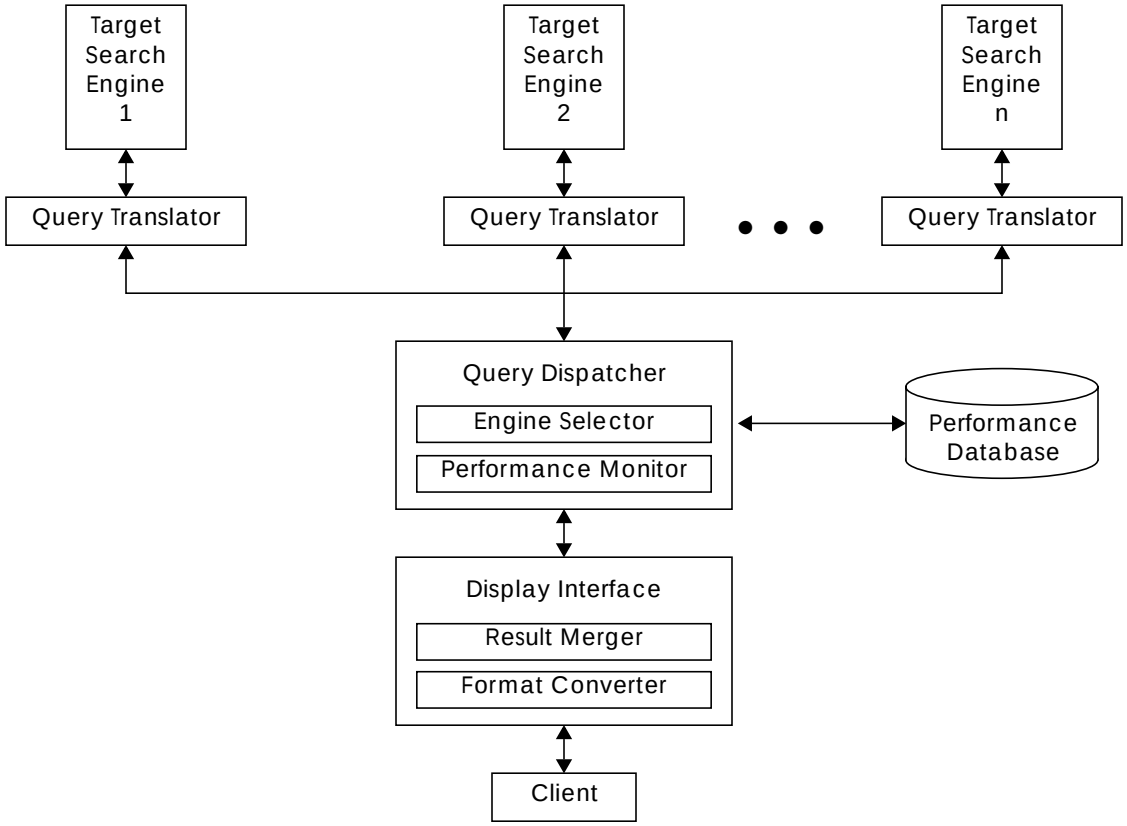


Figure 4: Overall architecture of MetaSEEk.

As discussed in the previous section, we envision a major transformation in multimedia search engines thanks to the development of the MPEG-7 standard. Future systems will accept not only queries by example and by sketch, but also queries by MPEG-7 multimedia descriptions. Users will be able to submit MPEG-7 descriptions of multimedia content as the query input to search engines, without the need of submitting the multimedia content itself. In this case, search engines will work on a best effort basis to provide appropriate results: search engines unfamiliar with some descriptors in the query multimedia description may just ignore those descriptors; others may try to translate them to local descriptors. Furthermore, queries will result in a list of matched images as well as their MPEG-7 descriptions (partial or complete). Each search engine will also make available the description scheme of its content and maybe some proprietary code.

We envision the connection between the individual search engines and the metaseach engine to be a path for MPEG-7 streams, which will enhance the performance of metasearch engines. In particular, the ability of the proposed description schemes to dynamically download programs for feature extraction and similarity matching by using linking or code embed-

ding will open the door to improved metasearching capabilities. Metasearch engines will use the description schemes of each target search engine to learn about the content and the capabilities of each search engine. This knowledge also enables meaningful queries to the repository, proper decisions to query optimal search engines, efficient ways of merging results from different repositories, and intelligent display of the search results from heterogeneous sources.

Consider a metasearch engine that receives a query by MPEG-7 multimedia description, queries by example and by sketch can be easily expressed using MPEG-7 standard.

- First, the dispatcher will match the query description to the description schemes of each search engine to ensure the satisfaction of the user preferences in the query (features selection, annotations, etc.). It will then select the target search engines to be queried by consulting the performance database. If the user wants to search by color and one search engine does not support any color descriptors, it will not be useful to query that particular search engine. The current prototype of MetaSEEk already considers the specific features supported by each search engine; however, they are fixed parameters and have to be maintained manually. On the contrary, a MPEG-7 compliant metasearch engine could adapt automatically to additions of new features in the target search engines by consulting their description schemes regularly.

- Then, the query translators will adapt the query description to descriptions conforming to each selected search engine. This translation will also be based on the description schemes available from each search engine. This task may require executing extraction code for standard descriptors or downloaded extraction code from specific search engines to transform descriptors. For example, if the user specifies the color feature of an object using a color histogram of 166 bins, the query translator will translate it to the specialized color descriptors used by each search engine (e.g. dominant color and color histogram of 128 bins). If the direct transformation of features is not possible, the metasearch engine may download the extraction code from the target search engine to extract the compatible features from the query object.

- Before displaying the results to the user, the query interface will merge the results from each search engine by translating the descriptions of the result images into a homogeneous one. Again, code for standard descriptors or specialized one may need to be executed. User preferences and/or past performance information may determine how the results are merged and displayed to the user.

We are currently developing the necessary components to begin the evaluation and test of the metasearch engine testbed based on the self-describing schemes proposed in this paper.

5. CONCLUSIONS

The increasing availability of digital multimedia information requires an interoperable multimedia content description for its efficient processing, filtering, and searching, among others. This is the objective of the MPEG-7 standard. In this paper, we have presented self-describing schemes for image and video content that we plan to propose to MPEG-7. To ensure maximum interoperability and flexibility, our descriptions use the eXtensible Markup Language (XML), developed by the World Wide Web Consortium. Under the proposed self-describing schemes, an image is represented as a set of relevant objects that are organized in one or more object hierarchies. Similarly, a video is viewed as a set of relevant events that can be combined hierarchically in one ore more event hierarchies. Both, objects and events, are described by some feature descriptors that can link to external extraction and similarity code.

The proposed self-describing schemes are intuitive, flexible, and efficient. We demonstrate the feasibility of our self-describing schemes in our MPEG-7 testbed. In the testbed, we are using the self-describing schemes for descriptions of images and videos that are generated by a wide variety of image/video indexing systems. We are developing two systems which are enabled and enhanced by our approach for multimedia content descriptions. The first system is an intelligent search engine with an associated expressive query interface. The second system is a new version of MetaSEEk, a metasearch system for mediation among multiple search engines for audio-visual information.

Currently, we are also undertaking interoperability experiments with several research groups using the proposed XML-based DS and our MPEG-7 testbed.

6. REFERENCES

[1] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain, and C. Shu, "Virage image search engine: an open framework for image management", *Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases IV*, **IS&T/SPIE'96**, San Jose, CA, Feb. 1996; web site http://www.virage.com/cgi-bin/query-e.

[2] A. B. Benitez, M. Beigi, and S.-F. Chang, "Using Relevance Feedback in Content-Based Image Metasearch", *IEEE Internet Computing*, Vol. 2, No. 4, pp. 59-69, Jul./Aug. 1998; web site http://www.ctr.columbia.edu/metaseek/.

[3] J. Bosak, "XML, Java, and the future of the Web", web site http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm.

[4] S.-F. Chang, J. R. Smith, M. Beigi, and A. B. Benitez, "Visual Information Retrieval from Large Distributed On-line Repositories", *Communications of the ACM*, Vol. 40, No. 12, pp. 63-71, Dec. 1997.

[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] S.-F. Chang, J. R. Smith, H. J. Meng, H. Wang, and D. Zhong, "Finding Images/Videos in Large Archives", *CNRI Digital Library Magazine*, Feb. 1997.

[7] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System", *IEEE Computer Magazine*, Vol. 28, No. 9, pp. 23-32, Sep. 1995; web site http://wwwqbic.almaden.ibm.com/.

[8] A. Gupta and R. Jain, "Visual Information Retrieval", *Communications of the ACM*, Vol. 40, No. 5, pp. 70-79, May 1997.

[9] A. G. Hauptmann and M. Smith, "Text, Speech and Vision for Video Segmentation: The Informedia Project", *AAAI Fall Symposium, Computational Models for Integrating Languageand Vision*, Boston, MA, Nov. 1995.

[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] J. Meng and S.-F. Chang, "CVEPS: A Compressed Video Editing and Parsing System", *ACM Multimedia Conference*, Boston, MA, Nov. 1996.

[13] J. Meng, Y. Juan, and S.-F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence", *Symposium on Electronic Imaging: Science & Technology*, **IS&T/SPIE'95**, Vol. 2417, San Jose, CA, Feb. 1995.

[14] MPEG-7's Call for Proposals for Multimedia Content Description Interface, web site at http://drogo.cselt.it/mpeg/public/mpeg-7_cfp.htm.

[15] S. Paek, A. B. Benitez, and S.-F. Chang, "Document Type Definitions for the Self-Describing Image/Video Schemes", ADVENT Project Technical Report #1998-02, Columbia University, Nov. 1998.

[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] Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing Table of Contents for Videos", *ACM J. of Multimedia Systems*, 1998.

[18] B. Shahraray and D. C. Gibbon, "Automatic Generation of Pictorial Transcript of Video Programs", **SPIE**, Vol. 2417, pp. 512-518, 1995.

[19] J. R. Smith and S.-F. Chang, "Integrated Spatial and Feature Image Query", *ACM Multimedia Systems Journal*, 1997; web site http://disney.ctr.columbia.edu/safe/.

[20] H. Wang and S.-F. Chang, "A Highly Efficient System for Automatic Face Region Detection in MPEG Video Sequences," *IEEE Trans. on Circuits and Systems for Video Technology*, special issue on Multimedia Systems and Technologies, Vol. 7, No. 4, pp. 615-628, Aug. 1997.

[21] World Wide Web Consortium's (W3C) XML web site http://www.w3.org/XML.

[22] M. M. Yeung and B. L. Yeo, "Video Content Characterization and Compaction for Digital Library Applications", *Symposium on Electronic Imaging: Science & Technology - Storage and Retrieval for Still Image and Video Databases V*, **IS&T/SPIE'97**, San Jose, CA, Vol. SPIE 3022, pp 45-58, Feb. 1997.

[23] D. Zhong and S.-F. Chang, "AMOS: An Active System for MPEG-4 Video Object Segmentation", *IEEE International Conference on Image Processing*, Chicago, IL, Oct. 1998.