1

Introduction

Multimedia Applications and Data Management Requirements

Among countless others, applications of multimedia databases include personal and public photo/media collections, personal information management systems, digital libraries, online and print advertising, digital entertainment, communications, long-distance collaborative systems, surveillance, security and alert detection, military, environmental monitoring, ambient and ubiquitous systems that provide real-time personalized services to humans, accessibility services to blind and elderly people, rehabilitation of patients through visual and haptic feedback, and interactive performing arts. This diverse spectrum of media-rich applications imposes stringent requirements on the underlying media data management layer. Although most of the existing work in multimedia data management focuses on content-based and object-based query processing, future directions in multimedia querying will also involve understanding how media objects affect users and how they fit into users' experiences in the real world. These require better understanding of underlying perceptive and cognitive processes in human media processing. Ambient media-rich systems that collect diverse media from environmentally embedded sensors necessitate novel methods for continuous and distributed media processing and fusion schemes. Intelligent schemes for choosing the right objects to process at the right time are needed to allow media processing workflows to be scaled to the immense influx of real-time media data. In a similar manner, collaborative-filtering–based query processing schemes that can help overcome the semantic gap between media and users' experiences will help the multimedia databases scale to Internet-scale media indexing and querying.

1.1 HETEROGENEITY

Most media-intensive applications, such as digital libraries, sensor networks, bioinformatics, and e-business applications, require effective and efficient data management systems. Owing to their complex and heterogeneous nature, management, storage, and retrieval of multimedia objects are more challenging than the management of traditional data, which can easily be stored in commercial (mostly relational) database management systems.

Querying and retrieval in multimedia databases require the capability of comparing two media objects and determining how similar or how different these two objects are. Naturally, the way in which the two objects are compared depends on the underlying data model. In this section, we see that any single media object (whether it is a complex media document or a simple object, such as an image) can be modeled and compared in multiple ways, based on its different properties.

1.1.1 Complex Media Objects

A complex multimedia object or a document typically consists of a number of media objects that must be presented in a coherent, synchronized manner. Various standards are available to facilitate authoring of complex multimedia objects:

- **SGML/XML.** Standard Generalized Markup Language (SGML) was accepted in 1986 as an international standard (ISO 8879) for describing the structure of documents [SGML]. The key feature of this standard is the separation of document content and structure from the presentation of the document. The document structure is defined using *document type definitions (DTDs)* based on a formal grammar. One of the most notable applications of the SGML standard is the HyperText Markup Language (HTML), the current standard for publishing on the Internet, which dates back to 1992.

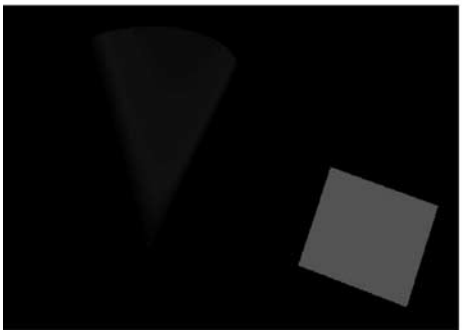
  Extensible Markup Language (XML) has been developed by the W3C Generic SGML Editorial Review Board [XML] as a follow-up to SGML. XML is a subset of SGML, especially suitable for creating interchangeable, structured Web documents. As with SGML, document structure is defined using DTDs; however, various extensions (such as elimination of the requirement that each document has a DTD) make the XML standard more suitable for authoring hypermedia documents and exchanging heterogenous information.
- **HyTime.** SGML and XML have various multimedia-oriented applications. The Hypermedia/Time-based Structuring Language (HyTime) is an international multimedia standard (ISO 10744) [HyTime], based on SGML. Unlike HTML and its derivatives, however, HyTime aims to describe not only the hierarchical and link structures of multimedia documents, but also temporal synchronization between objects to be presented to the user as part of the document. The underlying event-driven synchronization mechanism relies on timelines (Section 2.3.5).
- **SMIL.** Synchronized Multimedia Integration Language (SMIL) is a synchronization standard developed by the W3C [SMIL]. Like HyTime, SMIL defines a language for interactive multimedia presentations: authors can describe spatiotemporal properties of objects within a multimedia document and associate hyperlinks with them to enable user interaction. Again, like HyTime, SMIL is based on the timeline model and provides event-based synchronization for multimedia objects. Instead of being an application of SGML, however, SMIL is based on XML.
- **MHEG.** MHEG, the Multimedia and Hypermedia Experts Group, developed a hypermedia publishing and coding standard. This standard, also known as the MHEG standard [MHEG], focuses on platform-independent interchange and presentation of multimedia presentations. MHEG models presentations as a

collection of objects. The spatiotemporal relationships between objects and the interaction specifications form the structure of a multimedia presentation.

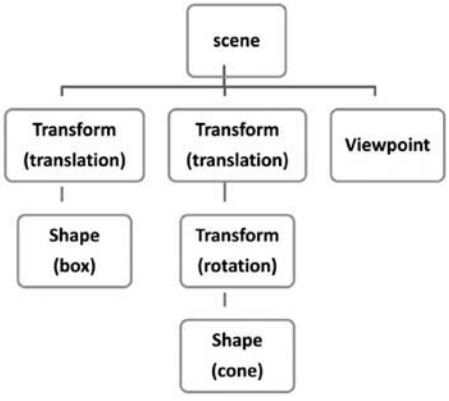
- **VRML and X3D.** Virtual Reality Modeling Language (VRML) provides a standardized way to describe interactive three-dimensional (3D) information for Web-based applications. It soon evolved into the international standard for describing 3D content [Vrml]. A VRML object or world contains various media (including 3D mesh geometry and shape primitives), a hierarchical structure that describes the composition of the objects in the 3D environment, a spatial structure that describes their spatial positions, and an event/interaction structure that describes how the environment evolves with time and user interactions. The Web3D consortium led the development of the VRML standard and its XML representation, X3D standard [X3D].
- **MPEG7 and MPEG21.** Unlike the standards just mentioned, which aim to describe the content of authored documents, the main focus of the MPEG7 (Multimedia Content Description Interface) [MPEG7] is to describe the content of captured media objects, such as video. It is a follow-up to the previous MPEG standards, MPEG1, MPEG2, and MPEG4, which were mainly concerned with video compression. Although primarily designed to support content-based retrieval for captured media, the standard is also rich enough to be applicable to synthetic and authored multimedia data. The standard includes content-based description mechanisms for images, graphics, 3D objects, audio, and video streams. Low-level visual descriptors for media include color (e.g., color space, dominant colors, and color layout), texture (e.g., edge histogram), shape (e.g., contours), and motion (e.g., object and camera trajectories) descriptors. The standard also enables description of how to combine heterogeneous media content into one unified multimedia object. A follow-up standard, MPEG21 [MPEG21], aims to provide additional content management and usage services, such as caching, archiving, distributing, and intellectual property management, for multimedia objects.

Example 1.1.1: As a more detailed example for nonatomic multimedia objects, let us reconsider the VRML/X3D standard, for describing virtual worlds. In X3D, the world is described in the form of a hierarchical structure, commonly referred to as the scene graph. The nodes of the hierarchical structure are expressed as XML elements, and the visual properties (such as size, color, and shine) of each node are described by these elements' attributes. Figure 1.1 provides a simple example of a virtual world consisting of two objects. The elements in this scene graph describe the spatial positions, sizes, shapes, and visual properties of the objects in this 3D world. Note that the scene graph has a *tree* structure: there is one special node, referred to as the root, that does not have any ancestors (and thus it represents the entire virtual world), whereas each node except this root node has one and only one parent.

The internal nodes in the X3D hierarchy are called grouping (or transform) nodes, and they bring together multiple subcomponents of an object and describe their spatial relationships. The leaf nodes can contain different types of media (e.g., images and video), shape primitives (e.g., sphere and box), and their properties (e.g., transparency and color), as well as 3D geometry in the form of polyhedra (also called meshes). In addition, two special types of nodes, sensor and script nodes,



(a)



(b)

```
<Scene>
<Transform translation='-1 0 -6'>
 <Shape>
  <Appearance>
     <Material
        ambientIntensity='0.800'
        shininess='0.800'
        diffuseColor='3 0 0'/>
  </Appearance>
  <Box size='1 1 1'/>
 </Shape>
</Transform>
<Transform translation='-3 0 -6'>
   <Transform rotation='3 1 3 3'>
     <Shape>
       <Appearance>
          <Material
             ambientIntensity='0.800'
             shininess='0.600'
             diffuseColor='0 0 1'/>
       </Appearance>
       <Cone height='2.000' bottomRadius='1.000'/>
     </Shape>
   </Transform>
</Transform>
<Viewpoint
   fieldOfView='1'
   position='-5 -1 1'
   orientation='-0.2 -0.2 -0.7 -.4'/>
</Scene>
```

(c)

Figure 1.1. An X3D world with two shape objects and the XML-based code for its hierarchical scene graph: (a) X3D world, (b) scene graph, (c) X3D code. See color plates section.

can be used to describe the interactivity options available in the X3D world: sensor nodes capture events (such as user input); script nodes use behavior descriptions (written in a high-level programming language, for example, JavaScript) to modify the parameters of the world in response to the captured events. Thus, X3D worlds can be rich and heterogeneous in content and structure (Figure 1.2):

- *Atomic media types:* This category covers more traditional media types, such as text, images, texture maps, audio, and video. The features used for media-based retrieval are specific to each media type.

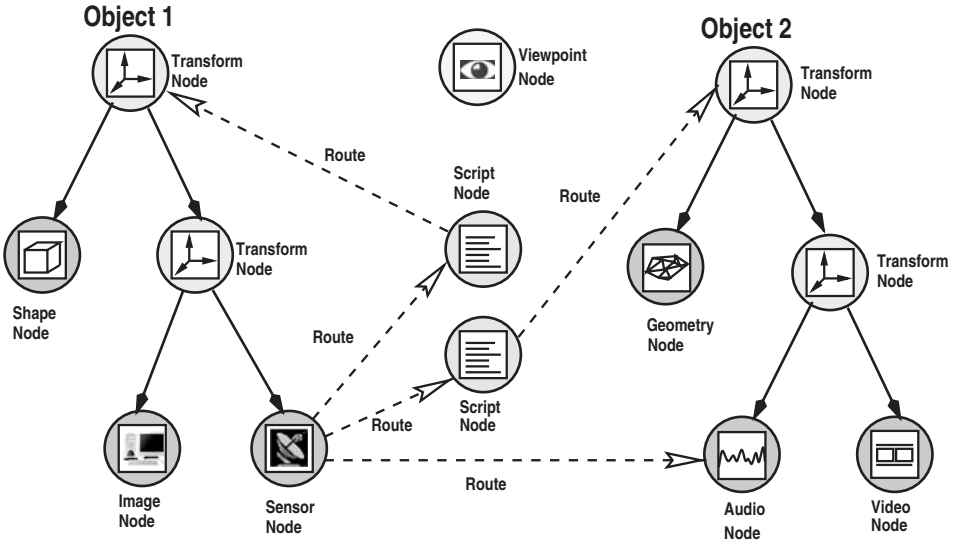


Figure 1.2. The scene graph of a more complex X3D world.

- *3D mesh geometry:* This category covers all types of polyhedra that can be represented using the X3D/VRML standard. Geometry-based retrieval is a relatively new topic, and the features to be used for retrieval are not yet well understood.
- *Shape primitives:* This category covers all types of primitive shapes that are part of the standard, as well as their attributes and properties.
- *Node structure:* The node structure describes how complex X3D/VRML objects are put together in terms of the simpler components. Because objects and subobjects are the main units of reuse, most of the queries need to have the node structure as one of the retrieval criteria.
- *Spatial structure:* The spatial structure of an object is related to its node structure; however, it describes the spatial transformations (scaling and translation) that are applied to the subcomponents of the world. Thus queries are based on spatial properties of the objects.
- *Event/interaction structure:* The event structure of a world, which consists of sensor nodes and event routes between sensor nodes and script nodes, describes causal relationships among objects within the world.
- *Behaviors:* The scripting nodes, which are part of the event structure, may be used for understanding the behavioral semantics of the objects. Because these behaviors can be reused, they are likely to be an important unit of retrieval. The standard does not provide a descriptive language for behaviors. Thus, retrieval of behaviors is likely through their interfaces and the associated metadata.
- *Temporal structure:* The temporal structure is specified through time sensors and the associated actions. Consequently, the temporal structure is a specific type of event structure. Because time is also inherent in the temporal media (such as video and audio) that can be contained within an X3D/VRML object, it needs to be treated distinctly from the general event structure.

- *Metadata:* This covers everything associated with the objects and worlds (such as the textual content of the corresponding files or filenames) that cannot be experienced by the viewers. In many cases, the metadata (such as developer's comments and/or node and variable names) can be used for extracting information that describes the actual content.

The two-object scene graph in Figure 1.2 contains an image file, which might be used as a surface texture for one of the objects in the world; an audio file, which might contain the soundtrack associated with an object; a video file, which might be projected on the surface of one of the objects; shape primitives, such as boxes, that can be used to describe simple objects; and 3D mesh geometry, which might be used to describe an object (such as a human avatar) with complex surface description. The scene graph further describes different types of relationships between the two nodes forming the world. These include a composition structure, which is described by the underlying XML hierarchy of the nodes constituting the X3D objects, and events that are captured by the sensor nodes and the causal structure, described by script nodes that can be activated by these events and can affect any node in the scene graph. In addition, temporal scripts might be associated to the scene graph, enabling the scene to evolve over time. Note that when considering the interaction pathways between the nodes in the X3D (defined through sensors and scripts), the structure of the scene graph ceases to be a tree and, instead, becomes a *directed graph*.

Whereas an X3D world is often created and stored as a single file, in many other cases the multimedia content may actually not be available in the form of a single file created by a unique individual (or a group with a common goal), but might in fact consist of multiple independent components, possibly stored in a distributed manner. In this sense, the Web itself can be viewed as a single (but extremely large) multimedia object. Although, in many cases, we access this *object* only a page (or an image, or a video) at a time, search engines treat the Web as a complex whole, with a dynamic structure, where communities are born and evolve repeatedly. In fact, with Web 2.0 technologies, such as blogs and social networking sites, which strongly tie the users to the content that they generate or annotate (i.e., tag), this vast object (i.e., the entire Web) now also includes the end users themselves (or at least their online representations).

1.1.2 Semantic Gap

It is not only the complex objects (described using hypermedia standards, such as X3D, SMIL, MPEG7, or HTML) that may necessitate structured, nonatomic models for representation. Even objects of relatively simple media types, such as images and video, may embed sub-objects with diverse local properties and complex spatiotemporal interrelationships. For example, an experimental study conducted by H. Nishiyama *et al.* [1994] shows that users are viewing paintings or images using two primary patterns. The first pattern consists of viewing the whole image roughly, focusing only on the layout of the images of particular interest. The second pattern consists of concentrating on specific objects within the image. In a sense, we can view a single image as a compound object containing many sub-objects, each

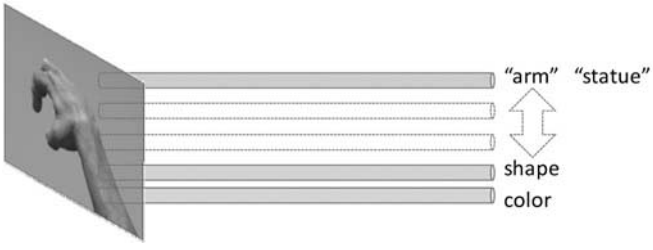


Figure 1.3. Any media object can be seen as a collection of channels of information; some of these information channels (such as color and shape) are low-level (can be derived from the media object), whereas others (such as semantic labels attached to the objects by the viewer) are higher level (cannot be derived from the media object without external knowledge). See color plates section.

corresponding to regions of the image that are visually coherent and/or semantically meaningful (e.g., car, man), and their spatial relationships.

In general, a *feature* of a media object is simply any property of the object that can be used for describing it to others. This can include properties at all levels, from low-level properties (such as color, texture, and shape) to semantic features (such as linguistic labels attached to the parts of the media object) that require interpretation of the underlying low-level features at much higher semantic levels (Figure 1.3). This necessity to have an interpretive process that can take low-level features that are immediately available from the media and map to the high-level features that require external knowledge is commonly referred to as the *semantic gap*.

The semantic gap can be bridged, and a multimedia query can be processed, at different levels. In content-based retrieval, the low-level features of the query are matched against the low-level features of the media objects in the database to identify the appropriate matches (Figure 1.4(a)). In semantic-based retrieval, either the high-level query can be restated in terms of the corresponding low-level features for matching (Figure 1.4(b)) or the low-level features of the media in the database can

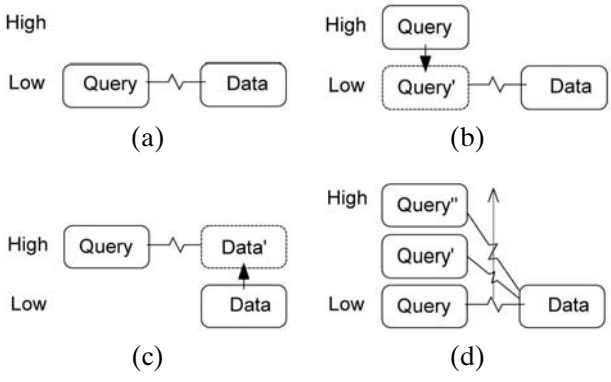


Figure 1.4. Different query processing strategies for media retrieval: (a) Low-level feature matching. (b) A high-level query is translated into low-level features for matching. (c) Low-level features are interpreted for high-level matching. (d) Through relevance feedback, the query is brought higher up in semantic levels; that is, it is increasingly better at representing the user's intentions.

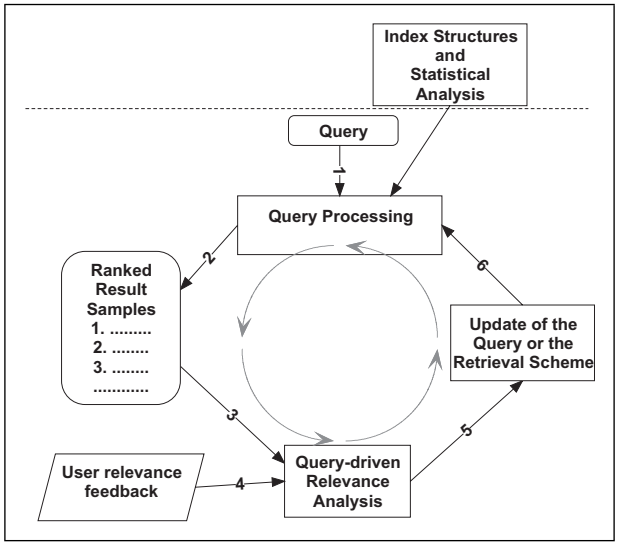


Figure 1.5. Multimedia query processing usually requires the semantic gap between what is stored in the database and how the user interprets the query and the data to be bridged through a relevance feedback cycle. This process itself is usually statistical in nature and, consequently, introduces probabilistic imprecision in the results.

be interpreted (for example through classification, Chapter 9) to support retrieval (Figure 1.4(c)). Alternatively, user relevance feedback (Figure 1.5 and Chapter 12) and collaborative filtering (Sections 6.3.3 and 12.8) techniques can be used to rewrite the user query in a way that better represents the user's intentions (Figure 1.4(d)).

1.2 IMPRECISION AND SUBJECTIVITY

One common characteristic of most multimedia applications is the underlying uncertainty or imprecision.

1.2.1 Reasons for Imprecision and Subjectivity

Because of the possibly redundant ways to sense the environment, the alternative ways to process, filter, and fuse multimedia data, the diverse alternatives in bridging the semantic gap, and the subjectivity involved in the interpretation of data and query results, multimedia data and queries are inherently imprecise:

- *Feature extraction algorithms that form the basis of content-based multimedia data querying are generally imprecise.* For example, a high error rate is encountered in motion-capture data and is generally due to the multitude of environmental factors involved, including camera and object speed. Especially for video/audio/motion streams, data extracted through feature extraction modules are only statistically accurate and may be based on the frame rate or the position of the video camera related to the observed object.
- *It is rare that a multimedia querying system relies on exact matching.* Instead, in many cases, multimedia databases need to consider nonidentical but *similar*

Table 1.1. Different types of queries that an image database may support

Find all images created by "John Smith"
Find all images that look like "query.gif"
Find top-5 images that look like "im_ex.gif"
Find all images that look like "mysketch.bmp"
Find all images that contain a part that looks like "query.gif"
Find all images of "sunny days"
Find all images that contain a "car"
Find all images that contain a "car" and a man who looks like "mugshot.bmp"
Find all image pairs that contain similar objects
Find all objects contained in images of "sunny days"
Find all images that contain two objects, where the first object looks like "im_ex.gif," the second object is something like a "car," and the first object is "to the right of" the second object; also return the semantic annotation available for these two objects
Find all new images in the database that I may like based on my list of preferences
Find all new images in the database that I may like based on my profile and history
Find all new images in the database that I may like based on access history of people who are similar to me in their preferences and profiles

features to find data objects that are reasonable matches to the query. In many cases, it is also necessary to account for semantic similarities between associated annotations and partial matches, where objects in the result satisfy some of the requirements in the query but fail to satisfy all query conditions.

- *Imprecision can be due to the available index structures, which are often imperfect.* Because of the sheer size of the data, many systems rely on clustering and classification algorithms for sometimes imperfectly pruning search alternatives during query processing.
- *Query formulation methods are not able to capture the user's subjective intention perfectly.* Naturally the query model used for accessing the multimedia database depends on the underlying data model and the type of queries that the users will pose (Table 1.1). In general, we can categorize query models into three classes:
  - *Query by example (QBE):* The user provides an example and asks the system to return media objects that are similar to this object.
  - *Query by description:* The user provides a declarative description of the objects of interest. This can be performed using an SQL-like ad hoc query language or using pictorial aids that help users declare their interests through sketches or storyboards.
  - *Query by profile/recommendation:* In this case, the user is not actively querying the database; instead the database predicts the user's needs based on his or her profile (or based on the profiles of other users who have similar profiles) and recommends an object to the user in a proactive manner.

  For example, in Query-by-Example (QBE) [Cardenas *et al.*, 1993; Schmitt *et al.*, 2005], which features, feature value ranges, feature combinations, or similarity notions are to be used for processing is left to the system to figure out through feature significance analysis, user preferences, relevance feedback [Robertson

```
select image P, imageobject object1, object2 where
        contains(P, object1) and contains(P, object2) and
        (semantically_similar(P.semanticannotation, "Fuji Mountain") and
        visually_similar(object1.imageproperties, "Fujimountain.jpg")) and
        (semantically_similar(P.semanticannotation, "Lake") and
        visually_similar(object2.imageproperties, "Lake.jpg")) and
        above(object1, object2).
```

Figure 1.6. A sample multimedia query with imprecise (e.g., semantically_similar(), visually_similar(), and above()) and exact predicates (e.g., contains()).

and Spark-Jones, 1976; Rui and Huang, 2001] (see Figure 1.5), and/or collaborative filtering [Zunjarwad *et al.*, 2007] techniques, which are largely statistical and probabilistic in nature.

1.2.2 Impact on Query Formulation and Processing

In many multimedia systems, more than one of the foregoing reasons for imprecision coexist and, consequently, the system must take them into consideration collectively. Degrees of match have to be quantified and combined, and results have to be filtered and ordered based on these combined matching scores. Figure 1.6 provides an example query (in an SQL-like syntax used by the SEMCOG system [Li and Candan, 1999a]) that brings together imprecise and exact predicates. Processing this query requires assessment of different sources of imprecision and merging them into a single value for ranking the results:

Example 1.2.1: Figure 1.7(a) shows a visual representation of the query in Figure 1.6. Figures 1.7(b), (c), (d), and (e) are examples of candidate images that may match this query. The values next to the objects in these candidate images denote the similarity values for the object-level matching. In this hypothetical example, the evaluation of spatial relationships is also fuzzy (or imprecise) in nature.

The candidate image in Figure 1.7(b) satisfies object matching conditions, but its layout does not match the user specification. Figures 1.7(c) and (e) satisfy the image layout condition, but the features of the objects do not perfectly match the query specification. Figure 1.7(d) has low structural and object matching. In Figure 1.7(b), the spatial predicate and in Figure 1.7(d), the image similarity predicate for the lake, completely fail (i.e., the degree of match is 0.0). A multimedia database engine must consider all four images as candidates and must rank them according to a certain unified criterion.

The models that can capture the imprecise and statistical nature of multimedia data are many times fuzzy and probabilistic in nature. Probabilistic models (Section 3.5) rely on the premise that the sources of imprecision in data and query processing are inherently statistical and thus they commit onto probabilistic evaluation. Fuzzy models (Section 3.4) are more flexible and allow different semantics, each applicable under different system requirements, to be selected for query evaluation.