MULTIMEDIA INFORMATION SYSTEMS

Multimedia information systems afford users conventional database functionalities in the context of multimedia data, including audio, image, and video data. Thus multimedia data can be queried on the basis of their semantic contents. Since such contents, in general, are not described in words as in conventional databases, conventional data indexing and search mechanisms cannot be used for processing queries on such data. How can we employ technology in order to obtain full database functionality from multimedia data stores? This article attempts to answer this question by describing the challenges, progress to date, and future directions in the area of multimedia information systems.

Multimedia information technology will allow users to store, retrieve, share, and manipulate complex information composed of audio, images, video as well as text. A variety of fields, including business, manufacturing, education, computer-aided design (CAD)/computer-aided engineering (CAE), medicine, weather, and entertainment, are expected to benefit from this technology. A broad range of applications includes remote collaboration via video teleconferencing, improved simulation methodologies for all disciplines of science and engineering, and better human-computer interfaces (1). There is a potential for developing vast libraries of information including arbitrary amounts of text, video, pictures, and sound more efficiently usable than traditional book, record, and tape libraries of today. These applications are just a sample of the kinds of things that may be possible with the development and use of multimedia.

As the need for multimedia information systems is growing rapidly in various fields, management of such information is becoming a focal point of research in the database community. Multimedia data possess certain distinct characteristics from conventional data, as shown in Table 1. This proliferation of applications also explains partly why there is an explosion of research in the areas related to the understanding, development, and utilization of multimedia-related technologies.

Table 1. Differences Between Conventional and Multimedia Data

Conventional Data	Multimedia Data
Types known to programming languages (character, integer, real)	Not generally known
Relatively small size	Large size (memory and bandwidth)
Fixed size atomic units	Variable size atomic units
Not highly interactive	Highly interactive
No special temporal requirements	Temporal synchronization needed
No special interface for querying	Special interface for querying
Frequent updating	Mostly archival

Depending on the application, multimedia data may have varying *quality* of presentation requirements. For example, in medical information systems, electronic images such as X rays, MRIs, and sonograms may require high-resolution storage and display systems. Systems designed to store, transport, display, and manage multimedia data require considerably more functionality and capability than conventional information management systems handling textual and numeric data. Some of the hardware problems faced include the following: Storage devices, which are usable on-line with the computers, are not "big" enough. The speed of retrieval from the available storage devices, including disks, is not sufficiently fast to cope with the demands of many multimedia applications. Conversely, storing multimedia data on disk is also relatively slow. Cache memories are a precious resource, but they are too small when it comes to multimedia, hence even greater demands for efficient resource management. Communication bandwidth tends to be another problem area for multimedia applications. A single object may demand large portions of bandwidth for extended periods of time. The problems of communication are compounded because of the delay-sensitive nature of multimedia. Storage problems for multimedia and for similar high-performance applications have been identified as deserving high priority.

In multimedia information systems, mono-media may represent individual data entities that serve as components of some multimedia object such as electronic documents or medical records containing electronic images and sonograms. Furthermore these objects/documents can be grouped together for efficient management and access. It is essential that the user be able to identify and address different objects and to compose them both in time and space. The composition should be based on a model that is visually presentable to the user (see Fig. 1). It is therefore desirable that a general framework for spatiotemporal modeling should be available that can ultimately be used for composing and storing multimedia documents.

The article is organized as follows. First we introduce the basic concepts of multimedia data, including fundamental pragmatics of multimedia information systems. This is followed by a description of peculiarities of audio, image, and video data, leading to the necessity of handling the temporal

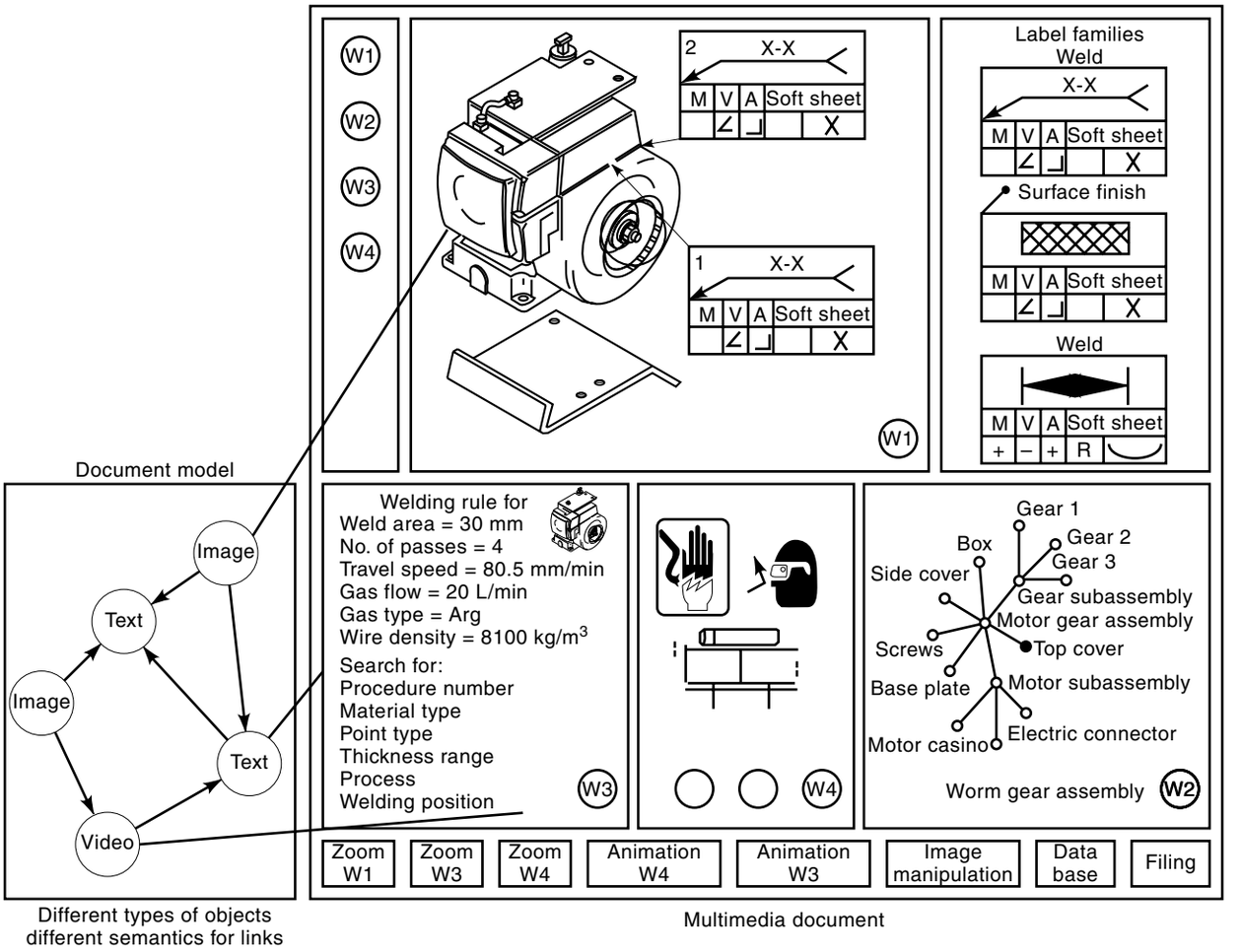

Figure 1. An example multimedia document from the manufacturing domain, along with its document model.

dimension in multimedia data processing. Then we introduce the vital issues inherent in content-based retrieval of image and video data. In order to allow content-based queries on multimedia data, designers must employ novel data modeling and processing techniques. These models and techniques are also covered in this section. Real-world multimedia documents consist of a mix of text, audio, image, and video data. Therefore, techniques and models specific to each of the component media need to be combined in order to handle multimedia documents. In the final section, we study models and techniques used to describe, author, and query complex multimedia documents consisting of several component media. Finally, we present conclusions and general reflections on the technical future of multimedia information systems.

REQUIREMENTS OF MULTIMEDIA INFORMATION SYSTEMS

From the systems point of view, because of the heterogeneous nature of the data, storage, transportation, display, and management of multimedia data must have considerably more functionalities and capabilities than the conventional information management systems. The fundamental issues faced by the multimedia information management researchers/designers are as follows:

- Development of models for capturing the media synchronization requirements. Integration of these models with the underlying database schema will be required. Subsequently, in order to determine the synchronization requirements at retrieval time, transformation of these models into a metaschema is needed. This entails designing of object retrieval algorithms for the operating systems. Similarly integration of these models with higher-level information abstractions such as Hypermedia or object-oriented models, may be required.
- Development of conceptual models for multimedia information, especially for video, audio, and image data. These models should be rich in their semantic capabilities for abstraction of multimedia information and be able to provide canonical representations of complex images, scenes, and events in terms of objects and their spatiotemporal behavior.
- Design of powerful indexing, searching, accessing, and organization methods for multimedia data. Search in

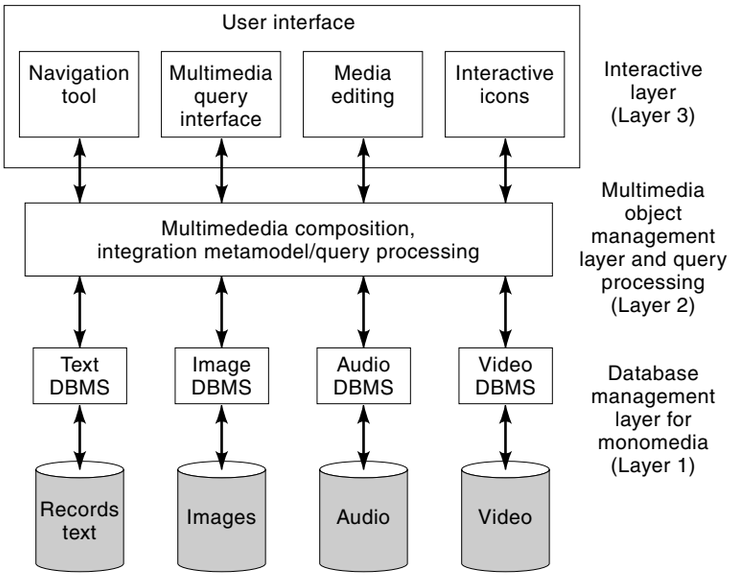

Figure 2. Example of multimedia information management system.

multimedia databases can be quite computationally intensive, especially if content-based retrieval is needed for image and video data stored in compressed or uncompressed form. Occasionally search may be fuzzy or based on incomplete information. Some form of classification/grouping of information may be needed to help the search process.

- Design of efficient multimedia query languages. These languages should be capable of expressing complex spatiotemporal concepts, should allow imprecise match retrieval, and should be able to handle various manipulation functions for multimedia objects.
- Development of efficient data clustering and storage layout schemes to manage real-time multimedia data for both single and parallel disk systems.
- Design and development of a suitable architecture and operating system support for a general purpose database management system
- Management of distributed multimedia data and coordination for composition of multimedia data over a network

Accordingly we can perceive an architecture for a general purpose multimedia information system as shown in Fig. 2. The architecture consists of three layers, which include a monomedia database management layer, an object management layer, and a user interface layer.

The monomedia database management layer provides the functionalities essential for managing individual media including formatted data (text and numeric) and unformatted data (audio, video, images). One of the key aspects of each database at this level is to maintain efficient indexing mechanism(s) and to allow users to develop semantic-based modeling and grouping of complex information associated with each media. The primary objective is to process content-based queries and facilitate retrieval of appropriate pieces of monomedia data, such as a video clip(s), parts of an image, or some desired audio segments. A major consideration at the time of retrieval is the quality of information that can be sustained by the system (both at the database site and the user site). Therefore it is important that some quality of presentation (QoP) parameters, such as speed, resolution, or delay bounds, be specified by the user and maintained by the system at this layer.

The middle layer provides the functionality of integration of monomedia for composing multimedia documents as well as integrating/cross-linking information stored across monomedia databases. Integration of media can span multiple dimensions including space, time and logical abstractions (e.g., Hypermedia or object oriented). Therefore the primary function of this layer is to maintain some metaschema for media integration along with some unconventional information, such as the QoP parameters discussed above. The objective is to allow efficient searching and retrieval of multimedia information/documents with the desired quality, if possible. Since there is a growing need for management of multimedia documents and libraries, the need for efficient integration models is becoming one of the key research issues in developing a general purpose multimedia DBMS.

The interactive layer consists of various user interface facilities that can support graphics and other multimedia interface functionalities. In this layer various database query and browsing capabilities can be provided.

NOTION OF TIME FOR MULTIMEDIA DATA

A multimedia object may contain real-time data like audio and video in addition to the usual text and image data that constitute present-day information systems. Real-time data can require time-ordered presentation to the user. A composite multimedia object may have specific timing relationships among the different types of component media. Coordinating the real-time presentation of information and maintaining the time-ordered relations among component media is known as temporal synchronization. Assembling information on the workstation is the process of spatial composition, which deals

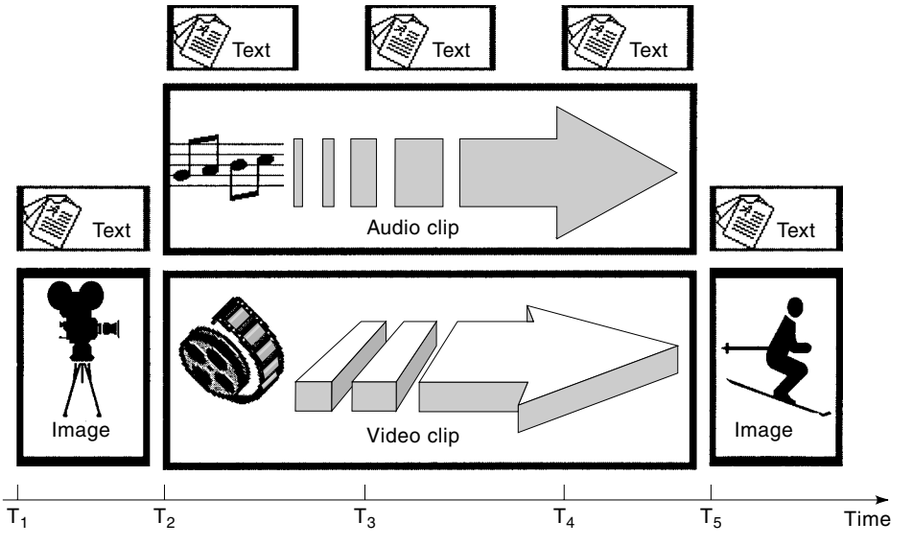

Figure 3. Time-ordered multimedia data.

basically with the window management and display layout interface.

For continuous media, the integration of temporal synchronization functions within the database management system is desirable, since it can make the storage and handling of continuous data more efficient for the database system. Also implementation of some standard format for data exchange among heterogeneous systems can be carried out more effectively. In this section we first elaborate on the problem of temporal synchronization of multimedia data for composing objects, followed by a discussion of modeling time. These models are then used to develop conceptual models for the multimedia data, as described in a later section.

Temporal Synchronization Problem

The concept of temporal synchronization is illustrated in Fig. 3 where a sequence of images and text is presented in time to compose a multimedia object. Notice in this figure that the system must observe some time relationships (constraints) among various data objects in order to present the information to the user in a meaningful way. These relationships can be *natural* or *synthetically created* (2). Simultaneous recording of voice and video through a VCR, is an example of *natural* relationship between audio and video information. A voice-annotated slide show, on the other hand, is an example of *synthetically created* relationship between audio and image information. In this case, change of an image and the end of its verbal annotation, represent a synchronization point in time.

A user can randomly access various objects, while browsing through a multimedia information system. In addition to simple forward play-out of time-dependent data sequences, other modes of data presentation are viable and should be supported by a multimedia database management system. These include reverse play-out, fast-forward/fast-backward play-out, and random access of arbitrarily chosen segments of a composed object. Although these operations are quite common in TV technology (e.g., VCRs), these capabilities are very hard to implement in a multimedia system. This is due to the nonsequential storage of multimedia objects, the diversity in the features of hardware used for data compression, the distribution of data, and random communication delays introduced by the network. Such factors make the provision of these capabilities infeasible with the current technologies.

Conceptually synchronization of multimedia information can be classified into three categories, depending on the "level of granularity of information" to be synchronized (3). These are the physical level, the service level, and the human interface level (3), as shown in Fig. 4.

At the physical level, data from different media are multiplexed over single physical connections or are arranged in physical storage. This form of synchronization can be viewed as "fine grain." The service level synchronization is "more coarse grain," since it is concerned with the interactions between the multimedia application and the various media, and among the elements of the application. This level deals primarily with intermedia synchronization necessary for presentation or play-out. The human interface level synchronization is rather "coarse grain," since it is used to specify the random user interaction to a multimedia information system such as viewing a succession of database items, also known as browsing.

In addition to time-dependent relational classification (i.e., synthetic/natural), data objects can be classified by their presentation and application lifetimes. A persistent object is one that can exist for the duration of the application. A nonpersistent object is created dynamically and discarded when ob-

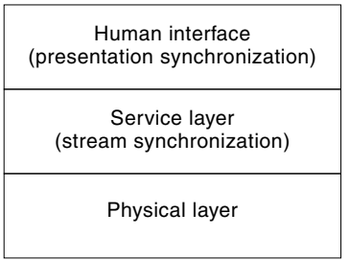

Figure 4. Levels of synchronization of multimedia data.

solete. For presentation, a transient object is defined as an object that is presented for a short duration without manipulation. The display of a series of audio or video frames represents a transient presentation of objects, whether captured live or retrieved from a database. Henceforth we use the terms *static* and *transient* to describe presentation lifetimes of objects, while persistence expresses their storage life in a database.

In another classification, multimedia data have been characterized as either continuous or discrete (4). This distinction, however, is somewhat vague, since time ordering can be assigned to discrete media, and continuous media are time-ordered sequences of discrete ones after digitization. We use a definition attributable to Ref. 4, where continuous media are represented as sequences of discrete data elements played out contiguously in time. However, the term *continuous* is most often used to describe the fine-grain synchronization required for audio or video.

Modeling Time

The problem of multimedia synchronizing at presentation, user interaction, and physical layers reduces to satisfying temporal precedence relationships among various data objects under real timing constraints. For such purpose, models to represent time must be available. Temporal intervals and instants provide a means for indicating exact temporal specification. In this section, we discuss these models and then describe various conceptual data models to specify temporal information necessary to represent multimedia synchronization.

To be applicable to multimedia synchronization, time models must allow synchronization of components having precedence and real-time constraints, and they must provide the capability for indicating laxity in meeting deadlines. The primary requirements for such a specification methodology include the representation of real-time semantics and concurrency, and a hierarchical modeling ability. The nature of presentation of multimedia data implies that a multimedia system has various additional capabilities such as to handle reverse presentation, to allow random access (at an arbitrary start point), to permit an incomplete specification of inter-media timing, to handle sharing of synchronized components among applications, and to provide data storage for control information. In light of these additional requirements, it is therefore imperative that a specification methodology also be well suited for unusual temporal semantics and be amenable to the development of a database for storing timing information.

The first time model is an instant-based temporal reference scheme which has been extensively applied in the motion picture industry, as standardized by the Society of Motion Picture and Television Engineers (SMPTE). This scheme associates a virtually unique sequential code to each frame in a motion picture. By assigning these codes to both an audio track and a motion picture track, inter-media synchronization between streams is achieved. This absolute, instant-based scheme presents two difficulties when applied to a multimedia application. First, since unique, absolute time references are assumed, when segments are edited or produced in duplicate, the relative timing between the edited segments becomes lost in terms of play-out. Furthermore, if one medium, while synchronized to another, becomes decoupled from the other, then the timing information of the dependent medium becomes lost. This instant-based scheme has also been applied using musical instrument digital interface (MIDI) time instant specification (5). This scheme has also been used to couple each time code to a common time reference (6).

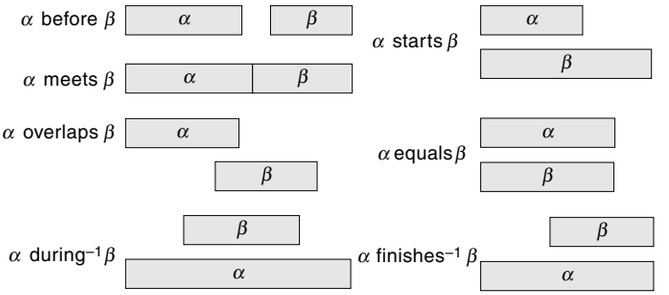

Figure 5. All possible temporal relations between two events.

In another approach, temporal intervals are used to specify relative timing constraints between two processes. This model is mostly applicable to represent simple parallel and sequential relationships. In this approach synchronization is accomplished by explicitly capturing each of the 13 possible temporal relations (2), shown in Fig. 5, that can occur between the processes. Additional operations can be incorporated in this approach to facilitate incomplete timing specification (4).

CONTENT-BASED RETRIEVAL OF MULTIMEDIA DATA

Image Data Modeling and Retrieval

Traditionally research in image database systems has been focused on image processing and recognition aspects of the data. The growing role of image databases for information technology has spurred tremendous interest in data management aspects of information. Many challenges are faced by the database community in this area, including development of new data models and efficient indexing and retrieval mechanisms. To date, the general approach for image data modeling is to use multilevel abstraction mechanisms and support content-based retrieval using such abstractions. The levels of abstraction require feature extraction, object recognition, and domain-specific spatial reasoning and semantic modeling, as shown in Fig. 6.

In this section, we use this figure as our focus of discussion and elaboration of few selected approaches proposed in the literature for developing such multilevel abstractions and associated indexing mechanisms. We discuss the important role played by the knowledge-based representation for processing queries at different levels.

Feature Extraction Layer. The main function of this layer is to extract object features from images and map them onto a multidimensional feature space that can allow similarity based retrieval of images using their salient features. Features in an image can be classified as: global or local. Global features generally emphasize "coarse-grained" similarity-based matching techniques for query processing. Example queries include "*Find images that are predominantly green,*" or "*Retrieve an image with a large round orange textured ob-*

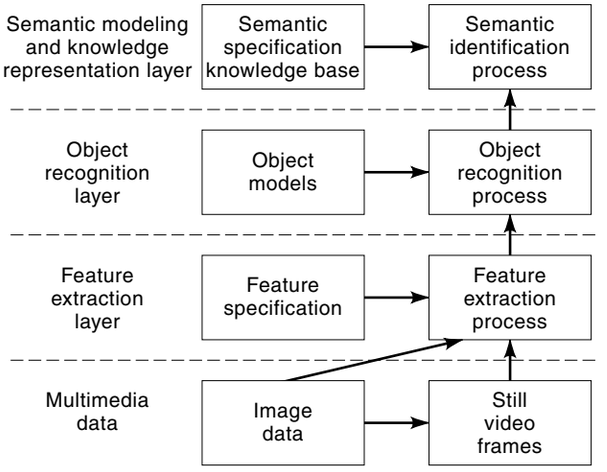

Figure 6. Processing and semantic modeling for image database.

ject." The global feature extraction techniques transform the whole image into a "*functional representation.*" The finer details among individual parts of the image are ignored. Color histograms, fast fourier transform, Hough transform, and eigenvalues are the well-known functional techniques that fall into this category.

Local features are used to identify salient objects in an image and to extract more detailed and precise information about the image. The approach is "*fine grained*" in the sense that images are generally segmented into multiple regions and different regions are processed separately to extract multiple features. In other words, local features constitute a multidimensional search space. Features in the form of encoded vectors provide the basis for indexing and searching mechanisms of image databases. Typical features include gray scale values of pixels, colors, shapes, and texture. Various combination of features can be specified at the time of formulating database queries. Incorporating domain knowledge with local features can provide more robust and precise indexing and search mechanisms using similarity based measures. Different kinds of measures have been proposed in the literature. These include, among others, Euclidean distance, Manhattan distance, weighted distance, color histogram intersection, and average distance. The performance of these similarity-based search strategies depends on the degree of imprecision and fuzziness introduced by the types of features used and the computational characteristics of the algorithms.

Choice of features, their extraction mechanisms, and the search process at this level are domain specific. For example, multimedia applications targeted for X-ray imaging, and geographic information systems (GIS) require spatial features such as shapes and dimensions. On the other hand, for applications involving MMR imaging, paintings, and the like, color features are more suitable. The feature extraction mechanism can be manual, automatic, or hybrid. The trade-off is between the complexity and robustness of the algorithm in terms of its precision and the cost incurred by the manual approach.

Various systems have been prototyped that use a feature extraction layer similar to the one shown in Fig. 6. For example, in the Query by Image Content (QBIC) system (7), color, shape, and texture features are used for image retrieval. In this system features are extracted using a fully automatic image segmentation method. A model is used to identify objects with certain foreground/background settings. The system allows querying of the database by sketching features and providing color information about the desired objects. A system that uses a combination of color features and textual annotation attributes for image retrieval is the Chabot system (8). The system uses the notion of "*concept query*" where a concept, like sunset, is recognized by analyzing images using color features. It uses a frame-based knowledge representation of image contents, which is pre-computed and stored as attributes in a relational data model. For improving the performance of the system, it uses textual annotation of images by keywords that are manually entered. A system that uses quantitative methods for edge detection to identify shape features in a radiological database, known as KMeD is presented in Ref. 9. This system employs a three-layer architecture, where the lowest layer, known as the representation layer, uses shapes and contours to represent features. This layer employs a semiautomatic feature extraction mechanism based on a combination of low-level image processing techniques and visual analysis of the image manually. From a functionality point of view, this layer reduces to the feature extraction layer of Fig. 6.

Object Recognition Layer. Features extracted at the lower level can be used to recognize objects and faces in an image database. Such a process is carried out by a higher layer as shown in Fig. 6. The process involves matching features extracted from the lower layer with the object models stored in a knowledge base. During the matching process, each model is inspected to find the "closest" match. Identifying an exact match is a computationally expensive task that depends on the details and the degree of precision possessed by the object model. Occlusion of objects and the existence of spurious features in the image can further diminish the success of matching strategies. As pointed out earlier, some fuzziness and imprecision must be incorporated in the similarity measure in order to increase the success rate of queries and not to exclude good candidates. For this reason, examining images manually at this level is generally unavoidable.

Identification of human faces is an important requirement in developing image databases. However, due to more inherent "structuredness" in human faces, models and features used for face recognition are different than those used for object recognition. Face recognition involves three steps: face detection whereby a face is located inside an image, feature extraction where various parts of a face are detected, and face recognition where the person is identified by consulting a database containing "facial models." Several face detection and recognition systems for multimedia environments have been proposed (10). For face recognition, most of these systems use information about various prominent parts of a face such as eyes, nose, and mouth. Another technique decomposes face images into a set of characteristic features called eigenfaces (10). This technique captures variations in a collection of face images and uses them to encode and compare individual features. A third approach, motivated by neurocomputing, uses global transforms, such as Morlet transform, to determine salient features present in human faces (10).

Extraction of features and object recognition are important phases for developing large-scale general purpose image database management systems. Significant results have been re-

Table 2. Survey of Different Image Database Systems

	Feature Extraction		Object Recognition		Spatial Semantics	
System	Process	Features	Process	Type of Knowledge Base	Process	Knowledge Base Support
QBIC	Automatic	Color, shape	Hybrid	—	—	—
Chabot	Automatic	Color	Keywords	Frame based	—	—
KMED	Hybrid	Shape	Hybrid	Attribute list of shape descriptors	Hybrid	Semantic nets
PICTION	Automatic	Facial shape	Automatic	Constraints	Automatic	Constraints
Yoshitaka et al.[a]	Automatic	Shape	Manual	—	Automatic	Inclusion hierarchies

ported in the literature for the last two decades, with successful implementation of several prototypes. However, the lack of precise models for object representation and the high complexity of image processing algorithms make the development of fully automatic image management and content-based retrieval systems a challenging task.

Spatial Modeling and Knowledge Representation Layer. The major function of this layer is to maintain the domain knowledge for representing spatial semantics associated with image databases. Queries at this level are generally descriptive in nature and are focused mostly on semantics and concepts present in image databases. For most of the applications, semantics at this level are based on "spatial events" (11) describing the relative orientation of objects with each other. Such semantics can provide high-level indexing mechanisms and support content-based retrieval for a large number of multimedia applications. For example, map databases and geographic information systems (GIS), are extensively used for urban planning and resource management. These systems require processing of queries that involve spatial concepts such as *close by, in the vicinity,* or *larger than.* In clinical radiology applications, relative sizes and positions of objects are critical for medical diagnosis and treatment. Some example queries in this application include "*Retrieve all images that contain a large tumor in the brain,*" or "*Find an image where the main artery is 40% blocked.*"

The general approach for modeling spatial semantics for such applications is based on identifying spatial relationships among objects once they are recognized and marked by the lower layer using bounding boxes or volumes. Spatial relationships can be coded using various knowledge-based techniques. These techniques can be used to process high-level queries as well as to infer new information pertaining to the evolutionary nature of the data. Several formal techniques have been proposed to represent spatial knowledge at this layer.

Table 2 summarizes the characteristics of several prototyped image database systems. Their key features are highlighted in the table. One of our observations from this table is that the underlying design philosophy of these systems is driven by the application domain. Development of a general purpose, automatic image database system capable of supporting arbitrary domains is a challenging task due to the limitations of existing image processing knowledge representation models.

Video Data Modeling and Retrieval

The key characteristic of video data that makes it different from temporal data such as text, image, and maps is its spatial/temporal semantics. Video queries generally contain both temporal and spatial semantics. For example in the query "*Find video clips in which the dissection of liver is described,*" *dissection* is a spatiotemporal semantic. An important consideration in video data modeling is how to specify such semantics and develop an efficient indexing mechanism. Another critical issue is how to deal with the heterogeneity that may exist among semantics of such data due to difference in the preconceived interpretation or intended use of the information given in a video clip by different sets of users. Semantic heterogeneity has proved to be a difficult problem for conventional databases, with little or no consensus on the way to tackle it in practice. In the context of video databases, the problem is exacerbated.

In general, most of the semantics and events in a video data can be expressed by describing the interplay among physical objects in time along with spatial relationships between these objects. Physical objects include persons, buildings, and vehicles. In order to model video data, it is essential to identify the component physical objects and their relationships in time and space. These relations may subsequently be captured in a suitable indexing structure, which may then be used for query processing.

In event-based semantic modeling and knowledge representation issues in video data, we consider two levels of modeling: low level and high level, as shown in Fig. 7. The low-level modeling is concerned with the identification of objects, their relative movements, and segmentation and grouping of video data using image processing techniques. The high-level modeling is concerned with identifying contents and event-based semantics associated with video data and representing these contents in conjunction with suitable structures for indexing and browsing. At this level, knowledge-based approaches can be used to process a wide range of content-based queries. Browsing models and structures can be used to allow users to navigate through groups of video scenes.

The current approaches for low-level video data modeling can be further classified into two categories based on the types of processing carried out on the raw video data. The first approach is *coarse grained* and uses various video parsing techniques for segmenting video data into multiple shots. These shots are subsequently grouped for building higher-level events. The second approach is *fine grained* and is primarily based on the motion analysis of objects and faces recognized in video data.

Coarse-Grained Video Data Modeling Based on Segmentation. In this approach based on global features, video data are analyzed using image processing techniques. These techniques are applied at the frame level, and any significant

Table 3. Survey of Different Video Database Models

	Spatial Temporal Models (Event Representation)	Modeling Approach	Mode of Capturing	Query Specification
Smoliar et al.	Predefined SCD-based model	Parsing, segmentation	Automatic	Visual browsing tool
Yeung et al.	Hierarchical scene transition graph	Parsing, segmentation	Semiautomatic	Visual browsing tool
Golshani et al.	Algebraic	Object identification and motion analysis	Automatic	Algebraic expressions
Day et al.	Spatiotemporal logic using objects and events	Object identification and motion analysis	Manual	Logical expressions
Bimbo et al.	Spatiotemporal logic using objects and events	Object identification and motion analysis	Semiautomatic	By sketch
Oomoto et al.	Algebraic using video objects	Segmentation	Manual	Visual SQL based
Weiss et al.	Algebraic using video expressions	Segmentation	Manual	Algebraic expressions

change in global features in a sequence of frames is used to mark a change in the scene. This process allows parsing and automatic segmentation of video into shots. For this reason it is often termed as *scene change detection* technique. Most of the existing approaches to scene change detection use color histograms as the global feature (13). In other words, a shot is defined as a continuous sequence of video frames that have no significant interframe difference in terms of their visual content (13). Subsequently shots are used to construct scenes and episodes and to build browsing structures for users to navigate through the video database.

In order to develop high-level semantics based on this technique (Fig. 7), scenes are clustered based on some desired semantics, and descriptions are attached to these clusters. There are several ways to build this abstraction. One possibility is to identify key objects and other features within each scene using either image processing techniques or textual information from video caption, in case it is available. Domain specific semantics can be provided in form of sketches or *reference frames* to identify video segments that are closely related to these frames. Reference 14 takes advantage of the well-structured domain of news broadcasting to build an *a priori* model of *reference frames* as a knowledge base to semantically classify the video segments of a news broadcast. Alternatively, the scenes of the segmented video can be examined manually in order to append appropriate textual description. Such description can then be used to develop high-level semantics and events present in different scenes.

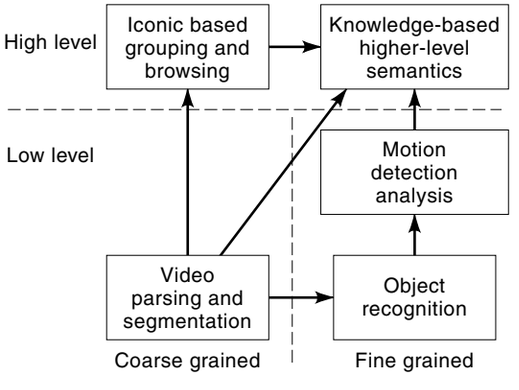

Figure 7. Semantic modeling of video data.

Video segmentation techniques are also suitable for building iconic-based browsing environments. In this case a *representative frame* of each scene can be displayed to the user in order to provide the information about the persons and possible event present in that scene (15).

Fine-Grained Video Data Modeling. In this approach, as shown in Fig. 7, detailed temporal information of objects and persons is extracted from the video data in order to identify high-level events and semantics of interest. In the following sections we elaborate on this modeling paradigm.

Low-Level Modeling. The main function of this layer is to identify key objects and faces and perform motion analysis to track their relative movements. For this purpose each video frame is analyzed either manually or using image processing techniques for automatic recognition of objects and faces. The major challenge in this approach is to track the motion of objects and persons from frame to frame and perform detailed motion analysis for temporal modeling.

Several approaches have been proposed in the literature to track motion of objects. Here we elaborate on two techniques. In one of these approaches the known compression algorithms are modified to identify objects and to track their motion. Such "semantic-based" compression approaches combine both image processing and image compression techniques. For example, in Ref. 16 a motion tracking algorithm uses forward and backward motion vectors of macroblocks used by an MPEG-encoding algorithm to generate trajectories for objects. These trajectories are subsequently used by the higher layer for semantic modeling.

The second approach for motion tracking uses a directed graph model to capture both spatial and temporal attributes of objects and persons. The proposed model, known as video semantic directed graph (VSDG), is used to maintain temporal information of objects once they are identified by image processing techniques. This is achieved by specifying the changes in the 3-D projection parameters associated with the bounding volume of objects in a given sequence of frames. At the finest level of granularity, these changes can be recorded for each frame. Although such a fine-grained motion specification may be desirable for frame-based indexing of video data, it may not be required in most of the applications. In addition the overhead associated with such detailed specification may be formidable. Alternatively, a coarse-grained temporal specification can be maintained by only analyzing

frames for motion tracking at some fixed distance apart. Such skip distance depends on the complexity of events. There is an obvious trade-off between the amount of storage needed for temporal specification and the detailed information maintained by the model. Both of these approaches, and several others, can be used to build high-level semantics, as discussed next.

Higher-Level Modeling of Video Data. Based on the information available from the low layer of Fig. 7, higher-level semantics can be built by the user to construct different views of the video data. There has been a growing interest in developing efficient formalisms to represent high-level semantics and event specifications as implied by the high level layer of Fig. 7. Several approaches have been proposed in the literature on this topic. The essence of these formalisms is the temporal modeling and specification of events present in video data. Semantic operators, which include logic, set, and spatiotemporal operators, are extensively used to develop such formalisms. Logical operators include the conventional boolean connectives such as *not, and, or, if-then, only-if,* and *equivalent-to.* Set operators like *union, intersection,* and *difference* are mostly used for event specification as well as for video composition and editing. Spatiotemporal operators, based on temporal relations, are employed for event specification and modeling. There are a total of 13 such possible operators, as shown in Fig. 5. In essence the approaches proposed in the literature use subsets and combinations of these operators.

Temporal Interval-Based Video Modeling. In this section we describe the approaches of video models based on temporal intervals. The first approach is based on spatiotemporal logic and uses temporal and logical operators for specifying video semantics. The second approach uses spatiotemporal operators with set-theoretic operators to specify video events in form of algebraic expressions. Such operations include merge, union, intersection, and so on. As a result of set-theoretic operations, this approach is also useful for video production environments. In this category we discuss three distinct models. In our opinion, these are among the most comprehensive frameworks that are representations of other models in the field.

Spatio-temporal Logic. An approach that uses spatial relations for representing video semantics is spatiotemporal logic (17). In this approach each object identified in a scene is represented by a symbol, and scenes are represented by a sequence of state assertions capturing the geometric ordering relationships among the projections of the objects in that scene. The assertions specify the dynamic evolution of these projections in the time domain. The assertions are inductively combined through the boolean connectives and temporal operators. Temporal and spatial operators, such as *temporal / spatial eventually* and *temporal / spatial always* are used for modeling video semantics in an efficient manner. Fuzziness and incomplete specification of spatial relationships are handled by defining multi-level assertions that provide general to specific detail of event specifications.

For temporal modeling of video data, Ref. 11 uses the notion of generalized temporal intervals initially proposed in Ref. 18. The temporal specification of events in this approach is equivalent to the detailed event specifications of the approach discussed above. A generalized relation, known as *n-ary* relation, is a permutation among n intervals, labeled 1 through n. The basis for this realization is that two consecutive intervals satisfy the same temporal relation, which is being generalized. The *n-ary* relations are used to build the video semantics in form of a hierarchy. For this purpose, *simple temporal events* are first constructed from spatial events with a special condition that the *n-ary* operators are of type *meets* and all operands of a certain operation belong to the same spatial event. This allows one to represent the "persistence" of a specified spatial event over a sequence of frames, which gives rise to a simple temporal event that is valid for the corresponding range of frames with some duration. In order to recognize whether or not a *simple event* is present in video data, the constructed event is evaluated using the spatial and motion information of objects, captured in the VSDG model.

Algebraic Models. These approaches use the temporal operators in conjunction with set operations to build formalisms that allow semantic modeling as well as editing capabilities for video data. For example, the framework discussed in Ref. 16 defines a set of algebraic operators to allow spatiotemporal modeling as well as video editing capabilities. In this framework temporal modeling is carried out by the spatiotemporal operators. These operators are usually defined through functions that map objects and their trajectories into temporal events. Based on lisplike operators for extracting items and lists, functions can be defined in order to perform various video editing operations such as inserting video clips, and extracting video clips and images from other video clips.

Another algebraic video model is proposed in Ref. 19. The model allows hierarchical abstraction of *video expressions* representing scenes and events, which can provide indexing and content-based retrieval mechanisms. A video expression, in its simplest form, consists of a sequence of frames defined on raw data, which usually represent a meaningful scene. Compound video expressions are constructed from simpler ones through algebraic operations, which include creation, composition, and description operators that form the basis of this formalism. Composition operators include several temporal and set operations. The set operators allow performing set operations on various video segments represented by expressions. These operators can be used to generate complex video expressions according to some desired semantics and description. Content-based retrieval is maintained through annotating each video expression with field name and value pairs that are defined by the user.

A similar approach is taken to develop an object-oriented abstraction of video data in Ref. 20. A video object in this approach is identical to a video expression in Ref. 21 and corresponds to semantically meaningful scenes and events. An object hierarchy is built using IS-A generalizations and is defined on instances of objects rather than classes of objects. Such generalizations allow grouping of semantically identical video segments. Hierarchical flow of information in this model is captured through interval inclusion based inheritance, where some attribute/value pairs of a video object A is inherited by video object B if the video raw data of B is contained in that of A. Set operators supporting composition operations, including interval projection, merge, and overlap constructs, are used for editing video data and defining new instances of video objects.

In modeling of video data, some degree of imprecision is intrinsic. To manage such imprecision, approach discussed in Ref. 17 uses a multilevel representation of video semantics as

mentioned earlier. The third level in that approach supports the most precise and detailed representation of spatio-temporal logic. This representation is equivalent to the one proposed in Ref. 17. The approach of Ref. 17, in practice, reduces to the one of Ref. 11 in the course of query evaluation. However, the approach of Ref. 17 is more pragmatic in terms of query formulation using visual sketches which provide an easier and more intuitive interface.

The model presented in Ref. 16 has a limitation in the sense that it puts the burden on the user to define semantic functions related to the video object. Furthermore these functions must be defined in terms of object trajectories. Ultimately the high-level functions need to be evaluated to provide precise results for query processing. This necessitates reductions to low-level evaluation of algebraic functions that must be specified by the users.

The other two algebraic approaches (19,20) require interactive formulation of video semantics by the user. These approaches afford us great flexibility in identifying the desired semantics. At the same time they put a great burden of responsibility on the user for such formulation, which may not be suitable for naive users. In addition these approaches may prove to be impractical for processing large amounts of video data because of the high cost of human interaction.

MULTIMEDIA DOCUMENT MODELING AND RETRIEVAL

An important problem that the multimedia community has to address is the management of multimedia documents. It is a general anticipation that parallel to the explosive growth in computer and networking technologies, multimedia repositories will soon become a reality and easy access to multimedia documents will make it essential to formally develop metaschema and indexing mechanisms for developing large-scale multimedia document management systems.

As mentioned earlier, an important issue for managing large volumes of multimedia documents is the support of efficient indexing techniques to support querying of multimedia documents. Searching information about a document can be multidimensional and may span over multiple documents. These include searching by spatiotemporal structures, by logical organization, or by contents. For example, the query "*Find documents that show a video clip of a basketball game accompanied by a textual information about other games' results*" requires searching documents by their spatiotemporal structures. Similarly the query "*Find documents that describe the assembly process of the transmission system of a car*" requires searching document database by contents. On the other hand, the query "*Find all the other sections in this book that refers to the image of the Himalayas of Chapter 7*" requires searching within a documents based on its logical structure.

Another crucial component of multimedia document management is the integration of the data, which requires both temporal and spatial synchronizations of monomedia data to compose multimedia documents. In addition to this, logical organization of document components is desired to facilitate browsing and searching within and across documents. For managing documents, representation of composition and logical information in form of a suitable metaschema is essential for designing efficient search strategies.

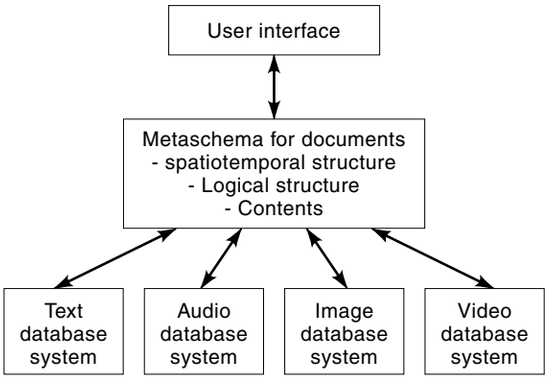

Figure 8. Generic architecture for multimedia document management system.

A generic architecture that highlights the overall process of document creation, management, and retrieval is shown in Fig. 8. Our focus here is on the second layer of this architecture which deals with the composition and management aspects of multimedia documents.

Temporal synchronization is the process of coordinating the real-time presentation of multimedia information and maintaining the time-ordered relations among component media. It is the process of ensuring that each data element appears at the required time and play-out for a certain time period. A familiar example is the voice-annotated slide show, where slides and voice data are played out concurrently.

Spatial composition describes the assembly process of multimedia objects on a display device at certain points in time. For text, graphics, image, and video, spatial composition includes overlay and mosaic, and it requires processing such as scaling and cropping. For audio data, spatial operations include mixing of signals, gain, tone adjustment, and selectively playing out various audio signals on multichannel outputs (stereo quad, etc.).

In the following sections we elaborate on two main aspects of document management; their spatiotemporal composition requirements and their organization models.

Composition Models for Multimedia Documents

In order to facilitate users to specify the spatiotemporal requirements, at the time of authoring a document, a composition model is needed. Recently various such models have been proposed in the literature, which include language-based models, time-interval based models, and object-oriented models (18,21–26).

Conceptual Models for Multimedia Objects. A number of attempts have been made to develop conceptual models for representing multimedia objects. These models can be classified into five categories: graphical models, Petri-Net based models, object-oriented models, language-based models, and temporal abstraction models. Some models are primarily aimed at synchronization aspects of the multimedia data, while others are more concerned with the browsing aspects of the objects. The former models can easily render themselves to an ultimate specification of the database schema, as briefly discussed

later in this section. Some models, such as those based on graphs and Petri-Nets, have the additional advantage of pictorially illustrating synchronization semantics, and they are suitable for visual orchestration of multimedia presentations. These models are discussed next.

Graphical Models. Labeled directed graphs have been extensively used to represent information (27). Hypertext systems provide an example of such a mechanism. This approach allows one to interlink small information units (data) and provides a powerful capability for users to navigate through a database. Information in such a system represents a "page" consisting of a segment of text, graphics codes, executable programs, or even audio/video data. All the pages are linked via a labeled graph, called hypergraph. The major application of this model is to specify higher-level browsing features of multimedia system. The essence of hypertext is a nonlinear interconnection of information, unlike the sequential access of conventional text. Information is linked via cross-referencing between keywords or subjects to other fragments of information. An application has been implemented (28) for interactive movies by using the hypertext paradigm.

Various operations, such as updating and querying, can be performed on a hypergraph. Updating means changing the configuration of the graph and the content of the multimedia data. Querying operations include navigating the structure, accessing pages (read or execute), showing position in the graph, and controlling side effects. Basically it is a model for editing and browsing hypertext.

The hypergraph model suffers from many shortcomings. The major drawback is that there is no specific mechanism to handle temporal synchronization among data items.

Language-Based Models. In this approach a scripting language is used to describe the spatiotemporal structure of multimedia documents. The leading example is the HyTime model that uses SGML (Standard Generalized Markup Language). HyTime has been recognized as an ISO standard for multimedia document modeling in 1986 (21). SGML has gained increasing popularity recently through the fame of its child, HTML, though it is a result of a decades long effort. SGML basically defines a framework to describe the logical layout of the information in a structured format through a user-defined markup language. Defining metastructures involves location addressing of entities within data, querying of the structure and content of documents, and most important, specification of measurement and scheduling of data contents along spatial and/or temporal axes. This last feature of the standard, and the deserved popularity of markup schemes in data representation, make HyTime the ideal choice for multimedia document specification (21). On the other hand, the multimedia technology still lacks "HyTime-aware" methodologies capable of creating and analyzing HyTime documents from the database management points of view.

A number of researchers have reported work involving SGML/HyTime structures (22,24). They mainly concentrate on document modeling and integrating HyTime-based information with databases. In their work, Özsu et al. describe a database application of SGML/HyTime documents for news-on-demand applications (24). The documents follow a fixed logical structure, and the document database is restricted to a certain schema. The document units are mapped into database objects in conformance with a predefined type hierarchy. Their work emphasizes the importance of spatial and temporal analysis and indexing of multimedia documents but does not propose any approach to address this issue.

In Reference 22 takes an alternative approach to the same problem: Storage and processing of structured documents within a DBMS framework is presented. This approach realizes the advantages of a general purpose scheme by a document insertion mechanism using super *Document Type Descriptors* that allows handling of arbitrary documents in the database (22). Like the new-on-demand application in Ref. 24, the scheme uses an object-oriented DB manager called VODAK. Spatiotemporal indexing is explicitly referenced as an important research problem, although no specific results have been reported. However, content-based and general indexing is briefly mentioned.

In sum, the HyTime standard is expected to play a major role in leading the research activities in multimedia document modeling. However, the management aspects of HyTime-based documents in terms of searching and indexing are open research issues.

Interval-Based Models for Multimedia Documents. Recently the use of Petri-Nets for developing conceptual models and browsing semantics of multimedia objects (18,25) has been proposed. The basic idea in these models is to represent various components of multimedia objects as places and describe their interrelations in the form of transitions. These models have been shown to be effective for specifying multimedia synchronization requirements and visualizing the composition structure of documents.

One such model is used to specify object-level synchronization requirements. It is both a graphical and mathematical modeling tool capable of representing temporal concurrency of media. In this approach Timed Petri-Net has been extended to develop a model that is known as Object Composition Petri-Nets (OCPNs); see Ref. 18. The particularly interesting features of this model are the ability to capture explicitly all the necessary temporal relations. Each place in this Petri-Net derivative represents the play-out of a multimedia object, while transitions represent synchronization points.

Several variations of the OCPN model have been proposed in the literature. One such variation deals with the spatial composition aspects of multimedia documents. For such composition, additional attributes are specified with each media place in the OCPN. These include the size and location of the display area for different media within a document, a priority vector that describes the relative ordering among changing background/foreground locations of intersecting spaces for media display with time; an ordered list of unary operations, such as crop and scale, applied to the data associated with the place, and a textual description about the contents of the media place.

As mentioned, the HyTime model suffers from a drawback, that the extraction of various spatiotemporal and content semantics from this model can be quite cumbersome. On the other hand, the OCPN model not only allows extraction of the desired semantics and generation of a database schema but also has the additional advantage of pictorially illustrating synchronization aspects of the information. In this regard this model is unique and therefore is also well suited for visual orchestration of multimedia document.

Organization Models for Multimedia Documents. From organizational structure point of view, a multimedia document

can be viewed as a collection of related information objects, such as books, chapters, and sections. The logical structure of objects can be maintained in the form of a metaschema associated with each document. Metainformation about such organization can be used for searching and accessing different parts of a document. Models for the logical structure of multimedia documents can be independent from the composition models. Such independence can support different presentation styles for a document that can be tailored to the target audience, as well as hardware display constraints.

The well-known organizational modeling paradigm of documents is based on hypermedia. There are basically three types of links used in a hypermedia environment. These include the base structure links for defining the organization of documents, the associative links for connecting concepts and accessing the same information from different contexts, and referential links that provide additional information on a concept within a document.

The HyTime model provides an elegant mechanism for the organizational structure of a document. Using SGML, a document's logical content is described by specifying the significant elements in that document along with the attributes associated with each such element, in a hierarchical manner. For example, an SGML specification of a textual report document may declare that it contains a title, an author, and a body. Each of these elements would in turn have attributes specifying their structure.

The hypermedia-based multimedia document models have several attractive features. For example, they allow efficient path-searching mechanisms for accessing information in various parts of the document (23). Furthermore they allow the development of object-oriented abstractions of documents. For this purpose the document components are represented in form of a set of nodes related to each other through IS-A, IS-PART-OF, and AGGREGATE relationships. Associated with each node is a concept or a topic, and the semantic relationships among nodes are based on concepts. In other words, each node in this model is an information unit, and object-oriented abstractions between two nodes can be represented using structural links.

Several hypermedia-based models of documents, with object-oriented abstractions have been proposed in the literature (22–25). The model presented in Ref. 22, in essence, is a HyTime model, as discussed earlier. Its hypermedia-based organization has been used to develop a multilayered architecture, known as VODAK. The layers consist of a conceptual schemata level for accessing several multimedia databases, a second level that supports document authoring environment by conceptualizing media objects, and a third level for the presentation of documents. The limitation in the design of VODAK system is that there is no explicit mechanism of supporting query based on contents associated with objects in a document.

Recently the researchers in Ref. 23, have proposed a hypermedia-based document model that uses the object-oriented paradigm. They describe a unique indexing scheme based on the underlying multistructure information of document to optimize the index structure and to provide efficient access document elements. The document data model can be implemented using object-oriented technology. The model is augmented with an object-oriented query language syntax.

CONCLUSION

We have covered several issues pertaining to rapidly evolving multimedia information technology, namely data modeling, storage, indexing and retrieval, and synchronization of multimedia data. It is now widely accepted that one of the main requirements of multimedia information systems is a data model more powerful and more versatile than the relational model, without compromising the advantages of the former. The relational data model exhibits limitations in terms of complex object management, indexing and content-based retrieval of video/image data, and facility for handling the spatiotemporal dimensions of objects. To address these issues, we have emphasized two key requirements for multimedia databases: the process of spatiotemporal modeling, and the computational needs for automatic indexing of spatiotemporal data. Enlisted were the general characteristics of a number of different media types with the notion of time identified among those as the major characteristic that also distinguishes multimedia data from traditional alphanumeric data. We have highlighted various challenges that need to be tackled before multimedia information systems become a reality. This area is expected to preserve it popularity into the next millenium and produce visible outcomes that will find direct and pragmatic usage in our lives.

BIBLIOGRAPHY

1. A. Ghafoor and P. B. Berra, Multimedia database systems, in B. Bhargava and N. Adams, (eds.), *Lecture Notes in Computer Science,* Vol. 759, New York: Springer-Verlag, 1993, pp. 397–411.
2. T. D. C. Little and A. Ghafoor, Synchronization and storage models for multimedia objects, *IEEE J. Select. Areas Commun.,* **8** (3): 413–427, 1990.
3. J. S. Sventek, An architecture for supporting multi-media integration, *Proc. IEEE, Comput. Soc. Office Automation Symp.,* 1987, pp. 46–56.
4. R. G. Herrtwich, Time capsules: An abstraction for access to continuous-media data, *Proc. 11th Real-Time Syst. Symp.,* 1990, pp. 11–20.
5. D. J. Moore, Multimedia presentation development using the audio visual connection, *IBM Syst. J.,* **29** (4): 494–508, 1990.
6. M. E. Hodges, R. M. Sasnett, and M. S. Ackerman, A construction set for multimedia applications, *IEEE Softw.,* **6** (1): 37–43, 1989.
7. M. Flickner et al., Query by image and video content: The QBIC system, *Computer,* **28** (9): 23–32, 1995.
8. V. E. Ogle and M. Stonebraker, Chabot: Retrieval from a relational database of images, *Computer,* **28** (9): 40–48, 1995.
9. C. C. Hsu, W. W. Chu, and R. K. Taira, A knowledge-based approach for retrieving images by content, *IEEE Trans. Knowl. Data Eng.,* **8**: 522–532, 1996.
10. M. Misra and V. K. Prasanna, Parallel computations of wavelet transforms, *Proc. Int. Conf. Pattern Recognition,* 1992.
11. Y. F. Day et al., Spatio-temporal modeling of video data for online object-oriented query processing, *IEEE Int. Conf. Multimedia Comput. Syst.,* 1995, pp. 98–105.
12. A. Yoshitaka et al., Knowledge-assisted content-based retrieval for multimedia database, *IEEE Multimedia,* **1** (4): 12–21, 1994.
13. A. Nagasaka and Y. Tanaka, Automatic video indexing and full video search for object appearances, *2nd Working Conf. Visual Database Syst.,* 1991, pp. 119–133.

14. S. W. Smoliar and H. Zhang, Content-based video indexing and retrieval, *IEEE Multimedia,* **1** (2): 62–74, 1994.

15. M. M. Yeung et al., Video browsing using clustering and scene transitions on compressed sequences, *Proc. IS&T / SPIE Multimedia Computing and Networking,* 1995, pp. 399–413.

16. F. Golshani and N. Dimitrova, Retrieval and delivery of information in multimedia database systems, *Inf. Softw. Technol.,* **36** (4): 235–242, 1994.

17. A. Del Bimbo, E. Vicario, and D. Zingoni, Symbolic description and visual querying of image sequences using spatio-temporal logic, *IEEE Trans. Knowl. Data Eng.,* **7**: 609–622, 1995.

18. T. D. C. Little and A. Ghafoor, Interval-based conceptual models for time-dependent multimedia data, *IEEE Trans. Knowl. Data Eng.,* **5**: 551–563, 1993.

19. R. Weiss, A. Duda, and D. K. Gifford, Composition and search with a video algebra, *IEEE Multimedia,* **2** (1): 12–25, 1995.

20. E. Oomoto and K. Tanaka, Ovid: Design and implementation of a video-object database system, *IEEE Trans. Knowl. Data Eng.,* **5**: 629–643, 1993.

21. ISO/IEC 10744, *Information Technology—Hypermedia / Time-Based Structuring Language (HyTime),* Int. Organ. for Standardization, 1992.

22. W. Klas, E. J. Neuhold, and M. Schrefl, Using an object-oriented approach to model multimedia data, *Comput. Commun.,* **13** (4): 204–216, 1990.

23. K. Lee, Y. K. Lee, and P. B. Berra, Management of multi-structured hypermedia documents: A data model, query language, and indexing scheme, *Multimedia Tools Appl.,* **4** (2): 199–223, 1997.

24. M. T. Ozsu et al., An object-oriented multimedia database system for a news-on-demand application, *ACM Multimedia Syst. J.,* **3** (5/6): 182–203, 1995.

25. M. Iino, Y. F. Day, and A. Ghafoor, Spatio-temporal synchronization of multimedia information, *IEEE Int. Conf. Multimedia Comput. Syst.,* 1994, pp. 110–119.

26. ISO 8613, *Information Processing—Text and Office Systems–Office Document Architecture (ODA) and Interchange Format,* Int. Organ. for Standardization, 1993.

27. F. W. Tompa, A data model for flexible hypertext database systems, *ACM Trans. Inf. Syst.,* **7** (1): 85–100, 1989.

28. R. M. Sasnett, Reconfigurable video, MS thesis, Massachusetts Inst. Technol., Cambridge, MA, 1986.

WASFI AL-KHATIB
M. F. KHAN
SERHAN DAĞTAŞ
ARIF GHAFOOR
Purdue University