Pakistan Journal of Applied Sciences 2(1): 109-113, 2002

Query Processing in Multimedia Databases

Oya Kalipsiz
Yildiz Technical University, Computer Science Department Yildiz, 80750 Istanbul, Turkey

Abstract: A multimedia database management system (DBMS) provides a suitable environment for using and managing the multimedia database. Multimedia objects, which can be defined as objects that integrate values of various media, have temporal and spatial relationships. Object DBMS have provided better capabilities to manage more complex data. Multimedia Query processing is relatively more complex than normal query processing in a traditional context in which only formatted data are considered. The main problem is the heterogeneity of data processed by the multimedia system. Data with different characteristics, such as video and text, are characterized by different storage techniques and different access structures. This article focuses on defining and discussing query processing in multimedia database systems.

Keywords: Information Systems, Multimedia Systems, Multimedia Databases

Introduction

A database is a collection of related data. A database management system (DBMS) is a general-purpose software system that facilitates the processes of defining, constructing and manipulating databases for various applications (Elmasri and Navathe, 1989; Korth and Silberschatz, 1991). Multimedia databases contain different data types such as text, images, graphic objects, animation sequences, audio and video. The different data types might require special methods for optimal storage, access, indexing and retrieval (Apers, Blanken and Houstma, 1997).

Normally, individual objects in an image or video frame have some kind of a spatial relationship between each other. These relationships produce some constraints when searching for objects in a database. Relational or object oriented data models have been proposed to cope with these constraints in the multimedia databases. An object oriented database management system is more suitable for developing multimedia applications. Object DBMS have provided better capabilities to manage more complex data requiring user- defined data types used in engineering designs and software configuration management (Pazandak and Srivastava, 1997).

To understand the requirements of a multimedia database, we need to know the types of multimedia data that can be stored and managed. Multimedia data is divided into two classes: continuous and discrete. Continuous media such as video and audio change with time. Discrete media are time independent. Common examples of discrete media are formatted text, unformatted text, still images and graphics (Kalipsiz, 2000).

Common multimedia data types found in a multimedia database are text, graphics, images, pictures and photographs, animation, video, audio (Adjeroh and Nwosu, 1997; Berra and Nwosu, 1997). Characteristics of multimedia data are:

- Lack of structure
- Temporality
- Massive Volume
- Logistics

The development of multimedia DBMS can benefit from traditional DBMS services such as:

- Data independence (data abstraction)
- High level access through query languages
- Application neutrality (openness)
- Controlled multi-user access (concurrency control)
- Fault tolerance (transactions, recovery)
- Restriction of unauthorized access and modification of stored data (privacy)

Multimedia objects have temporal and spatial relationships that must be taken into account for synchronization and display of information. These relationships should be modeled explicitly as part of the stored data. Thus, even if the multimedia data is stored in files, their relationships need to be stored as part of the metadata in some DBMS (Apers, Blanken and Houstma, 1997; Nikolovska and Martino, 1998).

A multimedia database management system (DBMS) provides support for the creation, storage, access, query and control of multimedia database. The following is a list of requirements for multimedia DBMS:

- Multimedia data modeling
- Multimedia object storage
- Multimedia indexing, retrieval, browsing
- Multimedia query support

Multimedia DBMS architecture meets the multimedia data requirements (Fig.1) (Berra and Nwosu, 1997). Developing a variety of multimedia database applications requires a powerful and flexible general-purpose hypermedia development environment. Multimedia databases have special requirements such as (Adjeroh and Nwosu, 1997; Apers, Blanken and Houstma, 1997; Berra and Nwosu, 1997):

- Traditional DBMS capabilities
- Huge capacity storage management
- Information retrieval capabilities
- Multimedia query support
- Media integration, composition and presentation
- Multimedia interface and interactivity

The query is one of the most important parts of DBMS. Multimedia query languages must deal with complex spatial and temporal relationships inherited in the wide range of multimedia data types. Powerful query languages could help manipulate multimedia DBMS and maintain the desired independence between the database and the application.

This paper describes the architecture and requirements of multimedia DBMS, object oriented multimedia databases and discusses query and indexing in multimedia DBMS

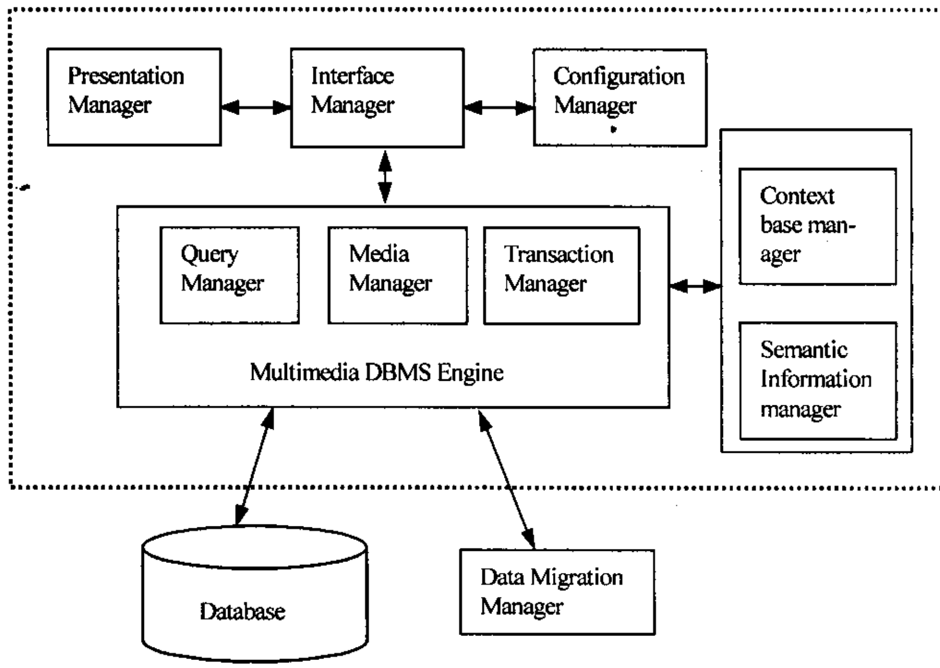

Fig. 1: The Architecture for Multimedia DBMS

Materials and Methods

Multimedia Data Model: Each component of multimedia entity has attributes and can participate in relationships. Using content-based attributes and relationships significantly expands the types of browsing and filtering operations you can invoke as a user. You could model the querying process as a sequence of filtering and browsing operations. In a multimedia information system, alphanumeric information, images, videos and audios are treated equally from the standpoint of query processing. Each may participate in a query, and each may be part of a query's output. We should represent multimedia information using an object oriented data model of some sort. There are five types of information such a data model should represent (Grosky, 1994; Pazandak and Srivastava, 1997):

- Uninterpreted multimedia information
- Multimedia-related, content-independent information
- Alphanumeric information
- Relationships between non-multimedia, real-world application entities and multimedia objects
- Methods of constructing and representing multimedia-world relationships

There are three repositories of information (Grosky, 1994):

- **A standard alphanumeric database:** Holds information concerning non-multimedia real world application objects.
- **A multimedia object database:** Contains the uninterpreted multimedia objects as well as multimedia-related content independent information.
- **A feature Database:** Contains features extracted by the feature processing module and used for content-based retrieval.

Object Multimedia Database: A data model must isolate users from the details of storage device management and data structures. In object-oriented data modeling approach, a database is considered as a collection of objects where each object represents a physical entity, a concept, an idea, or an event. In classical relational data models, data is viewed as a collection of record types, each having a collection of records stored in a file. In an object-oriented system, real-world objects are represented directly by database objects. Object identity is maintained via an object identifier. The interface between an object and the rest of the system is defined by a set of messages.

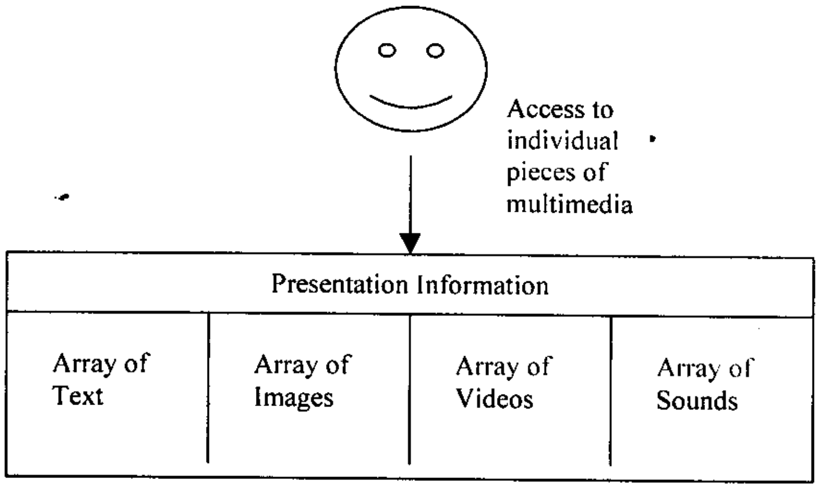

Fig.2. Modeling Stuructures Used To İncorporate Multimedia Data Types

A method, which is a body of code to implement, processes each message. A method returns a value as the response to the message. The process of classification involves classifying similar objects into object classes. A subclass inherits both data attributes and methods from its superclasses in the hierarchies (Gray, Kulkarni and Paton, 1992).

Multimedia data models capture the static and dynamic properties of the database contents. Static properties include the objects, their attributes and the relationships between the objects. Dynamic properties include interaction between the objects, operation on objects and user interaction. Relational and object-oriented data models are both used for multimedia databases (Grosky, 1997; Pazandak and Srivastava, 1997).

When stored as large objects, multimedia data can be directly associated with database objects much like textual or numerical data. This greatly simplifies database querying because multimedia data accessed directly from an object rather than having to link other relations. Relating several items of multimedia data to one object is still a problem for developers, one they can overcome through the built-in structured data type array. An object can be associated with type image or sound arrays. To implement the concept of packets for all related data, another attribute stores data in a multidimensional array that contains information for all objects in each packet (Fig. 2).

Object database management systems (ODBMS) have provided better capabilities to manage more complex data requiring user-defined data types. Supporting multimedia data will require changes to an ODBMS's software, as well as its operating systems, computer hardware and networks.

Some typical applications that might use an ODBMS to manage multimedia data are listed as (Pazandak and Srivastava, 1997):

- Data repositories: Data repositories do not need to understand the stored data formats because they don't operate on the data. The objects are stored as binary large objects. Examples of data repository applications are pseudo repository, electronic mail, and healthcare information systems.
- Intelligent data management: ODBMS can query multimedia object content. An example for intelligent data management is engineering design workflow.

ODBMS technology provides an extensible data model, which gives designers the ability to support dynamic schema changes. It provides easy hierarchical relationships like classification, generalization and aggregation between multimedia objects.

Query Processing and Indexing in Multimedia Database: A query that is expressed in a high-level query language, such as SQL, must first be scanned, parsed and validated. The scanner identifies the language components in the text of a query, while the parser checks the syntax of the query to determine whether it is formulated according to the syntax rules of the query language. An internal representation of a query is usually created as a tree or graph structure. The DBMS must then devise an execution strategy for retrieving the result of the query from the internal database files. An execution strategy is a plan for executing the query, accessing data and storing the intermediate results (Elmasri and Navathe, 1989; Korth and Silberschatz, 1991).

Multimedia Query Processing: Multimedia Query processing is relatively more complex than normal query processing in a traditional context in which only formatted data are considered. The main problem is the heterogeneity of data processed by the multimedia

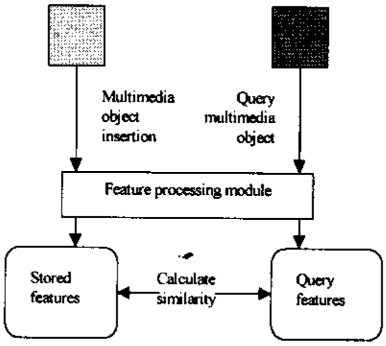

Fig. 3: A Multimedia Object As A Part Of Query

system. Data with different characteristics, such as video and text, are characterized by different storage techniques and different access structures. This variety greatly influences the choice of the best query execution strategy. Any MM Query Processing strategy can be classified according to (Apers, Blanken and Houstma, 1997):

- The used storage and access methods
- The kind of query considered predicates
- The use of a partial or exact matching between query conditions and data
- The approximation used both in defining data structures and computations
- The extensibility degree of the processor

Efficient query processing is mainly related to the following issues:

- Efficient storage methods
- Efficient access methods

Storage of multimedia data represents a variety of challenges and requires the development of methods to manage, distribute and transfer large volumes of data with varying requirements. The main differences with respect to the traditional storage methods can be summarized as follows:

- The need for high storage capacity
- The requirement for a very high transfer rate at playback time and physical organization techniques
- The need to combine and synchronize several multimedia data streams at presentation time

The definition of efficient access methods is a fundamental point to speed the evaluation of queries retrieving a small subset of data from a large database. Efficiency of query processing can be improved by allocating additional data structures to represent the different kinds of data and by constructing access structures to support random access.

In a multimedia Information system, multimedia objects can participate in any query. Since a feature-processing module extracts features of all inserted multimedia objects, this same module can similarly extract features from a multimedia object that is part of a query (Fig. 3).

Multimedia data queries can be subdivided into following types: keyword querying, semantic querying, visual query and video query. Keyword querying only uses well-defined queries. Semantic and visual querying designed to use the fuzzy query method. Visual query language is useful, because it can formalize complicated queries. Future multimedia retrieval systems will have to support access to information at different semantic levels to reflect diverse application needs and user queries.

Video Query: As digital video databases become more and more spread, finding video in large databases becomes a major problem. Because of the nature of video, accessing the content of such databases is inherently a time-consuming operation. Efficient retrieval of video in a terabyte database is a great challenge, solutions certainly involve technologies from various disciplines, including database management, image processing, pattern recognition, data security, networking, human-computer interactions and user interfaces (Gupta, Santini and Jain, 1992).

The indexes such as bibliographic, structure and content data are identified in order to satisfy the queries. Bibliographic data category includes information about the entire video and traditional metadata. A video query is more complicated than a traditional query of text databases. In addition to text, a video clip has visual and audio information as well as the dynamics associated with the presentation of such information. In any video retrieval system, most important concerns are as follows (Gupta, Santini and Jain, 1992; Yeo and Yeung, 1997):

- Queries must be natural and easy to formulate
- The user-computer interface should assist in the query-formulation process
- The search results must be presented in an organized and sensible fashion.
- The search should be performed quickly

The sequence of stages for a video query is navigating - searching - browsing and viewing (Bolle, Yeo and Yeung, 1998). Navigating is the stage where the user decides which category of video is to be searched. It is the capability to use metadata. Searching is the most important part of query. The result of a search is the list of candidate units that satisfy the constraints of the query. In the browsing stage, representations of video should be displayed and the user can quickly understand the video content. In the viewing stage, the usual functions of videocassette players and capabilities like semantic fast-forward should be available.

Results and Discussion

The problem we focus on in this article is the design of fast searching methods that will search a database of multimedia objects to locate objects that match a query object. Similarity queries can been classified into two categories as whole match and sub-pattern match. This application needs a multidimensional indexing method that works for large, disk-based databases.

Indexing of Multimedia Data: Multimedia data must be interpreted before it can be queried. This process demands sophisticated indexing schemes and image and audio analysis algorithms to generate content descriptions. Users may want to query for images that are similar to a certain object or involve specific actions such as running. Thus querying requires mechanisms to generate indices, interfaces and languages to pose queries, and underlying components to optimize queries.

The ideal indexing method should fulfill the following requirements (Apers, Blanken and Houstma, 1997; Elmasri and Navathe, 1989; Korth and Silberschatz, 1991):

- It should be fast
- It should be correct
- It should be require a small space overhead
- It should be dynamic

Querying image database by their content needs a multidimensional indexing method that works for large databases. The R-tree based methods seem to be most robust for higher dimensions. R-tree is an extension of the B-tree for multidimensional objects (Apers, Blanken and Houstma, 1997).
The proposed approach to indexing multimedia objects for fast similarity searching is presented by the Algorithm GEMINI (generic Multimedia object IndexIng). The Algorithm has four steps:

1. Determine a distance function *D()* which should measure the dis-similarity between two objects O_1, O_2
2. Find one or more numerical featurextraction functions, to provide *a* '**quick and dirty**' test
3. Prove that the distance in feature space **lower-bounds** the actual distance *D()*, to guarantee correctness
4. Choose a SAM (Spatial Access Methods) like the R*-trees, and use it to manage the *f-d* feature vectors.

Content - Based Indexing and Retrieval: Users of video-retrieval systems want to find videos on the basis of the semantic content of the video. Content-based image retrieval systems typically let users desired images from a collection on the basis of primitive features representing color, texture or shape (Bolle, Yeo and Yeung, 1998; Yeo and Yeung, 1997).
A retrieval system should embed a semantic level reflecting as much as possible the one human refers to during interrogation. The most common way to enrich a visual information retrieval system's semantics is to annotate pictorial information manually at storage time through a set of external keywords describing the pictorial content (Colombo, 1999).
Content-based image retrieval systems potentially have a wide application in areas such as engineering and architectural design, medical imaging, law enforcement, and journalism, fine art and fashion. Content-based search of an image database is well suited for searching scientific databases such as satellite-image, medical and seismic-data repositories (Boardman, Eakins and Graham, 1998; Löffler, 2000).

Conclusion

The capabilities of existing multimedia DBMS can be evaluated by their ability to support special functionalities required to manage multimedia data. The display of multimedia objects must be coordinated so that the display meets dynamic, temporal and spatial constraints.
Efficient access and storage methods are crucial for efficient query processing in multimedia database. Unlike traditional relational DBMS, multimedia DBMS require both a high storage capacity and a very high transfer rater due to its content which can be images, animation sequences, audio and video. That is why sophisticated indexing schemes and analysis algorithms are necessary to generate content descriptions and query.
Future trends and potential research topics on query processing in multimedia DBMS include; content-based and querying of multimedia objects - effective query method for spatial and temporal data - development of application specific query languages and performing indexing, retrieval and browsing directly on the compressed data. Query processing for multimedia databases is not a consolidated topic. Particular attention is given to problems related to the definition of multimedia cost models and to the interation of different approximation strategies.

References

Adjeroh D. and Nwosu K., 1997. "Multimedia Databases Management - Requirements and Issues", IEEE Multimedia.

Apers P., Blanken H. and. Houstma M, 1997. Multimedia Databases in Perspective, Springer Verlag.

Berra B., Nwosu K. and Thuraisingham K. 1997. "Multimedia Database Systems - A New Frontier", IEEE Multimedia.

Boardman J., Eakins J. and Graham M., 1998. "Similarity Retrieval of Trademark Images", IEEE Multimedia.

Bolle R., Yeo B. and Yeung M, 1998. "Video Query: Research Directions", IBM J. Res. Dev. Vol.42 No. 2.

Colombo C. *et al.*, 1999. "Semantics in Visual Information Retrieval", IEEE Multimedia.

Elmasri R. and Navathe S., 1989. Fundamentals of Databases Systems, The Benjamin/Cummings Pub.Comp.

Gray P., Kulkarni K. and Paton N. 1992. Object-Oriented Databases, Prentice Hall.

Grosky W.I., 1994. " Multimedia Information Systems", IEEE Multimedia, pp.12-24.

Grosky W.I., 1997. Managing Multimedia Information in Database Systems, Communication of the ACM, 40: 73-80

Gupta A., Santini S. and Jain R. 1997. "In Search of Information in Visual Media", Communication of the ACM, 40: 35-42.

Kalipsiz O., 2000. "Multimedia Databases", Proceedings IEEE International Conference on Information Visualisation, pp.111-116.

Korth H. and Silverschatz A. 1991. Database System Concepts, Mc Graw Hill.

Löffler J., 2000. "Content-based Retrieval of 3D Models in Distributed Web Databases by Visual Shape Information", Proceedings IEEE International Conference on Information Visualisation pp.82-87.

Nikolovska L. and Martino J., 1998. "Spatial Browsing to Retrieve Multimedia Information", IEEE Multimedia, pp.78-83.

Pazandak P. and Srivastava J., 1997. "Evaluating Object DBMSs for Multimedia", IEEE Multimedia.

Yeo B. and Yeung M., 1997. "Retrieving and Visualizing Video", Communication of the ACM, 40: 43-52.