A Distributed Database Server for Continuous Media[*]

W.G. Aref, A.C. Catlin, A.K. Elmagarmid, J. Fan, J. Guo, M. Hammad, I.F. Ilyas, M.S. Marzouk, S. Prabhakar, A. Rezgui, S. Teoh, E. Terzi, Y. Tu, A. Vakali, X.Q. Zhu

Department of Computer Sciences, Purdue University
West Lafayette IN 47907-1398
http://www.cs.purdue.edu/icds

1. Introduction

In our project, we adopt a new approach for handling video data. We view the video as a well-defined data type with its own description, parameters, and applicable methods. The system is based on PREDATOR, the open source object relational DBMS. PREDATOR uses Shore as the underlying storage manager (SM). Supporting video operations (storing, searching by content, and streaming) and new query types (query by examples and multi-features similarity search) requires major changes in many of the traditional system components. More specifically, the storage and buffer manager will have to deal with huge volumes of data with real time constraints. Query processing has to consider the video methods and operators in generating, optimizing and executing query plans. Our system includes the following features:

• A video processing tool to extract video shots, MPEG7 compatible visual descriptions (features), and semantic annotations of domain-experts (semantic descriptors).
• A general high-dimensional index mechanism to handle the extracted video features. An SR-tree index structure is implemented as our high-dimensional access path to the extracted visual features.
• A new rank-join query operator that implements a version of Fagin's optimal aggregate algorithm to support query by multiple examples and multi-feature queries.
• Search-based buffer management techniques to support continuous media streaming. A stream manager layer above the buffer manager to handle streams.
• Methods for handling different video storage hierarchies (buffer, disk, tertiary storage, and the Internet). An extention to the storage hierarchy in Shore to work with a DVD tertiary storage server.

[*]The support of the National Science Foundation under Grants IIS-0093116, EIA-9972883, IIS-9974255, and EIA-9983249, the Department of Navy under Grant N00164-00-C-0047 are gratefully acknowledged.

2. MPEG7 Compatibility

We follow the recent trend of representing the video content description in an XML-like format (according to the MPEG7 standard). The system offers a video processing toolkit which is used to process raw video streams and to extract scene cuts that partition the video into video shots. Video shots are then processed to extract the key frames and the MPEG7 visual features. The features extracted at the video shot level include texture, color, and motion. Semantic features are provided through annotations to the video streams by a domain expert and through automatic audio-to-text tools. The extracted features are saved in relational tables. We aim to include an XML wrapper to map MPEG7 files into object relational tables. In addition to the extracted content and semantic features, we store physical metadata such as resolution and quality, to enable the system to support quality-of-service.

Figure 1 provides an overview of our system architecture. It illustrates the functional components and the relationships among them. The modifications we made to the original system components to support the processing of video data are described in the following sections.

3. Query Processing

Our query model uses the features approach in accessing video by content. In the features approach, the extracted video features are mapped to a high-dimensional space, where each video unit (shot) is mapped to a point in this space. A high-dimensional index is needed to efficiently query these features.

As Shore does not provide a high-dimensional index, we extended the indexing capabilities of the Shore SM by implementing the Generalized Search Trees (GiST) general indexing framework to be used by the query processor layer in PREDATOR. We chose the GiST implementation of the

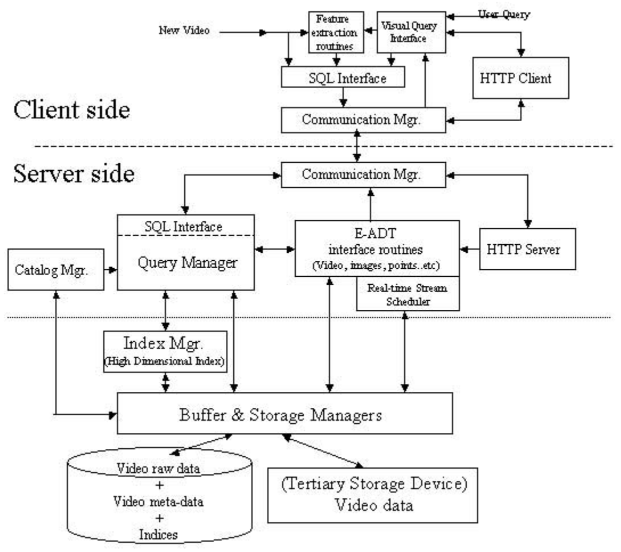
New Video
Feature extraction routines
Visual Query Interface
User Query
SQL Interface
HTTP Client
Client side
Communication Mgr.
Server side
Communication Mgr.
SQL Interface
Query Manager
E-ADT interface routines (Video, images, points..etc)
HTTP Server
Catalog Mgr.
Real-time Stream Scheduler
Index Mgr. (High Dimensional Index)
Buffer & Storage Managers
Video raw data + Video meta-data + Indices
(Tertiary Storage Device) Video data

Figure 1. System Architecture

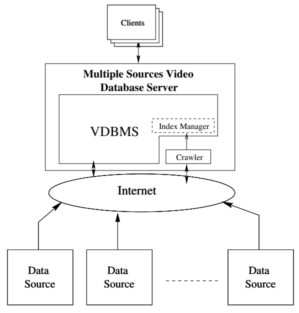

Figure 2. Multiple Sources Video Database Server Architecture

SR-tree as our high-dimensional index. Similarity search is performed by issuing nearest neighbor queries to the high-dimensional access path.

Similarity search usually involves more than one feature, hence, multi-feature similarity queries need to be handled. In our system, we propose a new query operator termed R-JOIN (rank join) for evaluating these types of queries. The operator implements a version of the optimal aggregate ranking algorithm by Fagin.

4. Buffer Management

The buffer pools are divided between the database buffer area and the streaming area where requests for streams are serviced. The buffer manager has been modified to deal with segment allocation instead of the traditional page-based approach. The replacement policy has also been modified to cache common pages between streams. A stream manager layer above the buffer manager provides support for the following: 1 - Admitting new streams requests if the maximum number of streams has not been reached. 2 - Serving current streams periodically, by issuing requests to the buffer manager. 3 - Sending the streams to the client according to a specific rate. 4 - Guiding the underlying buffer replacement and prefetching policies.

5. Storage Hierarchy

The Shore SM was extended to perform necessary video operations and to process both real-time and non real-time requests. We have also extended the storage hierarchy in Shore to work with a DVD tertiary storage server. We have implemented different caching levels on buffer and disk storage to enhance access for frequently referenced data. Tertiary resident data is directly accessible to the system. The jukeboxes can be daisy chained, allowing us to potentially manage terabytes of data. Issues that we are currently addressing include DVD data placement, prefetching, and caching.

In future and as the amount of continuous data available on the Internet is growing rapidly, we intend to consider the Internet as our source of video data. Therefore, we must address the challenges that will be introduced in the query and buffer managers. The remote *data sources* store the actual video data, and should have the corresponding content description. The *video database server* keeps summaries of numerous video and also *crawls* the Internet for new video sources. The multiple sources video database server is depicted in Figure 2.

The server needs to retrieve the video data (along with its description) and process it locally, before fully answering the query. The query manager should consider the continuous flow of video stream while processing the query. In addition, the query manager should adapt to fluctuation in the input flow and possibility of unpredictable delays.We plan to investigate the applicability of current approaches in handling video over the Internet and determine how they should be changed to meet the requirements of video data.

6. Description of the Demo

An interface has been developed to allow the user to submit queries. The system supports different types of queries like query by example(s), query by multiple visual features, query by camera motion, and query by key-words. A combination of these query types is also allowed. Our database contains 300 hours of medical data and the system features are tested against this data.