A Video Database Management System for Advancing Video Database Research[1]

Walid G. Aref, Ann C. Catlin, Jianping Fan, Ahmed K. Elmagarmid,
Moustafa A. Hammad, Ihab F. Ilyas, Mirette S. Marzouk, Xingquan Zhu

Abstract

The most useful environments for advancing research and development in video databases are those that provide complete video database management, including (1) video preprocessing for content representation and indexing, (2) storage management for video, metadata and indices, (3) image and semantic-based query processing, (4) real-time buffer management, and (5) continuous media streaming. Such environments support the entire process of investigating, implementing, analyzing and evaluating new techniques, thus identifying in a concrete way which techniques are truly practical and robust. In this paper we present a video database research initiative that culminated in the successful development of VDBMS, a video database research platform that supports comprehensive and efficient database management for digital video. We describe key video processing components of the system and illustrate the value of VDBMS as a research platform by describing several research projects carried out within the VDBMS environment. These include MPEG7 document support for video feature import and export, a new query operator for optimal multi-feature image similarity matching, secure access control for streaming video, and the mining of medical video data using hierarchical content organization.

Index terms: Video database management system, video preprocessing, feature extraction, query processing, rank-join algorithm, stream management, search-based buffer management, MPEG7 compliance, video data mining.

1.0 Introduction

A two-year video database research initiative at Purdue University has culminated in the successful development of a video-enhanced database research platform that supports comprehensive and efficient database management for digital video [1,2]. Our fundamental concept was to provide a full range of functionality for video as an intrinsic, well-defined, abstract database data type, with its own description, parameters and applicable methods. The functionality of system components, such as video preprocessing, query processing and stream management, have been implemented using algorithms and techniques developed by the VDBMS research group. A high level description of key system components is given here, and details can be found in the literature [13,18,21,38]. We define our system as a research platform, as it supports the easy implementation, integration and evaluation of new algorithms and components. Our system promotes the advance of video database technology by providing an open, flexible, robust environment for investigating new research areas related to video information systems. This concept is illustrated by a brief discussion of four research problems which were addressed within the VDBMS environment: MPEG7 document support for importing and exporting video features, a new multi-feature rank-join video query operator for image similarity matching,

[1] This work was supported in part by the National Science Foundation under Grants IIS-0093116, EIA-9972883, IIS-9974255, IIS-0209120 and by the NAVSEA/Naval Surface Warfare Center, Crane.

content-based access control for streaming video, and the mining of video data using hierarchical content organization.

VDBMS system functionality has been tested against more than 500 hours of medical videos obtained from Indiana University's School of Medicine. The videos are digitized, compressed into MPEG1 format, preprocessed off-line by VDBMS to generate image and content-based metadata, and then stored together with their metadata in the database.

2.0 The VDBMS Video Database Management System

A VDBMS query interface client supports end-user query and retrieval for the VDBMS medical video database. It connects to the database server which resides on a SUN Enterprise 450 with 4 UltraSparc II processors located at Purdue University. VDBMS is built on top of an open source system consisting of Shore [37], the object storage manager developed at the University of Wisconsin, and Predator [34], the object relational database manager from Cornell University. The VDBMS research group has developed the extensions and adaptations needed to support full database functionality for the *video* as a fundamental, abstract database data type. Key database extensions include high-dimensional indexing, video store and search operations, new video query types, real-time video streaming, search-based buffer management policies for continuous streaming, and support for extended storage hierarchies including tertiary storage. These extensions required major changes in many traditional system components. Figure 1 illustrates our layered system architecture with its functional components and their interactions. The system consists of the object storage system layer at the bottom, the object relational database management layer in the middle, and the user interface layer at the top.

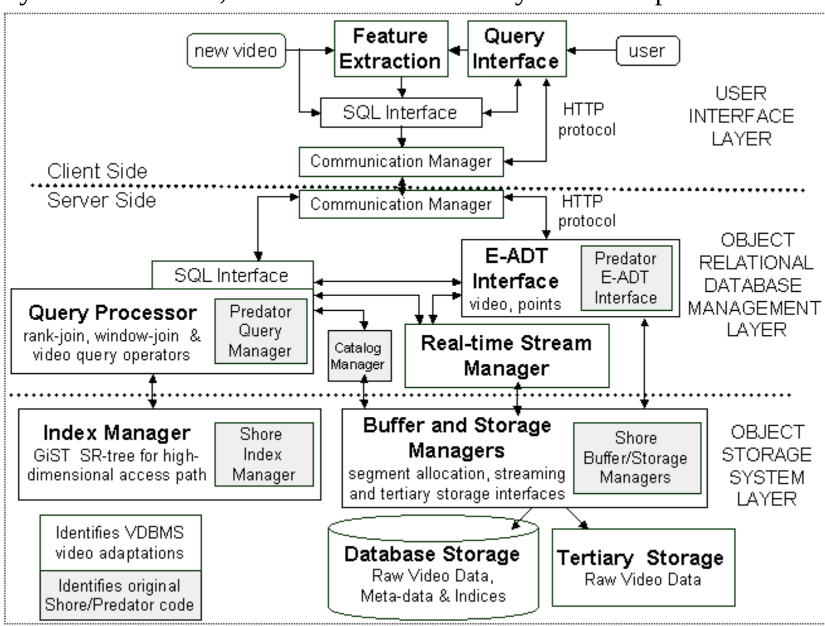


Figure 1. VDBMS layered system architecture

2.1 The User Interface and Application Layer

The top layer consists of the client side software for application layer processing. The VDBMS query interface supports content-based query, search, retrieval and real-time streaming. Queries are supported by a video preprocessing toolkit that applies image and semantic processing to partition raw video streams into shots associated with visual and semantic descriptors that represent and index video content for searching. Preprocessing algorithms detect the video scene boundaries that partition the video into meaningful shots, using a process that computes color histogram differences with a mechanism for dynamic threshold determination [15,25]. Video shots are then processed to extract representative key frames, MPEG7 compatible low-level visual feature descriptors [12,15], camera motion classification [38], spatial and temporal segmentation [14,16], and the semantic annotations of domain experts. The video, its features and indices are stored in the VDBMS database, along with physical metadata such as resolution for quality-of-service presentation. Our system follows the recent trend of representing the video content description in an XML-like format according to the MPEG7 [23] worldwide standard for multimedia content descriptors. MPEG7 standards are an integral part of VDBMS feature representation, and our video processing extracts nearly all low-level features defined as standard, including color histogram in both HSV and YUV formats, texture tamura, texture edges, color moment and layout, motion and edge histograms, dominant and scalable color, and homogeneous texture.

Figure 2. VDBMS query interface.

The query interface supports query by example, query by camera motion type, query by keywords, and SQL queries. In query by example, the user selects a query image and specifies the query features to match and the number of results to retrieve. The features of the query image are extracted online to construct an SQL statement that is sent to the server for execution. Images can be matched against specific frame features or aggregate shot features. Shot results are represented by key frames and ranked by degree of similarity. Users can navigate an image skim of the results, and play shots associated with frame or shot-based results.

2.2 Object Relational Database Management Layer

A new real-time stream manager component was introduced to the object relational database management layer to admit, schedule, monitor and serve concurrent video stream requests periodically [2]. The stream manager is implemented as multi-threaded modules, and has well-defined interfaces with the query engine, the buffer manager, and the Extensible Abstract Data Type (E-ADT) interface. The stream manager operates by issuing requests to the buffer manager, guiding the underlying buffer management policies, communicating with the query processor, and sending streams to clients at a specific rate. The stream manager uses the inherent connection between query results and streaming requests for a search-based buffer pre-fetching and replacement policy that uses both current and expected video streams. Top-ranked query results from the query processor are used to predict future video streaming requests, and a weight function [1,18] determines candidates for caching (e.g., references to current streams are more important than those for expected streams.) Experimental results show that the integration of knowledge from the query and streaming components significantly improves the caching of video streams, reducing initial latency and disk I/O during the streaming process [18].

The query processor was modified extensively to handle the new high-dimensional index schemes as well as to support new video query operators and their integration into the query execution plan. Query processing must take into account the video methods and operators in generating, optimizing and executing query plans. Our query model uses the features approach in accessing video by content. Extracted video features are mapped to a high-dimensional space and video shots are mapped to points in that space. A vector ADT was created to represent the high-dimensional visual feature fields, and a high-dimensional index mechanism [19,20,22] was developed to query the features efficiently. Similarity search is performed by issuing nearest neighbor queries to the high-dimensional access path.

2.3 The Object Storage System Layer

The storage and buffer managers must deal with huge volumes of data with real time constraints [29,32]. The VDBMS buffer pools are divided between the database buffer area and the streaming area where requests for streams are serviced. Extended buffer management handles multiple page requests with segment allocation (instead of the traditional page-based approach) for the large streaming requests from the stream manager. An interface between the buffer manager and the stream manager is used to exchange information that guides buffer caching for stream requests. The storage manager was extended to perform necessary video operations and process both real-time and non-real time requests. Methods for handling extended storage hierarchies [3,4] support real-time access to buffer, disk, and tertiary storage; different caching levels on buffer and disk storage enhance access for frequently referenced data. A tertiary storage server manages access to tertiary resident data, making it directly accessible to the VDBMS system. DVD jukeboxes can be daisy-chained, giving VDBMS access to terabytes of data.

To accommodate the high-dimensional indexing scheme for visual feature vectors, the indexing capabilities of the storage manager were extended by incorporating the Generalized Search Tree (GiST) general indexing framework [19]. We implemented the GiST SR-tree index structure [5,6,26] as our high-dimensional access path to the extracted features, and an instance of the SR-tree is used for the high-dimensional visual feature fields as the access path in image similarity searches.

3.0 The VDBMS Research Platform

This section identifies four recent research projects carried out within the VDBMS video database environment. The open flexible nature of VDBMS provides a foundation for our research, development, and experimentation in new areas of video database technology.

3.1 MPEG7 Document Compliance

We have created an XML wrapper to import user-supplied MPEG7 documents generated using multimedia description schemes for high level and low-level feature information. Document features described in MPEG7 format are extracted, parsed, and mapped to the VDBMS feature relational schema. The video and its documented MPEG7 features are then stored in VDBMS where they can be used for image and content-based queries. The wrapper also supports the export of VDBMS extracted features and metadata from the database in the format of an MPEG7 document, which can then be used by other video processing tools or databases. The wrapper is implemented using XML-DBMS and Java Packages for document transferal. The XML wrapper enables VDBMS to make use of any pre-extracted metadata formatted as MPEG7 documents. In addition, features that VDBMS cannot extract - most importantly event-based and other semantic features - can be integrated as VDBMS metadata via this mechanism and used for video browsing and retrieval.

3.2 A Rank-Join Query Operator

In multi-feature image similarity queries, users generally present a sample image and query the database for images "most similar" to the example based on some collection of visual features. Results must be determined according to a combined similarity order [11,17,30]. We have developed a practical, binary, pipelined query operator, NRA-RJ, which determines an output global ranking from the input ranked video streams based on a score function. Our algorithm extends Fagin's optimal aggregate ranking algorithm [11] by assuming no random access is available on the input streams. We created a new VDBMS query operator that encapsulates the rank-join algorithm in its *GetNext* operation; each call to *GetNext* returns the next top element from the ranked inputs. The output of NRA-RJ thus serves as valid input to other operators in the query pipeline, supporting a hierarchy of join operations and integrating easily into the query processing engine of any database system. The incremental and pipelining properties of our aggregation algorithm are essential for practical use in real-world database engines, and our new operator will help in implementing this type of join in ordinary query plans.

The *GetNext* operation is the core of the rank-join operator. The internal state information needed by the operator consists of a priority queue of objects encountered thus far, sorted on worst score in descending order. *GetNext* is binary, although this restriction is merely practical, and the algorithm holds for more than two inputs. Our modifications to the original NRA algorithm are the following: 1) The right input list is a source stream, which provides the operator with the ranked objects and their exact scores. The left input may not be a source list since it can be the output of another NRA-RJ operator. In this case, the score is expressed as a range, from worst to best. This means that *GetNext* must be able to handle a score range rather than an exact score from the left iterator. 2) The parameter k, the number of requested output objects, is not known in advance, rather it increases for each call to *GetNext*. The modified algorithm first checks if another object can be reported from the priority queue without violating the stopping condition, and if not, moves deeper into the input streams to retrieve more objects. 3) In each call to *GetNext*, the current depth of the caller is passed to the operator. This extra information assures synchronization among the pipeline of NRA-RJ operators.

We implemented the NRA-RJ and other state-of-the-art rank-join algorithms [11,31] as query operators in VDBMS for an extensive empirical study to evaluate operator performance and trade-off issues. We investigated scalability as well as time and space complexity between the algorithms for executing a join of multiple ranked inputs (any number and combination of features) on the stored video objects. An analysis of the resulting performance data showed that our algorithm suffered from computational overhead as the number of pipeline stages increased. Our solution was to unbalance the depth step in the operator children, thus reducing the local ranking overhead. This modification significantly improved the performance of our operator. The optimized NRA-RJ then outperformed the other rank-joins for both small and large numbers of ranked inputs; it was an order of magnitude faster, required less space, and had a comparable number of disk accesses [21].

As a development environment, VDBMS supports the integration and evaluation of our algorithms as well as those from other sources. The above study demonstrates how our system can be used to *test* the correctness and scope of algorithms, *measure* their performance in a standardized way, and *compare* the performance of difference implementation of an algorithm or system component.

3.3 Selective Access Control for Streaming Video

We have developed an access control model that provides selective, content-based access to streaming video data [7,9]. The three layers of control support 1) user credential classification, 2) content-dependent authorizations [27] for video objects, where access is given or denied to a class of users based on the semantic content of the video, and 3) mechanisms for varying the streaming granularity for authorized video elements based on the hierarchical structure of videos - the video stream, shot, frame, or object.

The components of our model are the video elements and potential users. The unit of authorization is a video element: either a sequence of video frames, a specific frame, or a video object that appears as part of a frame (e.g., a face.) Users are characterized by their credentials.

Video elements are specified either explicitly by identifiers or implicitly by their semantic content. Our framework for semantic content analysis is general, and supports either semantic or visual feature descriptors for content. The implicit content identification of video elements applies "concepts" (such as keywords from the video annotations, faces from a catalog, etc) to specify semantic information about the actual content of a set of video data objects. The concepts determine how to restrict access to videos dealing with the specified content. Our control algorithm determines the authorized portions of a video that a user may receive, given the user credentials and the video content description [8,10].

(a) (b)

Figure 3. (a) Content-based access control for streaming video: the face of the patient is blurred during streaming, since the end-user is not authorized to view this object. (b) User interface to the medical video library: index, store, access, query, retrieve, stream.

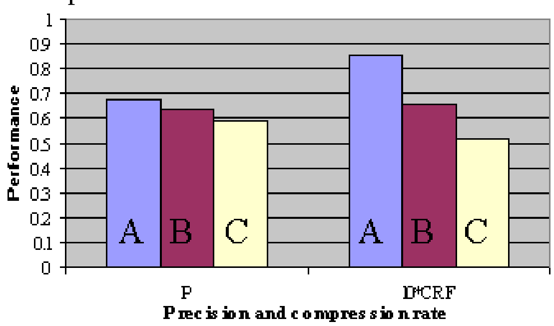
The control model was integrated into VDBMS using a set of authorization rules and control procedures that deploy the rules on a transactional level. Figure 3(a) shows a sample of an altered frame streamed to an end-user, where the user's view of frame content is restricted within the operational context for user/object authorization.

3.4 Medical Video Content Mining for Scalable Skimming

A prototype environment for teaching basic medical education in the regional centers of the Indiana University School of Medicine (IUSM) was created though a joint effort between the VDBMS research group and IUSM. The system uses VDBMS technology to support content-based query, search, retrieval, and presentation for a digital medical video library. Our user interface is shown in Figure 3(b).

To achieve more efficient video indexing and access for medical videos, we are developing a video content structure and event-mining framework [39]. Following shot segmentation and key-frame selection, algorithms for shot grouping, group merging, and scene clustering are applied to organize video content into a five-level hierarchical structure with increasing granularity: video, scene, group, shot, and frame. Video content structure mining is executed in three steps: 1) group detection, 2) scene detection, and 3) scene clustering. Video shots are first grouped into semantically richer units. Similar neighboring groups are then merged into scenes, and a pairwise clustering scheme is applied to eliminate repeated scenes throughout the video. For shot grouping, we apply techniques to identify shots that share similar backgrounds or have high correlation in time series. To segment spatially or temporally related shots into groups, a given shot is compared with shots that precede and succeed it, where similarity is determined using visual feature matching. We compute a shot-based separation factor to evaluate potential group boundaries; this factor is compared to a threshold that is determined automatically using a fast entropy technique [13]. To merge groups into scenes, we define the similarity between two groups to be the average similarity between shots in the benchmark (e.g. smaller) group and the most similar shots in the second group. Neighboring groups with similarity larger than a computed merging threshold are combined into scenes. We have developed a seedless pairwise scheme to cluster similar scenes into units. This scheme requires that the final number of clusters be specified a priori to determine the algorithm stopping point. We compute the optimal number of clusters using cluster validity analysis [24], which finds clusters that minimize intra-cluster distance while maximizing the inter-cluster distance.

After video shots have been parsed into scenes, an event mining strategy is applied to detect event information among scenes. Medical videos are used mainly for educational purposes, thus video content is generally recorded or edited using specific style formats: presentations about specific topics by medical experts, clinical operations, or doctor-patient dialogs. Our strategy begins with the visual feature processing of representative frames to extract semantically related visual cues. Currently, five types of special frame/regions are detected: slide or clip art, black frame, frame with face, frame with large skin regions, frame with blood-red regions. Algorithm details for frame/region processing can be found in [40,41]. Event mining identifies the following: 1) a *presentation scene* is a group of shots with slide or clip art frames, at least one group of temporally related shots, no speaker change between adjacent shots, and at least one shot with a face close-up (>10% of frame), 2) a *clinical operation scene* is a group of shots without speaker change, with at least one blood-red frame or close-up of skin region (>20% of frame), or with more than half of the shots containing skin regions, and 3) a *dialog scene* is a group of shots with both face and speaker changes, at least one group of spatially related shots, and at least one repeated speaker.



(a)

Events	Selected No.	Detected No.	Actual No.	PR	RE
Presentation	15	16	13	0.81	0.87
Dialog	28	33	24	0.73	0.85
Clinical	39	32	21	0.65	0.54
Avg.	82	81	58	0.72	0.71

(b)

Figure 4. (a) Scene detection performance results. (b) Video event mining results.

Experiments to validate video scene detection and event mining were carried out using six hours of medical videos describing face repair, nuclear medicine, laparoscopic kidney removal, skin examination, and laser eye surgery. For scene detection, we compared our method to other methods in the literature using the following rule: a scene is correctly detected if all shots in the scene belong to the same semantic unit, otherwise it is falsely detected. Precision is defined as *P=Correctly detected scenes/All detected scenes*. Clearly, without any scene detection (each shot is one scene), scene detection precision is 100%. We therefore define a compression factor *CRF=Number of detected scenes/Number of total shots.* Figure 4(a) shows the precision and compression factors computed for our algorithm (method A) and two methods from the literature identified as B [33] and C [28]. We make the following observations: 1) method A achieves the best precision, about 65% of the shots are assigned to the appropriate semantic unit, 2) method *C* achieves the highest compression rate, unfortunately its precision is also the lowest, and 3) as a tradeoff with precision, the compression ratio of method A is the lowest (*CRF*=8.6%, each scene consists of about 11 shots). We believe that in semantic unit detection, it is better to over-segment a scene than to fail to segment distinct boundaries. From this point of view, our method is better than other two methods.

After building the video content structure, we manually selected scenes that clearly belonged to the three event categories (presentation, dialog, and clinical operation) and used them as benchmarks. We then applied our event-mining algorithm to determine event category. The experimental results are shown in Figure 4(b), where *PR* and *RE* represent the precision and recall defined as follows: *PR= Actual Number/Detected Number*, *RE= Actual Number/Selected Number*. On the average, our system performed quite well, achieving 72% in precision and 71% in recall when mining these three types of events [39].

4.0 Conclusion

We introduced a video database research initiative that began in June 2000 and culminated in the successful development of a video-enhanced database system that supports comprehensive and efficient database management for digital video libraries. The VDBMS research group developed the extensions and adaptations needed to support full database functionality for the video as a fundamental, abstract database data type. Key database extensions include high-dimensional indexing, new video query operators including a multi-feature rank-join for image similarity search, real-time video streaming, search-based buffer management policies, and support for extended storage hierarchies. We define our system as a research platform, as it provides a foundation for research, development, and experimentation in new areas of video database technology. This concept was illustrated by a discussion of four research problems which were addressed within the VDBMS environment: MPEG7 document support, a new rank-join video query operator, content-based access control for streaming video, and the mining of medical video data using hierarchical content organization.

References

[1] Aref, W., Catlin, A.C., Elmagarmid, A., Fan, J., Guo, J., Hammad, M., Ilyas, I., Marzouk, M., Prabhakar, S., Rezgui, A., Teoh, S., Terzi, E., Tu, Y., Vakali, A. and Zhu, X. A distributed server for continuous media. In *ICDE'02 Proc. of the 18th International Conference on Data Engineering*. February 26-March 1. San Jose, California. February 2002.

[2] Aref, W., Catlin, A.C., Elmagarmid, A., Fan, J., Hammad, M., Ilyas, I., Marzouk, M., Zhu, X. Search and discovery in digital video libraries. *CDS TR #02-005, Computer Sciences Department, Purdue University*. February 2002.

[3] Aref, W., El-Bassyouni, K., Kammel, I. and Mokbel, M. Scalable QoS- aware disk scheduling. In *IDEAS-02 Proceedings of the International Database Engineering and Applications Symposium*. Alberta, Canada. July, 2002. To appear.

[4] Aref, W., Kamel, I. and Ghandeharizadeh, S. Disk scheduling in video editing systems. *IEEE Trans. on Knowledge and Data Engineering*. 13(6). pp. 933-950. November/December 2001.

[5] Beckmann, N., Kriegel, H., Schneider, R. and Seeger, B. The R* -tree: an efficient robust access method for points and rectangles. *SIGMOD Record, ACM Special Interest Group on Management of Data*, 19(2): pp. 322-331. 1990.

[6] Berchtold, S., Böhm, C., Jagadish, H., Kriegel, H-P. and Sander, J. Independent quantization: An index compression technique for high-dimensional data spaces. In *ICDE'00 Proc. of the 16th International Conference on Data Engineering*. San Diego, CA. pp. 577-588. February 2000.

[7] Bertino, E., Bettini, C., Ferrari, E. and Samarati. P. An access control model supporting periodicity constraints and temporal reasoning. *ACM Trans. on Database Systems*, 23(3). pp.231-285. 1998.

[8] Bertino, E., Elmagarmid, A. and Hacid, M-S. Quality of service in multimedia digital libraries. *SIGMOD Record*. 30(1), pp. 35-40, March 2001.

[9] Bertino, E., Hammad, H., Aref, W. and Elmagarmid, A. An access control model for video database systems. In *Proceeding of CIKM, Ninth International Conf. on Information and Knowledge Management*. pp. 336-343. November 2000.

[10] Bertino, E., Samarati. P. and S. Jajodia. An extended authorization model. *IEEE Trans. on Knowledge and Data Engineering*. 9(1). pp. 85-101. 1997.

[11] Fagin, R., Lotem, A. and Naor, M. Optimal aggregation algorithms for middleware. In *PODS'2001* Santa Barbara, CA. May 2001.

[12] Fan, J., Aref, W., Hacid, M-S. and Elmagarmid, A. An improved isotropic color edge detection technique. *Pattern Recognition Letters*. pp. 1419-1429, 2001.

[13] Fan, J., Aref, W., Elmagarmid, A., Hacid, M-S., Marzouk, M. and Zhu, X. Multiview: Multi-level video content representation and retrieval. *Journal of Electrical Imaging*, Vol. 10, No. 4, pp. 895-908, October 2001.

[14] Fan, J, Zhu, X. and Wu, L. Automatic model-based semantic object extraction algorithm. *IEEE CSVT*, 11(10). pp.1073-1084. Oct., 2000.

[15] Fan, J., Hacid, M-S. and Elmagarmid, A. Model-based video classification for hierarchical video access. *Multimedia Tools and Applications*. Vol. 15. October 2001.

[16] Fan, J., Yau, D., Elmagarmid, A. and Aref., W. Automatic image segmentation by integrating color edge detection and seeded region growing. *IEEE Trans. On Image Processing*. 10(10). pp.1454-1466. 2001.

[17] Guntzer, U., Balke, W-T. and Kiessling, W. Optimizing multi-feature queries for image databases. In *VLDB 2000, Proc. Of 26^{th} International Conf. On Very Large Databases*. September 10-14, 2000. Cairo, Egypt. p. 419-428. 2000.

[18] Hammad, M., Aref, W., and Elmagarmid, A. Search-based buffer management policies for streaming in continuous media. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. Lausanne, Switzerland. August 26-29, 2002. To appear.

[19] Hellerstein, J., Naughton, J. and Pfeffer, A. Generalized search trees for database systems. In *VLDB'95 Proc. of 21^{st} International Conf. on Very Large Data Bases*. September 11-15. Zurich, Switzerland. 1995.

[20] Ilyas, I. and Aref, W. SP-GiST: An extensible database index for supporting space partitioning trees. *Journal of Intelligent Systems (JIIS)*. 17(2-3). pp. 215-235. 2001.

[21] Ilyas, I, Aref, W, and Elmagarmid, A. Joining ranked inputs in practice. In *Proceedings of the 28th VLDB Conference*. Hong Kong, China. 2002. To appear.

[22] Ilyas, I. and Aref, W. An extensible index for spatial databases. In *Proceedings of the 13th International Conference on Statistical and Scientific Databases*. Virginia. July 2001.

[23] ISO/IEC/JTC1/SC29/WG11: Text of ISO/IEC 15938-3 Multimedia Content Description Interface – Part 3: *Visual. Final Committee Draft. Document No. N4062*. Singapore. March 2001

[24] Jain, A. K. Algorithms for clustering data. *Prentice Hall*. 1998.

[25] Jiang, H., Helal, A., Elmagarmid, A., and Joshi, A. Scene change detection for video database systems. *Journal on Multimedia Systems*, 6(2), pp.186–195. May 1998.

[26] Katayama, N. and Satoh, S. The SR-tree: An index structure for high dimensional nearest neighbor queries. *SIGMOD Record, ACM Special Interest Group on Management of Data*, 26(2). 1997.

[27] Kumar, P. and Babu, G. Intelligent multimedia data: data + indices + inference. *Multimedia Systems.* 6(6). pp.395-407. 1998.

[28] Lin, T. and Zhang, H. Automatic Video Scene Extraction by Shot Grouping. In *Proc. ICPR 2000*.

[29] Moser, F., Kraiss, A. and Klas, W. L. A buffer management strategy for interactive continuous data flows in a multimedia dbms. In *VLDB'95 Proc. of 21st International Conf. on Very Large Data Bases*. September 11-15. Zurich, Switzerland. pp. 275-286. 1995.

[30] Nepal, S., Ramakrishna, M. Query processing issues in image (multimedia) databases. In *ICDE'99 Proc. of the 15th International Conference on Data Engineering*. March 23-26. Sydney, Australia. p. 22-29. IEEE Computer Society, 1999.

[31] Natsev, A., Chang, Y-C., Smith, J., Li, G-S. and Vitter, J. Supporting incremental join queries on ranked inputs. In *VLDB'01 Proc. of 27th International Conf. on Very Large Data Bases*. Rome, Italy. 2001.

[32] Ozden, B., Rastogi, R. and Silberschatz, A. Buffer replacement algorithms for multimedia storage systems. In *Proc. of IEEE International Conf. on Multimedia Computing and Systems*. pp. 172-180. 1996.

[33] Rui, Y., Huang, T. and Mehrotra, S. Constructing table-of-content for video. *ACM MSJ*. 7(5). pp. 359-368. 1999.

[34] Seshadri, P. Predator: A resource for database research. *SIGMOD Record.* 27(1). pp. 16-20. 1998.

[35] Smith, J. Sequentiality and prefetching in database systems. *ACM Trans. on Database Systems*. 3(3). pp. 223-247. September 1978.

[36] Stonebraker, M. Operating system support for database management. *CACM*. 24(7). pp. 412-418. 1981

[37] Storage Manager Architecture. *Shore Documentation, Computer Sciences Department*. UW-Madison, June 1999.

[38] Zhu, X., Elmagarmid, A., Xiangyang, X. and Catlin, A. InsightVideo: Toward hierarchical content organization for efficient video browsing, summarization and retrieval. *IEEE Transactions on Multimedia Journal*. Submitted.

[39] Zhu, X., Fan, J., Aref, W. and Elmagarmid, A. ClassMiner: Mining medical video content structure and events towards efficient access and scalable skimming. In *Proceedings of the SIGMOD Workshop on Data Mining and Knowledge Discovery*. Madison, WI. June, 2002. To appear.

[40] Zhu, X., Fan, J., Elmagarmid, A. and Aref, W. Hierarchical video summarization for medical data. In *Proc. of IST/SPIE storage and retrieval for media database*. pp.395-406, 2002.

[41] Zhu, X., Fan, J. and Elmagarmid, A. Towards facial feature localization and verification for omni-face detection in video/images. In *Proc. of IEEE ICIP*. 2002.