IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4403

CONTENT-BASED RETRIEVAL OF SPATIO-TEMPORAL VIDEO EVENTS

M. Petkovic, W. Jonker, Computer Science Department, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
Phone: +31 53 4893725, Fax: +31 53 4892927,E-mail: {milan, jonker}@cs.utwente.nl

ABSTRACT

This paper addresses content-based video retrieval with an emphasis on spatio-temporal modeling and querying of events. Our approach is based on a layered model that guides the process of translating raw video data into an efficient internal representation that captures video semantics. We also present a video query language and show that the proposed model facilitates execution of different types of queries. The main ideas have been implemented in order to achieve a prototype of a video database management system. Furthermore, results on real tennis video data are presented demonstrating the validity of the approach.

1. INTRODUCTION

In the literature, video content is mostly approached either at the feature or at the semantic level [1]. Features, such as color histogram, shape orientation, or motion trajectory, characterize the low-level visual content, while the semantic content is described by high-level concepts such as objects and events. Extensive research efforts have been made with regard to the retrieval of video and image data based on their low-level visual content. Examples such as VisualSEEk, Photobook, Blobworld, IBM's Query by Image Content (QBIC), VideoQ, Virage video engine, and CueVideo are surveyed in [2-5]. Early approaches in video retrieval only added the functionality for segmentation and key-frame extraction to the existing image retrieval systems. After key-frame extraction, they apply similarity measurements on them based on features. This is not satisfactory because video is temporal media, so sequencing of individual frames creates new semantics that may not be present in any of the individual frames. Query by example approaches are suitable if a user has a similar image at hand, but they often would not perform well if the image is taken from a different angle or has a different scale. The naive user is interested in querying at the semantic level rather then having to use features to describe his concepts. Furthermore, good match in terms of the feature metrics may yield poor results in the context of concepts (multiple domain recall, etc.).

Modeling the semantic content is far more difficult then modeling the low-level visual content of a video. At the physical level video is a temporal sequence of pixel regions without direct relation to its semantics. Therefore, it is very difficult to explore semantic content from the raw video data. The simplest way is by using free text manual annotation. Textual descriptors are associated with portions of linear video stream, which are usually organized hierarchically [6, 7]. Some other approaches introduce additional video entities that should be annotated, because they are subjects of interest. Beside objects and actions introduced in [8], spatio-temporal relationships among video objects become first class citizens of the video model. A video object is associated with sub-frame regions, while spatio-temporal relations are used to describe objects in space and time and capture movements of objects. Attempts to include these high-level concepts into video models are made in [9-11].

Obviously, there is a big gap between two kinds of modeling approaches mentioned. On the one hand, feature-based models use automatically extracted features, which represent the content of a video, but they do not provide semantics that describe high-level video concepts. On the other hand, semantic models usually use free text/attribute/keywords annotation to represent the high-level concepts of the video content, which results in many drawbacks. The major limitation of these approaches is that the search process is based only on the predefined attribute information, which is associated with video segments in the process of annotation. Furthermore, manual annotation is tedious, subjective and time consuming. On the contrary, we propose an integrated approach that provides a framework for automatic mapping from features to high-level concepts. Instead of extending the basic feature-based retrieval with querying by keywords or captions as in Virage data model, our model is aiming at automatic extraction of concepts from visual features. We associate video objects with regions across frames, formalizing their spatio-temporal interactions as events. In order to achieve automatic extraction of objects and events, two grammars that accumulate explicit domain knowledge are introduced.

In this paper we address three issues of video databases: (1) data model, (2) query language, and (3) implementation. In particular, a new data model, called COBRA (COntent-Based RetrievAl), is introduced in order to overcome the problem of mapping low-level visual features to high-level concepts. Based on the model an OQL-like query language that supports homogeneous querying at different content levels is designed. Consequently, architectural and implementation issues are investigated. The paper is organized as follows. In the next section, the COBRA video data model is briefly described. The third section presents the COBRA query language and discusses the issues concerning the necessary extension of the underlying database management system that enables modeling and querying spatio-temporal events. In the fourth section we present a tennis case study, and we conclude the paper in the fifth section.

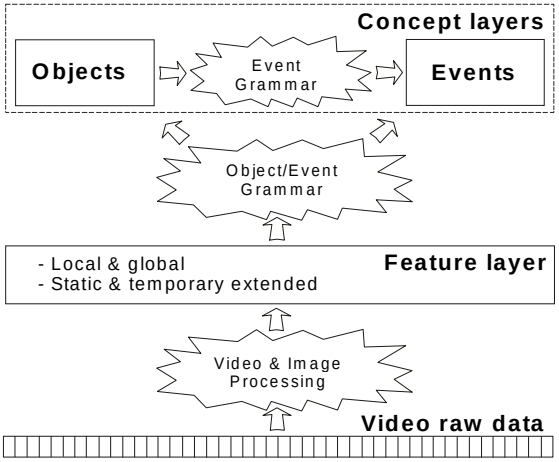


Fig. 1. The layered hierarchy of the COBRA video data model

2. THE COBRA VIDEO MODEL

In order to overcome the problem of mapping features to high-level concepts we propose the COBRA video data model (Fig. 1). It consists of four layers: the raw video data, feature, object and event layer. The raw video data layer comprises sequences of frames and regions (RVD), as well as some video attributes (A). The feature layer consists of domain-independent features (F) that can be automatically extracted from raw data characterizing colors, shapes, textures and motion. The object layer contains logical concepts characterized by a prominent spatial dimension, assigned to regions across frames and grouped together under some criteria defined by the domain knowledge. A region is a contiguous set of pixels that is homogeneous in texture, color, shape and motion properties. An object (O) should also satisfy some conditions so that it would be semantically consistent, representing one real-world object, and subject of interest to users or applications. Some examples of video objects are a specific player or the ball in a tennis game or a specific car in a car-race video. The event layer consists of events (E), i.e. entities that have a prominent temporal extent, describing the movements and interactions of different objects in a spatio-temporal manner. Other elements, which are also important for basic functionality of our model defined in (1), are: a set of audio segments (S); a signature (S) that defines types for F, O, E, and S; a set of map functions that map F, O, E, and S onto a power set of RVD; and a set of spatial, temporal, real-world, and video operations (a).

$$V = (\Sigma, A, RVD, F, O, E, S, \lambda, G_O, G_E, \alpha) \quad (1)$$

In the sequel, we will focus on the grammars (G_O, G_E) that are proposed to facilitate object and event extractions. For the automatic mapping from the feature/object layer to the event layer, we propose the event grammar that defines rules for describing spatio-temporal event types (S_E). The event types can be primitive and compound. There are two rules for primitive event types. The first one defines events using visual features of the raw video data and their spatio-temporal and similarity relations (s , t, and j), while the second one instead of raw data uses object types (S_O) together with their real-world relations (w). Audio segment types (S_S) may also be included. Hence, it is possible to define audio events, and compound audio-visual events. On the other hand, a compound event type is described by a power set of predefined event types, their temporal relations (t), as well as real-world (w) and spatial relations (s) among their objects. So, the event grammar rules are defined as follows:

$$\rho_{Eprimitve\_RVD}: (2^{\Sigma_{RVD}}, 2^{\Sigma_S}, 2^{\varphi}, 2^{\sigma}, 2^{\tau}) \rightarrow \Sigma_E;\ \rho_{Eprimitve\_O}: (2^{\Sigma_O}, 2^{\Sigma_S}, 2^{\omega}, 2^{\sigma}, 2^{\tau}) \rightarrow \Sigma_E;$$
$$\rho_{Ecompound}: (2^{\Sigma_E}, 2^{\omega}, 2^{\sigma}, 2^{\tau}) \rightarrow \Sigma_E.$$

Let us consider a simple rule for the 'Net_playing' event type as an example:

$\rho_{Net\_playing}$: ({o_1: player, o_2: net}, {}, {}, {distance(o_1, o_2) < 50}, {duration(this) > 300}).

There are two object types involved: Player and Net. There are no audio segment types and real-world relations among object types. But, there are a spatial relation (distance) and a temporal relation (duration). The temporal relation says that this event type should last a specific period, as well as that the spatial relation should be valid for that period of time.

As we can see in the literature [12-14] automatic detection of video objects in a known domain is feasible. For this purpose, we proposed an object grammar that consists of domain-dependent rules for object extractions. Despite the objects are defined as entities with a prominent spatial dimension, we take advantage of video temporality allowing the usage of temporal relations in the rules for object descriptions. Hence, they are defined as following:

$$\rho_{Oprimitive}: (2^{\Sigma_{RVD}}, 2^{\varphi}, 2^{\sigma}, 2^{\tau}) \rightarrow \Sigma_O;\ \rho_{Ocompound}: (2^{\Sigma_O}, 2^{\varphi}, 2^{\sigma}, 2^{\tau}) \rightarrow \Sigma_O.$$

The two grammars aim at formalizing descriptions of high-level concepts as well as to facilitate their extraction based on features that can be computed using existing techniques. At the same time the model is in line with the latest development of MPEG-7 [15], which means that it is independent of feature/semantic extractors, providing flexibility of using different video processing and pattern recognition techniques for those proposes.

3. THE COBRA QUERY LANGUAGE AND ITS IMPLEMENTATION

3.1. The COBRA Query Language

A capability of retrieving the video data specifying its content is very important for users. As video content can be approached at different levels (the raw data, feature, and concept level), it should be possible to be queried at all these levels too. In other words, a user should be able to query the video database combining the raw data, features, objects and events. As a standard query language does not support homogeneous querying at all these levels, we designed a new query language, called COBRA. The COBRA query language is based on the proposed data model, which means that the proposed grammars can be used inside a query to describe video objects and events. In order to build on standards and avoid dependence on a particular media or application, our language is designed as an extension of the Object Query Language (OQL) that has been proposed by Object Database Management Group (ODMG). We added seven predefined types to OQL, namely: (1) frame, (2) region, (3) frame sequence, (4) video, (5) features as descriptors for color, shape, texture and motion, (6) objects, and (7) events. This basic set of types is used as meta-data that describes video content.

Another extension is in the WHERE clause that specifies criteria that the query result should satisfy. The COBRA query language allows a user to specify four additional expressions as criteria: (1) video expression, (2) feature expression, (3) object expression, and (4) event expression. The video expression deals with video predicates, including the contain predicate. The contain predicate specifies a set of particular objects/events that should occur in the retrieved frame sequence. The feature expression includes feature types (S_F) and predicates (j) commonly used in feature-based models. A user can also specify an object or event expression. These two expressions define new objects/events observing the rules from the object/event grammar. In order to be able to evaluate event descriptions, we had to define a basic set of spatio-temporal relations. As far as spatial relations are concerned, we use the Minimum Bounding Rectangle (MBR) approximation to represent the object geometry in order to increase efficiency. Based on that, we define fundamental topological (equal, inside, cover, overlap, touch, disjoint and two inverse covered_by and contains) and directional relations (north, south, west, east, north-east, north-west, south-east and south-west), as well as the Euclidean distance relation. Definitions of these relations, as well as interval temporal relations are skipped, because they are already defined in [16, 17]. As far as temporal relations are concerned, we defined basic relations of interval algebra (before, meets, overlaps, during, starts, finishes, equal plus six inverse relations), as well as point temporal algebra (<, =, >). The mapping between them is

solved by introducing aggregates that operate on sets such as 'make intervals', 'start interval', 'end interval', as well as operations on the interval data type such as duration, intersect, and union.

Expressiveness of the grammars that are integrated into the query language allows a user to specify very detailed complex queries using a combination of feature, spatial (topological, directional and distance), and temporal relations. For example, a user can specify the queries that are reported in [9, 10] : "Find a video clip in which a dog approaches Mary from the left" or "Find video clips in which a police car with siren on is chasing a red Porsche and hit on it". Compared to the MOQL [9] and the query language based on the LHDVM model [10], our query language integrates querying at different content levels: the raw data, feature, object and event level. It has potential to deal with raw video data, features, and semantics. Contrary to the mentioned approaches, we do not assume that the dog from the first query example is manually annotated and that video has already been segmented into clips, but, as the advantage, the COBRA query language facilitates object extractions based on video features and segments the video dynamically. In contrast to spatio-temporal query languages from the literature that are rather complex for end-users, our object-event based querying is flexible, providing users with the possibility to define new concepts step-by-step (primitive and compound descriptions). In addition, we provide our query language with a graphical user interface as is described in the fourth section.

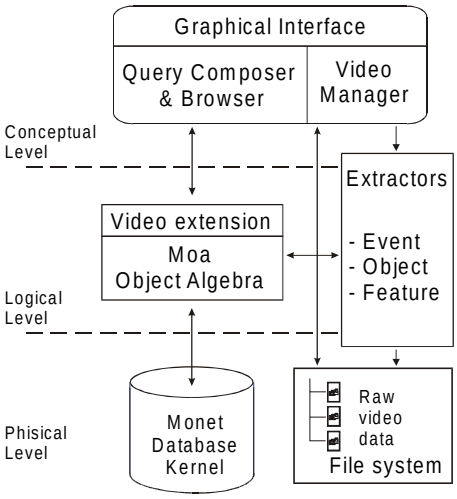


Fig 2. The video database architecture

3.2. Implementation Issues

The COBRA query language is implemented within our prototype video database system that follows the well-known three-level DBMS architecture (Fig. 2). The proposed query language is used at the conceptual level where we developed a query composer and browser with a graphical user interface. It rewriters a graphical or textual query into a Moa query and manages it further processing according to an algorithm shown in Fig. 3. The algorithm optimizes query evaluation using as the advantage our platform's ability of parallel query execution.

1. Let ρ be the description of a spatio-temporal event.
2. If there is any common operand between the spatial and temporal relations of ρ, set *Com*=true and do steps 3 to 5 sequentially, otherwise set *Com*=false and do steps 3 and 4 simultaneously with step 5.
3. Evaluation of the spatial part:
   a. Select objects that are involved in spatial relations;
   b. Perform spatial operations on selected objects using the sub-frame class;
   c. Project over the target list to obtain frame numbers.
4. Convert a set of frame numbers into a set of temporal intervals (frame sequences) using the "make_intervals" aggregate operation.
5. Evaluation of the temporal part:
   a. Select objects that are involved in temporal relations;
   b. Perform interval temporal operations on selected objects.
6. If (!*Com*) Perform the union interval operation on frame sequences obtained as results of the spatial and temporal part of the event description.
7. Exit

Fig. 3. The algorithm for query evaluation

The Moa object algebra [18] is used at the logical level. It accepts all base types of the underlying physical storage system and allows their orthogonal combination using the structure primitives set, tuple, and object. This provides data independence between the logical and physical level, as well as extra optimization possibilities during query execution. For each Moa operation, there is a program written in the interface language understood by the physical layer. So, a Moa query is rewritten into Monet Interface Language (MIL), which is understood by Monet [19] – an extensible parallel database kernel that is used at the physical level of our system. Monet supports a binary relational model, main memory query execution, extensibility with Abstract Data Types (ADTs) and new index structures, as well as parallelism. Extensibility of our implementation platform at all levels allows us to define proposed operations and optimization at the logical and even at the physical level. This is an important advantage over approaches that use commercial DBMSs and implement video extension at the application level, which results in much slower systems.

4. A CASE STUDY FOR TENNIS VIDEOS

In order to employ the object/event grammar, feature extraction has to be done in advance. For that propose we developed a special tool for the tennis domain. After entering of some initial data (type of tennis court, colors of players' dresses, names, etc.), we preprocess the raw data based on dominant color and select only video segments that contain tennis court. Then, the tool segments player regions from the raw data (Fig. 4).

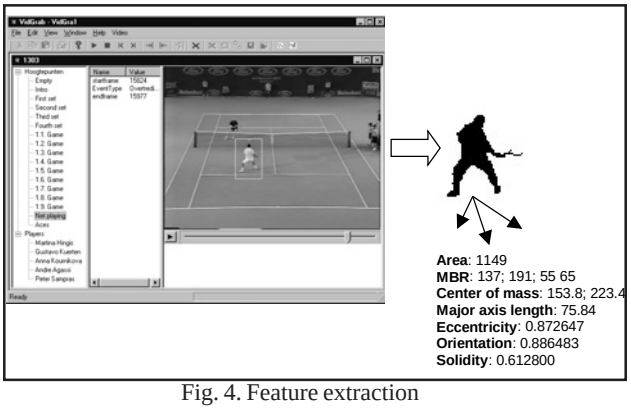


Fig. 4. Feature extraction

Regions that represent objects are selected using the domain knowledge accumulated in descriptions from the object grammar. For example, an object description that extracts a player closer to camera from a typical tennis shot using shape and color features is defined as following:

r_{Player}: ({r_1: body, r_2: rect(0, 144, 384, 288)}, {1000<area(r_1)<1400, dominant_color(r_1)=white, 0.75<eccentricity(r_1)<0.95, 70<orientation(r_1)<120}, {contain(r_2, r_1)}, {}).

Some query examples with integrated event descriptions are presented in the sequel. The first query retrieves all video segments where Sampras is near the net for a given period of time. The new "player_near_the_net" event type is defined in terms of spatio-temporal object interactions assuming that objects like players and the net have been already extracted using the object grammar, external extractors for specific objects, MPEG 4, or other techniques. Query 1 also shows how event descriptions can be parameterized (a user might be interested not only in Sampras, but in different players playing on the net).

Query 1: SELECT Frame_seq: f
FROM Sampras_vs_Agassi_AustraliaOpen00
WHERE EVENT_TYPE: f.e=($r_{Player\_near\_the\_net}$: ({o_1:player, o_2:net}, {},{}, {y_distance(o_1, o_2)<50}, {duration (this)>60}), o_1.name='Sampras')

A user can also reuse already defined event types in order to define compound ones. For example, possible lobs can be retrieved by describing a new event type using the already defined events ('Player_near_the_net' and 'Player_near_the_base-line') and some additional criteria (Query 2).

Query 2: SELECT Frame_seq: f
FROM Sampras_vs_Agassi_AustraliaOpen00
WHERE EVENT_TYPE: f.e=(r_{Lobs}: ({e_1:Player_near_the_net, e_2:Player_near_the_base-line}, {$e_1.o_1=e_2.o_1$}, {}, {meet(e_1, e_2)}))

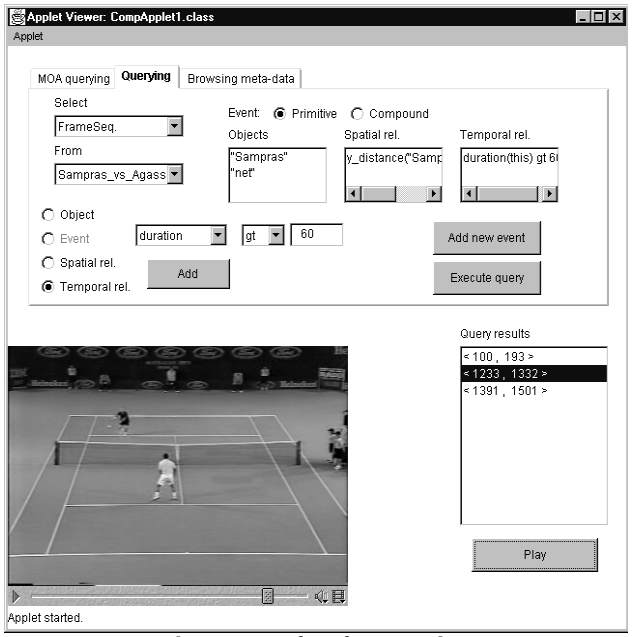
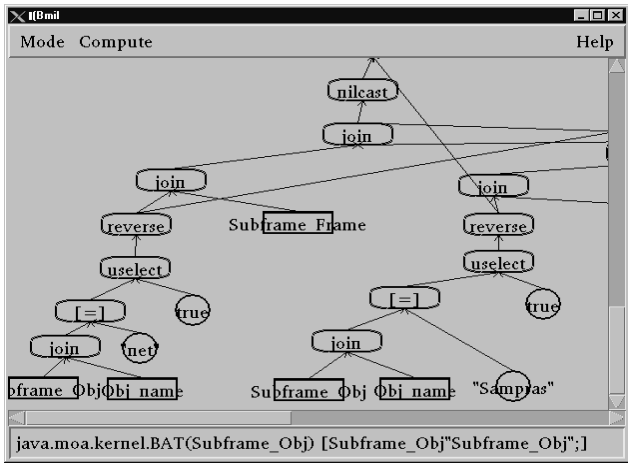


Fig. 5. The user interface for event descriptions

In order to simplify the process of describing semantic events, a graphical user interface is developed. Fig. 5 shows how the 'Net_playing' query can be expressed interactively on a screen. After posing a query, the system rewrites it into the Moa object algebra (Fig. 6a) and consequently into the Monet algebra (Fig. 6b). Note that the Fig. 6(b) shows only a small part of the very large MIL graph. The user can also update meta-data by adding the newly defined event, which will allow him to define more complex compound events that comprise this event. This will also speed up future querying of this event, because it will be resolved in the event layer without looking at other layers.

```
select[gt(-(attr(THIS, 1), attr(THIS, 0)), 60)] (Intv)
WITH
(miv (map[attr(attr(%0, 6), 0)] (select[le(-(attr(%1, 1),
   attr(%0, 1)), 50)](SFs))):Intv)
WITH
(join[%6, %6](SFs1, SFs2):SFs)
WITH
(select[=(attr(attr(THIS, 5), 0),"net")]
(SubframeExt):SFs2)
WITH
(select[=(attr(attr(THIS, 5), 0), "Sampras")]
(SubframeExt):SFs1);
```

(a)



(b)

Fig. 6. The Moa query (a) and a part of the MIL graph (b)

5. CONCLUSIONS

The COBRA system integrates feature-based and annotation-based approaches, in such a way that annotations, i.e. high-level concepts, are extracted automatically from visual features. The underlying video data model provides a framework for automatic mapping from features to semantic concepts, integrating audio and video primitives. In order to bridge this gap we take advantage of the domain knowledge that is accumulated in reusable sets of object and event descriptions defined by the object and event grammars. Additionally, the formalization of an event as a spatio-temporal description of object interactions results in easier capturing of high-level concepts, and allows queries that are closer to user way of thinking (users' cognitive maps of a video). Based on our video model we designed the COBRA video query language that integrates the new data types and the object and event grammars, supporting homogeneous querying at different content levels (the raw data, feature, and concept level). The three-level architecture of our prototype video database system and extensibility of its all levels allow us to efficiently implement the COBRA query language. Consequently, we propose the algorithm for query evaluation, which exploit our platform's ability of parallel query execution, and demonstrate the validity of our approach in the case study for tennis videos. Using the very small set of simple rules that are manually created, we are able to automatically extract spatio-temporal events such as "Net_playing", "Lobs", etc.

In order to overcome the problem of creating object and event rules manually, which might be difficult, especially in case of

some object rules that require familiarity with features, we are investigating how some well-defined statistical algorithms can be used to automatically extract high-level semantics (e.g. events) from video data. In particular, we are focusing on the use of Hidden Markov Models (HMMs) for automatic knowledge extraction from the raw video data. As HMMs are effective tools with solid theoretical basis for modeling time varying patterns, finding greatest use in fields like speech and gesture recognition, we believe that they can be also very effectively applied in recognition of video events.

REFERENCES

[1] A. Hampapur, R. Jain, "Video Data Management Systems: Metadata and Architecture" in *Multimedia Data Management*, A. Sheth, W. Klas (eds.), McGraw-Hill, 1998.

[2] M. Petkovic, W. Jonker, "Overview of Data Models and Query Languages for Content-based Video Retrieval", *International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, l`Aquila, Italy, 2000.

[3] A. Yoshitaka, T. Ichikawa, "A Survey on Content-Based Retrieval for Multimedia Databases", *IEEE Transactions on Knowledge and Data Engineering*, 11(1), pp. 81-93, 1999.

[4] B. Perry, S-K. Chang, J. Dinsmore, D. Doermann, A. Rosenfeld, S. Stevens, *Content-based Access to Multimedia Information: From Technology Trends to State of the Art*, Kluwer, 1999.

[5] P. Aigrain, H. Zhang, D. Petkovic, "Content-based Representation and Retrieval of Visual Media: A State-of-the-Art Review", *Multimedia Tools and Applications*, Kluwer, 3(3), pp. 179-202, 1996.

[6] R. Weiss, A. Duda, D. K. Gifford, "Content-based Access to Algebraic Video", *Int. Conf. on Multimedia Computing and Systems*, IEEE Press, pp. 140-151.

[7] E. Oomoto, K. Tanaka, "OVID: Design and Implementation of a Video-Object Database System", *IEEE Trans Knowl Data Eng*, 5(4), pp. 629-643, 1993.

[8] S. Adali, K. S. Candan, S-S. Chen, K. Erol, V. S. Subrahmanian, "Advanced Video Information System: Data Structure and Query Processing", *Multimedia System*, 4(4), Aug. pp. 172-186, 1996.

[9] J. Z. Li, M. T. Ozsu, D. Szafron, "Modeling of Video Spatial Relationships in an Object Database Management System", *Int. Workshop on Multi-media Database Management Systems*, pp. 124-132, 1996.

[10] H. Jiang, A. Elmagarmid, "Spatial and temporal content-based access to hypervideo databases", *VLDB Journal*, 7(4), pp. 226-238, 1998.

[11] Y. F. Day, S. Dagtas, M. Iino, A. Ghafoor, " An Object-Oriented Conceptual Modeling of Video Data", IEEE International Conference on Data Engineering, pp. 401-408, 1995.

[12] Y. Gong, L. T. Sin, C. H. Chuan, H-J. Zhang, M. Sakauchi, "Automatic Parsing of TV Soccer Programs", *IEEE Int. Conference on Multimedia Computing and Systems*, Washington D.C., pp. 167-174, 1995.

[13] A. Woudstra, D.D. Velthausz, H.J.G. de Poot, F. Moelaert El-Hadidy, W. Jonker, M.A.W. Houtsma, R.G. Heller, J.N.H. Heemskerk, "Modelling and Retrieving Audiovisual Information - A Soccer Video Retrieval System -", *4th International Workshop on Multimedia Information Systems*; Istanbul, Turkey, September 1998.

[14] G. P. Pingali, Y. Jean I. Carlbom, LucentVision: "A System for Enhanced Sports Viewing", *Proc. of Visual'99*, Amsterdam, pp. 689-696, 1999.

[15] Overview of the MPEG-7 Standard, *ISO/IEC JTC1/SC29/WG11 MPEG2000/N3445*, Geneva, CH, June 2000.

[16] D. Papadias, Y. Theodoridis, T. Sellis, and M. Egenhofer , "Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-Trees", SIGMOD '95, M. Carey and D. Schneider (eds.), *SIGMOD RECORD* 24 (2), pp. 92-103, 1995.

[17] J.F. Allen, "Maintaining knowledge about temporal intervals", *Communications of ACM*, 26(11), pp. 832-843, 1983.

[18] P. Boncz, A.N. Wilschut, M.L. Kersten, "Flattering an object algebra to provide performance", *IEEE International Conference on Data Engineering*, Orlando, pp. 568-577, 1998.

P. Boncz, M.L. Kersten, Monet: "An Impressionist Sketch of an Advanced Database System", *Basque International Workshop on Information Technology*, San Sebastian, 1995.