Spatio-Temporal Querying in Video Databases

Mesru Köprülü[1], Nihan Kesim Çiçekli[1,2] and Adnan Yazıcı[1]

[1] Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
mesruk@superonline.com, yazici@ceng.metu.edu/tr
http://www.ceng.metu.edu.tr/~yazici

[2] Department of Computer Science, University of Central Florida, Orlando, USA
nihan@cs.ucf.edu
http://www.cs.ucf.edu/~nihan

Abstract: In this paper a video data model that allows efficient and effective representation and querying of spatio-temporal properties of objects is presented. The data model is focused on the semantic content of video streams. Objects, events, activities performed by objects are main interests of the model. The model supports fuzzy spatial queries including querying spatial relationships between objects and querying the trajectories of objects. The model is flexible enough to define new spatial relationship types between objects without changing the basic data model. A prototype of the proposed model has been implemented. The prototype allows various spatio-temporal queries along with the fuzzy ones and it is prone to implement compound queries without major changes in the data model.

1 Introduction

The uninterruptible advances in computer technology, which have made the storage and processing capabilities increase while the costs decrease, brought multimedia to our computers. As a result, high technology multimedia systems like video on demand services, geographical information systems based on image and video data, digital video libraries, teleconferences, security systems with video support, etc., have become ordinary parts of our life. With the high usage of multimedia systems in our daily life, some new requirements have arisen, like querying the multimedia data. Content-based queries on images have been studied for a relatively long time and many solutions have been developed for storing and querying images in databases. Querying video databases however is a newer concept than image databases.

Although querying video databases is a newer concept, there have been several studies on querying the content of videos [1,8,11,12,13,15,16,17,18,19,20]. There exist mainly two different approaches for content-based query processing in video systems. One approach is to use image-processing techniques in evaluating the queries. The second approach deals with the semantic content of the video data, where the semantic data is stored as metadata with the video. Similar to querying image databases, querying video databases involves queries for spatial relationships among objects. In addition, we need to deal with temporal and spatio-temporal queries in video databases.

Several studies have been done on developing a video model that allows spatial, temporal and spatio-temporal queries. Some of them use annotation-based modelling [7,17], some use physical level video segmentation approach [2,10,19,20], and some have developed object based modelling approaches [1,6,8,13,15,18]. Annotation-based video models allow free text or attribute/keyword annotations in order to describe semantic and spatial attributes of video data. The physical level video segmentation approach does not address semantic concepts, such as, objects, events, roles, players, etc. Instead the video data is described as a stream of small segments with application specific temporal and spatial properties. Object based models focus on the modelling of semantic content of the video data using object-oriented modelling techniques.

Among the object-based models, the Advanced Video Information System (AVIS) [1] introduces indexing video data, based on the objects of interest and events in the video. This index is represented with an association map and then transformed into an augmented interval tree (frame segment tree) for efficient interval access. The main focus in the AVIS is the existence of objects and activities in the video. Objects and activities are stored in special index structures. The model is capable of performing complex and compound content-based queries within the well-defined index structure. However it lacks spatial information and thus, it does not support spatial and spatio-temporal queries.

Since multimedia data is a rich information source, it contains various data types, which are either structured or unstructured. Especially for unstructured data types, handling uncertainty (and fuzziness) is an important issue. Fuzzy queries on multimedia data have already been studied by some researchers [5, 9]. Among these, Aygun and Yazici [4,5], use a fuzzy object oriented conceptual data model [21] to represent fuzzy information and object-oriented concepts [22] of video data.

In this study we present a spatio-temporal data model for video data. The data model is developed as an extension to the AVIS model [1]. We have chosen to use AVIS as the basis for our study, since AVIS is a flexible model that can be used for any kind of video data. It is application-independent, and allows the user to store as much semantic meta-data as required. Our extended data model has the following properties and contributions: (1) it allows to model semantic content of video data including the spatial properties of objects. The extensions to AVIS include extending the association map, so that it can represent spatial properties of objects, being able to store spatial data together with objects in the frame segment tree, and introducing an array structure for objects to store spatial data associated with them, (2) the model allows to perform spatio-temporal queries on the video, including querying spatial relationships between objects in the video and querying moving objects in the video, (3) the model also allows to include fuzziness in spatial and spatio-temporal queries; therefore, one is able to do fuzzy querying, and (4) a prototype implementation of the proposed model is developed. The prototype implementation includes, in addition to all basic queries supported by AVIS, spatial relationship queries, fuzzy spatio-temporal and spatial relationship queries.

The rest of the paper is organized as follows: Section 2 describes the data model in AVIS in detail. In Section 3, we introduce the primary extensions to AVIS, which can handle spatial and temporal properties and relationships between objects. Section 4 discusses the query types that are supported by our model. The last section concludes the paper with some remarks for future extensions.

2 AVIS – Advanced Video Information System

AVIS (Advanced Video Information System) [1], is an object-based video data model that can be used for any kind of video data. In the model, the main focus is on objects, events, and activities, called *entities*, appearing in the video. Objects are the individual entities that may be characters in a movie such as Rambo, James Bond, or they may be objects such as, car, tree, and ball. An activity type may be accepted as the subject of a given frame sequence. It is the type of the action performed in a video (e.g. walking, eating, or kicking a ball). An event is an instance of an activity type. It consists of an activity type, roles in the activity, and objects as the actors of roles in the activity. For example, in the event "Philip is eating a cake", the activity type is eating, the role is eater, and the actor is Philip.

A video stream is composed of a huge set of video frames. A frame sequence is defined as a set of contiguous frames containing semantically important data, like an object or an event. Associated with each entity is a set of frame sequences. These are the frames in which that entity appears in the video. The information about entities and frames is indexed with special index structures based on an association map. Fig. 1 illustrates an association map generated for a

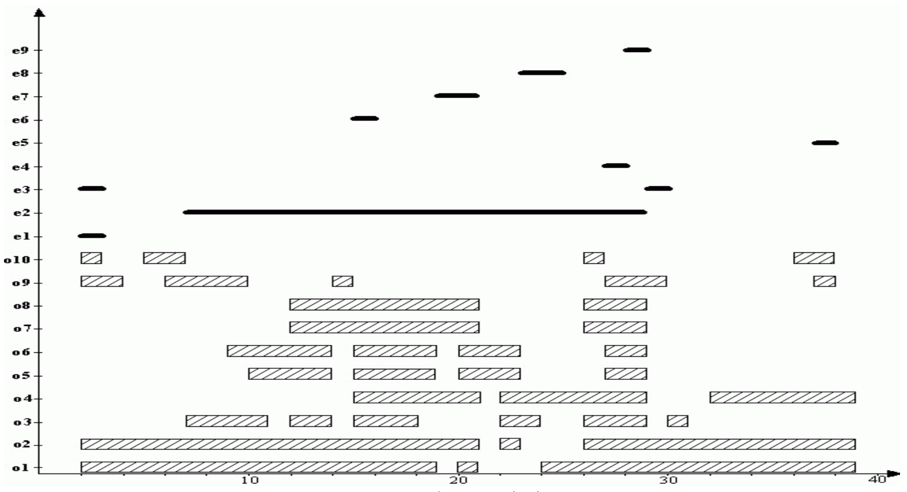

Fig. 1. A sample association map

movie. The x-axis represents the frames of video and y-axis represents the set of entities in the movie. Only two entity types are presented in the figure, *objects* and *events*. For each entity, the corresponding line segments represent the frames in which they appear. For example, the two line segments of object 7 (*o7*) means that, object 7 appears in frame segments [12…21) and [26…29).

Indexing in AVIS is based on the *frame-segment tree* (FST), which is derived from the association-map. Overlapping line segments for different entities are considered as separate nodes in the frame segment tree. A list of entities that appears in the time interval represented by a node is stored along with the node in the tree. Fig. 2 illustrates the segment tree generated from the map in Fig. 1 for the interval [1-11]. The set of numbers placed in a node denotes the identities of video

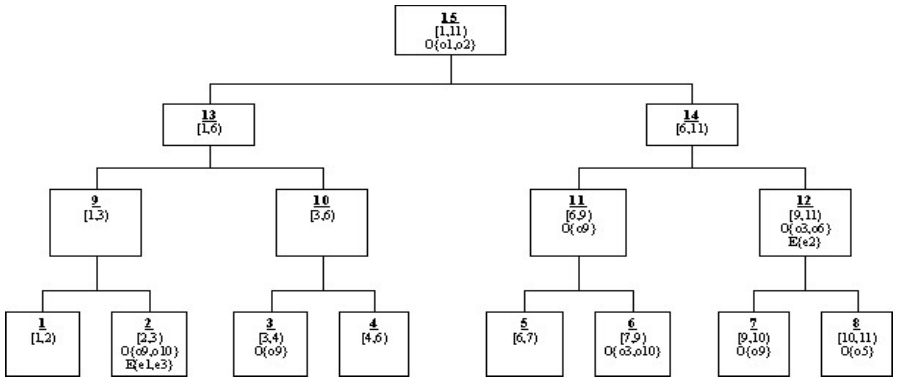

Fig. 2. Frame Segment Tree

objects and events that appeared in the entire frame-sequence associated with that node. If an object exceeds the frame interval of a node, then the object is represented in more than one node. For any node, if an object does not appear in all frames of that node, then, the object is inserted into the lists of sub-nodes of the node. For example, the appearance of object *o9* exceeds the time interval represented by node 2. Therefore *o9* is listed in the object list of both node 2 and node 3. (i.e. the intervals [2,3) and [3, 4)). However *o9* cannot be included in the list of node 13, because node 13 represents the interval [1,6) and *o9* does not appear in all frames of that interval.

In addition to the frame segment tree there are some additional index structures to access the nodes of the FST for a given entity. For instance for a given object, it is possible to traverse these index structures to locate the relevant nodes in the frame segment tree, thus to locate the frame sequences in which that object appears.

3 Modeling Spatio-Temporal Properties and Relationships

Handling spatial relationships in video databases is more difficult than in image databases, since video data has time-dependent properties. Some examples of spatial queries that can be applied to video are as follows:

- Find all frames, where first lady stands *close to* President Bush.
- Find all frames, where Bulent Ecevit is *at the right of* George Bush, and Tony Blair is *at the left*.
- Find the name of the Formula1 pilot that is *behind* Michael Schumacher between frames 100 and 120.

In this paper we introduce some extensions to AVIS in order to handle such kind of spatial queries. The extended data model, called spatio-temporal AVIS (ST-AVIS), handles spatial properties of objects in a video and allows the user to express spatial and spatio-temporal queries. In the rest of this section, we first discuss the representation of spatial properties of objects and how different spatial relationships between objects are computed and then show how we incorporate spatial querying into AVIS.

3.1 Spatial Properties of Video Objects

Spatial properties of objects in a video can be represented in different ways. In our model we use 2-dimensional coordinate system, in which the spatial property of the object is its position as appears on the screen. A common strategy is to use approximations for describing an object spatially. The most efficient method for the two-dimensional coordinate system is to use MBRs (Minimum Bounding Rectangles). We make use of a minimum rectangle that covers all parts of an object. This approach is similar to the MBR approach in approximating the location of objects.

The granularity of the coordinate system is an important issue, since a minimum of pixel-level granularity makes it hard to apply to video data. In our model, the user-specified rectangular areas on the screen represent spatial properties. We define the spatial property of an object as follows:

Definition 1. The spatial property of an object *A* is a *tuple (R, I)*, where, *R* is a rectangular area (a region) that covers all area in which object *A* appears during the time interval $I=[t_i, t_f]$. *R* is not exactly the minimum-bounding rectangle of *A*. At any time *t* in $[t_i, t_f]$, object *A* may be located anywhere in *R*.

Object *A* may be either static or moving in the rectangle *R* during the interval *I*. When object *A* is spatially static in *R*, then the rectangle *R* is practically the minimum-bounding rectangle of *A*. When *A* is moving in *R*, then we assume that the rectangle *R* is chosen so that *A* has a uniform movement in *R*, i.e. in a reasonable distance and as a single moving character. For example, while focusing on an object in the frame, if camera zooms out the object while it is moving, or if the object itself goes away, the size of the rectangle that covers the object gets smaller. In this case, for finer results, the scenes of zoom out should be handled in separate rectangles.

3.2 Spatial Relationships

Since the spatial property of an object is the minimum region that contains the object, the spatial relationship between two objects is obtained from the spatial relationship between the regions of each object. This property gives one the ability to find spatial relationships between any two objects.

Li et. al. [16] have a formal definition of each spatial relation in a rule base. They extended Allen's temporal interval algebra [3] into two-dimensional space in order to define spatial relationships between rectangular areas. They divide spatial relations into two groups; *directional* and *topological*. Directional relations include *south, north, west, east, northwest, northeast, southwest,* and *southeast*. Topological relations include *equal, inside, contain, cover, covered by, overlap, touch,* and *disjoint*. We use the notation of [16] in our formalization of spatial relations. In our notation, we use *left, right, top,* and *bottom* instead of *west, east, north*, and *south*, respectively. Our rule base covers the relations *top, bottom, right, left, top-right, top-left, bottom-right, bottom-left, overlaps, equal, inside, contain, touch*, and *disjoint*. Any two rectangular areas satisfying the definition of any of these spatial relations have those spatial relationship(s) between themselves.

Such a rule base can help the calculation of spatial relationships between two static objects, like in images, accurately. However in videos, objects are not static. They may be moving or they may seem to be moving because of a moving camera or zooming in or out. Therefore we need to deal with spatial relationships that change over time. At a certain snapshot of the video, one object may appear to be exactly to the left of another object. But within an interval of time, being its left might not be that exact all the time. Therefore we introduce fuzziness to the definition of spatial relationships between moving objects in videos. We define this fuzzy spatio-temporal relationship as follows:

Definition 2. Let A_i and A_j be two objects. Their *fuzzy spatio-temporal relationship* during time interval I_k is $A_i(\alpha,\mu,I_k)A_j$ where α is one of the spatial relations supported by our model (top, left, etc.), and μ is the value of membership. $A_i\alpha\mu A_j$ is true during the interval I_k.

Notice that this definition is similar to the definition of spatial relations between two moving objects given by Li et. al [16]. Their definition includes both directional and topological relations. Two different operators are used representing each relation type. We have used only one operator α for both directional and topological relations. However we use a fuzzy membership value, μ which is in [0,1], for any spatial relationship. The membership value determines how much the spatial position between two objects satisfies a given spatial relation.

We present some samples for LEFT relation between two objects in Fig. 3. In this figure, two objects, namely A and B, are assumed. Each occurrence of object A in the figure has a LEFT relationship with object B with different membership values. For a given time interval I and a membership μ, this relationship will be specified by A (LEFT, μ, I) B.

The membership value is used in the second step for the calculation of a relationship. The first step is to determine if any two objects satisfy the conditions stated in the rule base. If these conditions are not satisfied, membership value μ is assumed to be 0. Otherwise, we calculate the

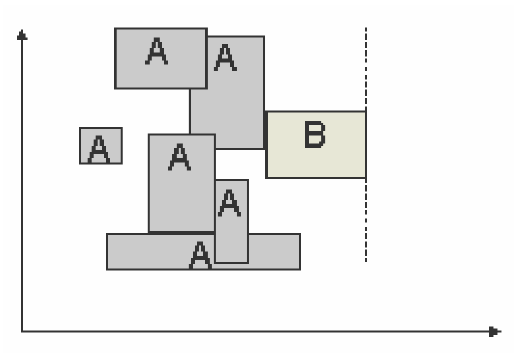

Fig. 3. Examples for LEFT relationship for two objects

Table 1. Relation between the membership value and angle between the centers of rectangles

Relation	**Angle** (*)	**Membership Value**
TOP	arctan(x/y)	1- (angle/90)
LEFT	arctan(y/x)	angle/90
TOP-LEFT	arctan(x/y)	1 – (abs(angle-45) / 45)
TOP-RIGHT	arctan(y/x)	1 – ((angle – 45) / 45)
* x is the horizontal distance between centers of two rectangles. * y is the vertical distance between centers of two rectangles.		

membership value of the relationship using the centres of rectangles. The angle between the x-axis and the line passing through the centres of the rectangles are used to calculate the membership value. Table 1 illustrates the relation between the angle and the membership value for each relation. Notice that, in this table, we represent only relations, *top, left, top-left*, and *top-right*. The reason is that, these relations are inverses of *bottom, right, bottom-right,* and *bottom-left* respectively. That is, TOP(A,B) is equal to BOTTOM(B,A), and so on. Therefore, only these four relations are enough for the calculation. The calculation of membership values allows us to perform fuzzy queries on video data. The user is able to specify the preciseness of the result of any query.

3.4 Extensions to AVIS

In this section we discuss how we incorporate spatial properties of video data into the AVIS data model. We have extended the contents of the association map and therefore the frame segment tree. Additional indexes to the frame segment tree are also modified.

In AVIS the association map shows the frame sequences in which the entities appear. However an object can appear in different locations in a single frame sequence. Therefore, we need to divide each frame sequence into sub-intervals according to the locations of the object. As a result, we represent *(interval, region)* pairs for objects in the extended association map. Fig. 4 presents a sample drawing of an extended association map where we used different patterns to indicate different locations of an object within a frame sequence. For example, in the figure, the boy appears in two separate frame sequences, but he is seen in three different locations in the first frame sequence and five different locations in second frame sequence.

In AVIS, each node in the frame segment tree has a list of objects existing in the interval represented by this node. In order to implement the region concept, the node structure in the frame segment tree is modified. In our model, a tree node is defined to contain a list of *(object, region)* pairs. This means that, each *(o_i, r_i)* pair in the list states that, "object o_i appears in the rectangular area r_i within the time interval represented by that node".

Storing objects with their regions within the time intervals had an important effect on the frame segment tree. An object *x* may appear in more than one region during a set of continuous frames. For example, a plane crossing the screen from left to right, visits more than one region. In the segment tree of AVIS, this entity is represented in only one FST-node, since the frames are continuous. However, in spatio-temporal model (ST-AVIS), we divide this single interval into

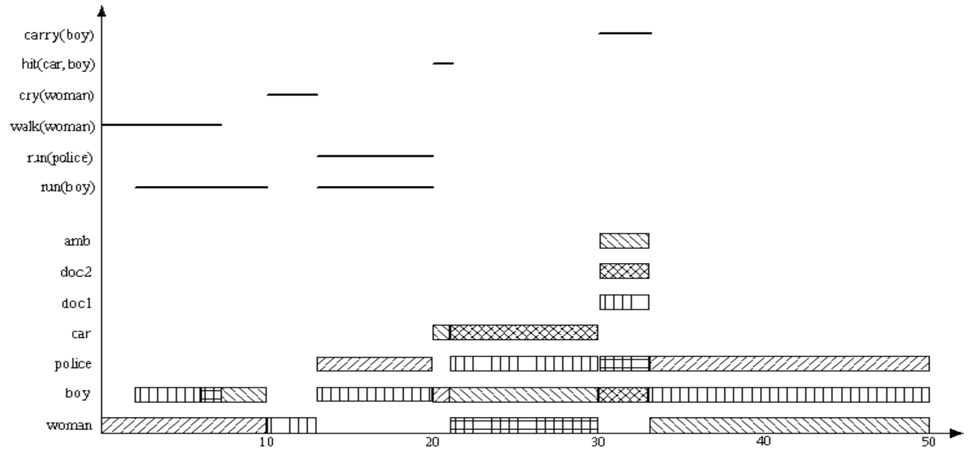

Fig. 4. The extended association map

sub-intervals, when the region of the object changes. This means that, each of these sub-intervals is represented with a separate tree node. As a result, the number of nodes in the tree increases. This affects the cost of operations on the tree proportional to the order of *log n*, where *n* is the number of nodes in the tree.

In order to achieve fast access to the nodes of the extended frame segment tree for a given object, we also modified the auxiliary indexes for objects. Each element in the object index points to a list of *(interval, region)* pairs, showing the intervals during which the object appears in the stated region. Additionally, for each *(interval, region)* pair, there is a list of pointers each of which points to the nodes in the frame segment tree, which is covered by the interval.

4 Supported Query Types

The model introduced in this paper allows all types of queries that are listed in [1]. In addition to these query types, our model supports querying the spatial properties of objects and the spatio-temporal relations between objects as well. We list the basic spatial queries into three categories:

- *Regional Queries*: Given an object and time interval, one may query the regions it appears. Or, given an object and its region one may query the frames.
- *(Fuzzy) Spatial Relationship Queries:* Given two objects and the spatial relationship between them we can query the frames in the video. Here we use fuzziness in the definition of the spatial relationship.
- *(Fuzzy) Spatio-temporal queries* (also called *Trajectory Queries*): Given an object we may query its trajectories.

The supported spatial query types are discussed in the following along with examples.

4.1 Regional Queries and Fuzzy Spatial Relationship Queries

Given object and interval, find regions: This is a rather simple query to handle with the data model of this study. An example query for this type can be "Give a list of regions, where the ball appears in, between 5th and 10th minutes". While processing the query, we first locate the object in the object index. Then, we trace the *(interval, region)* list of the object and retrieve regions of all intervals that intersect with the given interval in the query. A list of regions is returned as a result. The regions are represented with their *x-y* coordinates.

Given object and region, find frames: An example to this type of query is "List all frame sequences during which a plane appears in a given region". The computation is very similar to the previous one, since regions and intervals are kept in the same structure: The object is located in the index, and then all items in *(interval, region)* list are visited to find the given region. When the region is found, the corresponding interval is added to the resulting list. The intervals are sorted and consecutive ones are merged to obtain a solid list of intervals. When searching for the region we apply fuzziness since the region given in the query is only an approximation of the exact location of the object. Thus the equality of the regions is checked within a threshold.

Given objects and a spatial relation in between, find frames: This type of queries are of the form "Find the set of time intervals, in which object A is seen at left of object B, with a threshold value of 0.7". This is a fuzzy spatial relationship query. The model allows us to calculate this query with a cost proportional to the number of occurrences of the objects. We trace the regions of A, and for each interval, we look at the corresponding tree node, to find the object B in a suitable region. If B occurs and if its region satisfies the relationship within the threshold value, then the interval of current node is returned. As the result of the query, we have a solid set of frame intervals in which the given relationship occurs at least %70.

4.2 Trajectory Queries

Trajectory is the route of an object on the video in consecutive frames. In a trajectory query, we search both the time interval and the location of an object. A trajectory is composed of a set of sorted *(interval, region)* pairs, such that, any interval except the first one is the successor of the previous one in time, and any region except the first one is a neighbour of its predecessor. A formal definition of trajectory is given as the following:

Definition 3. A trajectory of an object o is a set $T_o = \{(I_n, R_n) \mid n > 1\}$ such that;

i. I_n is the time interval $[t_{ns}, t_{nf}]$,

ii. R_n is the rectangular area defined for object o in time interval I_n,

iii. For any i, where $1 < i \leq n$,

- $t_{(i+1)s} = t_{if} + 1$,
- R_{i+1} is a neighbour of R_i, where neighbourhood is defined in Definition 4.

Definition 4. Given two rectangles R_1 and R_2, with coordinates $(r_1x_1, r_1x_2, r_1y_1, r_1y_2)$ and $(r_2x_1, r_2x_2, r_2y_1, r_2y_2)$ respectively, R_1 is a neighbour of R_2 if and only if one of the following conditions is satisfied:

i. $r_1x_1 = r_2x_2 \wedge [r_1y_1, r_1y_2]$ and $[r_2y_1, r_2y_2]$ *overlaps*
ii. $r_1x_2 = r_2x_1 \wedge [r_1y_1, r_1y_2]$ and $[r_2y_1, r_2y_2]$ *overlaps*
iii. $r_1y_1 = r_2y_2 \wedge [r_1x_1, r_1x_2]$ and $[r_2x_1, r_2x_2]$ *overlaps*
iv. $r_1y_2 = r_2y_1 \wedge [r_1x_1, r_1x_2]$ and $[r_2x_1, r_2x_2]$ *overlaps*

where, *overlaps* is taken from Allen's temporal algebra [3]. Since this algebra can be applied to any one-dimensional space, we used it for x-axis and y-axis.

As an example of querying trajectories, consider a query of the form: "Find the trajectory of an object going from the left to the right of screen", where left and right of the screen are specified in the query interface (typically with the mouse). Here the left of screen and the right of screen are not precisely defined regions. The spatial property definitions of the object may not exactly fit in these stated regions. Therefore we use uncertainty in processing this kind of queries. We match the actual object regions with the imprecise regions given in the query by using approximation techniques. A degree of preciseness is requested with each query, in order to match the starting and the ending locations of the trajectory.

The neighbourhood of two rectangles is another point of uncertainty. We implemented a constant uncertainty degree, which is proportional to the areas of two rectangles. This degree may be user-specified or a commonly accepted value.

In our data model it is possible to find trajectories that start at any region on the screen and end at any region on the screen. Our model allows us to find the trajectories of objects that pass the screen from left to right, or diagonally, or from top to bottom, or in the reverse directions. The returned trajectories may not only be linear trajectories but also non-linear, even spiral.

5 Conclusions and Future Work

In this paper we presented a spatio-temporal video data model, which is an extension of the model named AVIS [1]. Our model represents the semantic content of video streams and allows users to model any kind of objects, events, and activities of interest in the video. The model presented here identifies spatial properties of objects with rectangular areas (regions) resembling MBRs. It is possible to compute spatial relationships between two rectangular areas, hence the objects covered by those rectangles. We can handle spatial relations left, right, top, bottom, top-left, top-right, bottom-left, bottom-right, as directional relations, and overlaps, equal, inside, contain, touch, and disjoint as topological relations. We also allow querying the trajectory of an object given the start and stop regions for the required trajectory.

A prototype of the proposed model is developed and implemented [14]. The implementation covers a data entry interface that allows users to do detailed frame-by-frame investigation on the video stream. It is also possible to see the current association map of any video stream graphically within the implementation. The application stores data for more than one video stream in its database. In case of a query development, there is a query interface to generate spatial and non-spatial queries on the video.

One advantage of this study is that, the proposed model is flexible in both design and implementation, so that it is easy to extend it with new spatial relations. New query types on object trajectories can be developed without much effort.

As a future study, the prototype can be improved to be a complete video database system. The other topological relationships such as *near, touch, cover*, can be defined without changing the proposed data model. We did not discuss the implementation of compound and conjunctive queries. However, we think that we have developed a base for these query types, although a further work is necessary for more efficient algorithms.

References

1. Adali, S., Candan, K.S., Chen, S., Erol, K., Subrahmanian, V.S.: The Advanced Video Information System: data structures and query processing, Multimedia Systems, vol. 4, 1996, pp. 172-186.
2. Ahanger, G., Little, T.D.C.: Data Semantics for Improving Retrieval Performance of Digital News Video Systems, IEEE Transactions on Knowledge and Data Engineering, Vol.13 No.3, pp. 352-360, June 2001.
3. Allen, J.F.: Maintaining Knowledge About Temporal Intervals, Comm. ACM, vol.26, pp.832-843, Nov. 1983.
4. Aygün, R.S., Yazıcı, A., Arıca, N.: Conceptual Data Modeling of Multimedia Database Applications, Proc. of IEEE Intl. Workshop on Multimedia DBMS, Ohio, pp. 182-189, August 1998.
5. Aygün, R.S., Yazıcı: A.: Modeling and Management of Fuzzy Information in Multimedia Database Applications, Multimedia Tools and Applications (to appear).
6. Chen, Y.R., Meliksetian, D.S., Chang, M.C.S., Liu, L.J.:Design of a Multimedia Object-Oriented DBMS, Multimedia Systems, vol. 3, pp. 217-227, 1995.
7. Davenport, G., Smith, T.A., Pincever, N.: Cinematic Primitives for Multimedia, IEEE Computer Graphics & Applications, pp. 67-74, July 1991.
8. Dönderler, M.E., Ulusoy, O., Güdükbay, U.: A rule-based video database system architecture, Information Sciences, Vol. 143, No. 1-4, pp. 13-45, June 2002.
9. Fagin, R.: Fuzzy Queries in Multimedia Database Systems, Proc. of 17th ACM SIGACT-SIGMOD-SIGART Symp. on PODS, Seattle, June 1998.
10. Gibbs, S., Breiteneder, C., Tsichritzis, D.:Data Modeling of Time-Based Media, Proc. ACM SIGMOD Conf. On Management of Data, Minnesota, pp. 91-102, June 1994.
11. Hacid, M.S., Decleir, C., Kouloumdijan, J.:A Database Approach for Modeling and Querying Video Data, IEEE Transactions on Knowledge and Data Engineering, Vol.12 No.5, pp. 729-750, October 2000.
12. Hjelsvold, R., Midtstraum, R.: Modeling and Querying Video Data, Proc. Intl. Conf. on Very Large Databases, Santiago, Chile, pp. 686-694, September1994.
13. Jiang, H., Elmagarmid, A.K.: Spatial and Temporal Content-Based Access To Hypervideo Databases, The VLDB Journal, vol. 7, pp. 226-238, 1998.
14. Koprulu, M.:A Spatio-Temporal Video Data Model, MSc Thesis, Middle East Technical University, Department of Computer Engineering, 2001.

15. Kuo, T.C.T., Chen, A.L.P.: A content-based video query language for video databases, Proc. of IEEE Int. Conf. on Multimedia Computing and Systems, Japan, IEEE Computer Society Press, pp. 209-214, June 1996.
16. Li, J.Z., Özsu, M.T., Szafron, D.: Modeling of Moving Objects in a Video Database, Proc. of IEEE Intl. Conf. on Multimedia Computing and Systems, Canada, pp. 336-343, 1997.
17. Little, T.D.C., Ahanger, G., Folz, R.J., Gibbon, J.F., Reeve, F.W., Shelleng, D.H., Venkatesh, D.: A Digital on Demand Video Service Supporting Content-Based Queries, Proc. ACM Multimedia, pp. 427-436, 1993.
18. Oomoto, E., Tanaka, K.: OVID: Design and Implementations of a Video-Object Database System, IEEE TKDE, 5,4, pp. 629-643, August 1993.
19. Swanberg, D., Shu, C.F., Jain, R.: Architecture of a multimedia information system for content-based retrieval, Audio Video Workshop, San Diego, California, 1992.
20. Weiss, R., Duda, A., Gifford, D.: Content-based access to algebraic video, Proc. First IEEE Int. Conf. On Multimedia Computing and Systems, 1994, Boston, MA. IEEE Computer Society Press, pp. 140-153, 1994.
21. Yazıcı, A. et al.: Handling Complex and Uncertain Information in the ExIFO and NF^2 Data Model, IEEE Transactions on Fuzzy Systems, Vol.7, No.6, December 1999.
22. Koyuncu, M., Yazici, A.: IFOOD: An Intelligent Fuzzy Object-Oriented Database Architecture, IEEE Trans. on Knowledge and Data Engineering (to appear).