University of Alberta

Modeling Video Spatial Relationships in an Object Model

by

John Z. Li, M. Tamer Özsu, Duane Szafron
Laboratory for Database Systems Research
Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada T6G 2H1
{zhong,ozsu,duane}@cs.ualberta.ca

Technical Report TR 96–06
March 1996

DEPARTMENT OF COMPUTING SCIENCE
The University of Alberta
Edmonton, Alberta, Canada

Modeling Video Spatial Relationships in an Object Model *

March 1996

Abstract

Video modeling has become a topic of increasing interest in the area of multimedia research. One of the key aspects in the video medium is spatial relationships. In this paper we propose a spatial representation, based on the temporal interval algebra, for specifying the spatial semantics of video data. Based on such a representation, a set of comprehensive spatial relationships for salient objects are defined in supporting qualitative and quantitative spatial properties. Further, both topological and directional spatial relationships are captured within the proposed model. We present a novel way of incorporating the spatial model into a video model, called a *common video object tree*, and integrating the abstract video model into an objectbase management system which has rich multimedia temporal operations. The integrated video objectbase management system supports a broad range of spatial queries and is extensible, thus allowing the easy incorporation of new features into the system. Our focus here is in supporting different types of spatial queries including direct spatial queries, hybrid

*This research is supported by a grant from the Canadian Institute for Telecommunications Research (CITR) under the Network of Centre of Excellence (NCE) program of the Government of Canada.

spatial queries, complex spatial queries, computational spatial queries, and temporal spatial queries. The integrated model is further enhanced by a spatial inference engine. The powerful expressiveness of our video model are validated by many concrete query examples.

Keywords: multimedia, spatial, object-oriented, database, video model, query, clips

Contents

List of Figures

List of Tables

1 Introduction

Management of multimedia data poses special requirements for database management systems. In a broad sense *multimedia data* includes the following data types: numeric data, character strings, graphics, images, audio, video, and animation. Many applications depend on spatial relationships among multimedia data. There is significant research on spatial relationships in image databases and geographic information systems (GIS) [OM88, RFS88, Ege91, CIT+93, AEG94, CSE94, PS94, SYH94, Ege94, NSN95, PTSE95]. On the other hand, very little research has been done on spatial modeling in the context of video data. Most work on videos [LG91, Mas91, OT93, LG93, WDG94, SW94, GBT94, HR95, LGÖS96] is concentrated on temporal relationships which are certainly the most striking characteristic of video data. However, this does not mean that spatial relationships are not important. Numerous query examples exist in which video retrieval must be done based on users' spatial specifications. For example, "*Find a video clip in which person A is at the left of person B*". A video spatial model is an essential part of an abstract multimedia information system model which can be used as the basis for declarative queries.

The information about the spatial semantics of a video must be structured so that indexes can be built to efficiently retrieve data from a video database. A *video* consists of a number of *clips*. A *clip* is a consecutive sequence of *frames*, which are the smallest units of video data. In this paper, we concentrate on *spatial relationships* in video data.

Spatial data pertains to spatial-oriented objects in a database including: points, lines, squares, polygons, surfaces, regions, and volumes. Spatial relations have been classified [PE88] into several types, including *topological relations* that describe neighborhood and incidence (e.g., overlap, disjoint), *directional relations* that describe order in space (e.g., south, northwest), and *distance relations* that describe space range between objects (e.g., far, near). These types of spatial relations have been studied independently and in association with each other. We focus on the first two types, i.e., topological and directional relations, because the distance relations are domain dependent and they are not as challenging as the other two.

How to handle user queries is one of the most important issues in modeling video spatial relationships. The special requirements of multimedia query languages in supporting spatial relationships have been investigated within the context of specific applications such as image database systems and geographic information systems [RFS88, SA95]. In our opinion, from a user's point of view the following requirements are necessary for supporting spatial queries in a multimedia information system:

- Support should be provided for object domains which consist of *complex* (structured) spatial objects in addition to simple (unstructured) points and alphanumeric domains. References to these spatial objects through their spatial domains must be directed by pointing to or describing the space they occupy and not by referencing their encodings.

- Support should exist for *direct spatial searches*, which locate the spatial objects in a given area of images. This can resolve queries of the form "*Find all the faces in a given area within an image or a video frame*".

- It should be possible to perform *hybrid spatial search*, which locates objects based on some attributes and some associations between attributes and the spatial objects. This can resolve queries of the form "*Display the person's name, age, and an image in which he/she is riding on a horse if the person is wearing blue jeans*". The riding horse image may be extracted from a frame of a video.

- Support should exist for *complex spatial searches*, which locate spatial objects across the database by using set-theoretic operations over spatial attributes. This can resolve queries of the form "*Find all the roads which pass through city X*" where one may need to get the location coordinates of city X and then check road maps to see which ones contain the coordinates.

- Support should be provided to perform *direct spatial computations*, which compute specialized simple and aggregate functions from the images. This can resolve queries of the form "*Tell me the area of this object and find another object which is closest to this one*".

- Finally support should exist for *spatio-temporal queries* which involve not only spatial relations, but temporal relations as well. This can resolve queries of the form "*Find a clip in which a dog is approaching someone from the left*".

We use the Common Video Object Tree model (CVOT) [LGÖS96] to build an abstract model. This abstract CVOT model is integrated into a powerful temporal object model to provide concrete objectbase management system (OBMS)[1] support for video data. The system that we use in this work is TIGUKAT[2] [ÖPS+95] which is an experimental system under development at the University of Alberta. Actually, any OBMS providing object-oriented techniques can be used here. The major contributions of this paper are the introduction of a unified representation of spatial objects, a complete set of definitions of both topological and directional relations, comprehensive support for all user spatial queries that we have elaborated above and support for user spatio-temporal queries. The unified representation is based on Allen's temporal interval algebra [All83] and a broad range of spatial topological and directional relations are supported. This is further enhanced by a rich set of spatial inference rules incorporated into the CVOT model in order to fully support complex spatial relationships between objects.

The rest of the paper is organized as follows. Section 2 reviews the related work in object spatial representations in image and video data. Section 3 introduces our representation of object spatial properties and relationships. Section 4 describes a new video model which captures common objects in videos. Furthermore, the requirements of an OBMS support are listed and a novel integration of the new model into an OBMS is also presented. Section 5 shows the expressiveness of our spatial representation by discussing many query examples. Section 6 summarizes our concluding remarks and possible future work.

[1] We prefer the terms "objectbase" and "objectbase management system" over the more popular terms "object-oriented database" and "object-oriented database management system", since the objects that are managed include code as well as data. Furthermore, we are using the term *video objectbase*, instead of *video database*.

[2] TIGUKAT (tee-goo-kat) is a term in the language of Canadian Inuit people meaning "objects." The Canadian Inuits (Eskimos) are native to Canada with an ancestry originating in the Arctic regions.

2 Related Work

Egenhofer [Ege91] has specified eight fundamental topological relations that can hold between two planar regions. These relations are computed using four intersections over the concepts of *boundary* and *interior* of pointsets between two regions embedded in a two-dimensional space. For example, let A^0 and B^0 be the interiors of objects A and B respectively and ∂A and ∂B be the boundaries of A and B respectively, then the combinations of intersection $(A^0 \cap B^0, A^0 \cap \partial B, \partial A \cap B^0, \partial A \cap \partial B)$ between interiors and boundaries define a set of topological relations. These four intersections result in eight topological relations: *disjoint, contains, inside, meet, equal, covers, covered_by*, and *overlap*. A spatial SQL [Ege94] based on this topological representation is proposed. The spatial SQL supports direct spatial search, hybrid spatial search, complex spatial search, and direct spatial computation.

Papadias et al. [PS94, PTSE95, GPP95] assume a construction process that detects a set of special points in an image, called *representative points*. Every spatial relation in the modeling space can be defined using only these representative points. Two kinds of representative points are considered: *directional representative points*, which are used to define directional relations, and *topological representative points*, which are used to define topological relations. For example, some possible directional representative points are the centroid of an object, the lower-left and upper-right corners of an object's minimum bounding rectangle (MBR), and a reference to a known object. Therefore, in the case of using two representative points the directional relations between objects can be defined as intervals which may facilitate the retrieval of spatial objects from a database using an R-tree based indexing mechanism [PTSE95]. Their topological reasoning work is based on Egenhofer's eight topological relations in two dimensional space. The topological relations are divided into three levels of resolution (high, medium, and low) according to the applications. The objective is to reduce the computational complexity whenever possible by using lower resolution. This approach transfers some burden to database designers.

Nabil et al. [NSN95] propose a two dimensional projection interval relationship (2D-PIR) to

represent spatial relationships based on Allen's interval algebra and Egenhofer's 4-intersection formalism. Then a graph representation for pictures based on 2D-PIR can be constructed. In order to overcome some problems of using the MBR with boundaries parallel to horizontal and vertical axes in the 2D-PIR representation, they propose two alternative solutions: slope projection and the introduction of topological relations. However, neither of these two solutions is complete in the sense that there still exist cases that the 2D-PIR representation cannot handle.

The Video Semantic Directed Graph (VSDG) model is a graph-based conceptual video model [DDI+95]. The most important feature of the VSDG model is an unbiased representation of the information that provides a reference framework for constructing a semantically heterogeneous user's view of the video data. The spatial property of an object in a VSDG is defined by a *bounding volume* $(MBR, depth, centroid)$. Here, MBR, *depth*, and *centroid* are the minimum bounding rectangle, the depth along the z-axis, and the centroid point of an object, respectively. The VSDG model also proposes to use Allen's temporal interval algebra to model spatial relations among objects. However, their definitions of such spatial relations are both incomplete and unsound.

Dimitrova and Golshani [DG94] describe a method to compute the trajectories of objects in a video database. Their objective is to discover motion using a dual hierarchy consisting of spatial and temporal parts for *video sequence* representation. Video sequences are identified by objects present in the scene and their respective motion. Their algorithm for motion detection uses the motion compensation component of the MPEG video encoding scheme. They focus on a high level abstraction of trajectories of objects, instead of spatial representations and spatial relations of objects.

Abdelmoty et al. [AEG94] extend the 4-intersection formalism [Ege91] for topological relations to represent *orientational* relations. The orientational relations always require a reference object called an *origin* to establish a spatial relation. Each object's bounding rectangle, together with four lines extending from the corners of the rectangle to the origin, are used to divide the space external to the object into four semi-infinite areas. The directional relations between two objects are defined using the intersections of these areas. One important result of this approach is that

the closer the objects, the stronger the dependency between the different relations. Hernández [Her94] defines the composition of topological and directional relations with the result being pairs of topological/directional relations. Composition is accomplished using *relative topological orientation nodes* as a store for intermediate results. This allows inferences such as if A *disjoint/right* B, B *disjoint/right-back* C then A *disjoint/right* or *disjoint/right-back* C. This work is extended in [CSE94] to handle composition of distance and directional relations.

3 Spatial Properties of Salient Objects

A *salient object* is an interesting physical object in a video frame. Each video frame usually has many salient objects, e.g. persons, houses, cars, etc. In this section we first describe the spatial representation of salient objects in our model and briefly introduce Allen's temporal interval algebra. Then, we provide complete definitions of spatial directional and topological relations, as well as some explanations. We also include a short discussion on integrating a set of spatial inference rules into our model. We use the term objects to refer to salient objects whenever this will not cause confusion.

3.1 Spatial Representations

It is a common strategy in spatial access methods to store object approximations and use these approximations to index the data space in order to efficiently retrieve the potential objects that satisfy the result of a query [PTSE95]. Depending on the application domain, there are several options in choosing object approximations. Minimum Bounding Rectangles (MBRs) have been used extensively to approximate objects because they need only two points for their representation. While MBRs demonstrate some disadvantages when approximating non-convex or diagonal objects, they are the most commonly used approximations in spatial applications. Hence, we use MBRs to represent objects in our system. We also assume there is always a finite set (possibly empty) of salient objects for a given video.

Definition 1 The *bounding box* of a salient object A_i is defined by its minimum bounding rectangle (X_i, Y_i, Z_i), where $X_i = [x_{s_i}, x_{f_i}], Y_i = [y_{s_i}, y_{f_i}], Z_i = [z_{s_i}, z_{f_i}]$. x_{s_i} and x_{f_i} are the salient object A_i's projection on the X axis with $x_{s_i} \leq x_{f_i}$ and similarly for y_{s_i} and y_{f_i}, z_{s_i} and z_{f_i}. The three intervals are represented by A_{ix}, A_{iy}, and A_{iz} respectively.

Definition 2 The *spatial property* of a salient object A_i is defined by a quadruple (X_i, Y_i, Z_i, C_i) where $X_i = A_{ix}, Y_i = A_{iy}, Z_i = A_{iz}$ and C_i is the centroid of A_i. The centroid is represented by a three dimensional point (x_i, y_i, z_i). This can be naturally extended by considering time dimension. I.e., the spatial property of a salient object A_i at time t is capture by $(X_i^t, Y_i^t, Z_i^t, C_i^t)$.

Basically, the spatial property of an object is described by its bounding box and a representative point, called the centroid or mass point. In video modeling we must also consider the time dimension as the spatial properties of an object may change over different time. For example, suppose the spatial property of A_i is $(X_i^{t_1}, Y_i^{t_1}, Z_i^{t_1}, C_i^{t_1})$ at time t_1 and the spatial property becomes $(X_i^{t_2}, Y_i^{t_2}, Z_i^{t_2}, C_i^{t_2})$ at time t_2. The displacement of A_i over time t_1 and t_2 is

$$DISPLACEMENT(A_i, t_1, t_2) \equiv \sqrt{(x_i^{t_1} - x_i^{t_2})^2 + (y_i^{t_1} - y_i^{t_2})^2 + (z_i^{t_1} - z_i^{t_2})^2}$$

which is the movement of the centroid of A_i. Also the distance between two objects A_i and A_j at time t_k is

$$DISTANCE(A_i, A_j, t_k) \equiv \sqrt{(x_i^{t_k} - x_j^{t_k})^2 + (y_i^{t_k} - y_j^{t_k})^2 + (z_i^{t_k} - z_j^{t_k})^2}$$

which is also characterized by the centroid of A_i and A_j. Our goal is to design a spatial representation that is powerful enough to support both quantitative and qualitative spatial retrieval.

3.2 Spatial Relationships

Spatial qualitative relations between objects are very important in multimedia objectbases because they implicitly support *fuzzy queries* which are captured by similarity matching or qualitative reasoning. It is well-known that precise matching usually generates no result in image or video

objectbases. Allen [All83] gives a temporal interval algebra (Table 1) for representing and reasoning about temporal relations between events represented as intervals. These temporal relations have been cited by others [Bee89, SF95, NSN95] for their simplicity and ease of implementation with constraint propagation algorithms. The elements of the algebra are sets of the seven basic relations that can hold between two intervals and the seven inverse relations.

Relation	Symbol	Inverse	Meaning
B before C	`b`	`bi`	BBB CCC
B meets C	`m`	`mi`	BBBCCC
B overlaps C	`o`	`oi`	BBB CCC
B during C	`d`	`di`	BBB CCCCC
B starts C	`s`	`si`	BBB CCCCC
B finishes C	`f`	`fi`	BBB CCCCC
B equal C	`e`	`e`	BBB CCC

Table 1: 13 Temporal Interval Relations

The temporal interval algebra essentially consists of the topological relations in one dimensional space enhanced by the distinction of the order of the space. The order is used to capture the directional aspects in addition to the topological relations. We consider 12 directional relations in our model and classify them into following three categories:

- *strict directional relations*: north, south, west, and east;
- *mixed directional relations*: northeast, southeast, northwest, and southwest;

- *positional relations*: above, below, left, and right.

The definitions of these relations in terms of Allen's temporal algebra are given in Table 2. The symbols $\wedge$ and $\vee$ are the standard logical *AND* and *OR* operators, respectively. A short notation $\{\}$ is used to substitute the $\vee$ operator over interval relations. For example $A_{ix}\,\{\mathtt{b},\mathtt{m},\mathtt{o}\}\,A_{jx}$ is equivalent to $A_{ix}\,\mathtt{b}\,A_{jx} \vee A_{ix}\,\mathtt{m}\,A_{jx} \vee A_{ix}\,\mathtt{o}\,A_{jx}$.

Among the Egenhofer's eight topological relations there are two inverse relations: *covers* vs *covered_by* and *inside* vs *contains*. Hence, only six topological relations are defined here as shown in the last part of Table 2. Note the definitions of directional and topological relations are based on two dimensional (2D) space since video frames are usually mapped into 2D images. To simplify our description, we only consider the 2D case. In 3D space, the depth of an object has to be considered and the extension is straightforward.

Figure 1 shows all the cases of A_i *north* of A_j ($A_i\,\mathtt{NT}\,A_j$). According to our definition if $A_i\,\mathtt{NT}\,A_j$, then $A_i\,\mathtt{AB}\,A_j$. In the case of $A_i\,\mathtt{NT}\,A_j \equiv A_{ix}\,\{\mathtt{d},\mathtt{di},\mathtt{s},\mathtt{si},\mathtt{f},\mathtt{fi},\mathtt{e}\}\,A_{jx} \wedge A_{iy}\,\{\mathtt{bi},\mathtt{mi}\}\,A_{jy}$, A_i's y interval must be always greater than or equal to A_j's y interval ($A_{iy}\,\{\mathtt{bi},\mathtt{mi}\}\,A_{jy}$). At the same time the intervals of A_{ix} and A_{jx} must satisfy one of the following conditions:

- A_{ix} and A_{jx} starts together but A_{jx} lasts longer ($A_{ix}\,\{\mathtt{s}\}\,A_{jx}$) or A_{ix} and A_{jx} starts together and A_{ix} lasts longer ($A_{ix}\,\{\mathtt{si}\}\,A_{jx}$);
- A_{ix} and A_{jx} finish at the same time with A_{jx} starting first ($A_{ix}\,\{\mathtt{f}\}\,A_{jx}$) or A_{ix} and A_{jx} finish at the same time with A_{ix} starting first ($A_{ix}\,\{\mathtt{fi}\}\,A_{jx}$);
- A_{ix} is a subinterval of A_{jx} ($A_{ix}\,\{\mathtt{d}\}\,A_{jx}$) or A_{jx} is a subinterval of A_{ix} ($A_{ix}\,\{\mathtt{di}\}\,A_{jx}$);
- A_{ix} and A_{jx} are equal ($A_{ix}\,\{\mathtt{e}\}\,A_{jx}$).

Figure 2 shows all the cases of A_i *northwest* of A_j ($A_i\,\mathtt{NW}\,A_j$). A_i *northwest* of A_j ($A_i\,\mathtt{NW}\,A_j$). Since the definition of northwest is $A_i\,\mathtt{NW}\,A_j \equiv (A_{ix}\,\{\mathtt{b},\mathtt{m}\}\,A_{jx} \wedge A_{iy}\,\{\mathtt{bi},\mathtt{mi},\mathtt{oi}\}\,A_{jy}) \vee (A_{ix}\,\{\mathtt{o}\}\,A_{jx} \wedge A_{iy}\,\{\mathtt{bi},\mathtt{mi}\}\,A_{jy})$, we may have following three cases:

Relation	Meaning	Definition
$A_i\,\texttt{ST}\,A_j$	South	$A_{ix}\,\{\texttt{d},\texttt{di},\texttt{s},\texttt{si},\texttt{f},\texttt{fi},\texttt{e}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{b},\texttt{m}\}\,A_{jy}$
$A_i\,\texttt{NT}\,A_j$	North	$A_{ix}\,\{\texttt{d},\texttt{di},\texttt{s},\texttt{si},\texttt{f},\texttt{fi},\texttt{e}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{bi},\texttt{mi}\}\,A_{jy}$
$A_i\,\texttt{WT}\,A_j$	West	$A_{ix}\,\{\texttt{b},\texttt{m}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{d},\texttt{di},\texttt{s},\texttt{si},\texttt{f},\texttt{fi},\texttt{e}\}\,A_{jy}$
$A_i\,\texttt{ET}\,A_j$	East	$A_{ix}\,\{\texttt{bi},\texttt{mi}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{d},\texttt{di},\texttt{s},\texttt{si},\texttt{f},\texttt{fi},\texttt{e}\}\,A_{jy}$
$A_i\,\texttt{NW}\,A_j$	Northwest	$(A_{ix}\,\{\texttt{b},\texttt{m}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{bi},\texttt{mi},\texttt{oi}\}\,A_{jy}) \vee (A_{ix}\,\{\texttt{o}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{bi},\texttt{mi}\}\,A_{jy})$
$A_i\,\texttt{NE}\,A_j$	Northeast	$(A_{ix}\,\{\texttt{bi},\texttt{mi}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{bi},\texttt{mi},\texttt{oi}\}\,A_{jy}) \vee (A_{ix}\,\{\texttt{oi}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{bi},\texttt{mi}\}\,A_{jy})$
$A_i\,\texttt{SW}\,A_j$	Southwest	$(A_{ix}\,\{\texttt{b},\texttt{m}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{b},\texttt{m},\texttt{o}\}\,A_{jy}) \vee (A_{ix}\,\{\texttt{o}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{b},\texttt{m}\}\,A_{jy})$
$A_i\,\texttt{SE}\,A_j$	Southeast	$(A_{ix}\,\{\texttt{b},\texttt{m}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{b},\texttt{m},\texttt{o}\}\,A_{jy}) \vee (A_{ix}\,\{\texttt{oi}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{b},\texttt{m}\}\,A_{jy})$
$A_i\,\texttt{LT}\,A_j$	Left	$A_{ix}\,\{\texttt{b},\texttt{m}\}\,A_{jx}$
$A_i\,\texttt{RT}\,A_j$	Right	$A_{ix}\,\{\texttt{bi},\texttt{mi}\}\,A_{jx}$
$A_i\,\texttt{BL}\,A_j$	Below	$A_{iy}\,\{\texttt{b},\texttt{m}\}\,A_{jy}$
$A_i\,\texttt{AB}\,A_j$	Above	$A_{iy}\,\{\texttt{bi},\texttt{mi}\}\,A_{jy}$
$A_i\,\texttt{EQ}\,A_j$	Equal	$A_{ix}\,\{\texttt{e}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{e}\}\,A_{jy}$
$A_i\,\texttt{IS}\,A_j$	Inside	$A_{ix}\,\{\texttt{d}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{d}\}\,A_{jy}$
$A_i\,\texttt{CV}\,A_j$	Cover	$(A_{ix}\,\{\texttt{di}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{fi},\texttt{si},\texttt{e}\}\,A_{jy}) \vee (A_{ix}\,\{\texttt{e}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{di},\texttt{fi},\texttt{si}\}\,A_{jy}) \vee$ $(A_{ix}\,\{\texttt{fi},\texttt{si}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{di},\texttt{fi},\texttt{si},\texttt{e}\}\,A_{jy})$
$A_i\,\texttt{OL}\,A_j$	Overlap	$A_{ix}\,\{\texttt{d},\texttt{di},\texttt{s},\texttt{si},\texttt{f},\texttt{fi},\texttt{o},\texttt{oi},\texttt{e}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{d},\texttt{di},\texttt{s},\texttt{si},\texttt{f},\texttt{fi},\texttt{o},\texttt{oi},\texttt{e}\}\,A_{jy}$
$A_i\,\texttt{TC}\,A_j$	Touch	$(A_{ix}\,\{\texttt{m},\texttt{mi}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{d},\texttt{di},\texttt{s},\texttt{si},\texttt{f},\texttt{fi},\texttt{o},\texttt{oi},\texttt{m},\texttt{mi},\texttt{e}\}\,A_{jy}) \vee$ $(A_{ix}\,\{\texttt{d},\texttt{di},\texttt{s},\texttt{si},\texttt{f},\texttt{fi},\texttt{o},\texttt{oi},\texttt{m},\texttt{mi},\texttt{e}\}\,A_{jx} \wedge A_{iy}\,\{\texttt{m},\texttt{mi}\}\,A_{jy})$
$A_i\,\texttt{DJ}\,A_j$	Disjoint	$A_{ix}\,\{\texttt{b},\texttt{bi}\}\,A_{jx} \vee A_{iy}\,\{\texttt{b},\texttt{bi}\}\,A_{jy}$

Table 2: Directional and Topological Relation Definitions

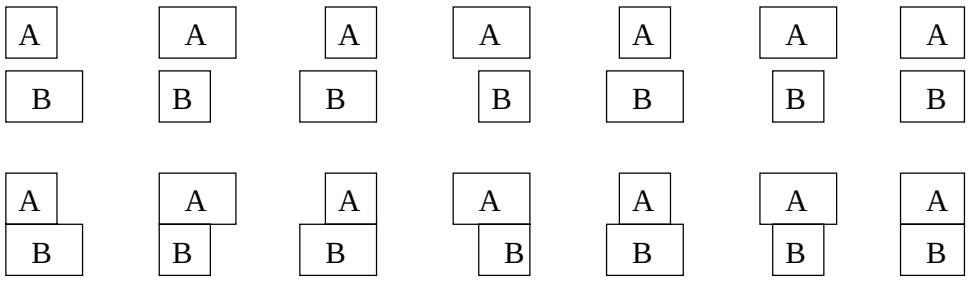

Figure 1: All the Cases of `NT`

- If A_{ix} is before A_{jx} (A_{ix} {`b`} A_{jx}), A_{iy} can be after, met by, or overlapped by A_{jy} (A_{iy} {`bi`, `mi`, `oi`} A_{jy}). These cases correspond (a), (b), and (c) of Figure 2 respectively.
- If A_{ix} meets A_{jx} (A_{ix} {`m`} A_{jx}), A_{iy} can be after, met by, or overlapped by A_{jy} (A_{iy} {`bi`, `mi`, `oi`} A_{jy}). These cases correspond (d), (e), and (f) of Figure 2 respectively.
- If A_{ix} overlaps with A_{jx} (A_{ix} {`o`} A_{jx}), A_{iy} can only be either after or met by A_{jy} (A_{iy} {`bi`, `mi`} A_{jy}). These cases correspond (g), and (h) of Figure 2 respectively.

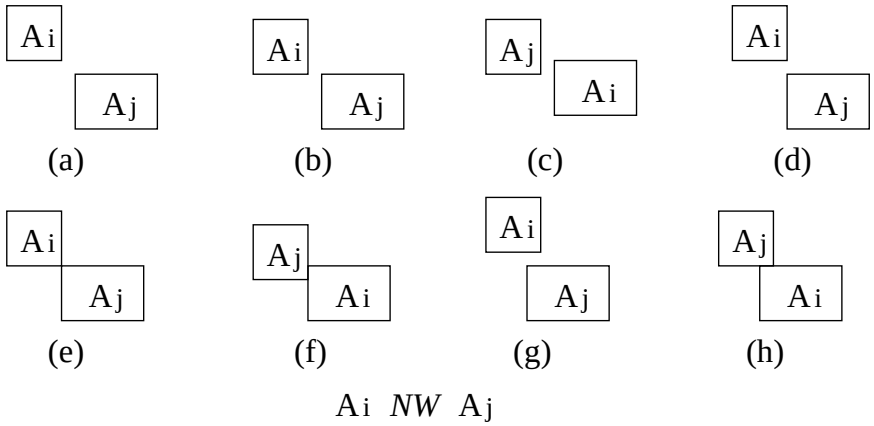

Figure 2: All the Cases of `NW`

Figure 3 shows all the topological relations. While any two spatial objects always have a topological relation, they may not have any directional relation. For instance, consider objects A_i and

A_j in the case of A_i OL A_j in Figure 3. A_i and A_j have no any directional relation. This coincides with our intuition about spatial objects.

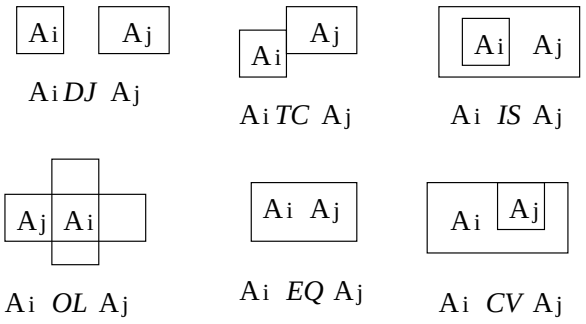

Figure 3: Definitions of Topological Relations

The definition of A_i *above* A_j (A_i AB $A_j \equiv A_{iy}$ {bi, mi} A_{jy}) requires that A_i's projection on the y-axis is greater than or equal to A_j's projection on the y-axis. The *above* relation includes A_i *north* of A_j (A_i NT A_j) because A_i *north* of A_j requires A_i's projection on the y-axis to be greater than or equal to A_j's projection on the y-axis and some restrictions on the x-axis projections. Furthermore, the *above* relation includes part of A_i *northwest* of A_j (A_i NW A_j) because the requirement of A_i's projection on the y-axis greater than or equal to A_j's projection on the y-axis is implied in relation *northwest* of in some cases. Similarly, the *above* relation includes part of A_i *northeast* of A_j (A_i NE A_j) for the same reason. Our positional relations are more general than those defined in [SYH94] because only the top half (A_i and A_j are not externally connected) satisfy the relation *above* among all the cases of *north* shown in Figure 1.

The definition of A_i *overlap* A_j (A_i OL A_j) indicates that object A_i shares some region with object A_j. If this shared region becomes just either a line or a point, then we say that object A_i touches object A_j (A_i TC A_j). A_i is *disjoint* A_j (A_i DJ A_j) means that object A_i shares no region with object A_j.

In our definition, if two objects overlap, they do not have any directional relation. This is certainly an arguable definition. Let us look at Figure 4. It is natural to say A_i *overlaps* A_j in (a) and A_i *west* of A_j in (c). However, it may not be reasonable to claim that these relations are still

hold in cases (b) and (d) respectively. The problem comes from the representation of the temporal interval algebra which does not distinguish the degree of the overlap regions in these cases. All overlaps are treated same. Even worse, in Figure 4(e) A_i and A_j do not have a clear directional relation. This may not be satisfactory in some fine-grain multimedia applications.

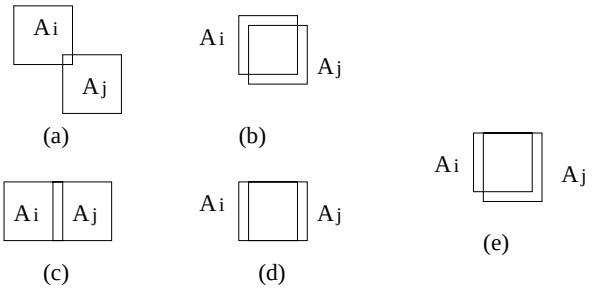

Figure 4: Some Non-directional Spatial Cases

Nevertheless, using the interval relations (algebra) to capture both directional and topological relations of spatial objects can offer more information about spatial relations than traditional methods [NSN95]. In other words, it has greater expressive power than traditional methods. Adopting such an interval algebra is especially attractive in multimedia objectbase systems, compared to GIS and image systems, because most multimedia systems already support Allen's temporal algebra in their temporal models. Hence, no special treatment is required for spatial intervals from an implementation point of view.

3.3 Reasoning about Spatial Relations

Logic-based representations, such as rules, are used in qualitative spatial reasoning since they provide a natural and flexible way to represent spatial knowledge [PS94]. Such a representation usually has well defined semantics and simple inference rules that can be integrated into any deductive system. For example, if there are A_1 *north* of A_2, and A_2 *overlap* A_3, and A_3 *north* of A_4, then we should have A_1 *above* A_4, which can be expressed as a rule

$$A_1 \,\mathtt{NT}\, A_2 \wedge A_2 \,\mathtt{OL}\, A_3 \wedge A_3 \,\mathtt{NT}\, A_4 \Rightarrow A_1 \,\mathtt{AB}\, A_4.$$

A spatial inference rule within a spatial DBMS can support spatial analysis without transforming any spatial knowledge into the domain of underlying coordinates and point-region representations. Instead, reasoning with imprecise and incomplete information may be achieved in a purely qualitative matter or, when necessary and available, augmented by quantitative information. Another major advantage of using spatial inference rules is to save space within video objectbases because it is not reasonable to explicitly store all the spatial relations between salient objects.

We have constructed a comprehensive set of spatial inference rules [LÖS96] and have proven the correctness of those rules. Both topological and directional relations are considered in the rules. Therefore, a broad range of qualitative spatial queries are supported. Since all the rules are propositional Horn clauses, they can be easily integrated into any multimedia objectbase by either using a simple inference engine or using a lookup table.

4 Video Modeling

Video modeling is the process of translating raw video data into an efficient internal representation which helps to capture video semantics. The procedural process of extracting video semantics from a video is called *video segmentation*. There are two approaches to video segmentation in an object-oriented context: *stream-based* and *structured*. In a stream-based approach, a clip is considered as a sequence of *frames* that are displayed at a specified rate. In a structured approach, a clip is considered as a sequence of scenes. Each approach has its own advantages and disadvantages as described in [Gha96]. However, very little work [Gha96] has been done on the structured approach because of its technical difficulties. On the other hand, the stream-based approach has received most of the research attention because of its technical feasibility. We concentrate on stream-based approaches. In this section we briefly introduce the Common Video Object Tree (CVOT) model, we have developed for video modeling, and its integration into a temporal OBMS.

4.1 The Common Video Object Tree Model

There are several different ways to segment a video into clips, e.g., by *fixed time intervals* or by *shots*. A *fixed time interval* segmentation approach divides a video into equal length clips using a predefined time interval (e.g. 2 seconds) while a *shot* is a set of continuous frames captured by a single camera action [HJW95]. Two common problems with existing models are restrictive video segmentation and poor user query support. The CVOT model [LGÖS96] is primarily designed to deal with these two problems. In the CVOT model, there is no restriction on how videos are segmented. Without loss of generality, we assume that any given video stream has a finite number of clips and any clip has a finite number of frames as shown in Figure 5. One unique feature of the CVOT model is that a clip overlap is allowed. This can bring a lot of benefit in modeling *events* which will be discussed in Section 4.3. Generally, a smooth transition of one event to another event requires to have some scene or activity overlap between the end of the previous event and the start of the next event. Such a transition phase is usually reflected in a few frames and this is shown in Figure 5.

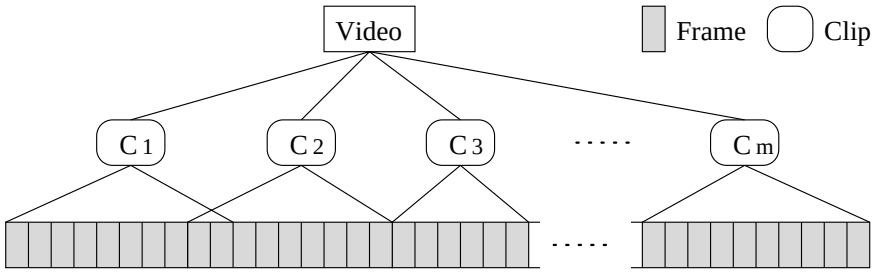

Figure 5: Stream-based Video Clips and Frames

The main idea of the CVOT model is to find all the common objects among clips and to group clips according to these objects. A tree structure is used to represent such a clip group. The *time interval* of a clip is defined according to the clip's starting frame and ending frame.

Example 1 Figure 6 shows a video in which John and Mary are walking toward their house. Later, Mary rides a horse on a ranch with her colt and dog. Let us assume that the salient objects are

$SO = \{\mathsf{john}, \mathsf{mary}, \mathsf{house}, \mathsf{tree}, \mathsf{horse}, \mathsf{colt}, \mathsf{dog}\}$. If the video is segmented as in Figure 6, then we have five clips $C = \{C_1, C_2, C_3, C_4, C_5\}$. Furthermore, john, mary, house, and tree are in C_1; john, house, and tree are in C_2; mary, horse, colt, and dog are in C_3; mary, horse, and colt are in C_4; and mary, horse, colt, and dog are in C_5.

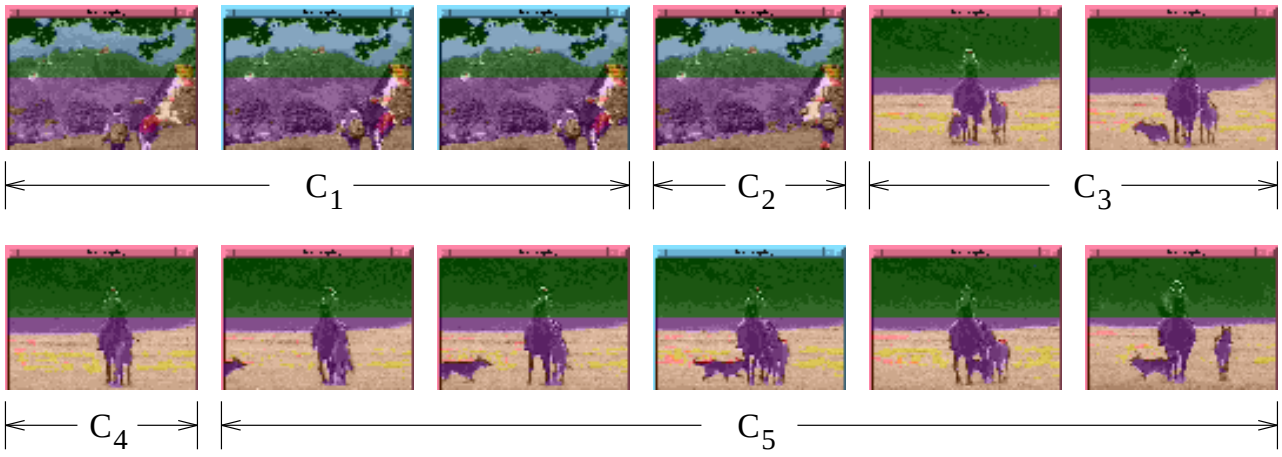

Figure 6: Salient Objects and Clips

Figure 7 shows a CVOT instance for Figure 6. In Figure 7, node C_1 has time interval $[1, 3]$ and a set of salient objects $\{\mathsf{john}, \mathsf{mary}, \mathsf{house}, \mathsf{tree}\}$; node C_2 has time interval $[4, 4]$ and a set of salient objects $\{\mathsf{john}, \mathsf{house}, \mathsf{tree}\}$; node C_3 has time interval $[5, 6]$ and a set of salient objects $\{\mathsf{mary}, \mathsf{horse}, \mathsf{colt}, \mathsf{dog}\}$; node C_4 has time interval $[7, 7]$ and a set of salient objects $\{\mathsf{mary}, \mathsf{horse}, \mathsf{colt}\}$; node C_5 has time interval $[8, 12]$ and a set of salient objects $\{\mathsf{mary}, \mathsf{horse}, \mathsf{colt}, \mathsf{dog}\}$; There are 3 common objects between C_1 and C_2 and this number is reduced to 0 if C_3 is added. Therefore, C_1 and C_2 have a parent node N_1 with a time interval $[1, 4]$ and a salient object set $\{\mathsf{john}, \mathsf{house}, \mathsf{tree}\}$. There are 3 common objects between C_3 and C_4 and this number is not reduced if C_5 is added. Therefore, C_3, C_4, and C_5 have a parent node N_2 with time interval $[5, 12]$ and a set of salient objects $\{\mathsf{mary}, \mathsf{horse}, \mathsf{colt}\}$. As there is no common object between N_1 and N_2, the *Root* node has time interval $[1, 12]$ with an empty salient object set. The CVOT model directly supports queries of the type "*Find all the clips in which a salient object appears*" and "*How long does a particular salient object occur in a video*".

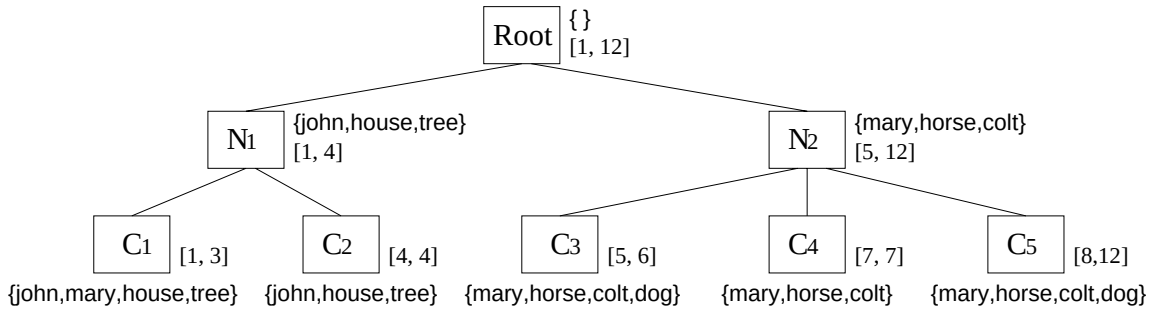

Figure 7: A Common Video Object Tree Built from Figure 3

4.2 The OBMS Support

CVOT is an abstract model; to have proper objectbase management support for continuous media, this model needs to be integrated with an object model. We choose an object model for this purpose for obvious reasons. In particular we work within the context of the TIGUKAT system [ÖPS+95]. In this section we introduce the TIGUKAT object model and its temporal extension.

The TIGUKAT object model [ÖPS+95] is purely *behavioral* with a *uniform* object semantics. The model is *behavioral* in the sense that all access and manipulation of objects is based on the application of behaviors to objects. The model is *uniform* in that every component of information, including its semantics, is modeled as a *first-class object* with well-defined behavior. Other typical object modeling features supported by TIGUKAT include strong object identity, abstract types, strong typing, complex objects, full encapsulation, multiple inheritance, and parametric types.

The primitive objects of the model include: *atomic entities* (reals, integers, strings, etc.); *types* for defining common features of objects; *behaviors* for specifying the semantics of operations that may be performed on objects; *functions* for specifying implementations of behaviors over types; *classes* for automatic classification of objects based on type[3]; and *collections* for supporting general heterogeneous groupings of objects. In this paper, a reference prefixed by "`T_`" refers to a type,

[3]Types and their extents are separate constructs in TIGUKAT.

"**C_**" to a class, "*B_*" to a behavior, and "T_X< T_Y >" to the type T_X parameterized by the type T_Y. For example, T_person refers to a type, **C_person** to its class, *B_age* to one of its behaviors and T_collection< T_person > to the type of collections of persons. A reference such as David, without a prefix, denotes some other application specific reference.

The primitive type system is a complete lattice with the T_object type as the root of the lattice and the T_null type as the base. T_null binds the lattice from the bottom. It is a subtype of every other type in the system. The access and manipulation of an object's state occurs exclusively through the application of behaviors. We clearly separate the definition of a behavior from its possible implementations (functions). The benefit of this approach is that common behaviors over different types can have a different implementation in each of the types. This provides direct support for behavior *overloading* and *late binding* of functions (implementations) to behaviors.

The model separates the definition of object characteristics (a *type*) from the mechanism for maintaining instances of a particular type (a *class*). A *type* defines behaviors and encapsulates behavior implementations and state representation for objects created using that type as a template. The behaviors defined by a type describe the *interface* to the objects of that type.

Temporality has been added to this model [GLÖS96] as type and behavior extensions of the type system discussed above. Figure 8 gives part of the time type hierarchy that includes the temporal ontology and temporal history features of the temporal model. Unary operators which return the lower bound, upper bound and length of the time interval are defined. The model supports a rich set of ordering operations among intervals, e.g., *before*, *overlaps*, *during*, etc. (see Figure 1) as well as set-theoretic operations viz *union*, *intersection* and *difference*[4]. A time duration can be added or subtracted from a time interval to return another time interval. A time interval can be expanded or shrunk by a specified time duration.

A *time instant* (*moment*, *chronon*, etc.) is a specific anchored moment in time. A time instant

[4]Note that the union of two disjoint intervals is not an interval. Similarly, for the difference operation, if the second interval is contained in the first, the result is not an interval. In the temporal model, these cases are handled by returning an object of the *null* type (T_null).

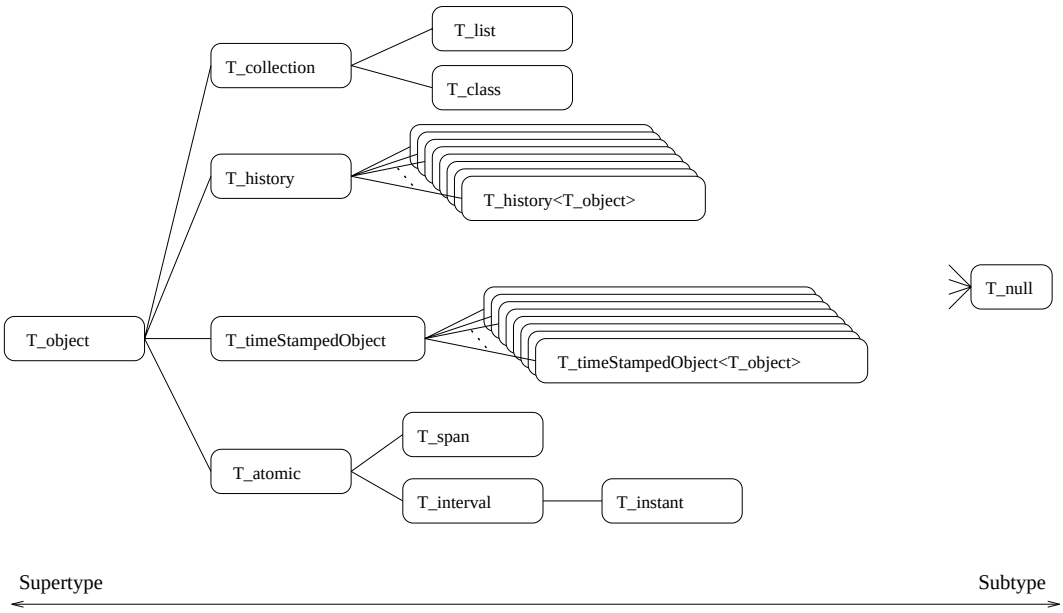

Figure 8: The Basic Time Type Hierarchy

can be compared with a time interval to check if it falls before, within or after the time interval. A *time span* is an unanchored relative duration of time. A time span is basically an atomic cardinal quantity, independent of any time instant or time interval. One requirement of a temporal model is an ability to adequately represent and manage histories of objects and real-world events. Our model represents the temporal histories of objects whose type is `T_X` as objects of the `T_history<T_X>` type as shown in Figure 8. A temporal history consists of objects and their associated timestamps (time intervals or time instants). A *timestamped object* knows its timestamp and its associated object (value) at (during) the timestamp. A temporal history is made up of such objects. Table 3 gives the behaviors defined on histories and timestamped objects. Behavior *B_history* defined on `T_history<T_X>` returns the set (collection) of all timestamped objects that comprise the history. Another behavior defined on history objects, *B_insert*, timestamps and inserts an object in the history. The *B_validObjects* behavior allows the user to get the objects in the history that were valid at (during) the given time.

Each timestamped object is an instance of the `T_timeStampedObject<T_X>` type. This type rep-

T_history<T_X>	*B_history*:	T_collection<T_timeStampedObject<T_X>>
	B_insert:	T_X,T_interval → T_boolean
	B_validObjects:	T_interval → T_collection<T_timeStampedObject<T_X>>
T_timeStampedObject<T_X>	*B_value*:	T_X
	B_timeStamp:	T_interval

Table 3: Behaviors on Histories and Time-stamped Objects

resents objects and their corresponding timestamps. Behaviors *B_value* and *B_timeStamp* defined on `T_timeStampedObject` return the value and the timestamp of a timestamped object, respectively.

4.3 System Integration

Integrated multimedia systems can result in a uniform object model, simplified system support and possibly better performance. In such a system, the multimedia component can directly use many functions provided by the OBMS, such as concurrency control, data recovery, access control etc. Figure 9 shows our video type system. The types that are in a grey shade are directly related to the CVOT model and they will be discussed in detail in following subsections.

4.3.1 Integrated System Model

We start by defining the `T_video` type to model videos. An instance of `T_video` has all the semantics of a video and is modeled as a history of clips. We model a clip set by defining the behavior *B_clips* in `T_video`. *B_clips* returns a history object of type `T_history< T_clip >`, whose elements are timestamped objects of type `T_clip` (`T_timeStampedObject < T_clip >`).

Example 2 Suppose myVideo is an instance (object) of `T_video`. Then,

- myVideo.*B_clips* returns an instance (object) of type `T_history< T_clip >`. Let this object be myVideoClipHistory.

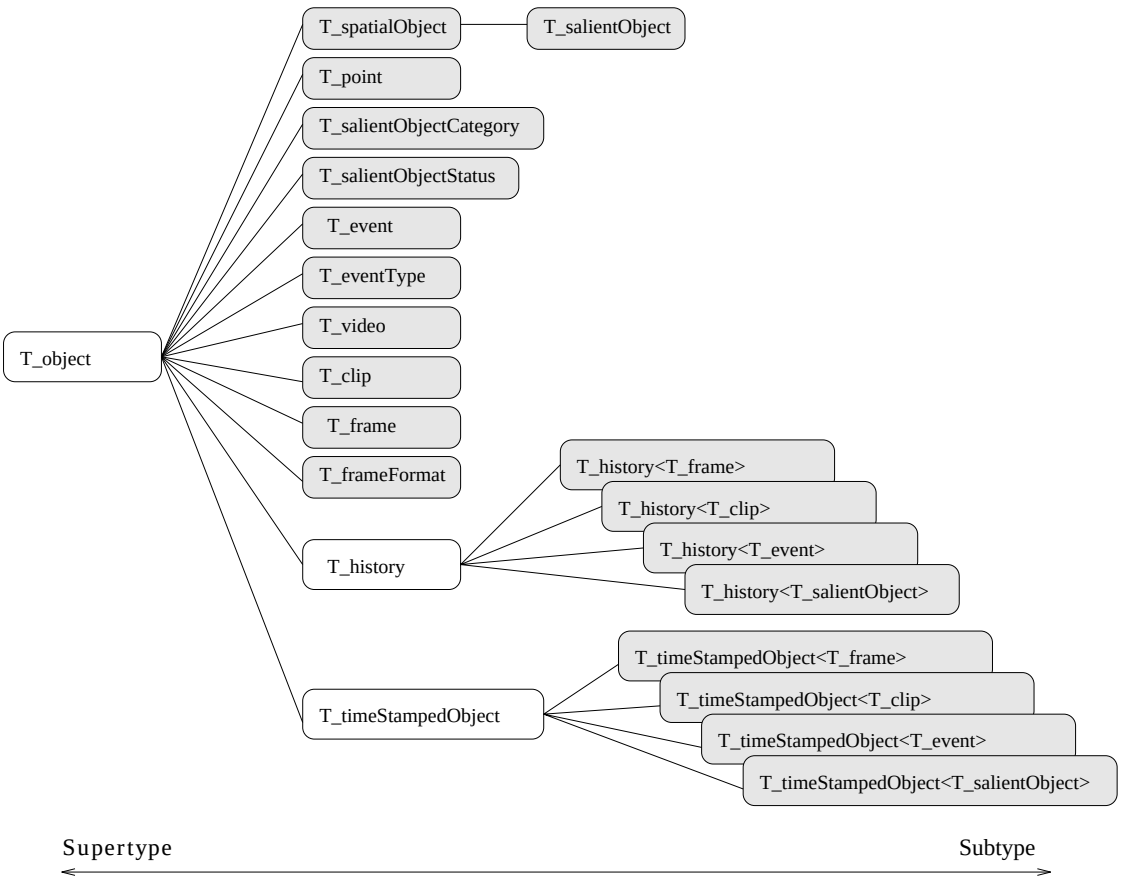

Figure 9: The Video Type System

- myVideoClipHistory.*B_history* returns a collection (clip set) which contains all the timestamped clip objects of type `T_timeStampedObject< T_clip >` in myVideo. As for Example 1 this collection is $\{C_1, C_2, C_3, C_4, C_5\}$. Let one of these clip history objects be myVideoCHOneClip.

- myVideoCHOneClip.*B_timeStamp* returns the time interval of myVideoCHOneClip. For example, C_3.*B_timeStamp* returns $[5, 6]$. myVideoCHOneClip.*B_value* returns the content of myVideoCHOneClip. Therefore, C_3.*B_value* returns C_3 without a time interval.

Table 4 gives the behavior signatures of videos.

`T_video`	*B_clips*:	`T_history<T_clip>`
	B_cvotTree:	`T_tree`
	B_search:	`T_salientObject, T_tree` → `T_tree`
	B_length:	`T_span`
	B_publisher:	`T_collection<T_company>`
	B_producer:	`T_collection<T_person>`
	B_date:	`T_instant`
	B_play:	`T_boolean`
`T_clip`	*B_frames*:	`T_history< T_frame >`
	B_salientObjects:	`T_collection<T_history<T_salientObject>>`
	B_events:	`T_collection<T_history<T_event>>`
`T_frame`	*B_location*:	`T_instant`
	B_format:	`T_videoFormat`
	B_content:	`T_image`

Table 4: Behavior Signatures of Videos, Clips, and Frames

The behavior *B_cvotTree* on `T_video` returns an instance of a CVOT for a video. A common question to myVideo would be its length (duration). This is modeled by the *B_length* behavior and it returns an object of type `T_span`. Video information should also include metadata, such as the publishers, producers, publishing date, etc. A video can also be played by using *B_play*[5].

Each clip has a set of consecutive frames, which is modeled by `T_history<T_frame>`. All the salient objects within a clip are grouped by the behavior *B_salientObjects* which returns an instance

[5] A full set of behaviors can, of course, be defined on `T_video` to enable typical actions, such as pause, fast forward, and rewind. We do not elaborate on these any further in this paper.

of T_collection< T_history < T_salientObjects >>. Similarly, All the events within a clip are grouped by the behavior *B_events* which returns an instance of T_collection< T_history < T_event >>.

The basic building unit of a clip is the frame which is modeled by T_frame in Table 4. A frame knows its location within a clip and such a location is modeled by a time instant (*B_location*), which can be a relative frame number. We model frames within a clip as a history which is identical to how we model clips within a video. It is possible to have different types of frames in a video objectbase, e.g. predicted frames, intracoded frames and bidirectional frames in MPEG videos [Gal91]. This is defined by the behavior *B_format* of T_frame. *B_format* is based on type T_frameFormat, an enumerate type, defines the format of a frame. The content of a frame, *B_content*, is an image which defines many image properties such as width, height and color.

4.3.2 Modeling Video Features

The semantics or contents of a video is usually expressed by its *features* which include video attributes and the relationships between these attributes. Typical video features are salient objects and *events*. An *event* is a kind of activity which may involve many different salient objects over a time period, like holding a part, walking, and riding a horse etc.

An event can occur in different places either within a clip or crossing multiple clips. For example, the event maryRide may occur in multiple clips. Additionally, this event may occur several times within a clip. Therefore, an appropriate representation is necessary to capture the temporal semantics of general events. A simple and natural way to model the temporal behavior of events is to use historical structure. Thus, we model histories of events as objects of type T_history< T_event >. Instances, such as maryRide, of T_history< T_event > consist of timestamped events. The time interval of an event does not have to be restricted to a clip interval so that an event can cross multiple clips. In the interest of tracking all the events occurring within a clip, the behavior *B_events* is included in T_clip.

Similarly, since salient objects can also appear multiple times in a clip or a video, we model the history of a salient object as timestamped object of type `T_history< T_salientObject >`. The behavior *B_salientObjects* of `T_clip` returns all the salient objects within a clip. Using histories to model salient objects and events results in powerful queries as will be shown in the next subsection. Furthermore, it enables us to uniformly capture the temporal semantics of video data because a video is modeled as a history of clips and a clip is modeled as a history of frames. Since any object occupying some space is an instance of `T_spatialObject`, `T_salientObject` is a subtype of `T_spatialObject`.

The behavior *B_activity* of `T_event`, shown in Table 5, identifies the type of events, while `T_eventType` and the behavior *B_roles* indicates all the salient objects which are involved in an event. *B_eventObjects* returns all the salient objects within an event. *B_inClips* indicates all the clips in which this event occurs. It is certainly reasonable to include other information, such as the location and the real-world time of an event, into type `T_event`, but they are not important to our discussion.

In type `T_salientObject`, the behavior *B_inClips* returns all the clips in which the salient object appears. *B_category* describes the category of salient objects, such as static objects (e.g. mountains, houses, trees) and mobile objects (e.g., cars, horses, boats). *B_status* may be used to define some other attributes of salient objects. For example, it is very useful to know whether an object is *rigid* or not if we want to track the motion of the object. Here `T_salientObjectStatus` is defined to capture this property. The rest of the behaviors are related to the directional and topological relations and they are self-explanatory. The spatial properties of salient objects are captured by spatial objects.

Table 5 also shows the behavior signatures of spatial objects. The behaviors *B_xinterval*, *B_yinterval*, and *B_zinterval* of type `T_spatialObject` define the x-interval, y-interval, and z-interval of an object respectively. These behaviors are computed from the projections of the object's bounding box over x, y, z axes. The behavior *B_centroid* returns the centroid of the object while the behavior *B_area* returns the region occupied by the object. The distance between objects at a

T_event	*B_activity*:	T_eventType
	B_roles:	T_collection<T_salientObject>
	B_inClips:	T_video → T_history< T_clip >
	B_eventObjects:	T_collection<T_salientObject>
T_salientObject	*B_inClips*:	T_video → T_history< T_clip >
	B_category:	T_salientObjectCategory
	B_status:	T_status
T_spatialObject	*B_xinterval*:	T_interval
	B_yinterval:	T_interval
	B_zinterval:	T_interval
	B_centroid:	T_point
	B_area:	T_real
	B_displacement	T_interval, T_interval → T_real
	B_distance	T_spatialObject, T_interval → T_real
	B_south:	T_spatialObject → T_boolean
	B_north:	T_spatialObject → T_boolean
	B_west:	T_spatialObject → T_boolean
	B_east:	T_spatialObject → T_boolean
	B_northwest:	T_spatialObject → T_boolean
	B_northeast:	T_spatialObject → T_boolean
	B_southwest:	T_spatialObject → T_boolean
	B_southeast:	T_spatialObject → T_boolean
	B_left:	T_spatialObject → T_boolean
	B_right:	T_spatialObject → T_boolean
	B_below:	T_spatialObject → T_boolean
	B_above:	T_spatialObject → T_boolean
	B_equal:	T_spatialObject → T_boolean
	B_inside:	T_spatialObject → T_boolean
	B_overlap:	T_spatialObject → T_boolean
	B_cover:	T_spatialObject → T_boolean
	B_touch:	T_spatialObject → T_boolean
	B_disjoint:	T_spatialObject → T_boolean
T_point	*B_xvalue*:	T_real
	B_yvalue:	T_real
	B_zvalue:	T_real

Table 5: Primitive Behavior Signatures of Events, Salient Objects, and Spatial Objects

certain time and the displacement of an object over time intervals are captured by *B_distance* and *B_displacement*, respectively.

Example 3 Let mary and dog be two timestamped salient objects. Their spatial relations at time t (or frame t) can be decided by first resulting mary and dog to a common time interval. That is, we assume t is a time interval t (whose starting time and ending time are t) and both t.*B_during*(mary.*B_timeStamp*) and t.*B_during*(dog.*B_timeStamp*) are true. Then, we compare the spatial intervals of mary and dog according to the definitions given in Table 2 to check what topological relations exist or what directional relations exist. These spatial intervals of mary can be extracted by mary.*B_value.B_xinterval* and mary.*B_value.B_yinterval*. Similarly we have dog.*B_value.B_xinterval* and dog.*B_value.B_yinterval* for the spatial intervals of dog. To measure the distance between mary and dog we have to access the objects' centroid which can be expressed as mary.*B_value.B_centroid.B_xvalue*, mary.*B_value.B_centroid.B_yvalue*, and mary.*B_value.B_centroid.B_zvalue*. Here, *B_xvalue*, *B_yvalue*, and *B_zvalue* are behaviors for getting x, y, and z values defined in `T_point`. It is trivial to compute the distance once two objects centroids are known.

5 Query Examples

In this subsection we present some examples to show the expressiveness of our model from the spatial properties point of view. We first introduce object calculus [Pet94]. The alphabet of the calculus consists of object constants (a, b, c, d), object variables (o, p, q, u, v, x, y, z), monadic predicates (C, P, Q), dyadic predicates $(=, \in, \notin)$, an n-ary predicate $(Eval)$, a function symbol (β) called *behavior specification* (Bspec), and logical connectives $(\exists, \forall, \wedge, \vee, \neg)$. The "evaluation" of a Bspec is accomplished by predicate $Eval$. A *term* is an object constant, an object variable or a Bspec. An *atomic formula* or *atom* has an equivalent Bspec representation. From atoms, *well-formed formulas* (WFFs) are built to construct the declarative calculus expressions of the language. WFFs are

defined recursively from atoms in the usual way using the connectives $\wedge, \vee, \neg$ and the quantifiers $\exists$ and $\forall$.

A query is an object calculus expression of the form $\{t_1, \ldots, t_n | \phi(o_1, \ldots, o_n)\}$ where $t_1, \ldots, t_n$ are the terms over the multiple variables $o_1, \ldots, o_n$. ϕ is a WFF. Indexed object variables are of the form $o[\beta]$ where β is a set of behaviors defined on the type variable o. The semantics of this construct is to project over the behaviors in β for o, meaning that after the operation only the behaviors given in β will be applicable to o.

We assume that all the queries are posted to a particular video instance myVideo and also salient objects and events are timestamped objects as discussed in Section 4. We also assume that all clips are timestamped clips and c $\in$ myVideo.*B_clips*.*B_history* where c is an arbitrary clip. myVideo.*B_clips* returns a history of all the clips in myVideo and myVideo.*B_clips*.*B_history* returns a collection of all the timestamped clips in myVideo. Since c is a timestamped clip, c belongs to the class **C_timeStampedObject** and the type of c is `T_timeStampedObject` < `T_clip` >. For simplicity, if a clip, salient object, or an event belongs to a timestamped object class **C_timeStampedObject**, we omit it in the query calculus expressions.

Query 1 What is the duration of clip c?

It is simply c.*B_timeStamp*.*B_length*. Similarly, the duration of salient object a (or an event e) is a.*B_timeStamp*.*B_length* (or e.*B_timeStamp*.*B_length*).

Query 2 Is the salient object a in clip c?

$\{q \mid q = a.B\_timeStamp.B\_during(c.B\_timeStamp)\}$.

The query checks whether the time interval of object a is a subinterval of clip c. Another way to express the same query is to use clips associated with a:

$\{o \mid o = a.B\_value.B\_inClips(\text{myVideo}).B\_history.B\_elementOf(c)\}$.

Here, a.*B_value*.*B_inClips*(myVideo) returns a history of all the clips containing a. Applying *B_history* to it returns the collection (set) of these clips. The behavior *B_elementOf*(c), defined in `T_collection`, checks whether c is an element of the collection.

For convenience, predicate $IN(\mathsf{o},\mathsf{c})$ is used to denote that object o is in clip c.

Query 3 Find all the clips in which *Mary* appears:

$$\{\mathsf{c} \mid \exists \mathsf{p}(\mathsf{p}.B\_value.B\_name = \text{`}Mary\text{'} \wedge \forall \mathsf{w}(\mathsf{w} \in \mathsf{p}.B\_value.B\_inClips(\mathsf{myVideo}).B\_history) \wedge \mathsf{c} = \mathsf{w}))\}$$

or

$$\{\mathsf{c} \mid \forall \mathsf{w}(\mathsf{p}.B\_value.B\_name = \text{`}Mary\text{'} \wedge IN(\mathsf{p},\mathsf{w}) \wedge \mathsf{c} = \mathsf{w})\}$$

where p is an instance of timestamped `T_person`.

Query 4 Find all the objects in a given area a at time t.

$$\{z \mid \exists \mathsf{c}(\mathbf{C\_interval}(t) \wedge \mathbf{C\_salientObject}(\mathsf{a}) \wedge \mathbf{C\_history}(\mathsf{x}) \wedge \mathbf{C\_collection}(\mathsf{y}) \wedge \mathsf{x} \in \mathsf{c}.B\_value.B\_salientObjects \wedge \mathsf{y} \in \mathsf{x}.B\_history \wedge \mathsf{t}.B\_during(\mathsf{y}.B\_timeStamp) \wedge \mathsf{z} = \mathsf{y}.B\_value \wedge \mathsf{z}.B\_inside(\mathsf{a}))\}$$

where c is an instance of timestamped clip, a is a spatial object, and t is a time interval. Suppose we can find a clip (c) in which some object (y) appears at time t ($\mathsf{t}.B\_during(\mathsf{y}.B\_timeStamp)$), then this object (y) is selected to check whether it is inside of area a. If an object which is partly within area a should also be included, we simply change the last predicate from $\mathsf{z}.B\_inside(\mathsf{a})$ into $(\mathsf{z}.B\_inside(\mathsf{a}) \vee \mathsf{z}.B\_overlap(\mathsf{a}))$.

Query 5 Find all the objects are very close to object a.

$$\{z \mid \exists \mathsf{y}(\mathbf{C\_history}(\mathsf{x}) \wedge \mathbf{C\_real}(\mathsf{h}) \wedge IN(\mathsf{a},\mathsf{c}) \wedge \forall \mathsf{x}(\mathsf{x} \in \mathsf{c}.B\_salientObjects \wedge \mathsf{y} \in \mathsf{x}.B\_history \wedge \mathsf{a}.B\_timeStamp.B\_during(\mathsf{y}.B\_timeStamp) \wedge \mathsf{y}.B\_value.B\_distance(\mathsf{a}.B\_value).B\_lessthan(h) \wedge \mathsf{z} = \mathsf{y}.B\_value))\}$$

where a is an instance of `T_timeStampedObject` $<$ `T_spatialObject` $>$ and h is a predefined threshold value for measuring *very close*. In this query formula we locate the clip c in which a appears and go through all the salient objects in c. If any object shows up at the time a shows up ($\mathsf{a}.B\_timeStamp.B\_during(\mathsf{y}.B\_timeStamp)$) then the distance between this object and a is computed and its value is compared with a predefined threshold h. It is either the objectbase designer or the end user to set the threshold value h.

Query 6 Find a clip in which object a is at left of object b and later they two exchange their positions.

$$\{\mathsf{c} \mid \exists \mathsf{x}\exists \mathsf{x}_2\exists \mathsf{x}_3\exists \mathsf{y}\exists \mathsf{y}_2\exists \mathsf{y}_3(\mathbf{C\_history}(x) \wedge \mathbf{C\_history}(y) \wedge \mathsf{x},\mathsf{y} \in \mathsf{c}.B\_value.B\_salientObjects \wedge$$
$$\mathsf{x}_2,\mathsf{x}_3 \in \mathsf{x}.B\_history \wedge \mathsf{y}_2,\mathsf{y}_3 \in \mathsf{y}.B\_history \wedge \mathsf{x}_2.B\_value = \mathsf{a} \wedge \mathsf{y}_2.B\_value = \mathsf{b} \wedge$$
$$\mathsf{x}_2.B\_timeStamp.B\_equal(\mathsf{y}_2.B\_timeStamp) \wedge \mathsf{x}_2.B\_value.B\_left(\mathsf{y}_2.B\_value) \wedge$$
$$\mathsf{x}_3.B\_value = \mathsf{a} \wedge \mathsf{y}_3.B\_value = \mathsf{b} \wedge \mathsf{x}_3.B\_timeStamp.B\_equal(\mathsf{y}_3.B\_timeStamp) \wedge$$
$$\mathsf{y}_3.B\_value.B\_left(\mathsf{x}_3.B\_value) \wedge \mathsf{x}_3.B\_timeStamp.B\_after(\mathsf{x}_2.B\_timeStamp))\}.$$

Suppose clip c is the one we are looking for. Then there must be two objects, denoted by x_2 and y_2 respectively, in c's salient object set so that x_2 is a and y_2 is b. Similarly, other two objects, denoted by x_3 and y_3 respectively, must be exist in c's salient object set so that x_3 is a and y_3 is b. The difference between x_2 and x_3 is only in their time stamps. Here we require that x_3 appears later than x_2 ($\mathsf{x}_3.B\_timeStamp.B\_after(\mathsf{x}_2.B\_timeStamp)$). Therefore, if x_2 is at the left of y_2 at time $\mathsf{x}_2.B\_timeStamp$ and y_3 is at the left of x_3 at time $\mathsf{x}_3.B\_timeStamp$, we are sure that a and b have exchanged their directional positions.

Query 7 Find a video clip in which a dog approaches Mary from the left.

$$\{\mathsf{c} \mid \exists \mathsf{x}\exists \mathsf{x}_2\exists \mathsf{x}_3\exists \mathsf{y}\exists \mathsf{y}_2\exists \mathsf{y}_3(\mathbf{C\_history}(\mathsf{x}) \wedge \mathbf{C\_history}(\mathsf{y}) \wedge \mathbf{C\_real}(\mathsf{h}_1) \wedge \mathbf{C\_real}(\mathsf{h}_2) \wedge$$
$$\mathsf{x},\mathsf{y} \in \mathsf{c}.B\_value.B\_salientObjects \wedge \mathsf{x}_2,\mathsf{x}_3 \in \mathsf{x}.B\_history \wedge \mathsf{y}_2,\mathsf{y}_3 \in \mathsf{y}.B\_history \wedge$$
$$\mathsf{x}_2.B\_value = \mathsf{dog} \wedge \mathsf{y}_2.B\_value = \mathsf{mary} \wedge \mathsf{x}_2.B\_timeStamp.B\_equal(\mathsf{y}_2.B\_timeStamp) \wedge$$
$$\mathsf{x}_2.B\_value.B\_left(\mathsf{y}_2.B\_value) \wedge \mathsf{x}_3.B\_value = \mathsf{a} \wedge \mathsf{y}_3.B\_value = \mathsf{b} \wedge$$
$$\mathsf{x}_3.B\_timeStamp.B\_equal(\mathsf{y}_3.B\_timeStamp) \wedge \mathsf{x}_3.B\_value.B\_left(\mathsf{y}_3.B\_value) \wedge$$
$$\mathsf{x}_3.B\_timeStamp.B\_after(\mathsf{x}_2.B\_timeStamp) \wedge$$
$$\mathsf{x}_2.B\_value.B\_displacement(\mathsf{x}_2.B\_timeStamp, \mathsf{x}_3.B\_timeStamp).B\_greaterThan(\mathsf{h}_1) \wedge$$
$$\mathsf{y}_2.B\_value.B\_displacement(\mathsf{x}_2.B\_timeStamp, \mathsf{x}_3.B\_timeStamp).B\_lessThan(\mathsf{h}_2))\}$$

where dog and mary are two instances of `T_salientObject`, Similar to the Query 6 we suppose clip c is what we are looking for and two salient objects, denoted by x_2 and x_3, are introduced to represent dog to reflect different time stamps. The same strategy is used for the object mary.

Then, we compute the dog's displacement over the time period and enforce this displacement to be greater than some predefined value h_1 to insure enough movement achieved. Furthermore, the displacement of mary is also computed and is required to be less than a predefined value h_2. This particular requirement to mary is to guarantee that it is the dog approaches Mary from the left, instead of that it is Mary approaches the dog from the right.

6 Conclusions

Spatial relationships play a very important role in the multimedia information systems. In this paper we explore the spatial properties of salient objects in a video objectbase. The major contribution of this work is that the proposed spatial model supports a comprehensive set of queries. Both the qualitative and quantitative spatial properties of objects are considered. In particular, we focus on the following issues:

- Support should be provided for object domains which consist of *complex* (structured) spatial objects in addition to simple (unstructured) points and alphanumeric domains. References to these spatial objects through their spatial domains must be directed by pointing to or describing the space they occupy and not by referencing their encodings.

- Support should exist for *direct spatial searches*, which locate the spatial objects in a given area of images.

- It should be possible to perform *hybrid spatial search*, which locates objects based on some attributes and some associations between attributes and the spatial objects.

- Support should exist for *complex spatial searches*, which locate spatial objects across the database by using set-theoretic operations over spatial attributes.

- Support should be provided to perform *direct spatial computations*, which compute specialized simple and aggregate functions from the images.

- Finally support should exist for *spatio-temporal queries* which involve not only spatial relations, but temporal relations as well.

We show that the integrated CVOT model supports the above requirements. The support for object spatial relationships is further strengthened by incorporating a rich set of spatial inference rules. A uniform approach to modeling video objects using histories is also discussed and the expressiveness of the CVOT model is demonstrated by means of example queries within the context of the TIGUKAT system.

There are two major directions for our future work on the spatial issues in the CVOT model. One is to extend the spatial model to capture the moving direction of an object and to combine it with the temporal model in order to perform video motion analysis [DG94]. We also intend to build a video query language based on the CVOT model. The spatial, temporal, and spatio-temporal queries can be translated into the query calculus and then the query algebra. Therefore, it is possible to optimize these queries using object query optimization techniques [MDZ93, ÖB95].

References

[AEG94] A. I. Abdelmoty and B. A. El-Geresy. An intersection-based formalism for representing orientation relations in a geographic database. In *Proceedings of the 2nd ACM Conference on Advances in GIS Theory*, Gaithersburg, MD, 1994.

[All83] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of ACM*, 26(11):832—843, 1983.

[Bee89] P. V. Beek. Approximation algorithms for temporal reasoning. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1291—1296, Detroit Michigan, August 1989.

[CIT+93] A. F. Cardenas, I. T. Ieong, R. K. Taira, R. Barker, and C. M. Breant. The knowledge-based object-oriented PICQUERY+ language. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):644—657, August 1993.

[CSE94] E. Clementini, J. Sharma, and M. J. Egenhofer. Modelling topological spatial relations: Strategies for query processing. *Computers and Graphics*, 18(6):815—822, 1994.

[DDI+95] Y. F. Day, S. Dagtas, M. Iino, A Khokhar, and A. Ghafoor. Object-oriented conceptual modeling of video data. In *Proceedings of the 11th International Conference on Data Engineering*, pages 401—408, Taipei, Taiwan, 1995.

[DG94] N. Dimitrova and F. Golshani. Rx for semantic video database retrieval. In *Proceedings of the 2nd ACM International Conference on Multimedia*, pages 219—226, San Francisco, CA, October 1994.

[Ege91] M. Egenhofer. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161—174, 1991.

[Ege94] M. Egenhofer. Spatial SQL: A query and presentation language. *IEEE Transactions on Knowledge and Data Engineering*, 6(1):86—95, January 1994.

[Gal91] D. L. Gall. MPEG: A video compression standard for multimedia applications. *Communications of ACM*, 34(4):46—58, 1991.

[GBT94] S. Gibbs, C. Breiteneder, and D. Tscichritzis. Data modeling of time-based media. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 91—102, Minneapolis, Minnesota, May 1994.

[Gha96] S. Ghandeharizadeh. Stream-based versus structured video objects: Issues, solutions, and challenges. In V. S. Subrahmanian and S. Jajodia, editors, *Multimedia Database Systems: Issues and Research Directions*, pages 215—236. Springer Verlag, 1996.

[GLÖS96] I. A. Goralwalla, Y. Leontiev, M. T. Özsu, and D. Szafron. Modeling time: Back to basics. Technical Report TR-96-03, Department of Computing Science, University of Alberta, February 1996.

[GPP95] M. Grigni, D. Papadias, and C. Papadimitriou. Topological inference. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 901–906, 1995.

[Her94] D. Hernández. *Qualitative Representation of Spatial Knowledge*. Springer-Verlag, New York, 1994.

[HJW95] A. Hampapur, R. Jain, and T. E. Weymouth. Production model based digital video segmentation. *Multimedia Tools and Applications*, 1(1):9—46, March 1995.

[HR95] S. Hibino and E. A. Rundensteiner. A visual query language for identifying temporal trends in video data. In *Proceedings of International Workshop on Multi-Media Database Management Systems*, pages 74—81, Blue Mountain Lake, New York, August 1995.

[LG91] T. C. C. Little and A. Ghafoor. Spatio-temporal composition of distributed multimedia objects for value added networks. *Computer*, 24(10):42—50, 1991.

[LG93] T. C. C. Little and A. Ghafoor. Interval-based conceptual models for time-dependent multimedia data. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):551—563, August 1993.

[LGÖS96] J. Z. Li, I. Goralwalla, M. T. Özsu, and D. Szafron. Video modeling and its integration in a temporal object model. Technical Report TR-96-02, Department of Computing Science, University of Alberta, January 1996.

[LÖS96] J. Z. Li, M. T. Özsu, and D. Szafron. Spatial reasoning rules in multimedia management systems. Technical Report TR-96-05, Department of Computing Science, University of Alberta, March 1996.

[Mas91] Y. Masunaga. Design issues of OMEGA: An object-oriented multimedia database management system. *Journal of Information Processing*, 14(1):60—74, 1991.

[MDZ93] G. Mitchell, U. Dayal, and S. B. Zdonik. Control of an extensible query optimizer: A planning-based approach. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 517—528, Dublin, Ireland, August 1993.

[NSN95] M. Nabil, J. Shepherd, and H. H. Ngu. 2D projection interval relationships: A symbolic representation of spatial relationships. In *Proceedings of 4th International Symposium on Large Spatial Databases*, pages 292—309, Portland, Maine, USA, August 1995.

[ÖB95] M.T. Özsu and J. Blakeley. Query optimization and processing in object-oriented database systems. In W. Kim, editor, *Modern Database Systems*, pages 146—174. Addison-Wesley, 1995.

[OM88] J. A. Orenstein and F. A. Manola. PROBE spatial data modeling and query processing in an image database application. *IEEE Transactions on Software Engineering*, 14(5):611—629, May 1988.

[ÖPS+95] M. T. Özsu, R. J. Peters, D. Szafron, B. Irani, A. Lipka, and A. Munoz. TIGUKAT: A uniform behavioral objectbase management system. *The VLDB Journal*, 4:100—147, 1995.

[OT93] E. Oomoto and K. Tanaka. OVID: Design and implementation of a video-object database system. *IEEE Transactions on knowledge and data engineering*, 5(4):629—643, August 1993.

[PE88] D. Pullar and M. Egenhofer. Toward formal definitions of topological relations among spatial objects. In *Proceedings of the 3rd International Symposium on Spatial Data Handling*, pages 165—176, Sydney, Australia, 1988.

[Pet94] R. Peters. TIGUKAT: A uniform behavioral objectbase management system. PhD thesis, Department of Computing Science, University of Alberta. Available as Technical Report TR-94-06, 1994.

[PS94] D. Papadias and T. Sellis. The qualitative representation of spatial knowledge in two-dimensional space. *The VLDB Journal (Special Issue on Spatial Databases)*, 4:100—138, 1994.

[PTSE95] D. Papadias, Y. Theodoridis, T. Sellis, and M. J. Egenhofer. Topological relations in the world of minimum bounding rectangles: A study with R-trees. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 92—103, San Jose, CA, May 1995.

[RFS88] N. Roussopoulos, C. Faloutsos, and T. Sellis. Spatial data models and query processing. *IEEE Transactions on Software Engineering*, 14(5):639—650, May 1988.

[SA95] H. Samet and W. G. Aref. Spatial data models and query processing. In W. Kim, editor, *Modern Database Systems*, pages 338—360. Addison-Wesley, 1995.

[SF95] J. Sharma and D. M. Flewelling. Inferences from combined knowledge about topology and directions. In *Proceedings of 4th International Symposium on Large Spatial Databases*, pages 279—291, Portland, Maine, USA, August 1995.

[SW94] G. A. Schloss and M. J. Wynblatt. Building temporal structures in a layered multimedia data model. In *Proceedings of ACM Multimedia'94*, pages 271—278, San Francisco, CA, 1994.

[SYH94] P. Sistla, C. Yu, and R. Haddack. Reasoning about spatial relationships in picture retrieval systems. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 570—581, 1994.

[WDG94] R. Weiss, A. Duda, and D. K. Gifford. Content-based access to algebraic video. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 140—151, 1994.