

# Spatio-temporal composition and indexing for large multimedia applications

Michael Vazirgiannis, Yannis Theodoridis, Timos Sellis

Computer Science Division, Department of Electrical and Computer Engineering, National Technical University of Athens, Zographou, GR-15773 Athens, Greece; e-mail: {mvazirg, theodor, timos}@cs.ntua.gr

**Abstract.** Multimedia applications usually involve a large number of multimedia objects (texts, images, sounds, etc.). An important issue in this context is the specification of spatial and temporal relationships among these objects. In this paper we define such a model, based on a set of spatial and temporal relationships between objects participating in multimedia applications. Our work exploits existing approaches for spatial and temporal relationships. We extend these relationships in order to cover the specific requirements of multimedia applications and we integrate the results in a uniform framework for spatio-temporal composition representation. Another issue is the efficient handling of queries related to the spatio-temporal relationships among the objects during the authoring process. Such queries may be very costly and appropriate indexing schemes are needed so as to handle them efficiently. We propose efficient such schemes, based on multidimensional (spatial) data structures, for large multimedia applications that involve thousands of objects. Evaluation models of the proposed schemes are also presented, as well as hints for the selection of the most appropriate one, according to the multimedia author's requirements.

## 1 Introduction

A multimedia application (MAP) involves a variety of individual multimedia objects presented according to the MAP scenario. The multimedia objects that participate in a MAP are transformed either spatially or temporally in order to be presented according to the author's requirements. Moreover, the author has to define the spatial and temporal ordering of objects within the application context and define the relationships among them. Finally, the way that users will interact with the application as well as the way that the application will treat application or system events have to be defined.

Real-world MAPs may be very large and complex with respect to the number of involved objects, transformations of the objects in the scope of an application, and relationships among them. We consider a MAP as a container that includes

objects that are transformed and interrelated in the MAP context. It is obvious that in a complex MAP it may be very difficult to describe all possible functionality and paths the user or the application may follow. Therefore, one can think of a MAP as an *event-based* environment, in which there is a rich set of events that may occur and define its flow. For instance, the end of a video clip, the spatial coincidence of two objects in the application window, or the occurrence of a pattern in a media object are events that may be exploited to trigger other actions in an application.

A crucial part MAPs' modeling is related to temporal and spatial composition of objects in the context of the application. The well-known sets of temporal [Hamb72, Alle83] and topological [Egen91] relationships are not adequate to represent all different semantics of multimedia objects composition, since they do not convey this kind of information. For instance, the topological relationship 'disjoint' between two spatial objects A and B (as in Fig. 1a) does not suffice to represent their relative position as well as their relative distance (i.e., B is on the right side of A at a distance of 8 cm). Another similar example is the successive presentation of two sounds (A1, A2) with a temporal gap of 8 s (see Fig. 1b).

It is desirable that the spatial and temporal specifications, defined by the author in a high-level GUI, are explicitly transformed to a uniform representation that retains the spatio-temporal relationships among the objects. Thus, the need for high-level declarative representation of multimedia object composition arises. Authoring complex MAPs (for instance, 3D synthetic movies [Krie96]) that involve a large number of objects (typically  $\geq 10^4$ ) may be a very complicated task, keeping in mind the large set of possible spatio-temporal relationships that may be encountered in the application context. Normally, in a 90-min synthetic movie, the number of participating objects is expected to be  $10^4$  or more with respect to the order of magnitude. Taking into account the vast number of possible events and their combinations based on (user and object) interaction, the number of the entities that have to be managed by the MAP authors is considerable.

A powerful authoring procedure should provide the tools for declarative high-level complete specification of the MAP.

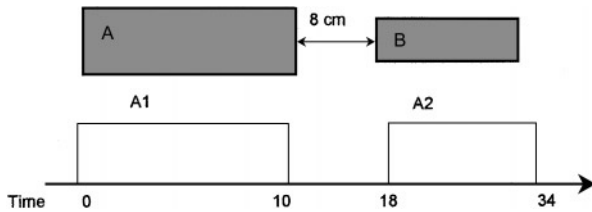


Fig. 1a,b. Simple media spatial and temporal relationships in the context of a MAP

Such tools are based on languages that reflect the underlying model primitives. Our objective is the definition of a model that could fulfill this need and also support the queries submitted by the authors for verification purposes during MAP development. Thus, during the development of a digital movie, the authors/directors would perhaps submit queries related to:

- spatial (screen) layout at a specific time instance during the movie,
- temporal layout of the movie in terms of temporal intervals,
- spatio-temporal relationships among objects (actors) (i.e., “does object A spatially overlap with object B?” or “which objects temporally overlap with object A?”)

In this paper, we define a model for the representation of spatio-temporal composition in MAPs and we also propose and evaluate indexing schemes to support queries related to the spatio-temporal content of a MAP.

The model is based on a set of spatial and temporal relationships between media objects. Our work exploits existing approaches for spatial [Papa97] relationships. We extend those relationships in order to cover the specific requirements of MAPs and we integrate the results in a uniform framework for spatio-temporal composition representation. Moreover, we define a set of operators for representation of temporal compositions. These operators are *declarative* and *complete* (i.e., able to represent any temporal scenario and convey semantics about the temporal composition).

We also propose indexing schemes (i.e., disk-resident structures organizing spatio-temporal features of media objects) for large MAPs in order to assist authors:

- manage the large number of objects in the MAP under development,
- acquire spatial and temporal layouts of the MAP under development, for verification purposes,
- submit queries regarding spatio-temporal relationships among objects.

The proposed indexing schemes are based on the R-tree index [Gutt84], which is widely used for indexing of spatial data in several applications, such as geographic information systems (GIS), CAD and VLSI design, etc. We adapt R-trees in order to index either spatial, temporal or spatio-temporal occurrences of objects and relationships between them. Moreover, we evaluate the proposed schemes against two simple indexing cases: the primitive one, based on serial storage of objects’ spatio-temporal coordinates, and a simple indexing scheme, which keeps disk-resident arrays of pre-sorted object coordinates according to each direction

(i.e., lower x- or y-coordinate and start point at the t-axis). We also provide hints to multimedia database designers, in order to select the most efficient scheme according to the requirements of MAP authors.

In the literature, there is no previous work, according to our knowledge, on indexing spatio-temporal characteristics of MAPs. Research has mainly focused on *content-based* image indexing, i.e., fast retrieval of objects using their content characteristics (color, texture, shape). In [Falo94a], a system, called *QBIC*, (coupling several features from machine vision with fast indexing methods from the database area) supports color-, shape- and texture-matching queries. Nearest neighbor queries (based on image content) are addressed in [Chiu94]. In general, indexing multimedia objects’ contents is an active research area, while indexing objects’ extents in the spatio-temporal coordinate system sets a new direction. In this paper, we do not consider some aspects of MAPs, such as interaction handling, network and distribution requirements and scenario rendering. These issues may be confronted during the implementation of such a system. Moreover, the proposed model refers mostly to specification and retrieval rather than to storage, and execution of a MAP.

The paper is organized as follows: In Sect. 2, we present background work on temporal and spatial relationships to be exploited. Furthermore, we discuss the requirements that a, specific to MAPs, spatio-temporal composition scheme should fulfill. In Sect. 3, we discuss and define the spatio-temporal relationships and a generic composition model for MAPs. In addition, we present a sample MAP and its representation according to the proposed model. In Sect. 4, we propose indexing schemes for MAPs to support queries involving the operators introduced in Sect. 3. In particular, we propose a simple one, based on sorted arrays and two more sophisticated ones, based on the R-tree spatial index structure, in order to support these operators. In Sect. 5, we evaluate analytically the proposed schemes for some reasonable spatio-temporal MAP configuration. We conclude in Sect. 6, by summarizing our work and giving hints for future research.

## 2 Related work

In the past, the term *synchronization* has been widely used to describe the temporal ordering of objects in a MAP [Litt93]. A MAP specification should describe both temporal and spatial ordering of objects in the context of the application. The spatial ordering (i.e., absolute positioning and spatial relationships among objects) issues have not been adequately addressed. We claim that the term “*synchronization*” is poor for MAPs. Instead we propose the term “*composition*” to represent both temporal and spatial ordering of objects. Hereafter, we review existing systems and approaches for temporal and spatial composition.

### 2.1 Temporal composition

Many existing models for temporal composition of multimedia objects in the framework of a MAP are based on 13 relations defined in [Hamb72, Alle83]: *before*, *meets*, *during*,

*overlaps*, *starts*, *ends*, *equal* and the inverse ones (does not apply to *equal*). These relations are not adequate for temporal composition description. They are descriptive, hence, they do not reflect causal dependencies between intervals. They depend on interval durations and may lead to temporal inconsistency. More specifically, the problems that may arise when trying to use these relations are the following [Duda95].

- The relations are designed to express relationships between intervals of fixed duration. In the case of MAPs, it is required that a relationship holds independently from the duration of the related object (i.e., the relationship should not change when the duration changes).
- Their descriptive character does not convey the cause and the result in a relationship.

Other models for temporal composition representation may be classified in two categories [Duda95]: *point-based* and *interval-based*. In point-based models, the elementary units are points in time and space. Each event has an associated time point. The time points arranged according to some relations (such as “precede”, “simultaneous” or “after”) form complex multimedia presentations. An example of the point-based approach is the *timeline*. Interval-based models consider elementary media entities as temporal intervals, ordered according to some relations. Existing models are mainly based on the relations defined in [Hamb72, Alle83] for expressing knowledge about time.

An interesting mechanism for temporal composition is presented in [Duda95]. This work presents a model that takes into account the semantics of temporal relationships between objects. The resulting set of operators represent the causal relations between intervals. In [Hirz95], a temporal model for interactive scenarios is presented. This model is based on the timeline approach and provides the primitives for specification of synchronous and asynchronous interactive multimedia temporal compositions. The timeline approach is extended to a tree of timelines. Each branch of timelines represents the different scenarios that may be selected by the user.

Other approaches use Allen’s relations [Alle83] to specify a multimedia database schema. Little and Ghafoor [Litt93] propose an OCPN (object composition Petri nets) model equivalent to Allen’s relations. This approach does not take into account the possible unknown durations of intervals. Thus, in order to prepare an instantiated presentation, the tree of interval relations must be traversed to obtain deadlines to be used in the presentation schedule. There are also other approaches based on interval temporal logic [King94]. Although such formalisms have a solid mathematical background, the specification of multimedia presentations is awkward, since the specification does not correspond explicitly to the author’s perception of the multimedia composition.

In [Hand96], a synchronization model is presented. This model covers many aspects of multimedia synchronization, such as: incomplete timing, hierarchical synchronization, complex graph type of presentation structure with optional paths, presentation time prediction and event-based synchronization. The events are considered as presentations constrained by unpredictable temporal intervals. There is neither the notion of event semantics nor the notion of a composition

scheme. In [Schn96], a presentation synchronization model is presented. Important concepts introduced and manipulated by the model are the object states (“Idle”, “Ready”, “In-process”, “finished”, “complete”). Although events are not explicitly presented, user interactions are treated. There are two categories of interaction envisaged: buttons and user skips (“forward”, “backward”).

As referred to in [Blak96], event-based representation of a multimedia scenario is one of the four categories for modeling a multimedia presentation. There, it is mentioned that events are modeled in HyTime [Newc91, Bufo96] and HyperODA. Events in HyTime are defined as presentations of media objects along with their presentation specifications and FCS coordinates. According to [Erf93], HyTime modeling primitives are sufficient for temporal composition representation. HyperODA events are instantaneous happenings mainly corresponding to the start and end of media objects or timers. All these approaches suffer from poor semantics conveyed by the events, and moreover they do not provide any scheme for composition and consumption architectures.

## 2.2 Spatial specification and composition

The issue of spatial composition modeling is rather under-addressed in the current multimedia authoring environments and synchronization models. One of the few efforts that integrate both spatial and temporal aspects is [Iino94], a model for spatio-temporal multimedia presentations. The temporal composition is handled in terms of Allen’s relationships, whereas spatial aspects are treated in terms of a set of operators for binary and unary operations. The model lacks the following features: there is no indication of the temporal causal relationships (i.e., what are the semantics of the temporal relationships between the intervals corresponding to multimedia objects). The spatial synchronization essentially addresses only two topological relationships: *overlap* and *meet*, giving no representation means for the directional relationships between the objects (i.e., object A is to the right of object B) and the distance information (i.e., object A is 10 cm away from object B). The modeling formalism in this approach is oriented more towards execution and rendering of the application rather than to authoring.

Moreover, spatial specification is addressed by document preparation and modeling systems like Latex, SGML and HyTime. In Latex [Lamp90], spatial specification is feasible including the direction and the distance (i.e., an image to be placed 2 cm west to another of 10 cm north to the text). The specifications are expressed in terms of “float parameters”. HyTime [Newc91, Bufo96] provides a rich specification spatio-temporal scheme with the FCS (finite coordinate space) space. FCS provides an abstract coordinate scheme of arbitrary dimensions and semantics, where the author may locate the objects of a MAP and define their relationships. Nevertheless, the formalism is awkward and, as practice has proved, it is rarely used for real-world applications. Hytime is not an efficient solution for MAP development, since “there are significant representational limitations with regard to interactive behavior, support for scripting language integration, and presentation aspects” [Bufo96].

### 3 Spatio-temporal composition model

As mentioned in the previous section, there is a lack of an integrated approach for representation of all functional aspects of multimedia presentations. Such an application involves

- transformation of objects, in order to be aligned with the presentation specifications,
- specification of the composition of objects in space and time (spatial and temporal ordering through the definition of relationships among the media objects),
- definition of the application functionality (i.e., the application scenario) which is of two kinds: *pre-orchestrated* and *event-based*. The term *pre-orchestrated* implies that certain actions will take place at specific time instants, while *event-based* implies that actions are triggered by events that occur in the application context either due to the user or the system or due to entities participating in the application (media objects, media compositions, etc.). The fundamental entities of the scenario are called *scenario tuples* [Vazi96b] and are triggered by the occurrence of an event (simple or complex).

#### 3.1 Temporal relationships

The topic of relations between temporal intervals was originally addressed by Hamblin [Hamb72] and, later, by Allen [Alle83], as we have already mentioned in Subject. 2.1. In this section, we define a set of concepts to be exploited for the representation of temporal composition in the context of a MAP. We consider the presentation of a multimedia object as a temporal interval (hereafter multimedia instance). We exploit the start- and end-points of a multimedia instance as events and distinguish the end of a multimedia instance in *natural* (i.e., when the media object finishes its presentation) and *forced* (i.e., when an event explicitly stops the presentation of a media object). Furthermore, we are interested in the well-known *pause* (temporary stop of presentation) and *resume* actions (start the presentation from the point where the pause operation took place).


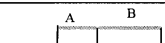
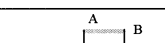
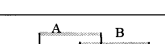

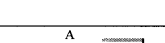
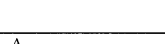
An important concept is the *temporal instance*: we consider it as an arbitrary temporal measurement, relative to some reference point (i.e., the application temporal starting point in our case, hereafter  $\hat{E}$ ). Based on the above descriptions, we define the following operators attached to the corresponding events.

**Definition 1.** Let  $A$  be a multimedia instance;  $A>$  represents the start of the multimedia instance,  $A<$  the natural end of the instance,  $A!$  the forced stop,  $A||$  the pause and  $A|>$  the resume actions.

**Definition 2.** Let  $A, B$  be two multimedia instances, then the expression  $Aop1 t Bop2$  represents all temporal relationships between the two multimedia instances, where  $op1 \in \{>, <, ||, |>\}$  and  $op2 \in \{>, !, ||, |>\}$  and  $t$  is a vacant temporal interval.

**Definition 3.** Let  $A$  be a multimedia instance; we define as  $t_{Aop}$  temporal instances corresponding to the events  $Aop$ , where  $Aop \in \{>, <, !, ||, |>\}$ .

**Table 1.** Temporal relationships and the corresponding operator expressions

Temporal relationship	Equivalent operator expression	Constraints
A before B 	$A < t B >$	
A meets B 	$A < 0 B >$	
A during B 	$A > t B >$	$t + d_A < d_B$
A overlaps B 	$A > t B >$	$t < d_A$
A starts B 	$A > 0 B >$	
A ends B 	$A < 0 B !$	
A equal B 	$A > 0 B >$	$d_A = d_B$

**Definition 4.** Let  $A$  be a multimedia instance; we define as  $d_A$  the temporal duration of the multimedia instance  $A$ .

The above operators are complete in the sense that all temporal relationships can be expressed using these operators as long as the appropriate conditions are fulfilled (Table 1). In addition, the proposed operators capture the semantics of the temporal relationships among the multimedia instances (i.e.,  $A$  meets  $B$  may be expressed as  $A < 0 B >$  or  $B > 0 A !$ ). Moreover, the proposed set of operators may be used for a high-level mechanism of temporal scenario specification.

#### 3.2 Spatial relationships

Another aspect of composition concerns the spatial ordering and topological features of the participating objects. Spatial composition aims at representing three aspects:

- the topological relationships between the objects (*dis-joint, meet, overlap, etc.*),
- the directional relationships between the objects (*left, right, above, above-left, etc.*)
- the distance/metric relationships between the objects (*outside 5 cm, inside 2 cm, etc.*).

As for the first aspect, a complete set of topological relationships between two objects, called *4-intersection model*,

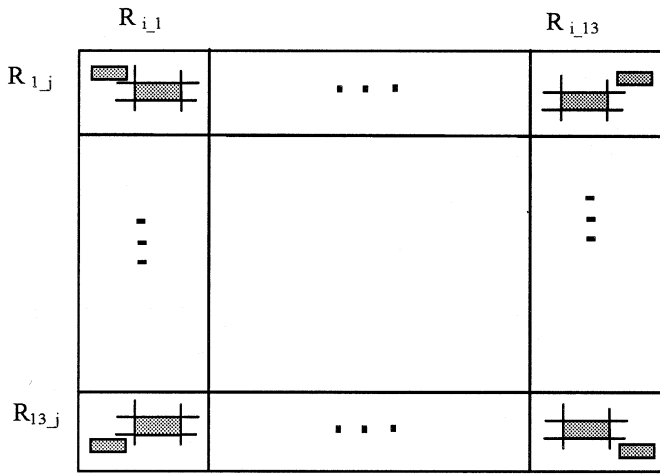


Fig. 2. Directional relationships between two spatial objects (including topological information)

was proposed in [Egen91]. Thus, two objects  $p$ ,  $q$  may coincide (*equal*), intersect (*overlap*), touch externally (*meet*), touch internally (*covers* and the reverse *covered\_by*), be *inside* (and the reverse *contains*), or be *disjoint*.

Concerning directional relationships, there is a complete set of relationships defined in [Papa97] (see Fig. 2). This set of 169 ( $13^2$ ) relationships  $R_{i_j}$  ( $i = 1, \dots, 13$  and  $j = 1, \dots, 13$ ) arises from exhaustive combination of the 13 relations defined in [Hamb72, Alle83] regarding relationships between temporal intervals. This set also covers topological relationships, since any topological relationship of the 4-intersection model could be expressed as a subset of the set of 169 relationships [Papa95].

In the context of a MAP, an author would like to place spatial objects (text windows, images, video clips, animation) in the application window in such a way that their relationships are clearly defined in a declarative way, i.e., “text window  $A$  is placed at the location  $(100, 100)$ , text window  $B$  appears 8 cm to the right and 12 cm below the upper side of  $A$ ” (see Fig. 3). This declarative definition should be transformed into an internal representation that captures the topological and directional relationships, as well as the distance between the objects in a uniform and correct way. In the next subsection, we propose a definition model to support these needs.

### 3.3 The model definition

Current MAP modeling schemes do not provide powerful tools for the complete description of the spatial and temporal composition that takes place in a complex application (an overview of related work was presented in Sect. 2).

We define a set of operators for representing temporal and spatial composition. Here, we have to make the distinction between pre-orchestrated and interactive applications. The term “*pre-orchestrated*” implies that certain actions will take place at specific time and/or spatial instants (i.e., temporal location relative to the applications start or spatial location in the application window), while “*event-based*” implies that actions are triggered by events that occur in the application context either by the user or the system or by entities

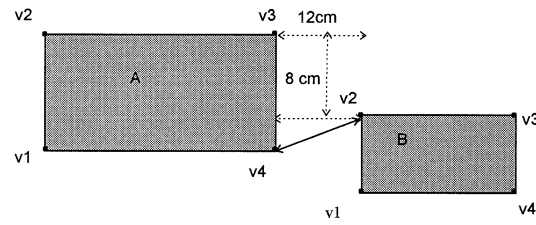


Fig. 3. Spatial composition generalized modeling

participating in the application (media objects, media compositions, etc.)

The resulting requirement is for a set of operators that allows users to represent any spatio-temporal relationship between objects in the context of a MAP in a declarative way. As for temporal composition of objects, we exploit the operators defined above. As far as it concerns spatial composition, we are based on the complete set of topological-directional relationships illustrated in Fig. 2, and propose the following generalized methodology for representing the distance between two spatial objects<sup>1</sup>. In order to achieve a uniform approach, we impose the constraint that the distance will be expressed in terms of distance between the ‘closest vertices’. For each spatial object  $O$ , we label its vertices as  $O.v_i$  ( $i = 1, 2, 3, 4$ ), starting from the bottom left vertex in a clockwise manner. As “closest”, we define the pair of vertices  $(A.v_i, B.v_j)$  with the minimum Euclidean distance.

The author of a MAP must be able to express spatial composition predicates in an unlimited manner. For instance (see Fig. 3), the author could describe the appearing composition as: “*object B to appear 12 cm lower than the upper side of object A and 8 cm to the right*”. The model we propose will translate such descriptions into minimal and uniform expressions, as imposed by the requirements for correct and complete representations.

For uniformity reasons, we define an object named  $\Theta$ , that corresponds to the spatial and temporal start of the application (i.e., the upper left corner of the application window and the temporal start of the application). Another assumption we make is that the objects that appear in the composition include their spatio-temporal presentation characteristics (i.e., size, duration, etc.) [Vazi95]. In the rest of this section, we exploit the EBNF formalism to represent the model primitives.

**Definition 1.** Assuming two spatial objects  $A$ ,  $B$ , we define the generalized spatial relationship between these objects as:  $S.R = (r_{ij}, v_i, v_j, x, y)$ , where  $r_{ij}$  is the identifier of the topological-directional relationship between  $A$  and  $B$  (derived from Fig. 2),  $v_i, v_j$  are the closest vertices of  $A$  and  $B$ , respectively, and  $x, y$  are the horizontal and vertical distances between  $v_i, v_j$ .

Next, we define a generalized operator expression to cover the spatial and temporal relationships between objects in the context of a MAP. It is important to stress the fact that, in some cases, we do not need to model a relationship between two objects but have to declare the spatial and/or

<sup>1</sup> We assume that spatial objects are rectangles. More complex objects can also be represented as rectangles by using their minimum bounding rectangle (MBR) approximation.

temporal position of an object relative to the application spatial and temporal start point  $\Theta$  (i.e., object A to appear at the spatial coordinates (110, 200) on the 10th second of the application).

**Definition 2.** We define a composite spatio-temporal operator that represents absolute spatial/temporal coordinates or spatio-temporal relationships between objects in the application:  $ST\_R(sp\_rel, temp\_rel)$ , where  $sp\_rel$  is a spatial relationship ( $S\_R$ ), while  $temp\_rel$  is a temporal relationship, as defined in Subsect. 3.1.

The spatio-temporal composition of a MAP consists of several independent fundamental compositions. The term ‘independent’ implies that objects participating in them are not related implicitly (either spatially or temporally), except for their implicit relationship to the start point  $\Theta$ . Thus, all compositions are explicitly related to  $\Theta$ . We call these compositions *composition\_tuples*, and these include spatially and/or temporally related objects.

**Definition 3.** We define the composition tuple in the context of a MAP as:  $composition\_tuple = A_i[\{ST\_RA_j\}]$ , where  $A_i, A_j$  are objects participating in the application,  $ST\_R$  is a spatio-temporal relationship (as defined in Definition 2).

**Definition 4.** We define the composition of multimedia objects in the context of MAPs as a set of composition\_tuples:  $composition = C_i\{C_j\}$ , where  $C_i, C_j$  are composition\_tuples.

The EBNF definition of the spatio-temporal composition based on the above definition follows:

```

composition ::=
    composition_tuple {[,composition_tuple]}
composition_tuple ::=
     $\Theta$  {[spatio_temporal_relationship action]}
action ::=
    object {[spatio_temporal_relationship object]}
    — (“ object spatio_temporal_relationship object “)
    — object
    — spatio_temporal_instance
spatio_temporal_relationship ::=
    “[([spatial_operator — spatial_instance“),
    (“temporal_operator — temporal_instance“)]”
temporal_operator ::=  $\Theta$  — t_event t_interval TAC_operation
t_event ::= “>” — “<” — “!” — “—>” — “——”
TAC_operation ::= “>” — “!” — “—>” — “——”
spatio_temporal_instance ::=
     $\Theta$ 
    — (spatial_instance, temporal_instance)
spatial_instance ::=
    (“x “,” y “”)
temporal_instance ::=
    TIME — event
spatial_operator ::=
    ( $r_{ij}, v_i, v_j, x, y$ )
x ::=
    INTEGER
y ::=
    INTEGER
 $\Theta$  ::=
    application start: (0, 0, 0)

```

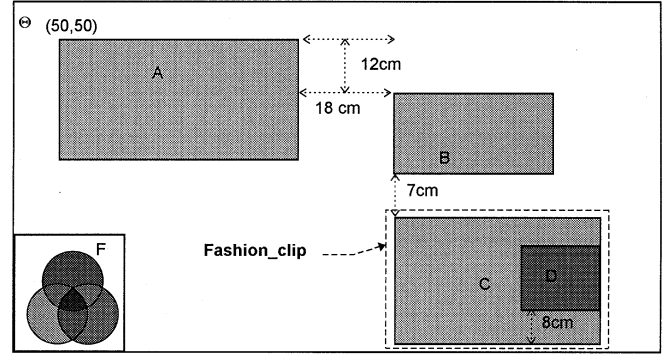


Fig. 4. Spatial composition of the ‘News\_clip’ MAP

where  $r_{ij}$  denotes a topological-directional relationship (from Fig. 2) between two objects and  $v_i, v_j$  denote the closest vertices of the two objects (see definition above).

In this section, we proposed a model for representing spatio-temporal composition in pre-orchestrated MAPs. In the next subsection, we illustrate the potential of the model by means of a sample application.

### 3.4 A sample multimedia composition

In this section, we describe a composite MAP corresponding to a TV news clip in terms of spatio-temporal relationships as defined above. The high-level scenario of the application is the following.

“The *News\_clip* starts with presentation of image A (located at point 50, 50 relative to the application origin  $\Theta$ ). At the same time, a background music E starts. Ten seconds later a video clip B starts. It appears to the right side (18 cm) and below the upper side of A (12 cm). Just after the end of B, another MAP starts. This MAP (called *Fashion\_clip*) is related to fashion. The *Fashion\_clip* consists of a video clip C that presents the highlights of a fashion show and appears 7 cm below (and left-aligned to) the position of B. Three seconds after the start of C, a text logo D (e.g., the designer’s logo) appears inside C, 8 cm above the bottom side of C, aligned to the right side. D will remain for 4 s on the screen. Meanwhile, at the 10th second of the News clip, the TV channel logo (F) appears at the bottom-left corner of the application window. F disappears after 3 s. The application ends when music background E ends.”

The spatial composition (screen layout) of the above scenario is illustrated in Fig. 4, while the temporal one is illustrated in Fig. 5.

The objects to be included in a composition tuple of a MAP are those that are spatially and/or temporally related. In our example (News\_clip), A and B and Fashion\_clip should be in the same composition tuple, since A relates to B and B relates to Fashion\_clip. On the other hand, F is not related to any other object, neither spatially nor temporally, so it composes a different tuple. The above spatial and temporal specifications defined by the author in a high-level GUI are transformed into the following representation according to the model primitives defined in Subsect. 3.3.

// News Clip

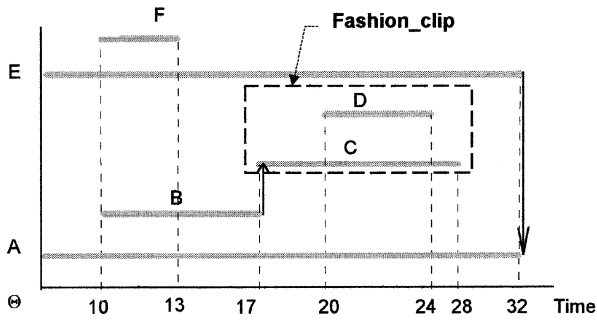


Fig. 5. Temporal composition of the 'News\_clip' MAP

```

composition = {r1, r2}
r1 = Θ [(→, →, →, →), (>0>)]
     E [(→, →, →, →), (<0!)]
     News
r2 = Θ [(r1_1, →, v2, 50, 50), (>0>)]
     A [(r11_13, v3, v2, 18, 12), (>10>)]
     B [(r13_6, v1, v2, 0, -7), (>0>)]
     Fashion_clip
r3 = Θ [(→, →, v1, 0, 300), (>10>)]
     F
// Fashion clip
composition = {r4}
r4 = Θ [(→, →, v2, 0, 0), (>0>)]
     C [(r9_10, v4, v4, 0, 8), (>3>)]
     D

```

It is important to stress that  $\Theta$  in composition tuple  $r_4$  represents the spatio-temporal origin of the Fashion clip. In this example, we have a composition of MAPs. It has to be stressed that, when the host MAP (i.e., News\_clip) ends, all the MAPs started by it are also stopped (i.e., Fashion\_clip). There is an issue regarding the mapping of the spatio-temporal specifications into the composition tuples: the classification of involved objects. The proposed procedure is the following. For each object  $A_i$ , we check whether it is related to objects already classified in an existing tuple. If the answer is positive,  $A_i$  is classified in the appropriate composition tuple (a procedure that possibly leads to reorganization of the tuples). Otherwise, a new composition tuple, composed by  $\Theta$  and  $A_i$ , is created.

The composition model should satisfy the following criteria:

- *completeness*: i.e., the available operators of the model suffice for representing any spatio-temporal relationship between objects, and
- *correctness*: i.e., each specification of a spatial composition  $s_i$  leads to a different representation expression  $r_i$ .

We claim that the proposed model is *complete* and *correct*. In particular, it is complete because the set of operators that was exploited may represent all spatio-temporal relationships among objects in a MAP. However, we do not provide formal proof in this paper. This is an issue of our current research.

The objects to be included in a composition tuple are those that are spatially and/or temporally related to each

other. During the application development process, it is probable (especially in the case of complex and large applications) that authors would need information related to the spatio-temporal features of the MAP (TV clip in the case of the example). The related queries, depending on the spatio-temporal relationships that are involved, may be classified in the following categories.

- pure spatial or temporal query: only a temporal or a spatial relationship is involved in the query. For instance, “*which objects temporally overlap the presentation of test logo D?*”, “*which objects spatially lie above object D in the application window?*”,
- spatio-temporal query: where such a relationship is involved. For instance, “*which objects spatially overlap with object D during its presentation?*”.
- layout query: spatial or temporal layouts of the application. For instance, “*what is the screen layout at the 22nd second of the application?*”, “*which objects are presented between the 10th and the 20th second of the application?*” (temporal layout).

A simple serial storage scheme which includes the objects’ spatial and temporal coordinates is an inefficient solution, since typical MAPs include thousands of objects. Hence, indexing techniques that would efficiently handle spatial and temporal characteristics of objects need to be adopted. In the next section, we propose such efficient indexing mechanisms, in order to support such queries in the context of MAP authoring. Here, it should be stressed that, in this research work, we do not deal with queries related to the content of the objects (*content-based queries*), but only to their spatio-temporal extents and relationships.

#### 4 Indexing techniques for large MAPs

As discussed in previous sections, MAPs usually involve a large number of media objects, such as images, video, sound and text. The quick retrieval of a qualifying set, among the huge amount of data, that satisfies a query based on spatio-temporal relationships is necessary for the efficient construction of a MAP. The multimedia scenario represented in terms of composition tuples essentially represents the spatio-temporal relationships among the multimedia objects according to the authors requirements. From these relationships the absolute objects’ spatio-temporal coordinates may be determined. Spatial and temporal features of objects are then identified by six coordinates: the projections on x- (points  $x_1, x_2$ ), y- (points  $y_1, y_2$ ), and t- (points  $t_1, t_2$ ) axes<sup>2</sup>. A serial storage scheme, maintaining the objects characteristics as a set of seven values (id,  $x_1, x_2, y_1, y_2, t_1, t_2$ ) and organizing them into disk pages, is not an efficient solution, since lack of ordering leads to the access of all pages for answering any query, like the example queries of Sect. 3. However, this scheme will be used as the baseline for the evaluation of our proposals in Sect. 5.

A more efficient, but still simplistic, solution (as will be presented next) is based on the maintenance of three disk

<sup>2</sup> We adopt a unified 3D workspace for space (two dimensions) and time (one dimension) features.

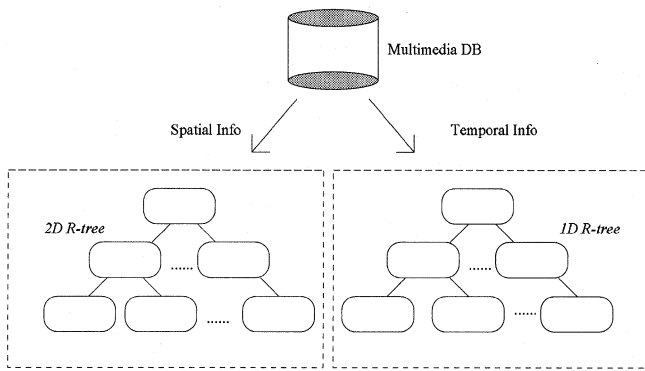


Fig. 6. A simple (spatial and temporal) indexing scheme

arrays that keep low coordinates of objects (i.e.,  $x_1$ ,  $y_1$ , and  $t_1$ ) separately in a sorted order<sup>3</sup>. Several queries involving spatio-temporal operators, among the ones presented at the end of Sect. 3, require the retrieval of one array only, using “divide-and-conquer” techniques. Temporal layout queries (such as query 5) belong to this group. However, the majority of queries involves information about more than one axis. Hence, the retrieval of more than one array and the subsequent combination of the answer sets is necessary for such cases. As a conclusion, efficient indexing mechanisms that could combine spatio-temporal characteristics of objects in order to efficiently support a wide range of spatio-temporal operators need to be present in a MAP authoring tool. In the next subsections, we propose two indexing schemes and their retrieval procedures.

#### 4.1 Indexing schemes

##### 4.1.1 A simple spatial and temporal indexing scheme

A simple indexing scheme that could be able to handle spatial and temporal characteristics of media objects consists of two indices:

- a *spatial (2D) index* for spatial characteristics (id, and  $x_1$ ,  $x_2$ ,  $y_1$ ,  $y_2$  values) of the objects, and
- a *temporal index* for temporal characteristics (id, and  $t_1$ ,  $t_2$  values) of the objects.

In the literature concerning the area of *spatial databases*, several data structures have been proposed for the manipulation of spatial data (a survey can be found in [Same90]). Among others, R-trees [Gutt84, Beck90] seem to be the most efficient ones. On the other hand, the manipulation of temporal information can be supported either by one-dimensional versions of the above data structures (since all of them have been designed for  $n$ -dimensional space in general) or by specialized temporal data structures (e.g., segment trees [Bent75]). For uniformity reasons, we select a single multi-dimensional data structure (R-tree) to play the role of the spatial (2D R-tree) and temporal (1D R-tree) index. The above indexing scheme is illustrated in Fig. 6.

<sup>3</sup> Instead of using low- coordinates one can select high- coordinates (or six arrays with low- and high- coordinates). It is a decision that affects neither the discussion that will follow nor its conclusions.

The R-tree is a height-balanced tree which consists of intermediate and leaf nodes. Objects’ approximations (commonly MBRs) are assumed to be stored in the leaf nodes of the tree. Intermediate nodes are built by grouping rectangles (or hyper-rectangles, in general) at the lower level. An intermediate node is associated with some rectangle which encloses all rectangles that correspond to lower level nodes. Formally,

- a *leaf node* is of the form  $(oid, RECT)$ , where *oid* is an object identifier and is used to refer to an object in the database and *RECT* is the MBR approximation of the data object, i.e., it is of the form  $(p_{l-1}, p_{l-2}, \dots, p_{l-n}, p_{u-1}, p_{u-2}, \dots, p_{u-n})$  which represents the  $2n$  coordinates of the lower left ( $p_l$ ) and the upper right ( $p_u$ ) corner of an  $n$ -dimensional (hyper-) rectangle  $p$ , and
- an *intermediate node* is of the form  $(ptr, RECT)$ , where *ptr* is a pointer to a lower level node of the tree and *RECT* is a representation of the rectangle that encloses spatially the children nodes.

Currently, the R-tree index is integrated in commercial DBMSs such as ILLUSTRATE [Ubel94]. In the case of other traditional database systems (like ORACLE, SYBASE, etc.), the R-tree code cannot be integrated, thus an interface layer should be implemented so that the R-tree implementation could communicate with the database. Concerning performance, in principle, the architecture does not affect the performance, since R-tree is a disk-resident structure and the only factor that is taken into account for estimating performance is the number of disk accesses.

We claim that the adoption of the above indexing scheme improves the retrieval of spatio-temporal operators compared to the “sorted arrays” scheme. Even for complex operators, where both tree indices need to be accessed (e.g., for the *overlap\_during* operator), the cost of the two indices’ response times are expected to be lower than the retrieval cost of the (three) arrays.

A weak point of the above scheme has been already mentioned. The retrieval of objects according to their spatio-temporal relationships (e.g., the *overlap\_during* one) with others demands access to both indices and, in a second phase, the computation of the intersection set between the two answer sets. Access to both indices is usually costly and, in many cases, most of the elements of the two answer sets are not found in the intersection set. In other words, most of the disk accesses to each index separately are useless. An efficient solution to that problem is the merging of the two indices (the spatial and the temporal one) in a unified mechanism. This scheme is proposed in the next subsection.

##### 4.1.2 A unified spatio-temporal indexing scheme

The proposed unified indexing scheme consists of only one index: a *spatial (3D) index* for the complete spatio-temporal information (location in space and time coordinates) of the objects. If we assume that the R-tree is an efficient spatial indexing mechanism, then the unified scheme is illustrated in Fig. 7.

The main advantages of the proposed scheme, when compared to the previous one, are the following:



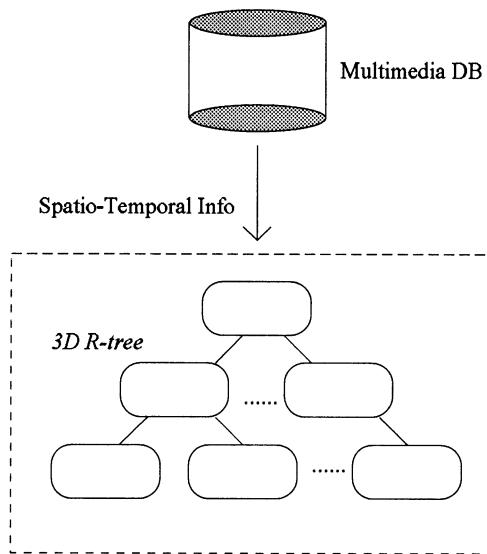


Fig. 7. A unified (spatio-temporal) indexing scheme

- the indexing mechanism is based on a unified framework. Only one spatial data structure (e.g., the R-tree) needs to be implemented and maintained.
- Spatio-temporal operators are more efficiently supported. Using the appropriate definitions, spatio-temporal operators are implemented as 3D queries and retrieved using the 3D index. So the need for the intersection procedure is eliminated.

The evaluation of the two proposed indexing schemes against each other, against the “sorted-arrays” and the serial storage ones, will be described in Sect. 5, where analytical models that predict the performance of each scheme will be presented. In the rest of this section, we will describe the retrieval process of such operators when the unified indexing scheme is available within a MAP authoring tool.

#### 4.2 Retrieval of spatio-temporal operators using R-trees

The majority of multidimensional data structures, such as the R-tree family, have been designed as extensions of the classic alphanumeric index, the B-tree. They usually divide the plane into appropriate sub-regions and store these sub-regions in hierarchical tree structures. Objects are represented in the tree structure by an approximation (the MBR approximation being the most common one) instead of their actual scheme, for simplicity and efficiency reasons.

Unfortunately, the relative position of two MBRs does not convey full information about the spatial (topological, direction, distance) relationship between the actual objects. For this reason, spatial queries involve the following two-step strategy [Oren86].

- *Filter step.* The tree structure is used to rapidly eliminate objects that could not possibly satisfy the query. The result of this step is a set of candidates which includes all the results and possibly some false hits.
- *Refinement step.* Each candidate is examined (by using computational geometry techniques). False hits are detected and eliminated.

In order to retrieve objects that belong to the answer set of a spatio-temporal operator, with respect to a reference object  $q$ , we have to specify the MBRs that could enclose such objects and then search the R-tree nodes that could contain such MBRs. This technique was proposed and implemented in [Papa97], in order to support spatial operators of high resolution (e.g., *meet*, *contains*) that are popular in GIS applications.

As an example, Fig. 8 shows how the MBRs corresponding to the representations of the objects are grouped and stored in the 3D R-tree of our unified scheme. We assume a branching factor of 4, i.e., each node contains at most four entries. At the lower level, MBRs of objects are grouped into two nodes  $R_1$  and  $R_2$ , which, in turn, compose the root of the index. Assume a spatio-temporal query involving the *overlap\_during* operator, with  $D$  being the reference object  $q$ . In order to answer this query, only  $R_2$  is selected for propagation. Among the entries of  $R_2$ , objects  $C$  and (obviously)  $D$  are the ones that constitute the qualified answer set. Note that only the right sub-tree of the R-tree index of Fig. 8a was propagated in order to answer the query. The rate of the accessed nodes heavily depends on the size of the reference object  $q$  and, of course, on the kind of the operator (more selective operators result in smaller number of accessed nodes).

Consider now a spatial query involving the *overlap* operator, with  $D$  being the reference object  $q$ . Since the query gives no temporal information on the reference object, the unified scheme transforms it to a large cube that covers the whole t-axis. In this case, the simple scheme, could be more efficient, since the 2D R-tree which is dedicated to spatial information of objects is able to answer the query. Similarly, a query involving the *during* operator could also be efficiently supported by the simple scheme.

A special type of queries, which are of interest in MAP authoring, includes *spatial* or *temporal* layout retrieval. In other words, queries of the type “Find the objects and their position on screen at the  $T_0$  second” (spatial layout) or “Find the objects that appear in the application during the  $(T_1, T_2)$  temporal segment and their temporal duration” (temporal layout) need to be supported by the underlying scheme. As we will present next, both types of queries are efficiently supported by the unified scheme, since they correspond to the *overlap\_during* operator and an appropriate reference object  $q$ : a rectangle  $q_1$  that intersects the t-axis at point  $T_0$ , or a cube  $q_2$  that overlaps the t-axis at the  $(T_1, T_2)$  segment, respectively. The reference objects  $q_1$  and  $q_2$  are illustrated in Fig. 9a. In a second step, the objects that compose the answer set are filtered in main memory in order to design their positions on the screen (spatial layout) or the intersection of their t-projections to the given temporal segment (temporal layout).

The ‘layout’ type of queries could be processed as described above. In particular, the screen layout at the 22nd second of the application, could be obtained by exploiting the reference object  $q_1$  at the specific time instance  $T_0 = 22$  s. The result would be a list of objects (the identifiers of the objects, their spatial and temporal coordinates) that are displayed at that temporal instance on the screen. This result may be visualized as a screen snapshot with the objects that are included in the answer set drawn in that (Fig. 9b). As for

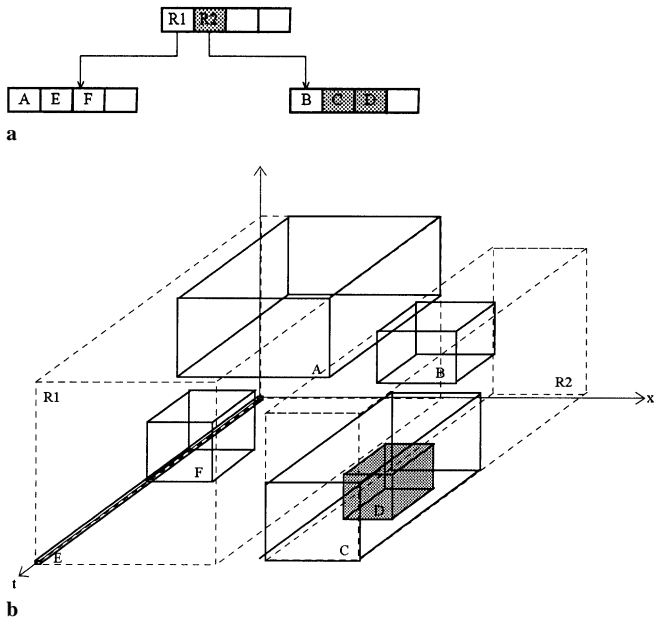


Fig. 8a,b. Retrieval of *overlap.during* operator using 3D R-trees

temporal layout query of Subsect. 3.4, it could be answered using a 3D cube  $q_2$  with area  $(X_{\max} - 0) \cdot (Y_{\max} - 0) \cdot (T_2 - T_1)$  as the reference object, where  $X_{\max} \cdot Y_{\max}$  is the screen area and  $(T_2 - T_1)$  is the requested temporal interval;  $T_1 = 10$  and  $T_2 = 20$  in our example. The result would be a list of objects (the identifiers of the objects, their spatial and temporal coordinates) that are included or overlapped with cube  $q_2$ . This result can be visualized towards a temporal layout by drawing the temporal line segments of the retrieved objects that lie within the requested temporal interval  $(T_2 - T_1)$  (Fig. 9c).

On the other hand, the simple indexing scheme (consisting of two index structures), as well as the ‘sorted-arrays’ scheme, are not able to give straightforward answers to the above layout queries, since information stored in multiple indices needs to be retrieved and combined.

In this section, we proposed several schemes for the indexing of objects that appear in MAPs and presented the retrieval procedure that concerns spatio-temporal operators on these objects. In the next section, all schemes will be analytically evaluated and compared to each other. Their comparison will result in general conclusions on the advantages and disadvantages of each solution.

## 5 Estimation of the retrieval cost

We present an analytical model that estimates the performance of R-trees on the retrieval of  $n$ -dimensional queries. The analytical formula is applicable to both R-tree-based indexing schemes, if we keep in mind that the simple one consists of one 2D R-tree and one 1D R-tree, while the unified one consists of one 3D R-tree. Using this model, we can estimate the performance of both schemes and compare their efficiency using several spatio-temporal operators.

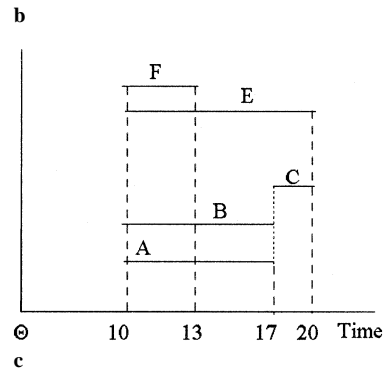
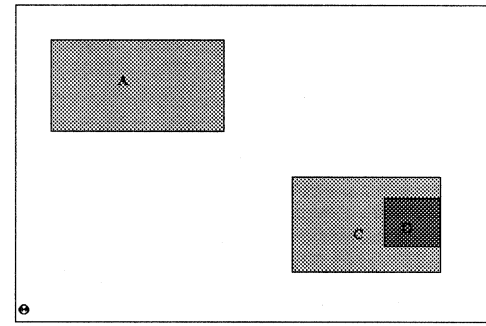
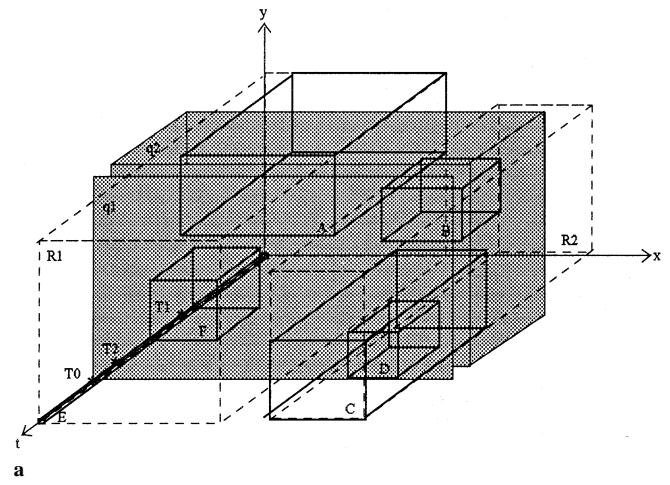


Fig. 9a-c. Spatial and temporal layout retrieval using 3D R-trees. a Query windows for spatial and temporal layout. b Spatial layout. c Temporal layout

### 5.1 Cost analysis of R-trees

Most of the work in the literature has dealt with the expected performance of R-trees for processing *overlap* queries, i.e., the retrieval of data objects  $p$  that share a common area with a query window  $q$  [Page93, Fal94b, Theo96b]. More particularly, let  $N$  be the total number of data objects indexed in an R-tree,  $D$  the density of the data objects in the global space and  $f$  the average capacity of each R-tree node. If we assume that the average size of a query window  $q$  is  $\prod_{i=1}^n q_i$ , then the expected retrieval cost (number of disk accesses) of an *overlap* query using R-trees is [Theo96b]:

$$C(q) = 1 + \sum_{j=1}^{1 + \lceil \log f \frac{N}{q} \rceil} \left\{ \frac{N}{f^j} \cdot \prod_{i=1}^n \left( \left( D_j \cdot \frac{f^j}{N} \right)^{1/N} + q_i \right) \right\}, (1)$$

where the average density of the R-tree nodes  $D_j$  at each level  $j$  is given by

$$D_j = \left\{ 1 + \frac{(D_j - 1)^{1/N} - 1}{f^{1/N}} \right\}^n. \quad (2)$$

Hence,  $D_j$  can be computed recursively using  $D_0$  which denotes the density  $D$  of the data MBRs, or, in other words, node density at each level of the R-tree is a function of the density of the dataset. Qualitatively, this means that the retrieval cost  $C(q)$  of an *overlap* query is estimated by only using knowledge of the dataset (number  $N$  and density  $D$  of data) and the query window  $q$  with no need to construct the tree structure.

Since Eq. 1 expresses the expected performance of R-trees on *overlap* queries using a query window  $q$ , in order to estimate the retrieval cost of a spatio-temporal operator  $R(p, q)$ , we need the following transformation:  $R(p, q) \Rightarrow \text{overlap}(p, Q)$ . In other words, the retrieval of a spatio-temporal operator using R-trees is equivalent (in terms of cost) to the retrieval of an *overlap* query using an appropriate query window  $Q$ . The necessary transformation  $Q$  for each operator  $R$  should take into consideration the corresponding constraint of the intermediate nodes, because only these nodes are important when estimating the retrieval cost [Papa95]. For the spatio-temporal operators that we consider in this paper, the appropriate query window  $Q = (Q_{x1}, Q_{x2}, Q_{y1}, Q_{y2}, Q_{t1}, Q_{t2})$  for the unified scheme (or  $Q = (Q_{x1}, Q_{x2}, Q_{y1}, Q_{y2})$ ,  $Q = (Q_{t1}, Q_{t2})$  for the simple scheme) is defined in Table 2, as a function of the original query window  $q = (q_{x1}, q_{x2}, q_{y1}, q_{y2}, q_{t1}, q_{t2})$ .

Using information from Table 2 and Eq. 1 we can estimate the expected cost for the query window  $Q$ , which equals the expected cost  $C(R)$  for the retrieval of a spatio-temporal operator  $R$ . The accuracy of the above analytical model has already been evaluated on spatial relationships of varying selectivity (e.g., *inside*, *near*, *northeast*, and combinations) in [Theo95]. Intuitively, we assume that the unified scheme should be the most efficient solution when both spatial and temporal information are included in the query, while, in the rest of the cases, the simple scheme seems to be preferable. The accuracy of these intuitive conclusions will be examined in the next subsection, where the above analytical model will be used as a basis for the analytical comparison of the proposed schemes.

## 5.2 Analytical comparison of the indexing schemes

In order to compare the efficiency of each proposed scheme on the retrieval of spatio-temporal operators, we assumed a MAP including 10,000 objects of the following distribution:

- a portion of 75% characterized by small projections on the three axes (x, y, t), e.g., text or video that cover a small space on the screen and last a short time,
- a portion of 15% characterized by zero projection on the two axes (x, y) and small projection on the third axis (t), e.g., sounds that cover zero space on the screen and last a short time,
- a portion of 5% characterized by small projections on the two axes (x, y) and large projection on the third axis

- (t), e.g., heading titles or logos that cover a small space on the screen and last a long time, and
- a portion of 5% characterized by large projections on the two axes (x, y) and small projection on the third axis (t), e.g., full text or background patterns that cover a large space on the screen and last a short time.

We consider the above distribution to be a typical distribution of media objects in a MAP and we use it for the comparison of the alternative indexing schemes. Different distributions of objects are also supported in a similar way by adapting their density  $D$ .

For the analytical estimates we used Eq. 1 and the following values: amount of data objects  $N = 10,000$  (8,500) for the 1D and 3D (2D) R-tree indices, density of data objects  $D = 145, 145, 1.6$  for the 1D, 2D, and 3D indices, respectively, and average node capacity  $f = 0.67 \cdot M$ , where  $M = 84, 50, 35$  for 1D, 2D, and 3D R-trees, respectively<sup>4</sup>. The sizes of the reference objects  $q$  varied from 0% up to 50% of the global space per axis, while the corresponding query windows  $Q$  for each combination of R-tree index and operator were formulated in Table 2. Table 3 summarizes the comparative results for the operators discussed in the paper. For uniformity reasons, we set the cost of serial retrieval to be 100% and express the costs of the “sorted-arrays” scheme and the indexing schemes proposed in Sect. 4 as portions of that value.

The cost of serial retrieval is computed as follows. Each object representation requires a space of 28 bytes (4 bytes  $\times$  7 numbers). If we set the size of a disk page to be 1024 bytes, then a page contains 36 (= 1024/28) objects. Hence, 278 pages are required to store 10,000 objects. All of these pages should be accessed in order to answer any spatio-temporal operator.

The cost of the ‘sorted-arrays’ scheme is computed as follows. The scheme consists of three arrays which contain the id plus the low (as primary key) and high (as secondary key) coordinate of each object per axis. Hence, each object representation requires a space of 12 bytes (4 bytes  $\times$  3 numbers). Since a page of 1024 bytes contains 85 (= 1024/12) objects, each array includes 118 (= 10,000/85) pages. The retrieval cost per operator is a ratio of the total amount of 118 pages and is computed by using classic “divide-and-conquer” techniques with respect to the constraints that characterize each operator (i.e., logarithmic cost per array for selective almost exact match queries, such as *during* and about 50% of the total cost per array for non-selective queries, such as *overlap*, *above*, *before*, etc.).

The costs of the indexing schemes have been already discussed in Subsect. 5.1, with Eq. 1 being used for their computation.

Several conclusions arise from the analytical comparison results presented in Table 3.

<sup>4</sup> The number of data objects stored in the 2D index is less than those stored in the 1D and 3D indices, because zero-space objects (e.g., sounds) are not included in the dataset of the 2D index. The  $D$  values are implied from the above distribution if we assume that small (large) space corresponds to 5% (50%) of the screen and a short (long) period of time corresponds to 1% (10%) of the whole duration of the application. The 67% capacity is a typical value for R-trees and variants, while the  $M$  values represent the maximum node capacity for pages of 1024 bytes.

**Table 2.** Query windows  $Q$  for spatio-temporal operators

Operator	1D R-tree	2D R-tree	3D R-tree
<i>overlap</i>	–	$Q = (q_{x1}, q_{x2}, q_{y1}, q_{y2})$	$Q = (q_{x1}, q_{x2}, q_{y1}, q_{y2}, 0, 1)$
<i>above</i>	–	$Q = (0, 1, q_{y2}, 1)$	$Q = (0, 1, q_{y2}, 1, 0, 1)$
<i>during</i>	$Q = (q_{t1}, q_{t2})$	–	$Q = (0, 1, 0, 1, q_{t1}, q_{t2})$
<i>before</i>	$Q = (0, q_{t1})$	–	$Q = (0, 1, 0, 1, 0, q_{t1})$
<i>overlap_during</i>	–	–	$Q = (q_{x1}, q_{x2}, q_{y1}, q_{y2}, q_{t1}, q_{t2})$
<i>overlap_before</i>	–	–	$Q = (q_{x1}, q_{x2}, q_{y1}, q_{y2}, 0, q_{t1})$
<i>above_during</i>	–	–	$Q = (0, 1, q_{y2}, 1, q_{t1}, q_{t2})$
<i>above_before</i>	–	–	$Q = (0, 1, q_{y2}, 1, 0, q_{t1})$

**Table 3.** Comparison of indexing schemes (with respect to serial storage cost)

Operator	“sorted-arrays” scheme	Simple scheme (one 1D plus one 2D R-tree)	Unified scheme (one 3D R-tree)
<i>overlap</i>	40%–45%	5%–10%	5%–15%
<i>above</i>	20%–25%	45%–50%	80%–95%
<i>during</i>	1%	2%–10%	25%–45%
<i>before</i>	20%–25%	25%–35%	80%–95%
<i>overlap_during</i>	40%–45%	5%–20%	1%–5%
<i>overlap_before</i>	60%–70%	35%–40%	3%–10%
<i>above_during</i>	20%–25%	55%–60%	15%–25%
<i>above_before</i>	40%–50%	70%–85%	50%–65%

- The intuitive conclusion that the simple R-tree scheme would outperform the unified one when dealing with operators that keep only temporal or spatial information, while the opposite would be the case for spatio-temporal operators, is valid. The first four operators are more efficiently supported by the simple scheme, while the cost of the unified scheme is usually two or three times higher. The reverse situation appears for the last four operators.
- Both schemes based on R-trees are much more efficient than the serial storage scheme for all operators. For the most selective ones (*overlap*, *during*, *overlap\_during*), the improvement is at a level of one or even two orders of magnitude, compared to the serial cost. For the least selective ones (*above*, *before*, *above\_before*), the cost of the most efficient scheme is a 1/4 up to a 1/2 portion of the serial cost.
- The ‘sorted-arrays’ scheme is shown to be a competitive solution. It always outperforms the serial storage scheme (its cost being usually a 1/5 up to a 3/5 portion of the serial cost). In comparison with the two indexing schemes based on R-trees, it is the winner when operators of very low selectivity (*above*, *before*, *above\_before*) are involved, while, for the rest of the cases, it remains an efficient alternative solution.

A graphical comparison of the three schemes as compared to the serial one appears in Fig. 10.

The above conclusions are, more or less, expected. However, in real-world cases, a mixture of temporal, spatial and spatio-temporal operators needs to be supported. Then, selecting the most efficient scheme for such mixed requirements arises. In [Theo96a], we propose guidelines for dealing with this issue. The average costs of the alternative indexing scheme based on R-trees are evaluated when: all eight operators are used, or only the most selective (*inclusive*) ones are involved, or when only the least selective (*exclusive*) operators are involved. The main conclusion from this

discussion is the following. If we distinguish between high and low selective operators, then the thresholds shift right (high selective operators) or left (low selective operators). In other words, when dealing with selective operators, the simple scheme is sometimes preferable, even if the majority (up to 65%) of the queries involve spatio-temporal information. It is a choice of the multimedia database designer to select the most preferable solution, with respect to the requirements of the MAP author.

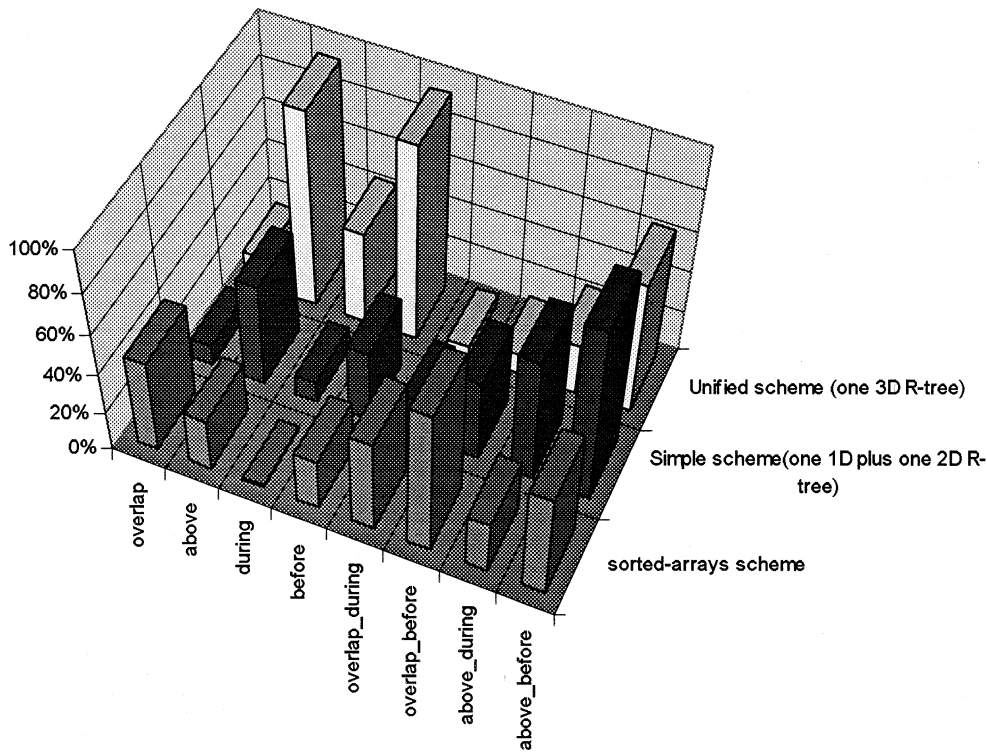
## 6 Conclusion

Authoring complex MAPs that involve a large number of media objects is a complicated task, keeping in mind the large set of possible relationships and events that may be encountered in the application context, as well as the various potential combinations of these parameters. Thus, the need for a scheme that will support the authors in managing the large number of objects and spatio-temporal relationships among them is required. Current authoring tools do not provide such facilities. The mechanism we propose provides support for queries, before application execution, related to the application scenario, and more specifically, to spatio-temporal relationships among media objects (i.e., “*does object A spatially overlap with object B in the application?*” or “*which objects temporally overlap with object A?*”). Moreover, authors may request spatio-temporal layouts of the application at specific spatial and/or temporal instances (i.e., “*which objects appear in the application at a specific time instance*”, or “*what is the spatial (screen) layout at a specific time instance during the application*”, or “*what is the temporal layout of the application in terms of temporal intervals*”).

In this paper, we presented

- a model for the declarative representation of spatio-temporal composition in the context of large MAPs,
- an efficient indexing mechanism for such applications based on R-trees.

With regard to the MAP model, the motivation for this research work was the lack of a complete declarative approach for representation of spatio-temporal composition of objects in current multimedia document standards (Hy-Time [Newc91], MHEG [ISO93]) and authoring tools (Apple / SCRIPTX [Scri96], Assymetrix / ToolBook [Assy94], Macromedia / MacroMind Director [Makr94]). An integrated high-level model to facilitate the above-mentioned requirements would be of benefit to application designers and developers and presents the following advantages.



**Fig. 10.** Comparison of the retrieval cost of the three indexing schemes (% of the serial storage cost)

- Explicit mapping between author high-level spatio-temporal specifications into a declarative uniform specification. Spatio-temporal relationships (instead of their absolute coordinates) are retained. This enables answering queries based on spatio-temporal relationships among the objects.
- Formal specification of a MAP, which will allow the use of software-engineering methodologies (quality and maintenance) in this area.
- Separation of application specification from the application content, which enables reusability of the MAP functionality specifications for other cases with similar functionality but different content.

As for the indexing schemes, we are based on indexing spatial and temporal presentation features of the media objects during the application. We propose two indexing schemes based on the R-tree data structure; the first scheme includes one 1D and one 2D R-tree that separately index temporal and spatial characteristics of objects, respectively, while the second scheme includes one 3D R-tree that indices the spatio-temporal characteristics of objects, considering time to be the third axis of the coordinate system. We evaluated the two schemes against the serial storage scheme and a scheme using disk-resident sorted arrays, and presented guidelines that help one to select the most appropriate solution.

The composition model we proposed has a considerable limitation: it does not support interaction handling in terms of events, while it covers the case of pre-orchestrated scenarios (i.e., the spatio-temporal ordering of objects in the application is pre-defined). Specifically for the indexing mechanism, a limitation of our approach is that it does not support interactive scenarios, due to the non-deterministic spatial and temporal occurrences of the objects.

We claim that there is a lot of potential in this area. We plan to address the following research issues in the future:

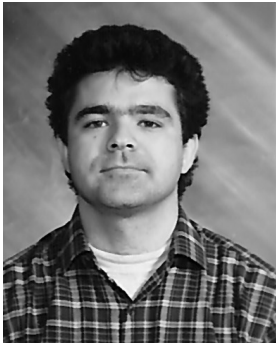
- *automatic composition tuple extraction* from spatio-temporal specifications.
- *design of a GUI* that fulfills the requirements of designing, verifying and testing complex spatio-temporal composition of objects in the context of MAPs,
- *modeling of events*, in order to provide the ability of handling user interactivity. Modern applications are heavily based on user interaction, which may be represented in terms of events. Embedding of such events in our model would result in a powerful tool for authors of interactive multimedia applications,
- *constraint checking*: a multimedia scenario involves a lot of objects which may be related in various ways. These relations may lead to inconsistencies. Such issues should be further investigated.

In a parallel way, the proposed unified indexing scheme could be further extended towards:

- *scenario rendering based on the indexing scheme*: the model we proposed could also be used during the execution phase of the scenario. In this case, the appropriate media would be quickly located on the basis of the scenario,
- *indexing of interactive scenarios*: the indexing scheme should be modified, in order to cover the case of interactive scenarios, where the spatio-temporal presence of an object depends on the occurrence of events.

## References

- [Alle83] Allen JF (1983) Maintaining Knowledge about Temporal Intervals. *Comm ACM* 26(11):832–843
- [Assy94] Assymetrix (1994) Assymetrix ToolBook User's Manual
- [Beck90] Beckmann N, Kriegel H-P, Schneider R, Seeger B (1990) The R\*-tree: An Efficient and Robust Access Method for Points and Rectangles. In: Garcia-Molina H, Tagadish HV (eds) Proceedings of ACM SIGMOD International Conference on Management of Data, ACM-Press
- [Bent75] Bentley JL (1975) Multidimensional Binary Search Trees Used for Associative Searching. *Comm ACM* 18:509–517
- [Blak96] Blakowski G, Steinmetz R (1996) A Media Synchronisation Survey: Reference Model, Specification, and Case Studies. *IEEE J Sel Areas Commun* 14(1):5–35
- [Bufo96] Buford J (1996) Evaluating HyTime: An Examination and Implementation Experience. In: Proceedings of ACM Hypertext '96 Conference, ACM-Press
- [Chiu94] Chiu T (1994) Content-Based Image Indexing. In: Proceedings of the 20th International Conference on Very Large Databases (VLDB)
- [Duda95] Duda A, Keramane C (1995) Structured Temporal Composition of Multimedia Data. In: Proceedings of the 1st IEEE International Workshop for MM-DBMSs, IEEE Computer Society, Los Alamitos, Calif
- [Egen91] Egenhofer M, Franzosa R (1991) Point-Set Topological Spatial Relations. *Int J Geogr Inf Sys* 5(2):161–174
- [Erf193] Erfle R (1993) Specification of temporal constraints in multimedia documents using HyTime. *Electronic Publishing* 6(4):397–411
- [Falo94a] Faloutsos C, Equitz W, Flickner M, Niblack W, Petkovic D, Barber R (1994) Efficient and Effective Querying by Image Content. *J Int Inf Sys* 3:1–28
- [Falo94b] Faloutsos C, Kamel I (1994) Beyond Uniformity and Independence: Analysis of R-trees Using the Concept of Fractal Dimension. In: Proceedings of the 13th ACM Symposium on Principles of Database Systems (PODS), ACM-Press
- [Gutt84] Guttman A (1984) R-trees: A Dynamic Index Structure for Spatial Searching. In: Yormark B (ed) Proceedings of ACM SIGMOD International Conference on Management of Data, ACM-Press
- [Hamb72] Hamblin C (1972) Instants and Intervals. Fraser JT (ed) *The Study of Time*. Springer, Berlin Heidelberg, pp 325–331
- [Hand96] Handl M (1996) A New Multimedia Synchronisation Model. *IEEE J Sel Areas Commun* 14(1):73–83
- [Hirz95] Hirzalla N, Falchuck B, Karmouch A (1995) A Temporal Model for Interactive Multimedia Scenarios. *IEEE Multimedia Magazine* 2(3):24–31
- [Iino94] Iino M, Day YF, Ghafoor A (1994) An Object Oriented Model for Spatio-Temporal Synchronization of Multimedia Information. In: Proceedings of the 1st IEEE Conference on Multimedia Computing and Systems (ICMCS), IEEE Computer Society, Los Alamitos, Calif
- [ISO93] ISO/IEC (1993) Information Technology – Coded representation of Multimedia and Hypermedia Information Objects (MHEG)
- [Krie96] Krieg P (1996) Digital Hollywood: The turbulent Marriage of Computer, Telecom and Media Industry. In: Proceedings of International Workshop on Multimedia Software Development (MMSD), IEEE Computer Society, Los Alamitos, Calif
- [King94] King PR (1994) Towards a temporal logic-based formalism for expressing temporal constraints in multimedia documents. Technical Report 942, LRI, Universite de Paris-Sud, Orsay, France
- [Lamp90] Lamport L (1990)  $\text{\LaTeX}$ : A Document Preparation System. Addison-Wesley, Reading, Mass.
- [Litt93] Little T, Ghafoor A (1993) Interval-Based Conceptual Models for Time Dependent Multimedia Data. *IEEE Trans Knowl Data Eng* 5(4):551–563
- [Macr90] Macromind Inc. (1990) Macromind Director, Interactivity Manual
- [Newc91] Newcomb S, Kipp N, Newcomp V (1991) The HyTime, Hypermedia Time based Document Structuring Language. *Comm ACM* 34(11):67–83
- [Oren86] Orenstein J (1986) Spatial Query Processing in an Object-Oriented Database System. In: Zanido C (ed) Proceedings of ACM SIGMOD International Conference on Management of Data, ACM-Press
- [Page93] Pagel B-U, Six H-W, Toben H, Widmayer P (1993) Towards an Analysis of Range Query Performance. In: Proceedings of the 12th ACM Symposium on Principles of Database Systems (PODS), ACM-Press
- [Papa95] Papadias D, Theodoridis Y, Sellis T, Egenhofer M (1995) Topological Relations in the World of Minimum Bounding Rectangles: a Study with R-trees. In: Carey M, Schneider D (eds) Proceedings of ACM SIGMOD International Conference on Management of Data, ACM-Press
- [Papa97] Papadias D, Theodoridis Y (1997) Spatial Relations, Minimum Bounding Rectangles, and Spatial Data Structures. *Int J Geogr Inf Sys* 11(2):111–138
- [Same90] Samet H (1990) *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, Mass.
- [Schn96] Schnepf J, Du DH-C (1996) Doing FLIPS: Flexible Interactive Presentation Synchronisation. *IEEE J Sel Areas Commun* 14(1):114–125
- [Scri96] ScriptX Technical Overview (1996) available at: <http://dev.info.apple.com/scriptx/techdocs.html>
- [Theo95] Theodoridis Y, Papadias D (1995) Range Queries Involving Spatial Relations: A Performance Analysis. In: Proceedings of the 2nd International Conference on Spatial Information Theory (COSIT), IEEE Computer Society, Los Alamitos, Calif
- [Theo96a] Theodoridis Y, Vazirgiannis M, Sellis T (1996) Spatio-Temporal Indexing for Large Multimedia Applications. In: Proceedings of the 3rd IEEE Conference on Multimedia Computing and Systems (ICMCS), IEEE Computer Society, Los Alamitos, Calif
- [Theo96b] Theodoridis Y, Sellis T (1996) A Model for the Prediction of R-tree Performance. In: Proceedings of the 15th ACM Symposium on Principles of Database Systems (PODS), ACM-Press
- [Ubel94] Ubell M (1994) The Montage Extensible Datablade Architecture. In: Snodgrass R, Winslet M (eds) Proceedings of ACM SIGMOD International Conference on Management of Data, ACM-Press
- [Vazi95] Vazirgiannis M, Hatzopoulos M (1995) Integrated Multimedia Object and Application Modeling Based on Events and Scenarios. In: Proceedings of the 1st IEEE International Workshop for MM-DBMSs, IEEE Computer Society, Los Alamitos, Calif
- [Vazi96a] Vazirgiannis M, Theodoridis Y, Sellis T (1996) Spatio-Temporal Composition in Multimedia Applications. In: Proceedings of International Workshop on Multimedia Software Development (MMSD), IEEE Computer Society, Los Alamitos, Calif
- [Vazi96b] Vazirgiannis M, Sellis T (1996) Event And Action Representation And Composition For Multimedia Application Scenario Modelling. In: Proceedings of ERCIM Workshop on Interactive Distributed Multimedia Systems and Services, Springer Verlag, Heidelberg



MICHAEL VAZIRGIANNIS received his Diploma degree in Physics in 1986 from the University of Athens, Athens, Greece. In 1989, he received the M.Sc. degree from Heriot Watt University, Edinburgh (UK) and in 1994 the Ph.D. degree from the University of Athens, Athens, Greece. In 1995, he joined the Knowledge & Data Bases Laboratory in the Computer Science Division of the National Technical University of Athens, Athens, Greece, where he is a postdoctoral researcher. His research interests include multimedia information systems, spatio-temporal databases and fuzzy logic. He has published articles in

refereed journals and international conferences in the above areas. Dr. Vazirgiannis worked as guest scientist in GMD/IPSI in Darmstadt, Germany (1996) and in Fernuniversitaet in Hagen, Germany (1997). Dr. Vazirgiannis is a member of ACM and IEEE.



YANNIS THEODORIDIS received his Dipl. Eng. (1990) and Dr. Eng. (1996) degrees from National Technical University of Athens (NTUA), Greece. He is currently a Research Engineer at the Knowledge and Data Bases Systems Laboratory, NTUA. His research interests include geographical databases, multimedia systems, spatial data structures and algorithms. He is a member of ACM and IEEE.



TIMOS SELLIS received his Diploma degree in Electrical Engineering in 1982 from the National Technical University of Athens, Athens, Greece. In 1983, he received the M.Sc. degree from Harvard University and in 1986 the Ph.D. degree from the University of California at Berkeley, where he was a member of the INGRES group, both in Computer Science. In 1986, he joined the department of Computer Science of the University of Maryland, College Park as an Assistant Professor, and became an Associate Professor in 1992. Between 1992 and 1996 he was an Associate Professor at the Computer Science Division of the

National Technical University of Athens, in Athens, Greece, where he is currently a Full Professor. His research interests include extended relational database systems, active database systems, and spatial, image and multimedia database systems. He has published several articles in refereed journals and international conferences in the above areas. Prof. Sellis is a recipient of a Presidential Young Investigator (PYI) award for 1990–1995. He is a member of the Editorial Boards of the International Journal on Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies, and Geoinformatica. Dr. Sellis is a member of IEEE and ACM.