Symbolic Description and Visual Querying of Image Sequences Using Spatio-Temporal Logic

Alberto Del Bimbo, *Member, IEEE*, Enrico Vicario, and Daniele Zingoni, *Member, IEEE*

Abstract—**The emergence of advanced multimedia applications is emphasizing the relevance of retrieval by contents within databases of images and image sequences. Matching the inherent visuality of the information stored in such databases, visual specification by example provides an effective and natural way to express content-oriented queries. To support this querying approach, the system must be able to interpret example scenes reproducing the contents of images and sequences to be retrieved, and to match them against the actual contents of the database. In the accomplishment of this task, to avoid a direct access to raw image data, the system must be provided with an appropriate description language supporting the representation of the contents of pictorial data.**

An original language for the symbolic representation of the contents of image sequences is presented. This language, referred to as Spatio-temporal Logic, comprises a framework for the qualitative representation of the contents of image sequences, which allows for treatment and operation of content structures at a higher level than pixels or image features. Organization and operation principles of a prototype system exploiting Spatio-temporal Logic to support querying by example through visual iconic interaction are expounded.

Index Terms—**Image sequence retrieval, visual querying by example, image sequence symbolic description, symbolic projection, spatial logic, temporal logic.**

I. Introduction

With the emergence of multimedia applications, a growing interest is being focused on databases of digital images and image sequences. The marked characterization of the information managed in such systems largely pervades retrieval methods and the way in which these are supported by appropriate content representation and querying languages.

The lack of a native structured organization and the inherent visuality of pictorial data advise against the use of conventional querying techniques based on textual keywords, and rather suggest the convenience of visual querying by example and iconic indexes [27]. In this approach, queries are expressed through visual examples reproducing visual features of searched images such as object shapes, object colors or spatial relationships, and retrieval is carried out by checking these features against those of the information units stored in the database. This largely reduces the cognitive effort of the user in the access to the database, and permits a natural exploitation of human capabilities in picture analysis and interpretation.

A number of techniques have appeared in the literature which deal with visual querying by example and representation of iconic indexes for single images, taking into account different facets of the informative contents of pictorial data. Querying approaches based on picture color distribution and texture organization have been proposed in [26], [6], and [25]; queries are expressed by selecting colors and textures from a menu, and retrieval is carried out by comparing them against color histograms and texture measures of the objects appearing in the images stored in the database.

Querying by visual sketch has been proposed in [20], [25], and [17]. In this approach, the user draws a rough sketch of an object shape, and retrieval is performed by considering shape features such as area, circularity, major axis orientation and moment invariants [20], [25] or through elastic template matching [17].

Semantic relationships among imaged objects have been used as iconic indexes by several authors [7], [8], [23], [10]. In particular, the expression of spatial relationships between imaged objects by means of the *Symbolic Projection* technique has been first proposed in [8] and then exploited and extended by several authors [21], [11], [13], [18], [15]. Using symbolic projections, the contents of an image are represented by a *2D-string* which encodes the positional relationships between the projections of the objects on two reference coordinate axes. Queries can be formulated visually by arranging object icons on a screen to reproduce the spatial relationships between imaged objects [9]. These relationships are encoded as 2D strings and retrieval matching is reduced to the comparison of symbolic strings.

As opposed to the large number of experiences on single images, only a few techniques have been proposed for content representation, indexing and querying of image sequences.

In [24], [19], [29], cuts between video clips are detected through the analysis of color histograms of subsequent frames; frame subsequences with continuity in color distribution are indexed by their initial frame. Full video search for object appearance is accomplished in [24] by comparing the color map of a sample object against color distributions within connected regions of the indexing frames.

Descriptions of sequence contents using semantic relationships among imaged objects have been proposed in [4], [28], [14]. Specifically, in [28], semantic networks capturing high-level spatio-temporal interactions among a limited number of objects are proposed to describe contents of short image sequences. In this approach, queries are expressed textually, and retrieval is carried out by checking a rule base against episode descriptions. In [14], video content representation is provided

Manuscript received Feb. 12, 1993; revised July 1, 1994.

The authors are with Dipartimento di Sistemi e Informatic, Università di Firenze, 3, via Santa Marta, 50139, Florence, Italy; e-mail: {delbimbo, vicario}@aguirre.ing.unifi.it. A. Del Bimbo is also with Dipartimento di Elettronica per l'Automazione, Università di Brescia, 38, via Branze, 25133, Brescia, Italy.

IEEECS Log Number K95052.

through handmade annotations reflecting the occurrence of significative situations. A retrieval system is presented in which queries are expressed by combination of situation icons associated with annotations. In [4], the perspective use of the Symbolic Projection approach and exploitation of its potential benefits in the representation and visual querying by example of image sequences are suggested. 2D strings are used to represent spatial relationships between objects in individual frames and time tags are employed to mark changes in 2D strings throughout the sequence. While effective to represent contents of image sequences at a finer level of detail with respect to the previous approaches, this technique does not provide a sufficient flexibility and expressivity in the representation of spatio-temporal relationships as needed in sequence querying.

In this paper, an original language for the symbolic representation of spatio-temporal relationships between objects within image sequences is presented, and its use within a prototype retrieval system supporting visual querying by example of image sequences is discussed.

The language, which is referred to as *Spatio-temporal Logic* (STL), extends basic concepts of *Temporal Logic* and *Symbolic Projection* to provide descriptions of sequence contents within a unified and cohesive framework. By inheriting from Temporal Logic the native orientation towards qualitative and uncomplete descriptions, STL permits (without imposing) representations with intentional ambiguity and detail refinement, which are especially needed in the expression of queries. Besides, by exploiting the Symbolic Projection approach, STL supports the description of sequence contents at a lower level of granularity with respect to semantic networks and annotation based approaches, thus allowing to focus on the actual dynamics represented in the sequence.

In the retrieval system, queries are expressed by the user through a visual iconic interface which allows for the creation of sample dynamic scenes reproducing the contents of sequences to be retrieved. Sample scenes are automatically described through STL assertions, and retrieval is carried out by checking them against the descriptions of the sequences in the database.

The rest of the paper is organized in four sections. Temporal and Spatial operators of STL are introduced in Section II, and their flexibility in the expression of spatio-temporal descriptions is discussed in Section III. The system for image sequence retrieval is presented in Section IV, expounding how STL is casted in the system and the way in which visual retrieval is performed. Conclusions and future research are discussed in Section V.

II. Spatio-temporal Logic

Spatio-temporal Logic is an original formulation of Temporal Logic, which encompasses both spatial and temporal expressivity within a unified framework for the representation of the contents of image sequences. Temporal properties are expressed as Temporal Logic assertions capturing the evolution over time of spatial relationships occurring in the single frames of the sequence. In turn, these relationships are expressed through an original language, referred to as *Spatial Logic*, which transposes concepts of Symbolic Projection and Temporal Logic itself to express ordering relationships between the objects of individual frames of a sequence.

A. Temporal Logic of Scene Sequences

Temporal Logic has been widely addressed as a language to describe and reason about the temporal ordering of the actions of parallel and concurrent systems [2], [22]. In this logic, *state assertions* capturing static properties of the individual states visited in a system execution sequence are combined through temporal operators to express *sequence assertions* capturing dynamic properties of the entire execution sequence. Model checking algorithms have been developed, which support the automatic verification of abstract properties expressed as Temporal Logic assertions against concrete system models expressed as state transition systems [12].

Temporal Logic of scene sequences can be regarded as a variant of the propositional Temporal Logic of linear and discrete time [3], in which state assertions capture the spatial arrangement of the objects in a scene. These state assertions are inductively combined through the Boolean connectives ($\neg$, $\wedge$, and their derived shorthands $\vee$, $\leftrightarrow$, $\rightarrow$, and $\leftarrow$) and the *temporal-until* operator (unt_t). Boolean connectives have their usual meaning and permit the combination of multiple assertions referring to any individual scene of the sequence. The *temporal-until* operator permits the expression of temporal ordering relationships between the scenes in which different state assertions hold. Specifically, *temporal-until* is a binary operator which permits the composition of two assertions θ_1 and θ_2 to express that θ_1 holds along the sequence at least until reaching a scene in which θ_2 holds.

Syntax: If σ is a scene sequence, a *temporal assertion* Θ on σ is expressed in the form:

$$\Theta := (\sigma, k) \models \theta \qquad (1)$$

where k is the index of a scene in σ, and θ a *temporal formula* which is formed by combining *spatial assertions* Φ through the Boolean connectives of negation and conjunction, and through the *temporal-until* operator unt_t:

$$\theta := \Phi \mid \neg\theta \mid \theta_1 \wedge \theta_2 \mid \theta_1\ unt_t \theta_2 \qquad (2)$$

Semantics: The satisfaction of a temporal assertion θ is interpreted over a sequence σ according to the following inductive semantic clauses:

- $(\sigma, k) \models \Phi$ iff the spatial assertion Φ holds in the kth scene of sequence σ;
- $(\sigma, k) \models \neg\theta$ iff the temporal assertion $(\sigma, k) \models \theta$ does not hold:

$$\neg((\sigma, k) \models \theta) \qquad (3)$$

- $(\sigma, k) \models \theta_1 \wedge \theta_2$ iff both $(\sigma, k) \models \theta_1$ and $(\sigma, k) \models \theta_2$ hold:

$$((\sigma, k) \models \theta_1) \wedge ((\sigma, k) \models \theta_2) \qquad (4)$$

- $(\sigma, k) \models \theta_1\ unt_t \theta_2$ iff θ_2 holds in a scene with index $k' > k$ and θ_1 holds in all the scenes from k to k':

$$\exists \Delta > 0. \quad \left((\sigma, k+\Delta) \models \theta_2\right) \wedge \quad \left(\forall \delta \in [0, \Delta-1], (\sigma, k+\delta) \models \theta_1\right) \tag{5}$$

Shorthands: Using the *temporal-until* operator in conjunction with Boolean connectives, a number of further temporal operators can be derived as shorthands. In particular, the *temporal-eventually* ($\Diamond_t$) and *temporal-always* ($\Box_t$) operators permit to express that a certain condition holds either in *some* or in *all* the future scenes of a sequence:

- $(\sigma, k) \models \Diamond_t\theta$ means that θ will hold in *some* scene subsequent to the kth one:

$$\Diamond_t\theta := \textit{true unt}_t\ \theta \tag{6}$$

- $(\sigma, k) \models \Box_t\theta$ means that θ holds in *all* the scenes subsequent to the kth one:

$$\Box_t\theta := \neg(\textit{true unt}_t\ (\neg\theta)) \tag{7}$$

Checking Temporal Formulae against Scene Sequences: In Fig. 1, a sequence made up of five scenes is schematized as a time line with five nodes. Each node is labeled with the spatial assertion that is satisfied in its corresponding scene: The spatial assertion Φ_1 holds in the scenes S_1, S_3, and S_4, while the assertion Φ_2 holds in the scene S_5. According to this, sequence σ satisfies the Temporal Logic assertion:

$$(\sigma, k) \models (\Phi_1\ \textit{unt}_t\ \Phi_2) \tag{8}$$

for the scenes with index k equal to 3 or 4, but not with k equal to 1 or 2. Since for each scene in the sequence there exists a future scene in which the spatial assertion Φ_2 holds, for any k in the interval [1, 5], the sequence σ also satisfies the temporal assertion:

$$(\sigma, k) \models \Diamond_t\Phi_2 \tag{9}$$

Besides, since one of the assertions Φ_1 and Φ_2 holds in all the scenes subsequent to scene S_3 (inclusive), for any value of k in the interval [3, 5], σ satisfies the assertion:

$$(\sigma, k) \models \Box_t(\Phi_1 \vee \Phi_2) \tag{10}$$

In general, for any temporal formula θ and any finite scene sequence σ, the satisfaction of θ on the states of σ can be automatically derived from the labeling of spatial assertions satisfied in the individual scenes through the Clarke's model checking algorithm [12]. This algorithm identifies every state S_i in σ such that $(\sigma, i) \models \theta$, and runs in linear time with respect to both the length of θ and the number of scenes in σ. Briefly described, the algorithm consists of two subsequent steps. In the first step, the temporal formula θ is recursively decomposed into subformulae of decreasing length until reducing it as a composition of state assertions. This decomposition follows the inductive semantic clauses of Temporal Logic of scene sequences. For instance, the temporal formula $\theta = \Phi_1$ $\textit{unt}_t\ \Phi_2$ is decomposed as $\theta = \theta_1\ \textit{unt}_t\theta_2$, where $\theta_1 = \Phi_1$, $\theta_2 = \Phi_2$. In the second step, subformulae are recursively checked, in order of growing length, against the states of the sequence, and the labeling of each state is progressively augmented with all the subformulae that it satisfies. Referring again to the above example, the initial labeling of the states with respect to the spatial assertions Φ_1 and Φ_2 is first exploited to identify and label the states in which the elementary statements θ_1 and θ_2 are satisfied (these are the states S_1, S_3, S_4, and the state S_5, respectively). This labeling is then used to identify and label the states in which $\theta_1\ \textit{unt}_t\theta_2$ is satisfied (these are the states starting from which θ_1 holds until a state in which θ_2 holds, i.e., states S_3 and S_4).

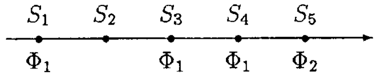


Fig. 1. The schematization of a sequence with five scenes.

B. Spatial Logic of Individual Scenes

Spatial Logic transposes the concepts of Temporal Logic to express geometric ordering relationships between the projections of the objects in a multi-dimensional scene.

The scene model of spatial logic assumes that a scene S is a triple $S = <V, Obj, F>$, where V is an N-dimensional Euclidean space ($N = 2$ and $N = 3$ being the significant cases), Obj is a set of objects in V, and F is a mapping from Obj onto the powerset of V, which associates each object p with the set of points of V over which it stands:

$$F(p) := \{r \in V \mid p \textit{ stands over } r\} \tag{11}$$

Mirroring the native discrete partitioning of the time axis, the space V of each individual scene can be partitioned into a grid of rectangular *regions* aligned with any given system of N orthogonal coordinate axes $E = \{e_n\}_{n=1,N}$. Each region is labeled with a number of *atomic spatial propositions* expressing the presence of objects in the region itself. With the assumption of one such partitionment, a scene $S = <V, Obj, F>$ can be represented as a discrete scene $S_E = <V_E, Obj, F_E>$, where

- V_E is a partitionment over V defined as the union of the set of regions $g(\bar{J})$:

$$V_E = \bigcup\left\{g(\bar{J}) \mid \bar{J} \in Z^N\right\} \tag{12}$$

  where, considering each coordinate axis e_n as partitioned into a sequence of adjacent subintervals $\{I_j^n\}_{j=-\infty,\infty}$, for any possible N–tuple of integer numbers $\bar{J} = (J_1, J_2, \ldots, J_N) \in Z^N$, $g(\bar{J})$ denotes the rectangular region obtained from the Cartesian product of subintervals $I_{j_1}^1, I_{j_2}^2, \ldots, I_{j_N}^N$.
- F_E is derived as the quantization of F over V_E by associating each object p with the set of regions over which it stands:

$$F_E(p) := \left\{g(\bar{J}) \in V_E \mid \exists r \in g(\bar{J}).r \in F(p)\right\} \tag{13}$$

As an example, the sketch of a bidimensional scene with a rectangular grid aligned with axes e_1 and e_2, and two objects p_1 and p_2, is depicted in Fig. 2.

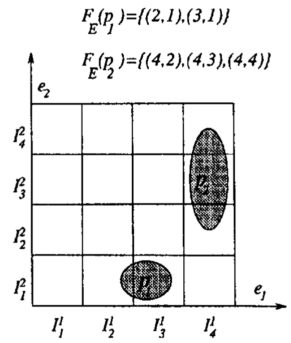


Fig. 2. The sketch of a bidimensional scene with a rectangular partitioning into 16 regions and two objects p_1 and p_2 standing over regions (2, 1), (3, 1) and (4, 2), (4, 3), (4, 4), respectively.

B.1. Region-Based Formulation

In the following, a *region-based* formulation of Spatial Logic is introduced which permits to express the positioning of a set of objects with respect to a single region of the scene. Relationships between objects possibly extending over multiple regions will be later accommodated in the treatment through the introduction of *context declarators.*

Syntax: If S_E is a discrete scene referred to the set of Cartesian axes $E = \{e_n\}_{n=1,N}$, a *Spatial Logic assertion* Φ on the contents of S_E is expressed in the form:

$$\Phi := \left(S_E, \bar{J}, e_n\right) \models \phi \qquad (14)$$

where $\bar{J}$ is an N–tuple of integer numbers identifying a region of the scene (namely, $g(\bar{J})$), e_n is one of the axes of the reference system E, and, ϕ is a *spatial formula*. Spatial formulae are formed by any possible combination of object identifiers p through the Boolean connectives and the *spatial-positive-until* and *spatial-negative-until* operators unt_{s+} and unt_{s-}:

$$\phi := p \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \phi_1\ unt_{s+}\phi_2 \mid \phi_1\ unt_{s-}\phi_2 \qquad (15)$$

It is worth noting that, according to this syntax, a Spatial Logic assertion refers to a single reference axis (namely, e_n). The expression of spatial conditions referring to multiple axes can be accomplished through the Boolean combination of multiple spatial assertions within a common Temporal Logic assertion.

Semantics: The satisfaction of a spatial assertion $(S_E, \bar{J}, e_n) \models \phi$ is interpreted according to the following five semantic clauses:

- $(S_E, \bar{J}, e_n) \models p$ iff the orthogonal projections on axis e_n of region $g(\bar{J})$ and object p have a nonempty intersection (see Fig. 3):

$$\exists \bar{K}\ .\left(K_n = J_n\right) \wedge g\left(\bar{K}\right) \in F_E(p) \qquad (16)$$

- $(S_E, \bar{J}, e_n) \models \neg\phi$ iff the spatial assertion $(S_E, \bar{J}, e_n) \models \phi$ does not hold:

$$\neg\left((S_E, \bar{J}, e_n) \models \phi\right) \qquad (17)$$

- $(S_E, \bar{J}, e_n) \models \phi_1 \wedge \phi_2$ iff both $(S_E, \bar{J}, e_n) \models \phi_1$ and $(S_E, \bar{J}, e_n) \models \phi_2$ hold:

$$\left((S_E, \bar{J}, e_n) \models \phi_1\right) \wedge \left((S_E, \bar{J}, e_n) \models \phi_2\right) \qquad (18)$$

- $(S_E, \bar{J}, e_n) \models \phi_1 unt_{s+}\phi_2$ iff there exists a region $g(\bar{J}')$ which is reached from region $g(\bar{J})$ moving along the positive direction of axis e_n such that assertion ϕ_2 holds in region $g(\bar{J}')$ and ϕ_1 holds in all the regions from $g(\bar{J})$ to $g(\bar{J}')$:

$$\begin{aligned}&\exists\Delta > 0.\\ &\left(\left(S_E, \bar{J}+\Delta\cdot\bar{e}_n, e_n\right) \models \phi_2\right) \wedge \qquad (19)\\ &\left(\forall\delta \in [0, \Delta-1], \left(S_E, \bar{J}+\delta\cdot\bar{e}_n, e_n\right) \models \phi_1\right)\end{aligned}$$

where $\bar{J}+\delta\cdot\bar{e}_n$ denotes the N–tuple obtained by increasing by δ the nth component of the N–tuple $\bar{J}$;

- $(S_E, \bar{J}, e_n) \models \phi_1 unt_{s-}\phi_2$ iff there exists a region $g(\bar{J}')$ which is reached from region $g(\bar{J})$ moving along the negative direction of axis e_n such that assertion ϕ_2 holds in region $g(\bar{J}')$ and ϕ_1 holds in all the regions from $g(\bar{J})$ to $g(\bar{J}')$:

$$\begin{aligned}&\exists\Delta > 0.\\ &\left(\left(S_E, \bar{J}-\Delta\cdot\bar{e}_n, e_n\right) \models \phi_2\right) \wedge \qquad (20)\\ &\left(\forall\delta \in [0, \Delta-1], \left(S_E, \bar{J}-\delta\cdot\bar{e}_n, e_n\right) \models \phi_1\right)\end{aligned}$$

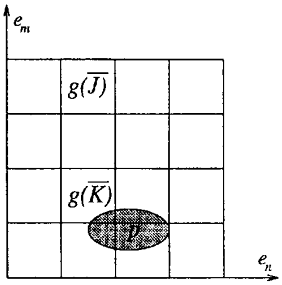


Fig. 3. The atomic proposition of Spatial Logic $(S_E, J, e_n) \models p$ expresses the existence of a nonempty intersection of the projections of an object p and a region $g(\bar{J})$ on a reference axis e_n.

Shorthands: As in Temporal Logic, the operators *spatial-eventually* ($\Diamond_{s\pm}$) and *spatial-always* ($\Box_{s\pm}$) can be derived as shorthands by combination of *spatial-until* operators and Boolean connectives:

- $(S_E, \bar{J}, e_n) \models \Diamond_{s\pm}\phi$ means that ϕ will eventually hold in some of the regions encountered moving from region $g(\bar{J})$ along axis e_n (either in the positive or in the negative direction according to the sign ±):

$$\Diamond_{s\pm}\phi := true\ unt_{s\pm}\ \phi \qquad (21)$$

- $(S_E, \bar{J}, e_n) \models \Box_{s\pm}\phi$ means that ϕ holds in all the regions encountered moving from region $g(\bar{J})$ along axis e_n:

$$\square_{s\pm}\phi := \neg(true\ unt_{s\pm}(\neg\phi)) \tag{22}$$

B.2. Object-Based Formulation

To permit the expression of spatial relationships between pairs of objects, Spatial Logic is augmented with *context declarators* which allow the expression of assertions of the form $(S_E, q, e_n) \models \phi$, q being an object, possibly standing over more than one region.

Context Declarators: In the conventional formulations of Temporal Logic, the context declarator *in* is used to deal with actions lasting over a finite temporal interval. Specifically, if *Act* is an action with a finite duration, the clause *in Act* is satisfied at any instant of time along the execution of *Act*.

This temporal semantics can be transposed in the spatial domain so as to fit in the Spatial Logic framework: if q is an object, the assertion $(S_E, q, e_n) \models \phi$ (read: "ϕ holds in q") will express that $(S_E, \bar{J}, e_n) \models \phi$ holds in any region $g(\bar{J})$ containing q:

$$\forall \bar{J} \;.\; g(\bar{J}) \in F_E(q),\ (S_E, \bar{J}, e_n) \models \phi \tag{23}$$

According to this, the assertion $(S_E, q, x) \models p$ (which will be referred to as *spatial alignment* with respect to the x axis) expresses that the projection of q on the x axis is entirely contained in the projection of p (in Fig. 4a), all the possible mutual positions of a pair of objects q and p which satisfy this assertion are shown). Besides, the assertion $(S_E, q, x) \models \Diamond_{s+} p$ expresses that every point of the projection of q on the x axis has at least one point of the projection of p to its right side (see Fig. 4b); and the assertion, $(S_E, q, x) \models q\ unt_{s+}\ p$ expresses that the projection of q extends at least until one point is found that belongs to the projection of p (see Fig. 4c).

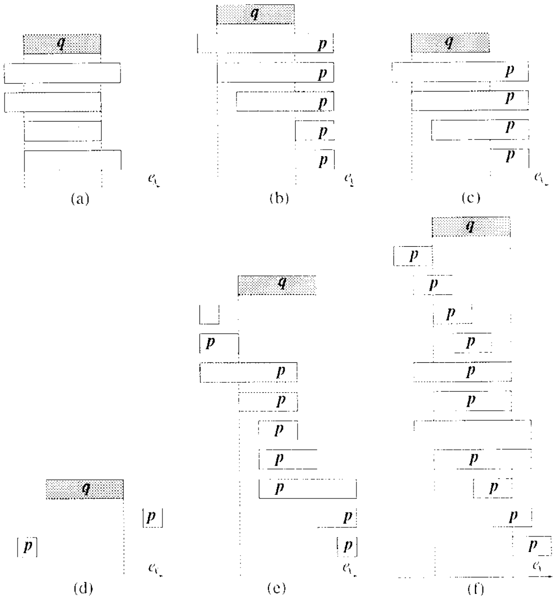


Fig. 4. The possible mutual positions of p and q which satisfy the assertions: $(S_E, q, x) \models p$ (a), $(S_E, q, x) \models \Diamond_{s+} p$ (b), $(S_E, q, x) \models q\ unt_{s+}\ p$ (c), $(S_E, q, x) \models \neg p$ (d), $\neg((S_E, q, x) \models p)$ (e), and $\neg((S_E, q, x) \models \neg p)$ (f).

It is worth noting that negation does not distribute with respect to the context declaration, i.e.,

$$(\neg\ (S_E, q, e_n) \models \phi) \neq ((S_E, q, e_n) \models \neg\phi) \tag{24}$$

For instance, the assertion $(S_E, q, x) \models \neg p$ expresses that the projection of q on the x axis does not intersect the projection of p (see Fig. 4d). Whereas, the assertion $\neg((S_E, q, x) \models p)$ expresses that part of the projection of q does not intersect the projection p on the x axis (see Fig. 4e).

By exploiting the nondistributivity of negation over context declarators, Spatial Logic assertions can be formed to express properties holding in weak contexts, i.e., assertions holding in *some* rather than in *all* the regions occupied by an object. For instance, the assertion $\neg((S_E, q, x) \models \neg p)$ means that some of the regions of q are aligned with some of the regions of p (see Fig. 4f).

Checking Spatial Formulae Against Discrete Scenes: In general, given a spatial formula ϕ and a discrete scene S_E containing an object q, the satisfaction of ϕ in the regions of S_E occupied by q can be automatically verified through a model checking algorithm which transposes in the spatial domain the same steps followed by the temporal model checking algorithm described in Section II.A. After the spatial formula ϕ has been recursively decomposed in subformulae until it has been reduced into atomic propositions of Spatial Logic, subformulae are recursively checked, in order of growing length, against the regions of the sequence. In this bottom-up checking, the labeling of each region is progressively augmented with all the subformulae that it satisfies until obtaining a final labeling which identifies every region of the scene in which ϕ is satisfied. If object identifiers are associated with sets of regions it is possible to decide whether a scene satisfies or not a spatial assertion capturing spatial relationships between appearing objects. Spatial reasoning and inferencing are effectively supported as well. Similarly to the temporal model checker, the spatial checking algorithm runs in linear time with respect to both the length of ϕ and the number of regions in the scene S_E. To reduce the complexity, adjacent regions sharing a common ordering relationship with respect to all the objects in the scene can be grouped together into macro rectangles characterized by common sets of labeling formulae.

III. Sequence Descriptions Using STL

Image sequences usually represent 3D dynamic real-world scenes with three dimensional motions of multiple objects.

To avoid ambiguities in the representation of the spatial contents of individual frames, according to [15], descriptions referring to the original 3D imaged scene (*3D scene-based* descriptions) are generally needed. *2D image-based* descriptions can be considered only in the cases where all the objects lay on a common plane and the camera is in a normal position with respect to it.

In 3D scene-based descriptions, two different descriptions are possible, depending on the reference systems on which symbolic projections are taken. On the one hand, spatial relationships between objects can be derived by considering the Cartesian coordinate system originated in a privileged point of view corresponding to the vantage point of the viewing camera (*observer-centered* description). In this case, images of the same scene, taken from different viewpoints, are associated with distinct 3D scene descriptions. On the other hand, spatial relationships can be derived by projecting each object on the coordinate systems associated with the other objects in the scene (*object-centered* description). The overall description of the scene is obtained as the composition of multiple object-centered descriptions, each capturing how one object sees the rest of the scene. Since in the object-centered approach descriptions are independent of the observer point of view, images of a scene, taken from distinct viewpoints, are all associated with the same 3D description.

Considering the evolution over time of scene-based descriptions, observer-centered representations lead to a sequence description which is dependent on the observer point of view. Object motion is correctly represented only if the camera is fixed. In the presence of a moving camera, objects may be associated with apparent motions. For instance, camera zooming results in expansive or contractive motions. As a consequence, the description associated with the sequence may include changes in spatial relationships that are not due to the actual motion of the objects. Whereas, object-centered scene descriptions always result in the representation of the actual motion of the objects, both with a fixed and with a moving camera. In this case, the sequence description does not depend on the observer point of view and only represents actual changes in the spatial relationships between objects as occurring in the original imaged scene.

The object-based formulation of Spatial Logic naturally encompasses observer-centered as well as object-centered scene representations with different dimensionalities (both 2D and 3D). For the first type of representation, the scene is described with respect to a set of privileged axes E_{obs} that are chosen to coincide with those of the observing camera. Whereas, to obtain an object-centered description, each object p will be associated with a set of reference axes E_p, and its perception of the overall scene will be described by the relative scene $\langle V_{E_p}, Obj, F_{E_p} \rangle$.

A. An Example

In Fig. 5, a synthetic scene sequence is sketched which describes the motion of a car c between a house h and a tree t. All the objects in the scene lay on a common plane.

Considering object-centered descriptions and referring to the reference system E_c associated with the car c, the spatial positions of the house h as perceived by the car c along the course of the scene can be described by the following spatial assertions which only use *spatial alignment* relationships between objects:

$$\Phi_1 := \left(S_{E_c}, c, x\right) \models h \tag{25}$$

$$\Phi_2 := \neg\left(\left(S_{E_c}, c, x\right) \models \neg h\right) \tag{26}$$

$$\Phi_3 := \left(S_{E_c}, c, y\right) \models h \tag{27}$$

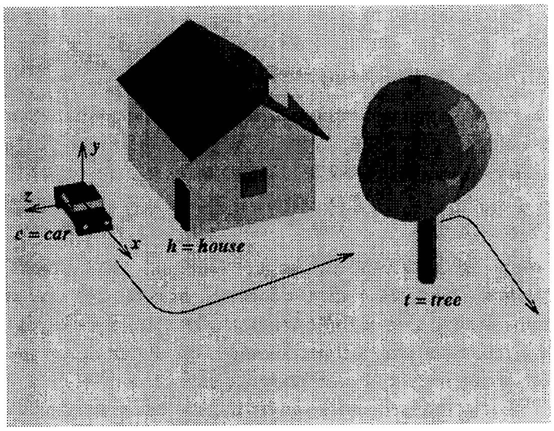


Fig. 5. The sketch of a scene sequence with a car moving between a house and a tree.

$$\Phi_4 := \left(S_{E_c}, c, z\right) \models h \tag{28}$$

$$\Phi_5 := \neg\left(\left(S_{E_c}, c, z\right) \models \neg h\right) \tag{29}$$

Assertions (25), (27), and (28) express that *all* the points of the car are aligned with points of the house with respect to the x, the y, and the z axis, respectively; (26) and (29) express that *some* of the points of the car are aligned with points of the house with respect to the x and the z axis.

By temporal combination of these spatial assertions, the evolution over time of the the position of the house h with respect to the car c can be described through the following STL assertion:

$$\begin{aligned}(\sigma, 0) \models & \left[\Phi_1 \wedge \Diamond_t(\Box_t \neg\Phi_2)\right] \\ & \wedge \left[\Box_t \Phi_3\right] \wedge \left[\neg\Phi_5 \wedge \Diamond_t(\Box_t \Phi_4)\right]\end{aligned} \tag{30}$$

The expression in the first pair of square brackets describes the dynamics of the ordering relationships between the projections on the x axis of the house and the car: initially, Φ_1 holds and eventually a frame will be reached after which Φ_2 will never hold again. The evolution of the projections of the house on the y axis is described by the expression in the second pair of square brackets which states that Φ_3 holds along the entire course of the scene. Finally, the evolution of the projections on the z axis is described by the expression in the third pair of square brackets: initially, Φ_5 does not hold and, eventually, a frame will be reached from which Φ_4 will always hold.

B. Description Refinement

The STL assertion in (30) provides a rough description of the contents of the sample sequence. A finer description can be obtained, which describes more closely the spatial contents of the individual scenes and the temporal evolution of the sequence by using more fitting operators in spatial and temporal descriptions. Temporal and spatial description refinements have an independent, orthogonal, impact on the level of detail and completeness of the resulting spatio-temporal description.

Refining the Spatial Description: Referring to the object-centered description of (30), the spatial assertions Φ_1 through Φ_5 express the spatial alignment of the car with the house with respect to the three reference axes.

The *spatial-eventually* operator can be used instead of the simple spatial alignment condition in order to define also on which side the house is with respect to the projection of the car. To this end, with the introduction of two new assertions Φ_6 and Φ_7:

$$\begin{aligned} \Phi_6 &:= \left(S_{E_c}, c, x\right) \models \Diamond_{s\_}h \\ \Phi_7 &:= \left(S_{E_c}, c, z\right) \models \Diamond_{s\_}h \end{aligned} \tag{31}$$

the assertions $\neg\Phi_2$ and $\neg\Phi_5$ can be refined as $(\neg\Phi_2) \wedge \Phi_6$ and $(\neg\Phi_5) \wedge \Phi_7$, which express that the house is *before* the car with respect to the x and z axes, respectively. Using these statements, (30) can be replaced by:

$$\begin{aligned} (\sigma, 0) \models &\left[\Phi_1 \wedge \Diamond_t \Box_t \left((\neg\Phi_2) \wedge \Phi_6\right)\right] \\ &\wedge \left[\Box\Phi_3\right] \wedge \left[\left((\neg\Phi_5) \wedge \Phi_7\right) \wedge \Diamond_t(\Box_t\Phi_4)\right] \end{aligned} \tag{32}$$

Through the joint use of Boolean connectives and spatial operators, several different assertions about relationships between object projections can be defined to allow for different levels of detail and refinement. Some of these will be expounded in Section IV with reference to the retrieval system which exploits STL as its internal representation formalism.

Refining the Temporal Description: An assertion of the form:

$$\Phi_1 \wedge \Diamond_t \Phi_2 \tag{33}$$

states that Φ_1 holds and that Φ_2 will eventually hold, thus allowing for the existence of a period of time during which neither Φ_1 or Φ_2 hold. The use of *temporal eventually* permits the avoidance of a complete specification of all the subsequent spatial relationships between two objects in a sequence, and allows the concealment of subsequences that are not essential for the description.

Trading simplicity for a finer description, spatial assertions can be composed through the *temporal-until* to prevent the occurrence of intermediate concealed relationships that are not captured by the description. Referring again to the example of Fig. 5, this refinement leads to the replacement of the assertion in (30) with the following temporal statement:

$$\begin{aligned} &(\sigma, 0) \models [\Phi_1 unt_t((\Phi_2 \wedge \neg\Phi_1) unt_t(\Phi_1 unt_t(\Phi_2 \wedge (\neg\Phi_1)) \\ &unt_t(\Phi_1 unt_t((\Phi_2 \wedge \neg\Phi_1) unt_t(\Box_t \neg\Phi_2))))))] \\ &\wedge [\Box_t\Phi_3] \wedge [(\neg\Phi_5) unt_t((\Phi_5 \wedge \neg\Phi_4) unt_t(\Box_t\Phi_4))] \end{aligned} \tag{34}$$

where the expressions in the three square brackets still refer to the three axes x, y, and z, respectively. The joint use of *temporal-eventually* and *temporal-until* operators in the temporal description of the same sequence permits to obtain descriptions focusing on selected details of specific interest.

The assertion in (34) provides an *asynchronous* description of the temporal evolution of the projections on the three axes. For instance, it does not specify whether the assertion Φ_5 (referring to the z axis) holds before, during, or after the time interval in which the assertion $\neg\Phi_2$ (referring to the x axis) holds. This ambiguity can be removed by *synchronizing* the temporal structure of different assertions. For instance, the relationships along the x, y, and z axes appearing in (34), can be combined within a single temporal formula:

$$\begin{aligned} &(\sigma, 0) \models (\Phi_1 \wedge \Phi_3 \wedge \neg\Phi_5) unt_t \\ &((\Phi_2 \wedge \Phi_3 \wedge \neg\Phi_5) unt_t((\Phi_1 \wedge \Phi_3 \wedge \neg\Phi_5) unt_t \\ &((\Phi_2 \wedge \Phi_3 \wedge \neg\Phi_5) unt_t((\Phi_1 \wedge \Phi_3 \wedge \neg\Phi_5) unt_t \\ &((\Phi_2 \wedge \Phi_3 \wedge \neg\Phi_5) unt_t((\neg\Phi_2 \wedge \Phi_3 \wedge \neg\Phi_5) unt_t \\ &((\neg\Phi_2 \wedge \Phi_3 \wedge \Phi_5) unt_t(\Box_t(\neg\Phi_2 \wedge \Phi_3 \wedge \Phi_4))))))))) \end{aligned} \tag{35}$$

which makes explicit that the time interval during which Φ_5 holds is a subset of the time interval during which Φ_2 is not satisfied.

IV. A System for Image Sequence Retrieval

STL has been exploited as the internal representation formalism in a protoype system supporting visual retrieval by contents from a database of image sequences.[1]

The database contains raw digital image sequences; in accordance with considerations expounded in Section III, each sequence is associated, at storage time, with a symbolic description capturing the ordering relationships among the objects in the 3D scenes appearing in its frames. Queries are expressed through the visual composition of sample iconic scenes. The user reproduces the temporal evolution of the spatial relationships between the objects appearing in the sequences, by arranging 3D icons in a 3D virtual space. Sample scenes are interpreted and translated by the system into STL statements, which are then checked against the descriptions in the database to accomplish the retrieval.

In the rest of this section, iconic querying and image sequence retrieval are discussed, and operation examples are described.

A. Iconic Querying

Visual Querying: Following the specification-by-example paradigm, queries are expressed in terms of sample scenes that are created through the use of a visual *Scene Editor*.

The interface of the editor is shown in Fig. 6. It features two windows (the main window in the central part of the screen and the smaller window in the top-right part of the screen) showing the sample scene under construction as perceived from two virtual cameras. To ease the construction of the sample scene, the positioning of the two cameras can be varied independently (using the command list close to the left of each window) by panning, tilting and zooming in/out. For the camera associated with the main window, flying through the sample scene is also supported.

Icons to appear in the sample scene are picked from a

[1] The system was implemented in C++ and presently runs under the AIX 3.1 operating system on an IBM RISC 6000 equipped with a 7235 Power GTO accelerator for 3D graphics support. Interfacing is based on IBM GraPhigs and X-Window-Motif.

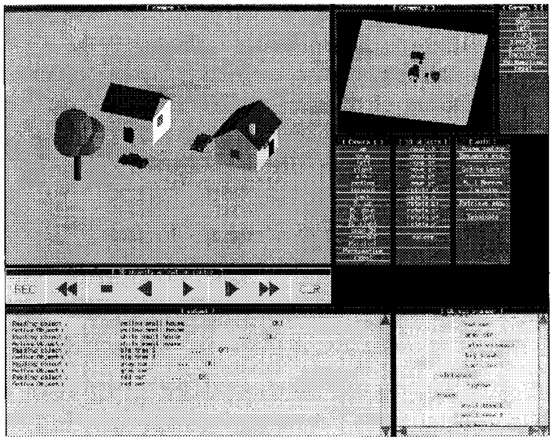


Fig. 6. The user interface of the retrieval system with 3D icons.

3D Icon Hierarchy (in the bottom right part of the screen) which collects the labels associated with all the different objects appearing in the sequence descriptions stored in the database. Selected icons are placed in the virtual space through the commands in the *3D Objects* list, and dragged by the user to reproduce the dynamics of the scene imaged in the sequences to be retrieved. The animation of multiple moving objects is defined according to a multi-track recorder metaphor: the user records the motion of one object at a time during the play-back of the animation of the objects that have been previously recorded. Synchronizations between the trajectories of multiple objects are expressed by adjusting relative speeds with which icons are dragged by the user. While not providing a fine method to define synchronization between motions, the multi-track metaphor gives the user a visual qualitative feeling of the relative speeds of the objects and allows for the approximate reproduction of real conditions without requiring an explicit knowledge of their quantitative details.

Automatic Parsing: The sample scene created through the visual Scene Editor is translated into an STL description by the *Spatio-temporal Parser.* This parser must cope with two contrasting requirements: it must limit the native flexibility of STL to obtain an univocal interpretation of a visual sample scene, but it must also permit the generation of more or less detailed descriptions to match the actual degree of prior knowledge of the user about the contents of the sequences to be retrieved.

The limitation of the expressivity of STL is obtained through the introduction of constraints on the number of objects considered in each spatial assertion and on the structure of the Boolean composition of spatial and temporal operators. Each scene description generated by the parser is made up of the conjunction of spatial assertions capturing the 3D spatial ordering relationships between every couple of objects. Specifically, for each object A, the scene description includes all the binary relationships capturing the mutual positioning of A and every other object in the scene with respect to the object-centered reference system of A itself. Relationships are expressed with reference to object-centered coordinate axes.

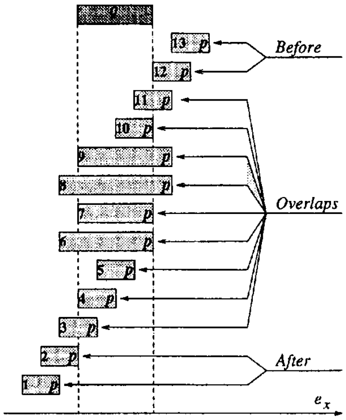


Fig. 7. Spatial relationships distinguished by *level 1* operators.

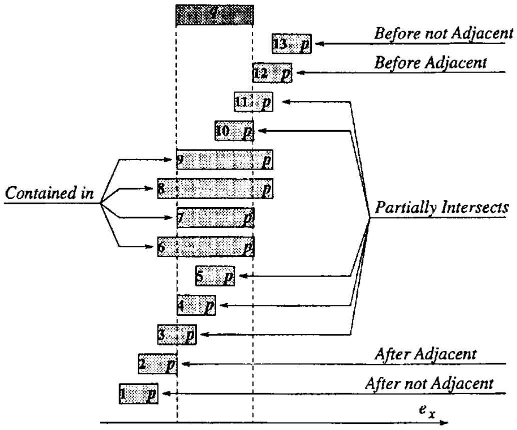


Fig. 8. Spatial relationships distinguished by *level 2* operators.

Therefore, the resulting query does not depend on the vantage point from which the querying example has been visualized but only expresses relative positions between objects and their change over time. This independenc copes with the fact that, when specifying the query, the user seldom knows the exact viewpoint from which the searched sequence has been grabbed.

In order to allow for different levels of detail in the interpretation of sample scenes, the spatial relationships between the projections of a couple of objects can be expressed using three different alternative sets of assertions. *Level 1* assertions provide the coarsest description by distinguishing only *before, after* and *overlapping* conditions (see Fig. 7). *Level 2* assertions provide a finer description by evidencing adjacency conditions and overlapping with either complete inclusion or partial intersection (see Fig. 8). Finally, *level 3* assertions provide the finest description, by distinguishing all the thirteen possible distinct mutual positions between two objects [2] (identified by the numerical labels in Figs. 7 and 8). The STL expressions for the assertions of each level, with reference to the x axis, are reported in Fig. 9. It is worth noting that the three levels of description are related through a specialization hierarchy by which a set of assertions of level 3 corresponds to a unique assertion of level 2, and a set of assertions of level 2 to a unique assertion of level 1.

Level 1

$$\begin{aligned}
&q\ \mathit{Before}\ p && := (S_E, q, x) \models (\neg\Diamond_{s-}p)\\
&q\ \mathit{Overlaps}\ p && := \neg((S_E, q, x) \models \neg(\Diamond_{s+}p \wedge \Diamond_{s-}p))\\
&q\ \mathit{After}\ p && := (S_E, q, x) \models (\neg\Diamond_{s+}p)
\end{aligned}$$

Level 2

$$\begin{aligned}
&q\ \mathit{Before\_Not\_Adjacent}\ p :=\\
&\qquad (q\ \mathit{Before}\ p) \wedge ((S_E, q, x) \models \neg p)\\
&q\ \mathit{Before\_Adjacent}\ p :=\\
&\qquad (q\ \mathit{Before}\ p) \wedge \neg((S_E, q, x) \models \neg p)\\
&q\ \mathit{Contained\_in}\ p :=\\
&\qquad (q\ \mathit{Overlaps}\ p) \wedge ((S_E, q, x) \models p)\\
&q\ \mathit{Partially\_Intersects}\ p :=\\
&\qquad (q\ \mathit{Overlaps}\ p) \wedge \neg((S_E, q, x) \models p)\\
&q\ \mathit{After\_Adjacent}\ p :=\\
&\qquad (q\ \mathit{After}\ p) \wedge \neg(S_E, q, x) \models \neg p)\\
&q\ \mathit{After\_Not\_Adjacent}\ p :=\\
&\qquad (q\ \mathit{After}\ p) \wedge ((S_E, q, x) \models \neg p)
\end{aligned}$$

Level 3

$$\begin{aligned}
&\mathit{Case}13 := q\ \mathit{Before\_Not\_Adjacent}\ p\\
&\mathit{Case}12 := q\ \mathit{Before\_Adjacent}\ p\\
&\mathit{Case}11 := (q\ \mathit{Partially\_Intersects}\ p) \wedge\\
&\qquad ((S_E, q, x) \models \Diamond_{s+}p)\\
&\mathit{Case}10 := (q\ \mathit{Partially\_Intersects}\ p) \wedge\\
&\qquad \neg((S_E, q, x) \models \neg(p\ \mathit{unt}_{s+}(\neg p \wedge \neq q)))\\
&\mathit{Case}\ 9 := (q\ \mathit{Contained\_in}\ p) \wedge\\
&\qquad ((S_E, q, x) \models p\ \mathit{unt}_{s-}(\neg q \wedge \neg p)) \wedge\\
&\qquad \neg((S_E, q, x) \models p\ \mathit{unt}_{s+}(\neg q \wedge \neg p))\\
&\mathit{Case}\ 8 := (q\ \mathit{Contained\_in}\ p) \wedge\\
&\qquad \neg((S_E, q, x) \models p\ \mathit{unt}_{s-}(\neg q \wedge \neg p)) \wedge\\
&\qquad \neg((S_E, q, x) \models p\ \mathit{unt}_{s+}(\neg q \wedge \neg p))\\
&\mathit{Case}\ 7 := (q\ \mathit{Contained\_in}\ p) \wedge\\
&\qquad ((S_E, q, x) \models p\ \mathit{unt}_{s-}(\neg q \wedge \neg p)) \wedge\\
&\qquad ((S_E, q, x) \models p\ \mathit{unt}_{s+}(\neg q \wedge \neg p))\\
&\mathit{Case}\ 6 := (q\ \mathit{Contained\_in}\ p) \wedge\\
&\qquad \neg((S_E, q, x) \models p\ \mathit{unt}_{s-}(\neg q \wedge \neg p)) \wedge\\
&\qquad ((S_E, q, x) \models p\ \mathit{unt}_{s+}(\neg q \wedge \neg p))\\
&\mathit{Case}\ 5 := (q\ \mathit{Partially\_Interesects}\ p) \wedge\\
&\qquad ((S_E, q, x) \models \neg(p\ \mathit{unt}_{s+}\neg q) \wedge \neg(p\ \mathit{unt}_{s-}\neg q)))\\
&\mathit{Case}\ 4 := (q\ \mathit{Partially\_Intersects}\ p) \wedge\\
&\qquad \neg((S_E, q, x) \models \neg(p\ \mathit{unt}_{s-}(\neg q \wedge \neg p)))\\
&\mathit{Case}\ 3 := (q\ \mathit{Partially\_Intersects}\ p) \wedge\\
&\qquad ((S_E, q, x) \models \Diamond_{s-}p)\\
&\mathit{Case}\ 2 := q\ \mathit{After\_Adjacent}\ p\\
&\mathit{Case}\ 1 := q\ \mathit{After\_Not\_Adjacent}\ p
\end{aligned}$$

Fig. 9. STL formal expressions of the spatial operations for the three levels of description.

In the description of the temporal evolution of the contents of the sample scene, the two composition structures of Section III.B are supported to permit either the explicit representation of all the subsequent conditions in the sequence (using the *temporal until* operator) or the concealment of intermediate subsequences (using the *temporal eventually* operator). Spatial assertions referring to the axes x, y, and z are collected as a single assertion within the temporal assertion to provide a synchronous description for the evolution of objects projections along the three axes.

The three levels of spatial assertions and the two temporal composition structures result into a total number of six alternative fragments of STL that can be selected by the user at querying time to direct the parsing to match his actual knowledge about the contents of sequences in the database.

The parsing is accomplished by the system in two subsequent phases. First, for every frame, the parser asks each object to evaluate its position with respect to the other objects in the scene according to its own reference system. A binary spatial assertion is derived for each couple of objects and for each of the three axes of the first object. Afterwards, subsequent frames are collected into states sharing a common spatial description, and the spatial assertions referring to the individual states are collected to form an assertion capturing the temporal evolution of the spatial relationships between the objects.

B. Retrieval from Database

In the retrieval phase, the query specification derived from the interpretation of the querying example sequence is checked against sequence descriptions stored in the database.

Sequence Representation: The descriptions associated with the sequences in the database must include a sufficient information base to permit the checking of the sequence contents against every possible assertion generated by the Spatio-temporal Parser. To this end, it is sufficient that descriptions in the database define the object-centered spatial relationships between any couple of objects in every frame of the sequence, and that these are expressed at the finest level of detail, i.e., using the spatial assertions of *level 3* and the temporal *until* operator. In the system, one such description is created manually, when the sequence is stored in the database, with the assistance of the *Sequence Recorder* and the *Spatio-temporal Parser.* Using the Sequence Recorder, the system operator reproduces the contents of the sequence in a virtual iconic scene which is then interpreted by the Spatio-temporal Parser to derive the symbolic description stored in the database.

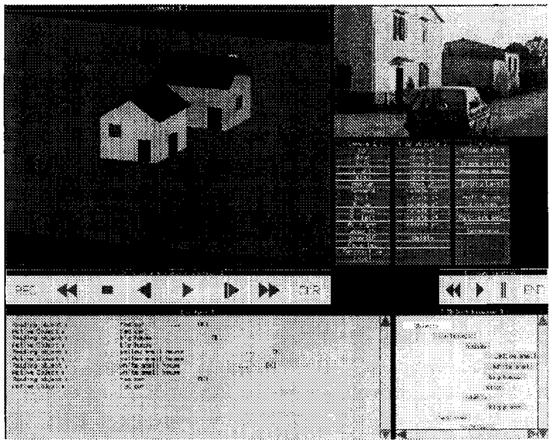


Fig. 10. The user interface for the iconic reproduction of sequence contents.

The visual interface of the Sequence Recorder is shown in Fig. 10. In the left window, the system plays the image sequence to be described, while, in the right window, the user creates a 3D iconic reproduction of the sequence contents that are considered of interest. To this end, after a preliminary inspection of the sequence, the user selects the icons that represent objects of interest and places them in their initial positions. Hence, he concentrates on the left window to inspect the spatial relationships between objects over time and reproduces them in the right window by dragging icons as described for the Visual Scene Editor used in the querying stage. Buttons are provided to advance, stop, rewind and play back the image sequence, and to permit the user to focus on relevant details of complex dynamics.

Through the same steps followed in the automatic parsing of visual queries, the iconic sequence created by the operator is interpreted by the system and translated into an object-centered STL description at the finest level of interpretation detail.

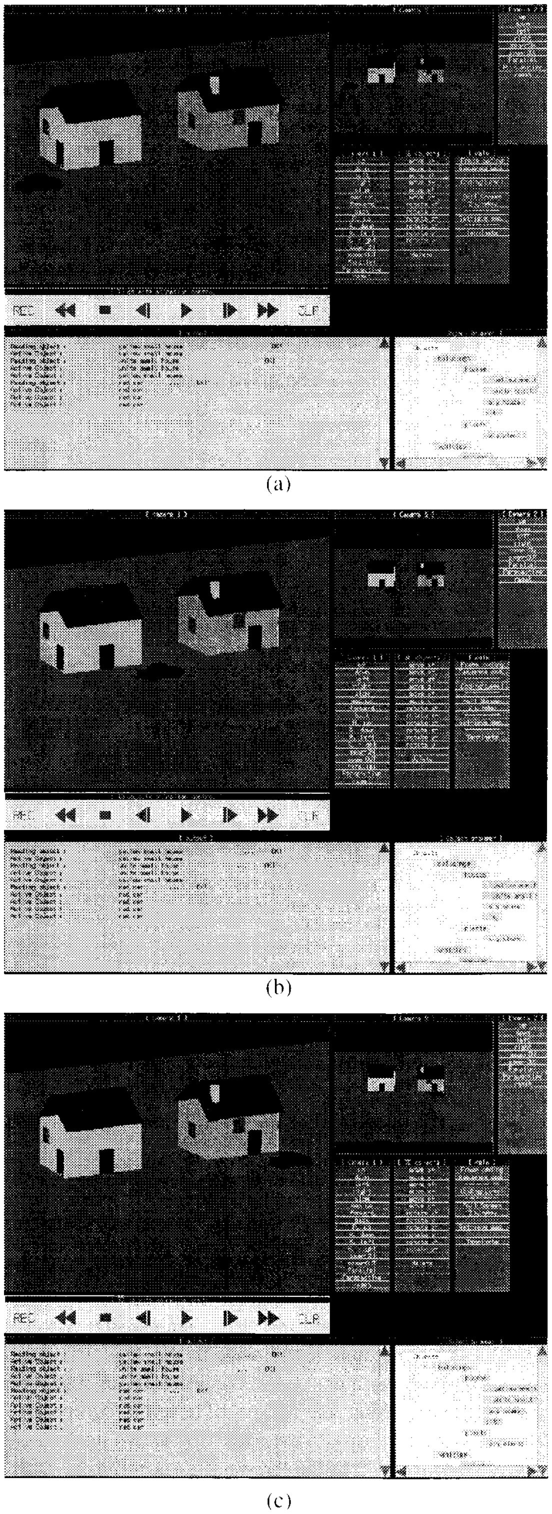
In this representation, the description of the spatial contents of a 3D scene with N interesting objects is made up of $3 \times N \times (N-1)$ relationships, each corresponding to one out of the 13 conditions considered at level 3. Thus, for a sequence with T states, each corresponding to a three-dimensional scene with N objects, the space-complexity of the description is equal to $3 \times N \times (N-1) \times T$.

Note that, since subsequent frames with equivalent spatial descriptions are grouped into states, the resulting description is independent of the actual duration of the sequence stored and of the dwelling time between any two subsequent states. Besides, since it is object-centered, it is also independent of the vantage point from which the original scene has been imaged in the sequence.

B. Sequence Retrieval: The matching of the query specification against the description of a sequence in the database is accomplished through two subsequent steps.

In the first step, a spatial checker identifies the states of the database description which satisfy the spatial relationships occurring in the states of the query specification. This could be accomplished through the general model checking algorithm mentioned in Section II.B.2. However, to exploit the simplification deriving from the limitations assumed in the automatic generation of spatial assertions by the Spatio-Temporal Parser, spatial checking is carried out through a one-to-one matching of the binary relationhips in the query against those of the sequence description. Specifically, a state s_q in the query corresponds with a state s_d of the sequence description if and only if every binary relationship of s_q is equal to or is a specialization of any of the relationships contained in s_d.

In the second step, a temporal checker, implementing the Clarke's algorithm described in Section II.A, identifies in which states of the database description there exists a matching with the temporal evolution of the spatial contents of the querying specification example. In this procedure, matching is decided with time-complexity linear in both the number of states of the query and the number of states in the sequence description. Search effort is reduced by avoiding the complete scanning of all the scenes sequences stored in the database through the use of indexes capturing the types of the objects appearing in the sequences.



(a)

(b)

(c)

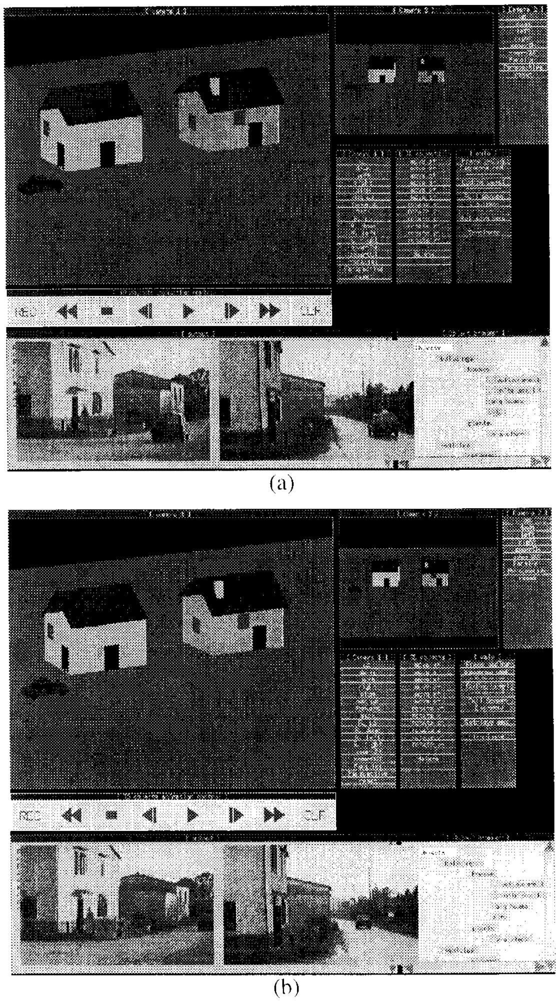
Fig. 11. Visual query by example. Specification of a simple dynamic scence: a) the initial scene, b) and c) specification of motion by the dragging of the car icon.

C. Examples of Operation

In Fig. 11, an example of operation of the system is reported. In the example, image sequences are queried which represent scenes where a car is moving in a straight direction near a pair of houses. After the icons representing the objects have been selected and placed in the appropriate positions, the car trajectory is drawn in the plane where motion takes place. In this example, spatial and temporal relationships in the query scene are encoded at the coarsest level of spatial detail (using the spatial operators of *level 1*) in conjunction with the finest level of temporal composition (using the *temporal until* operator).

The result of the retrieval is shown in Fig. 12, where two sequences are displayed in two separate windows. The contents of both the sequences correspond to the example scene created by the user, even if they are taken from different vantage points. Both are retrieved since descriptions used are object-centered and thus independent of the camera viewpoint.

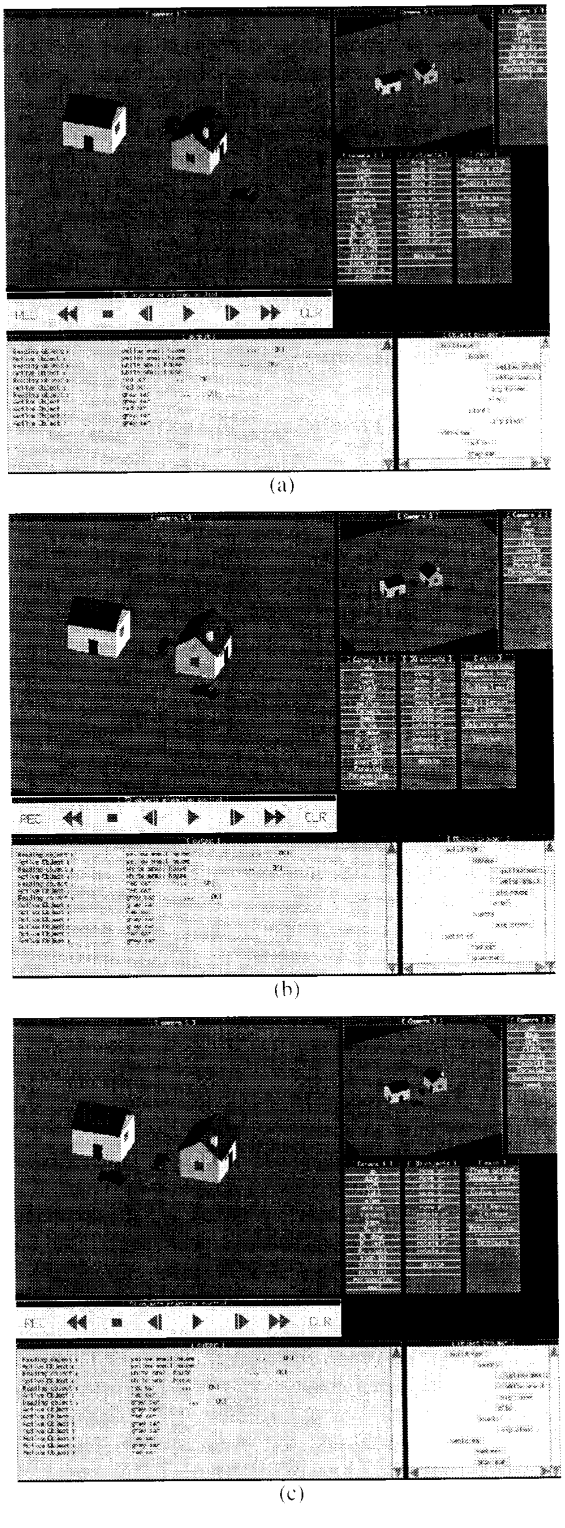
In Fig. 13, a more complex example involving synchroniza-



(a)

(b)

Fig. 12. Result of the retrieval for the scene of Fig. 11 (two frames of the sequence).



(a)

(b)

(c)

Fig. 13. Visual query by example. Specification of a more complex dynamic scene: a), b), and c) definition of the motion of the dark gray car during the play-back of the motion of the light gray car.

tion between two moving objects in a problem of crossroad traffic surveillance is shown. In this case, sequences are queried where a car has not given the right of way to another car coming from its right side. In the specification of the query, the trajectory of one car (the light gray car) is recorded first. Hence, it is rewound and played-back during the specification of the trajectory of the second car (the dark gray car). The two cars are made to come to a crossroad, at approximately the same time, and the dark gray car is made to pass before the light gray car without giving it the right of way. The query is expressed using the coarsest level in spatial relationships and the finest level in temporal relationships. The result of retrieval is shown in Fig. 14, where a single matched sequence is visualized.

V. Conclusions and Future Work

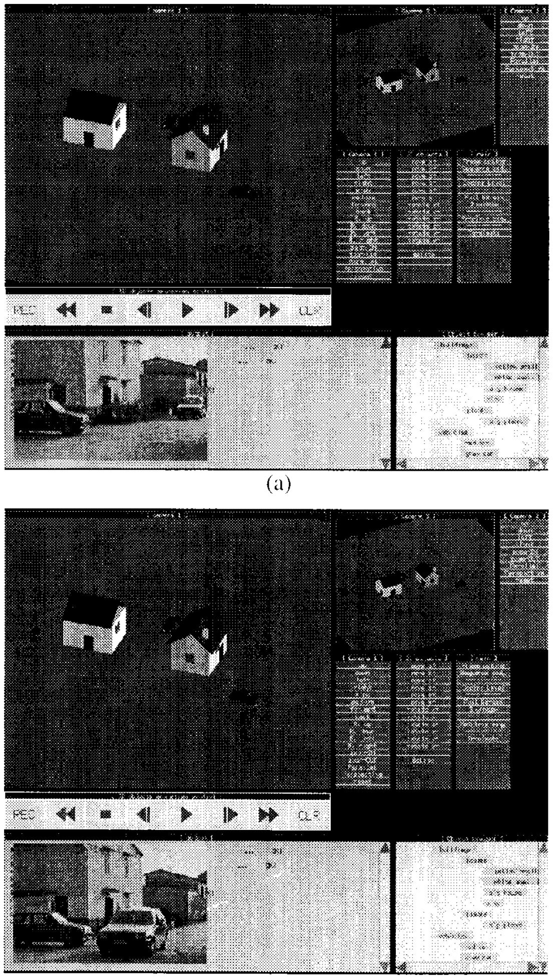
Spatio-Temporal Logic is a description language based on Temporal Logic and Symbolic Projection providing a framework for the qualitative representation of the contents of image sequences. It allows for the treatment and operation of content structures at a higher level than pixels or image features and the focusing on the actual dynamics of imaged objects. With respect to other description techniques based on Symbolic Projection, STL permits a cohesive treatment of both spatial and temporal conditions. Moreover, by permitting the combined use of spatial, temporal and Boolean operators, STL provides a wide flexibility and expressivity as needed for the description of concrete image sequences and for the specification of queries by content. Fragments of STL have been exploited as the internal representation formalism of a retrieval system supporting visual querying by example of image sequences. The system allows the expression of queries at different levels of spatial and temporal detail, thus permitting to match the actual user's knowledge about the contents of sequences to be retrieved.

Developments of this research have been started in several distinct directions including automatic derivation of STL descriptions from image data, extensions of STL expressivity and use of STL for descriptions of video clips within a system for video retrieval by content.

For the automatic derivation of STL descriptions from image sequences, the current state of image processing and machine vision techniques does not permit a full interpretation and understanding of the contents of digital images. We developed a prototype system which reconstructs 3D moving objects and their mutual spatial relationships from optical flow fields [1] of a sequence of monocular images [5]. In practice, given one image sequence, information about 3D imaged objects which is extracted automatically is incomplete and must be integrated manually by the operator. The completeness of reconstructions of objects and their spatial relationships strongly depends on the kind of sequence at hand.

Extensions of STL with metric qualifiers to take into account distances, speeds and the like, allow more fitting assertions on visual data and support more precise descriptions of space and time. An extended version of STL referred to as XSTL (eXtended Spatio-Temporal Logic) has been developed by the authors and expounded in [16]. It has been applied to symbolically describe spatio-temporal relationships within virtual worlds where virtual agents exhibit autonomous behaviors based on spatio-temporal reasoning.

Finally, research is ongoing to integrate STL-based sequence descriptions and querying with video segmentation according to cuts and scenes originated by film editing operations, to support full video indexing and retrieval by content.



(a)

(b)

Fig. 14. Result of the retrieval for the scene of Fig. 13 (two frames of the sequence).

ACKNOWLEDGMENTS

This work was partially supported by MURST 40% project *Sviluppo di una workstation multimediale ad architettura parallela.*

REFERENCES

[1] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7, no. 4, pp. 525-542, July 1985.

[2] J.F. Allen, "Mantaining knowledge about temporal intervals," *Comm. ACM*, vol. 26, no. 11, pp. 832-843, Nov. 1983.

[3] R. Alur and T.A. Henzinger, "Logics and models of real time: A survey," Tech. Rep. No. 92-1262, Dept. of Computer Science, Cornell Univ., Ithaca, N.Y., 1992.

[4] T. Arndt and S.K. Chang, "Image sequence compression by iconic indexing," *IEEE VL '89 Workshop on Visual Languages*, pp. 177-182, Roma, Italy, Sept. 1989.

[5] A. Barone, "Derivazione di relazioni simboliche 3D da viste monoculari," doctoral thesis (in Italian), A. Del Bimbo and G. Bucci, advisors, Tech. Rep. No. 15-94, Dip. Sistemi e Informatica Univ. Firenze, Florence, Italy, 1994.

[6] E. Binaghi, I. Gagliardi, and R. Schettini, "Indexing and fuzzy logic-based retrieval of color images," *IFIP Trans. A-7, Visual Database Systems II*, Knuth, Wegner, eds., pp. 79-92, Elsevier, 1992.

[7] S.K. Chang and S.H. Liu, "Picture indexing and abstraction techniques for pictorial databases," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 4, pp. 475-484, July 1984.

[8] S.K. Chang, Q.Y. Shi, and C.W. Yan, "Iconic indexing by 2D strings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 3, pp. 413-427, July 1987.

[9] S.K. Chang, C.W. Yan, D.C. Dimitroff, and T. Arndt, "An intelligent image database system," *IEEE Trans. Software Engineering*, vol. 14, no. 5, pp. 681-688, May 1988.

[10] S.K. Chang, T.Y. Hou, and A. Hsu, "Smart image design for large image databases," *J. Visual Languages and Computing*, vol. 3, no. 4, Dec. 1992.

[11] S.K. Chang and E. Jungert, "Pictorial data management based upon the theory of symbolic projections," *J. Visual Languages and Computing*, vol. 2, no. 2, pp. 195-215, June 1991.

[12] E.M. Clarke, E.A. Emerson, and A.P. Sistla, "Automatic verification of finite-state concurrent systems using temporal logic specifications," *ACM Trans. Programming Languages and Systems*, vol. 8, no. 2, pp. 244-263, Apr. 1986.

[13] G. Costagliola, G. Tortora, and T. Arndt, "A unifying approach to iconic indexing for 2D and 3D scenes," *IEEE Trans. Knowledge and Data Engineering*, vol. 4, no. 3, pp. 205-221, June 1992.

[14] M. Davis, "Media dtreams, an iconic visual language for video annotation," *Telektronik*, no. 4, pp.5 9-71, 1993 (also appeared in reduced version in *Proc. IEEE VL'93 Workshop on Visual Languages*, Bergen, Norway, Aug. 1993).

[15] A. Del Bimbo, M. Campanai, and P. Nesi, "A three-dimensional iconic environment for image database querying," *IEEE Trans. Software Engineering*, vol. 19, no. 10, pp. 997-1011, Oct. 1993.

[16] A. Del Bimbo and E. Vicario, "A logical framework for spatio-temporal indexing of image sequences," *Proc. Workshop on Spatial Reasoning*, Bergen, Norway, Aug. 1993, also to appear in *Spatial Reasoning*, S.K. Chang, E. Jungert, eds., Plenum Press.

[17] A. Del Bimbo, P. Pala, and S. Santini, "Visual image retrieval by elastic deformation of object shapes," *Proc. IEEE VL '94, Int'l Symp. Visual Languages*, pp. 216-223, St. Louis, Mo., Oct. 1994.

[18] E. Jungert, "The observer's point of view, an extension of symbolic projections," *Proc. Int'l Conf. Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Pisa, Italy, Sept. 1992, Lecture Notes in Computer Science, pp. 179-195, Springer Verlag, 1992.

[19] A. Hampapur, R. Jain, and T. Weymouth, "Digital video indexing in multimedia systems," *Proc. AAAI '94 Workshop Indexing and Reuse in Multimedia*, pp. 187-198, Seattle, Wash., Aug. 1994.

[20] K. Hirata and T. Kato, "Query by visual example: Content-based image retrieval," *Advances in Database Technology—EDBT '92*, A. Pirotte, C. Delobel, G. Gottlob, eds., Lecture Notes on Computer Science, vol. 580, pp. 56-71, Springer Verlag, Berlin, 1992.

[21] S. Lee, M.K. Shan, and W.P. Yang, "Similarity retrieval of iconic image fatabase," *Pattern Recognition*, vol. 22, no. 6, pp. 675-682, 1989.

[22] Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems*. New York: Springer Verlag, 1992.

[23] L. Mohan and R.L. Kashyap, "An object-oriented knowledge representation for spatial information," *IEEE Trans. Software Engineering*, vol. 14, no. 5, pp. 675-681, May 1988.

[24] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full video search for object appearances," *IFIP Trans. Visual Database Systems II*, Knuth, Wegner, eds. , pp. 113-127, Elsevier, 1992.

[25] W. Niblack et al., "The QBIC project: Querying images by content using color, texture, and shape," Res. Report 9203, IBM Res. Div. Almaden Res. Center, Feb. 1993.

[26] M.J. Swain and D.H. Gallard, "Color indexing," *Int'l J. Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.

[27] S.L. Tanimoto, "An iconic/symbolic data structuring scheme," *Pattern Recognition and Artificial Intelligence*, C.H. Chen, ed., New York: Academic, 1976.

[28] I.M. Walter, R. Sturm, and P.C. Lockemann, "A semantic network based deductive database system for image sequence evaluation," *IFIP Trans. Visual Database Systems II*, Knuth, Wegner, eds., pp. 251-276, Elsevier, 1992.

[29] H.J. Zhang, A. Kankanhalli, and S.W. Smoliar, "Automatic partitioning of video," *Multimedia Systems*, vol. 1, no. 1, pp. 62-75, 1993.

Alberto Del Bimbo received the doctoral degree in electronic engineering from the University of Florence, Italy, in 1977. He was with IBM Italia from 1978 to 1988. He is now a full professor of computer systems at the University of Brescia and the University of Florence. Dr. Del Bimbo is a member of IEEE and of the International Association for Pattern Recognition (IAPR). He is on the board of the IAPR Technical Committee No. 8 (Industrial Applications) and is vice president of the IAPR Italian Chapter. He serves as associate editor of *Pattern Recognition* and of the *Journal of Visual Languages and Computing*. His research interests and activities are in the field of image analysis, image databases, visual languages, and virtual reality.

Enrico Vicario received the doctoral degree in electronic engineering and the PhD in computer engineering from the University of Florence, Italy, in 1990 and 1994, respectively. He is currently a researcher with Departimento di Sistemi e Informatica at the University of Florence. His research activities are in the field of software engineering, with a particular interest in visual formalisms, specification languages, and validation techniques for time-dependent systems.

Daniele Zingoni received the doctoral degree in electronic engineering from the University of Florence, Italy, in 1992. He is currently with Departimento di Sistemi e Informatica at the University of Florence, under a research grant. His research interests are in the fields of visual languages and virtual reality.