Appl Intell (2009) 31: 161–179
DOI 10.1007/s10489-008-0120-4

Logic-based interpretation of geometrically observable changes occurring in dynamic scenes

M.V. dos Santos · R.C. de Brito · H.-H. Park · P. Santos

Published online: 23 February 2008

Abstract The work presented here is about employing a theory of updates to study geometrically observable changes that occur in spatial information about image sequences of a dynamic scene. The logical framework consists of a formalism for specifying the geometrical content of a scene, as well as the changes that occur in this geometry, and an algorithm for constructing a description for such changes from logical deductions. In this approach, a database state represents the available sensor data at a particular time instant. Transitions in sensor data are modeled by changes in the database and interpreted based on axioms encoding commonsense spatial reasoning. The main contribution of this work is that it provides the theoretical foundations for symbolically interpreting long sequences of sensor data transitions. For testing the framework and its implementation, the problem of interpreting rotational movements of objects in a sequence of images was used. Our experiments show that the system correctly interprets rotational movements for objects of different colors and provides satisfactory results for interpreting such movements from perceptually indistinguishable objects.

M.V. dos Santos (✉)
Ryerson University, 350 Victoria Street, Toronto, Ontario M5B 2K3, Canada
e-mail: m3santos@ryerson.ca

R.C. de Brito · P. Santos
Centro Universitario da FEI, Sao Paulo, Brazil

R.C. de Brito
e-mail: rod.coura@uol.com.br

P. Santos
e-mail: psantos@fei.edu.br

H.-H. Park
Chung-Ang University, Seoul, South Korea
e-mail: hohyun@cau.ac.kr

Keywords Logic · Knowledge representation · Qualitative spatial reasoning · Machine vision · Reasoning about actions and change

1 Introduction

Although people are able to interpret image sequences of a dynamic scene effortlessly, this is still a challenging task for artificial vision systems. One reason is that low-level vision tasks, such as segmentation, object recognition, categorization, and 3D analysis are still performed unsatisfactorily. The other reason is that we do not have high-level knowledge structures and reasoning mechanisms for enabling far-reaching interpretations, *i.e.*, interpretations that may include propositions about parts of a scene not yet seen. In this paper we focus on the task of extending a computer's "understanding" of a scene beyond single-object recognition. Our problem is then to provide a logical account which allows for the interpretation of low level predicates denoting visual information about image sequences of a dynamic scene (scene for short) in terms of high level predicates representing commonsense concepts about space.

Our approach is inspired by some standard techniques from the fields of qualitative spatial reasoning, logic-based image understanding, and image sequence interpretation. In the following, we provide an overview of the basic literature of each of these fields.

1.1 Qualitative spatial reasoning (QSR)

Qualitative spatial reasoning (QSR) aims at the logical formalization of space from elementary entities such as regions [1], line segments [2, 3], directions [4, 5] amongst others [6,

7]. The purpose of this field is to provide clear representations and efficient automated reasoning methods for handling commonsense knowledge about space.

The approach to the representation of space and sensor transitions presented in this paper has been inspired by the QSR theory called *region connection calculus* (RCC), and also by its extension: the *region occlusion calculus* (ROC). RCC [1, 8, 9] is a many-sorted first-order axiomatization of spatial relations based on a (reflexive and symmetric) dyadic primitive relation of *connectivity* ($C/2$) between two regions. Informally, assuming two regions x and y, the relation $C(x, y)$ (read as *"x is connected with y"*) is true if and only if the closures of x and y have a point in common. From $C/2$ other basic relations can be defined, such as relations representing when two objects are disconnected, equal, partially overlapping each other, externally connected to each other, and relations for different kinds of tangential overlaps.

RCC has been applied in a variety of domains [10–15]. A rigorous analysis of the computation complexity (including a description of tractable subclasses) of RCC is presented in [16].

Due to its versatility in representing various niches of spatial knowledge, allied to its emphasis on spatial regions, RCC provides some useful insights on expressing high-level knowledge about image sequences, since objects are picked out as spatial regions in images and are usually engaged in complex spatial arrangements. As we shall see later in this paper, the basic part of the spatial reasoning theory developed in this paper assumes three RCC-style relations based on simple measurements executed on snapshots of the world.

The RCC, however, is independent of any observer's viewpoint. In contrast, the *lines-of-sight* calculus [17] takes the observer into account in order to define object interposition. Likewise, the *Region Occlusion Calculus* (*ROC*) [18] uses RCC to represent object occlusion. However, occlusion in ROC is a static concept, defined on a fine-grain level of description, which makes it hard to be applied on noisy vision data. In contrast the present paper follows the dynamical definition of interposition, defined in terms of qualitative changes in the sensor data as proposed in [19].

A solid theoretical and a deep philosophical investigation of qualitative spatial change is described in [20, 21], whereby a theory of movement is proposed that combines a theory of time, a theory of space, a theory of objects and a theory of position. A key point in that work is the construction of spatial theories from the concept of *dominance*, whereby the extreme points of subsequent time intervals where fluents hold obey a priority criteria. That framework also allows for the combination of sequences of spatial events to define movements and their occurrence conditions. The implications of assuming a theory of dominance in the framework described in the present paper is an interesting issue for future investigations.

The problem of characterizing complex object behavior in space-time has also been tackled by research on spatiotemporal databases [22, 23]. In [22], Erwig and Schneider define spatiotemporal predicates from a combination of pointset topology and temporal logic in order to cope with the integration of various kinds of data-sets into spatial databases. They propose a set of canonical spatiotemporal predicates that are lifted from purely spatial predicates using a temporal function. These canonical predicates are combined in order to build more complex structures (called *developments*). In [23] Erwig proposes a set of control structures (*combinators* in his terminology) to rule the construction of patterns of spatiotemporal predicates.

In a similar way, the work reported in this paper uses the spatiotemporal predicates proposed in [24] to build more complex patterns in order to describe sequences of images. The procedure for constructing these patterns is ruled by a general logic for state change (described in Sect. 3).

1.2 Logic-based image understanding

A rigorous logical account of image depiction was first proposed by Reiter and Mackworth [25]. Their approach is based on three sets of axioms that constrain the image interpretation process. Therefore, image interpretation is executed as a constraint satisfaction procedure.

The SIGMA system [26] uses the ideas set forth on the Reiter-Mackworth approach to, allied with an hypothesis-based reasoning [27], generate abductive explanations for aerial images. Some of the properties of the language underlying this system were explored in [28], which has been recently revisited in [29].

However, the origins of our work lies on the logic-based sensor data interpretation framework proposed in [30], where an attempt is made to supply a logical account of the transition from a robot's raw sensor data to symbols denoting the existence, location and shapes of objects. This earlier work, however, assumes the interpretation of static scenes described in an absolute frame of reference. On the other hand, the work presented in [19, 24, 31] assumes that the changes in a dynamic world, represented from the viewpoint of an observer, form the central element for sensor data interpretation. These approaches, however, fall short of interpreting sequences which include more than two snapshots. A solution to this issue is proposed in the present paper.

Following similar precepts, [32] presents a system for assimilating scenes using spatiotemporal histories (*i.e.*, regions of space-time representing the temporal development of topological relations amongst objects). This model is based on earlier approaches for automatically building event

models from visual input [33]. Also based on histories, the work proposed in [34] evaluates multiple possibilities of histories for explaining a given video sequence, electing (by a voting criteria) the most consistent one as the interpretation of the sequence.

In contrast to space-time regions representing the temporal evolution of the scene via histories, the present paper employs a logic of action and state change whose path-semantics elicits the changes that occur in spatial objects in a dynamic scene. Explicit use of logic semantics is usually missing from the research on image sequence evaluation, whose literature we overview below.

1.3 Image sequence evaluation

Systems for interpreting image sequences by high-level concepts date back to the late seventies [35]. Since then, a large number of approaches have been proposed. Some classical examples are: the ALVEN system [36, 37] for the abstraction of motion concepts from sequences of images from the human heart; and the VITRA system [38, 39] whose purpose is to link image sequence evaluation and natural language processing. Understanding sequences of traffic scenes based on optical flow has been investigated by [40–42] and [43]. Other related approaches are surveyed in [44] and [45].

In particular, [46] proposes a solution to tracking vehicles under occlusion from a static viewpoint which has many characteristics in common with our solution for object interposition. In that paper, the authors propose that, in order to track vehicles under interposition, contextual knowledge about occlusion is needed. A set of *occlusion predicates* is introduced, along with transition predicates constituting this contextual knowledge. A scene where there is occlusion between two vehicles is thus interpreted by comparing its *temporal development* to a transition diagram representing the possible changes in situations involving interposition.

Some other approaches for image understanding are characterized by the use of physical features of the observed scenes. One such system was proposed in [47], whose purpose was to recover the *causal structure* of scene elements, i.e., how they interact and respond to forces. Physical causality is obtained by analyzing the connectivity and free space between scene elements. Most of this research has been applied to the explanation of complex machines [47, 48] and to the problem of understanding images of object manipulation [49].

Closer to the framework developed in this paper, [50–52] propose a system for describing visually observed motion events from sequences of video images. Their scene interpretation process is based on notions of *support*, *contact* and *attachment* between scene objects; the system uses these notions to segment sequences of snapshots of the world into distinct events, such as *dropping*, *throwing*, *picking up*, and *putting down*. Four naive physical constraints, inspired by [53], are assumed in the interpretation process: the *substantiality constraint*, which states that objects cannot pass through each other, the *continuity constraint* stating that any change in the location of an object is due to a continuous motion, the *gravity constraint* that states that unsupported objects fall, and the *ground plane constraint* which states that the ground supports every object.

Similarly, the present work explicitly assumes the substantiality and the continuity constraints, as well as a similar use of motion verbs. In contrast to our framework, the system proposed in [50] assumes that scene objects are always visible, change in their size does not occur, and they never appear or disappear from the scene. Some of these restrictions are relaxed in the present research.

A framework for understanding dynamic scenes based on conceptual spaces [54] is proposed in [55]. That approach assumes the task of sensor interpretation in terms of three processes: on one level, a *sub-conceptual area* concerned with processing data from the sensors generates a description of the observed dynamic scene that is further input into the *conceptual level*, whose task is to generate a high-level description of the scene. This high-level description adopts the logic-based language proposed in [56], whose definitions constitute the third module in the system: the *linguistic area*. The framework developed in the present work resembles the formalism constituting this linguistic area. The system described in [55], however, assumes that the observer is static and that the objects in the environment are never occluded. The latter constraint is relaxed here.

1.4 This work in outline

In this work we employ a theory of updates to study changes that occurred in the spatial information contained in image sequences of a dynamic scene (scene, for short). A scene is then a chronological sequence of snapshots of the environment taken by a static camera. Changes that occurred in the environment are represented by differences between image regions in consecutive camera snapshots. In practice, a database state represents the available sensor data at a particular time instant. Transitions in sensor data are modeled by changes in the database and interpreted based on axioms encoding commonsense spatial reasoning.

To decrease the complexity of the spatial representation, thus eliciting the state changes underlying the reasoning process, the logical framework presented here includes a minimal spatial theory. Specifically, the logical framework consists of a formalism for specifying the geometrical content of a scene described by two functions: one that gives the distance between two images of objects and another that gives the area of each object image, and an algorithm for constructing a logic expression describing changes that

occurred in the scene. The main contribution of our approach, therefore, is that it provides the theoretical foundations for symbolically interpreting long sequences of sensor data transitions as database state transitions. Therefore, this work focuses on employing a theory of state change to interpret geometrically observable changes occurring in dynamic scenes. As such, distance information in this paper represents simply a case study of a broader work. In fact, everything developed in this paper could be extended in ways first hinted in [19]. Therefore, starting from a simplified spatial theory allows us to make explicit the update structure underlying the reasoning process.

The rest of this paper is organized as follows. We first motivate our work providing in Sect. 2 an example of the class of problems we aim to address. Then we present in Sect. 3 our research methods. There we propose a formalism, called $\mathcal{T}$-logic, to account for the interpretation of observable changes occurring in dynamic scenes. $\mathcal{T}$-logic is an instance of the Transaction Logic ($\mathcal{TR}$) [57, 58], a general logic of state change that accounts for the phenomenon of updating arbitrary logical theories. We also introduce an inference system which allows us to verify if a given formula is true in a given scene. Later in Sect. 3.3, we provide details about the implementation of this framework; the experiments we have performed to test our implementation are shown in Sect. 3.4. We present the results of our experiments in Sect. 4, and finally discuss the contributions of this paper in Sect. 5.

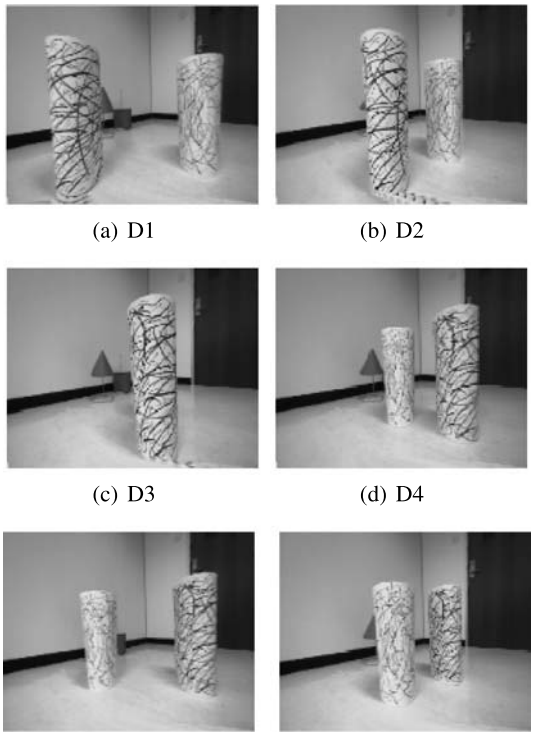
2 Motivating example: interpreting sequences of snapshots

To illustrate the classes of problems this paper addresses, Fig. 1 depicts a sequence of snapshots taken by a camera from an egocentric viewpoint.

Such image sequence can be interpreted as follows: assume that symbols p and q represent the spatial regions of the cylindrical objects depicted in Fig. 1. Thus, the transition from $D1$ to $D2$ (in Fig. 1) can be interpreted as "the spatial regions p and q are *approaching* each other". Similarly, the transition from $D2$ to $D3$ can be informally interpreted as "the spatial region q is *merging* into the spatial region p", from $D3$ to $D4$ as "the spatial region q is *emerging* from spatial region p ", from $D4$ to $D5$ as "p and q are *receding* from each other" and from $D5$ to $D6$ as "p and q are *approaching* each other". Interpreting such pairs of transitions was the purpose of the work described in [19, 24]. In a broader sense, however, the transitions from $D1$ to $D6$ should be interpreted as "the objects represented by p and q are *rotating around* each other".

Dynamic scenes may include objects that perform intrinsic movements, and objects that perform extrinsic movements. The former relates to movements that are perceived by the observer as changes of individual images of a single object in one snapshot with respect to its image in previous snapshots in the sequence. The latter regards movements that are perceived as changes in an object's position with respect to another (or others). In this work, we focus on *extrinsic* object movement. For example, when attempting to interpret a dynamic scene, such as the one depicted in Fig. 1, it is more important for us to be able to determine that object q and p are rotating around each other, than to find out that p, q, or both are performing rotations around their center of gravity.

To facilitate the treatment of extrinsic object motion, and to reduce the inherent complexity of dealing with object's shapes, this work assumes the environment is populated only by cylindrical objects. In fact, approximating object's shapes to cylinders is a traditional assumption in Computer Vision [59–61]. The high-level interpretation of the motion of arbitrary shaped objects depends on the assumption of models of shapes, which is one of the most elusive problems in qualitative spatial reasoning [6]. The investigation of reasoning about the dynamics of complex spatial objects (such as strings, holes and elongated objects) is the subject of our current investigations [62, 63].



(a) D1 (b) D2 (c) D3 (d) D4 (e) D5 (f) D6

Fig. 1 A chronological sequence of snapshots

3 Methods

In the context of qualitative spatial reasoning [7, 64], it has become apparent that a qualitative theory about observer-relative motion of objects in a dynamic environment is an essential part of an image interpretation system. In the research reported in this article we intend to show that a theory of updates (*i.e.*, actions and state change) provides a suitable logic foundation for the interpretation of geometrically observable changes in dynamic scenes.

The following materials were developed in this study:

1. a logical formalism, called $\mathcal{T}$-logic, for specifying changes in image sequences;
2. an algorithm that employs logical deductions for inferring changes that occurred in image sequences; and
3. a computer vision prototype system which uses items 1 and 2 above as foundations for interpreting scenes.

3.1 $\mathcal{T}$-logic

In this work we develop $\mathcal{T}$-logic, a specialized theory that provides a logical account for specifying, proving, and reasoning about object spatial relations of a scene (*i.e.*, a sequence of images). $\mathcal{T}$-logic is a dialect of the general language of Transaction Logic ($\mathcal{TR}$) [57] and was specifically designed to provide support for scene interpretation. In this section we introduce the basic concepts underlying $\mathcal{T}$-logic.

3.1.1 Syntax

We assume a many-sorted first-order language with sorts for spatial regions, physical objects and real numbers. The syntax includes also two infinite, enumerable sets of symbols: a set of function symbols and a set of predicate symbols. Constants, propositions, and terms are defined as usual in first-order logic. We adopt the Prolog convention that variables begin in upper-case, and predicate symbols (and constants) in lower-case roman letters. Atomic formulas are also called *atoms*. The logic extends first-order logic with a new connective, $\otimes$, called *serial conjunction*; given two first order formulas a and b, $a \otimes b$ means: "a occurs before b". In this paper we use the function term $i(X)$ that maps a physical body in the world to its spatial region in the camera snapshot.

Like classical logic, the underlying language has a Horn-like fragment (called *serial-Horn* [57]) with both a procedural and a declarative semantics. Serial-Horn rules, or *rules* for short, are formulas of the form $p \leftarrow a_1 \otimes a_2 \otimes \cdots \otimes a_n$, where each a_i is an atomic formula. A finite set of rules is a *rulebase*. A formula of the kind $a_i \otimes \cdots \otimes a_n$ is called a *serial goal*. The rule above may be read as "to compute p, it is sufficient to compute $a_1 \otimes a_2 \otimes \cdots \otimes a_n$". Later in this section we show examples on how to use rules to specify commonsense concepts about the movement of objects in space.

3.1.2 Semantics

In this work, formulas are interpreted on *paths* (*i.e.*, sequences of states), as in Process Logic [65]. A *state* is a set of spatial regions as noted by visual sensors in a particular time instant. To refer to states, we assume a countable collection $\mathbf{D}$ of symbols, called *state identifiers*, whereby each identifier in a collection is denoted by the symbol $\mathbf{D}_i$ (for a natural number i). A path represents a history of elementary changes on the world, and formulas represent what is true during periods of history. Classical connectives have their usual interpretations, except that they are interpreted on paths. For instance, $\alpha \wedge \beta$ means that α and β are both true on a path. The non-classical connectives allow a formula to relate to parts of a path. For instance, $\alpha \otimes \beta$ means that a given path can be split in two, where α is true on the prefix of the path and β is true on the suffix. Hence, it is convenient to define a *split* of a path π to be any pair of sub-paths, π_1 and π_2, such that $\pi_1 = \langle \mathbf{D}_1 \cdots \mathbf{D}_i \rangle$ and $\pi_2 = \langle \mathbf{D}_{i+1} \cdots \mathbf{D}_n \rangle$, where $\mathbf{D}_i$, $1 \leq i \leq n$. In this case we shall write $\pi = \pi_1 \circ \pi_2$. Moreover, we shall use the notation `state` to denote a constant that is true on any state, *i.e.*, on any path of length 1. Hence, ${}^{k}\otimes\texttt{state} \equiv \underbrace{\texttt{state} \otimes \cdots \otimes \texttt{state}}_{k}$ denotes a formula which is true on any path of length k. In Sect. 3.2.2, we shall use this particular kind of formula when interpreting dynamic scenes.

Next we present some examples illustrating the underlying syntax of the language.

Example 1 (Simple formulas) Assume X and Y are two distinct variables ranging over spatial regions, and p, q and w are three spatial regions of distinct objects detected at a given instant in the sensor data (*i.e.*, p, q and w are images of objects). Based on these assumptions, we show in Table 1 some simple formulas representing commonsense concepts about space.

Example 2 (Rules) In the context of this paper, rules represent *scene scripts*, *i.e.*, descriptions for events involving composite movements of objects (or images of objects).[1] For instance, for two given objects o_1 and o_2, we can write that o_1 is passing by o_2 (from left to right) if the image of o_1 is initially approaching the image of o_2, then it occludes the image of o_2 on the left, then it emerges on the right of the image of o_2, and finally recedes from the image of o_2. Formally, this case is represented by instantiating the following

[1] Henceforth, we shall use the term *scene script* when referring to this kind of formula.

Table 1 Simple formulas denoting commonsense concepts about space

$approaching(p, q)$	"p is approaching q"
$static(p, q)$	"p and q are static"
$approaching(p, q) \vee approaching(p, w)$	"p is approaching q or w"
$(\exists\, X)\ approaching(X, b)$	"some X is approaching b"
$\neg(\exists\, XY)\ static(X, Y)$	"no object image is static"
$approaching(p, q) \otimes static(p, q)$	"p is approaching q and then they are static"

formulae (for unifications O_1/o_1 and O_2/o_2):

$$passingBy(O_1, O_2) \leftarrow approaching(i(O_1), i(O_2)) \otimes mergeL(i(O_1), i(O_2)) \otimes emergeR(i(O_1), i(O_2)) \otimes receding(i(O_1), i(O_2)). \quad (1)$$

3.1.3 Primitive state operations

The general logic of $\mathcal{TR}$ [57] does not commit to a particular semantics of database state. One can think of $\mathcal{TR}$ as a logical framework, which can be instantiated as distinct specific logics in many ways. In $\mathcal{TR}$, a pair of oracles, called *state* and *transition oracles*, isolates elementary database operations from the logic used for combining, programming, and reasoning with them. The state oracle specifies a set of primitive state queries, *i.e.*, the *static* semantics of states; and the transition oracle specifies a set of primitive state *transitions*, *i.e.*, the *dynamic* semantics of states. In our approach, we specialize these oracles to provide us basic operations for interpreting visual sensor data. More specifically, here the state oracle encodes definitions used to translate visual sensor data into logic predicates describing spatial relations amongst objects. The state transition oracle encodes definitions used to translate transitions in sensor data into predicates describing higher-level commonsense concepts about space.

For the purposes of this work, in order to formalize the concepts of state data and state transition oracles we introduce the following functions. Function $dist/3$ defines the length of the shortest line separating any two distinct boundaries of object images in a state. Thus, $dist(x, y, \mathbf{D})$ means "the distance between regions x and y in state $\mathbf{D}$". Function $area(x, \mathbf{D})$ defines the area of an object image, discharging noisy regions (whose area is less than a given area threshold) and background region (whose area is greater than another given threshold). Thus, $area(x, \mathbf{D})$ means "the area of object x in $\mathbf{D}$", as described in Sect. 3.3. Future work shall tackle the task of defining these functions in terms of qualitative distances [66, 67] in order to avoid rounding errors, keeping the theory on the knowledge level [68]. Function $tlc(x, \mathbf{D})$ defines the coordinates of the leftmost top pixel of the image. Thus, $tlc(x, \mathbf{D})$ means "the coordinates of the top-left corner of the image of object x in $\mathbf{D}$".

As we shall see in Definition 1, function $area/2$ helps us identify the relation $ar(x)$, "area of an object image"; and function $tlc(x, \mathbf{D})$ helps us identify the relation $left(x, y)$, meaning "x is to the left of y".

Similarly, function $dist/3$ helps us identify three dyadic relations on images of objects: $disC(x, y)$, meaning "x is disconnected from y"; $extC(x, y)$, meaning "x is externally connected to y"; and $co(x, y)$, meaning "x is coalescent with y", as first defined in [24]. The relations $extC$, $disC$, and co form a jointly exhaustive, pairwise disjoint set of relations about distance, conform proved in [19].

The relations $extC$, $disC$, and co follow the example of RCC relations [1], however, the former are used to bridge the gap between sensor data and qualitative representation and reasoning, while the latter constitute an ontology about space assuming connectivity as the sole primitive. Therefore, it is worth pointing out that the relation *coalescent* is not equivalent to its RCC counterparts (such as *equality*, *partial overlap* and *tangential*), since *coalescing* represents a state whereby two regions cannot be distinguished (for instance, two image blobs of objects under occlusion), whereas the RCC relations assume that the regions can always be individualized.

Definition 1 (State oracle) Let δ, γ, and ω be a pre-defined distance, minimal and maximal area values, respectively. Let also *dist* and *area* be functions, $dist : S \times S \times D \to \Re$, $area : S \times D \to \Re$, where S is the set of object images, D is the set of state identifiers, and $\Re$ is the set of real numbers. For any pair of spatial regions x and y in a state $\mathbf{D}$, the state data oracle, $\mathcal{O}^d$, defines a mapping from x and y to one (and only one) relation $disC$, $extC$ or co, and to some relations ar and $left$ as follows:

$$disC(x, y) \in \mathcal{O}^d(\mathbf{D}) \leftrightarrow dist(x, y, \mathbf{D}) > \delta,$$

$$extC(x, y) \in \mathcal{O}^d(\mathbf{D}) \leftrightarrow dist(x, y, \mathbf{D}) \leq \delta \wedge dist(x, y, \mathbf{D}) \neq 0,$$

$$co(x, y) \in \mathcal{O}^d(\mathbf{D}) \leftrightarrow dist(x, y, \mathbf{D}) = 0,$$

$$ar(x) \in \mathcal{O}^d(\mathbf{D}) \leftrightarrow \gamma < area(x, \mathbf{D}) < \omega,$$

$$left(x, y) \in \mathcal{O}^d(\mathbf{D}) \leftrightarrow tlc(x, D) < tlc(y, D).$$

The diagram in Fig. 2 depicts a conceptual neighborhood diagram (CND) [69] for relations $disC$, $extC$ and co,

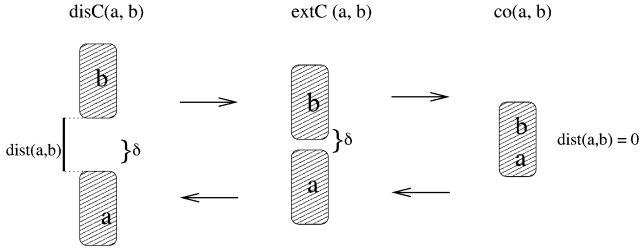


Fig. 2 Conceptual Neighborhood Diagram for the relations *disC*, *extC* and *co*

whereby arrows represent continuous transitions between these relations. The concept of continuous transitions, in this context, means that given two edge-connected relations, holding on spatial regions, there is no other possible relation (amongst *disC*, *extC* and *co*) that characterize the state of these regions.

Given a state identifier, the objective of this oracle is to inform what relations hold for objects in that state. Therefore, the state oracle provides a mapping from state identifiers (i.e. snapshots of the world) to ground instances of the relations presented in Definition 1.

It is worth pointing out that the reason for assuming distance as a primitive function is that the displacement between objects in the environment is a feature that can be easily extracted from the visual sensor data. It is for this reason, also, that the relations *disC*, *extC* and *co* (among the various possible relations between spatial regions) have a special status in this work.

Let us now consider elementary changes in the world. In our approach, elementary transitions are ground predicates denoting elementary perceptual changes in the world (as introduced in [19]). Therefore, assuming that two spatial regions x and y represent the images of two distinct objects, we use the following predicates to denote elementary transitions in spatial relations:

- $approaching(x, y)$, meaning "x and y are approaching each other";
- $receding(x, y)$, "x and y are receding from each other";
- $mergeR(x, y)$, "x is merging into the right of y";
- $mergeL(x, y)$, "x is merging into the left of y";
- $emergeR(x, y)$, "x is emerging to the right of y";
- $emergeL(x, y)$, "x is emerging to the left of y";
- $static(x, y)$, "x and y are static" (i.e. the distance separating them does not change in two subsequent states);
- $approachobs(x)$, "x approaches the observer";
- $recedeobs(x)$, "x recedes from the observer";
- $static(x)$, "x is static." (i.e. the area of x does not change in two subsequent states).

Here, left/right relations are mirrored on the observer's left/right sides.

In this context, given two states, the purpose of the state transition oracle is to inform what elementary transitions in spatial relations explain the difference in sensor data between the states. Formally:

Definition 2 (State transition oracle) Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be states, and $i(a)$ and $i(b)$ be the images of objects a and b respectively. Then,

$$
\begin{aligned}
&approaching(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) \\
&\quad \leftrightarrow (disC(i(a), i(b)) \in \mathcal{O}^d(\mathbf{D}_1) \\
&\qquad \vee extC(i(a), i(b)) \in \mathcal{O}^d(\mathbf{D}_1)) \\
&\qquad \wedge co(i(a), i(b)) \notin \mathcal{O}^d(\mathbf{D}_2) \\
&\qquad \wedge dist(i(a), i(b), \mathbf{D}_1) > dist(i(a), i(b), \mathbf{D}_2),
\end{aligned}
$$

$$
\begin{aligned}
&receding(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) \\
&\quad \leftrightarrow dist(i(a), i(b), \mathbf{D}_1) < dist(i(a), i(b), \mathbf{D}_2) \\
&\qquad \wedge (extC(i(a), i(b)) \in \mathcal{O}^d(\mathbf{D}_1) \\
&\qquad \vee disC(i(a), i(b)) \in \mathcal{O}^d(\mathbf{D}_1)),
\end{aligned}
$$

$$
\begin{aligned}
&static(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) \\
&\quad \leftrightarrow dist(i(a), i(b), \mathbf{D}_1) = dist(i(a), i(b), \mathbf{D}_2),
\end{aligned}
$$

$$
\begin{aligned}
&mergeR(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) \\
&\quad \leftrightarrow (disC(i(a), i(b)) \in \mathbf{D}_1 \vee extC(i(a), i(b)) \in \mathbf{D}_1) \\
&\qquad \wedge left(i(a), i(b)) \in \mathbf{D}_1 \wedge co(i(a), i(b)) \in \mathbf{D}_2,
\end{aligned}
$$

$$
\begin{aligned}
&mergeL(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) \\
&\quad \leftrightarrow (disC(i(a), i(b)) \in \mathbf{D}_1 \vee extC(i(a), i(b)) \in \mathbf{D}_1) \\
&\qquad \wedge left(i(b), i(a)) \in \mathbf{D}_1 \wedge co(i(a), i(b)) \in \mathbf{D}_2,
\end{aligned}
$$

$$
\begin{aligned}
&emergeR(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) \\
&\quad \leftrightarrow co(i(a), i(b)) \in \mathbf{D}_1 \wedge (disC(i(a), i(b)) \in \mathbf{D}_2 \\
&\qquad \vee extC(i(a), i(b)) \in \mathbf{D}_2) \wedge left(i(b), i(a)) \in \mathbf{D}_2,
\end{aligned}
$$

$$
\begin{aligned}
&emergeL(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) \\
&\quad \leftrightarrow co(i(a), i(b)) \in \mathbf{D}_1 \wedge (disC(i(a), i(b)) \in \mathbf{D}_2 \\
&\qquad \vee extC(i(a), i(b)) \in \mathbf{D}_2) \wedge left(i(a), i(b)) \in \mathbf{D}_2,
\end{aligned}
$$

$$
\begin{aligned}
&approachobs(i(a)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) \\
&\quad \leftrightarrow ar(i(a)) \in \mathcal{O}^d(\mathbf{D}_1) \wedge ar(i(a)) \in \mathcal{O}^d(\mathbf{D}_2) \\
&\qquad \wedge area(i(a), \mathbf{D}_1) < area(i(a), \mathbf{D}_2),
\end{aligned}
$$

$$recedeobs(i(a)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2)$$
$$\leftrightarrow ar(i(a)) \in \mathcal{O}^d(\mathbf{D}_1) \wedge ar(i(a)) \in \mathcal{O}^d(\mathbf{D}_2)$$
$$\wedge area(i(a), \mathbf{D}_1) > area(i(a), \mathbf{D}_2),$$

$$static(i(a)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2)$$
$$\leftrightarrow ar(i(a)) \in \mathcal{O}^d(\mathbf{D}_1) \wedge ar(i(a)) \in \mathcal{O}^d(\mathbf{D}_2)$$
$$\wedge area(i(a), \mathbf{D}_1) = area(i(a), \mathbf{D}_2).$$

In this way, the state transition oracle provides a built-in view to the transitions between spatial relations that have occurred during two consecutive states.

In a general way, the state data and the state transition oracles encode the axioms for sensor data assimilation presented in [19].

The relations presented in Definition 2 also form a conceptual neighborhood diagram, such as that shown in Fig. 2 [19]. In this context, the predicates *static* are necessary to guarantee the constraints of the dynamic qualitative simulation [70], since there is always a possibility of a static state happening between any transition from one relation to another. More specifically, $static/2$ guarantees this restriction for the binary predicates (representing transitions on distance) and $static/1$ for the unary predicates (that encode the transitions on the object's area).

It is worth noting that the axioms above hold only on images of objects. The mapping from predicates on images to their relative relations on bodies is accomplished by a second set of axioms that connect a disjunctive set of hypotheses on physical objects to explain a given state transition. The work presented in [19] describes this process in detail for pairs of images, and not on arbitrary long image sequences as we shall see further in this paper. However, an example on the mapping from images to bodies is in order. The axiom below (see (2)) is responsible for connecting hypotheses about physical objects and their relative images and illustrates two important issues: the representation of occlusion and the existence of multiple possible explanations.

$$\{occluded(A, B) \vee touching(A, B)\}$$
$$\leftarrow mergeR(i(A), i(B)) \vee mergeL(i(A), i(B)). \quad (2)$$

According to (2), the occlusion between two objects A and B ($occluded(A, B)$) can be inferred from the case where their respective images ($i(A)$ and $i(B)$) are merging. However, this is not the only possibility related to merging images as the objects could also be *touching* each other. These issues are discussed at length in [19], we shall now concentrate on the interpretation of long sequences of images.

Example 3 below shows that complex scripts describing elaborate scenes can be specified in the formalism. In the example, a set of Horn rules specifies a script for a scene in which an object moves between two other objects, performing semi rotations around each of the latter, developing an ∞-shaped pattern around them, as shown in Fig. 3.

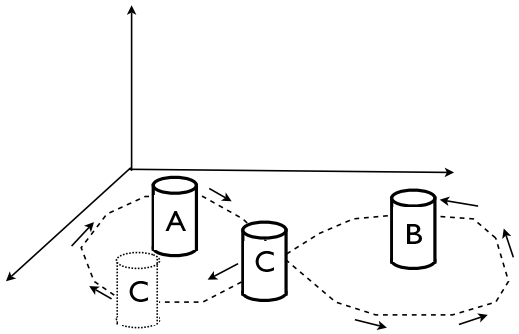


Fig. 3 An object 'C' moving around and between objects 'A' and 'B'

Example 3 (An object rotating around and passing between two objects) An object c is moving in an "∞"-shaped pattern around objects a and b if it initially rotates clockwise around a, then moves away from a, approaching b, then rotates counter-clockwise around b, as represented by the formulae below,[2] where: $drawing_{\infty}(C, A, B)$ means "C is moving in an ∞-shaped pattern around A and B"; $rotationCW(A, B)$ means "A is rotating clockwise around B"; $rotationCW(A, B)$ means "A is rotating counter-clockwise around B"; $passInFrontRL(A, B)$ means "A is passing in front of B from right to left (of the observer's view point)"; $passBehindLR(A, B)$ is analogous to $passInFrontRL$; and, $beginInFr(A, B)$, $beginBhd(A, B)$, $endInFr(A, B)$, $endBhd(A.B)$ represent transitions that occur at the beginning (end) of a "passing in front (behind)" movement. They serve as constraints that must be satisfied before (after) a $mergeRL(LR)$ occurs.

$$drawing_{\infty}(C, A, B)$$
$$\leftarrow rotationCW(C, A) \otimes receding(i(C), i(A))$$
$$\otimes approaching(i(C), i(B)) \otimes rotationCCW(C, B),$$
$$rotationCW(A, B)$$
$$\leftarrow passInFrontRL(A, B) \otimes passBehindLR(A, B),$$
$$rotationCCW(A, B)$$
$$\leftarrow passInFrontLR(A, B) \otimes passBehindRL(A, B),$$
$$passInFrontRL(A, B)$$
$$\leftarrow beginInFr(A, B) \otimes mergingRL(i(A), i(B))$$
$$\otimes endInFr(A, B),$$

[2] Due to space limitations, we do not provide rules specifying a counter-clockwise rotation. However, it is not difficult to see that they are analogous to the ones specifying a clockwise rotation.

$beginInFr(A, B)$

$\leftarrow approachobs(i(A)) \wedge approaching(i(A), i(B))$

$\wedge static(i(B))$,

$mergingRL(i(A), i(B))$

$\leftarrow mergeR(i(A), i(B)) \otimes emergeL(i(A), i(B))$,

$endInFr(A, B)$

$\leftarrow recedeobs(i(A)) \wedge receding(i(A), i(B))$

$\wedge static(i(B))$,

$passBehindLR(A, B)$

$\leftarrow beginBhd(A, B) \otimes mergingLR(i(A), i(B))$

$\otimes endBhd(A, B)$,

$beginBhd(A, B)$

$\leftarrow recedeobs(i(A)) \wedge approaching(i(A), i(B))$

$\wedge static(i(B))$,

$mergingLR(i(A), i(B))$

$\leftarrow mergeL(i(A), i(B)) \otimes emergeR(i(A), i(B))$,

$endBhd(A, B)$

$\leftarrow approachobs(i(A)) \wedge receding(i(A), i(B))$

$\wedge static(i(B))$.

Notice that according to predicate $rotationCW/2$, an object rotates clockwise around another stationary object if it passes first in front of the object from right to left (from the observer's view point), then behind it from left to right.

3.2 A deductive algorithm for interpreting image sequences

We begin by introducing the SDL-style resolution procedure that underlies the deductive algorithm for image sequence interpretation proposed later in this section.

The SDL-style refutation procedure designed for $\mathcal{T}$-logic differs from the inference system introduced in [57] where the path is a byproduct of the refutation procedure; here the path is given. In other words, in [57] the resolution may update the world state, thus creating a path, i.e., a sequence of world changes, whereas in our approach the path is given and the task of the refutation procedure is to determine the unsatisfiability of a formula in this path.

3.2.1 SDL-style resolution

The inference system manipulates expressions called *sequents*, which have the form

$$\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)\phi,$$

where $\mathbf{P}$ is the *background theory*, consisting of a rule-base specifying high (abstraction) level predicates representing commonsense concepts about space (e.g., see Example 3), and the pair of oracles (Definitions 1 and 2); $\pi_1 = \langle \mathbf{D}_1 \cdots \mathbf{D}_i \rangle$ and $\pi_2 = \langle \mathbf{D}_{i+1} \cdots \mathbf{D}_n \rangle$ are splits of a given path π and ϕ is a serial-goal. The informal meaning of a sequent is that formula $(\exists)\phi$ can be proved from state $\mathbf{D}_i$ and along π_2, *i.e.*, from the last state of π_1 followed by sequence of states π_2.

Let the expression $\leftarrow G_0$, denote a *goal clause*, where G_0 is the sequent

$$\mathbf{P}, \langle \mathbf{D}_1 \rangle \circ \langle \mathbf{D}_2 \cdots \mathbf{D}_n \rangle \vdash (\exists)\phi. \tag{3}$$

A SDL-style refutation of $\leftarrow G_0$ is a sequence of goal clauses $\leftarrow G_0 \cdots \leftarrow G_n$ where G_n is the *empty clause*, *i.e.*, the sequent $\mathbf{P}, \langle \mathbf{D}_1 \cdots \mathbf{D}_n \rangle \circ \langle\rangle \vdash ()$, where $\langle\rangle$ denotes the empty path, and $()$ denotes the empty formula. This sequent is an axiom of the inference system, which states that the empty formula is true on any path. Each $\leftarrow G_{i+1}$ is obtained from $\leftarrow G_i$ by using the following axiom and inference rules.

Definition 3 (Inference System)

Axiom: $\mathbf{P}, \pi_1 \circ \pi_2 \vdash ()$, for any path split π_1 and π_2.

Inference rules: In rules 1–3, σ is a variable substitution, a and b are atomic formulae, and ϕ and *rest* are serial goals.

1. *Applying rule definitions:* Suppose $a \leftarrow \phi$ is a rule in $\mathbf{P}$ whose variables have been renamed so that the rule shares no variables with $b \otimes rest$. If a and b unify with m.g.u. σ, then

$$\frac{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)(\phi \otimes rest)\sigma}{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)(b \otimes rest)}$$

2. *Querying the world state:* If $b\sigma$ and $rest\sigma$ share no variables, and $\mathcal{O}^d(\mathbf{D}_i) \models^c (\exists)b\sigma$, then

$$\frac{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)rest\sigma}{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)(b \otimes rest)},$$

where $\pi_1 = \langle \mathbf{D}_1 \cdots \mathbf{D}_i \rangle$.

3. *Verifying a state transition:* If $b\sigma$ and $rest\sigma$ share no variables, and $\mathcal{O}^t(\mathbf{D}_i, \mathbf{D}_{i+1}) \models^c (\exists)b\sigma$, then

$$\frac{\mathbf{P}, \pi_1' \circ \pi_2' \vdash (\exists)rest\sigma}{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)(b \otimes rest)},$$

where:
$\pi_1 = \langle \mathbf{D}_0 \cdots \mathbf{D}_i \rangle$, $\pi_2 = \langle \mathbf{D}_{i+1} \mathbf{D}_{i+2} \cdots \mathbf{D}_n \rangle$,
$\pi_1' = \langle \mathbf{D}_0 \cdots \mathbf{D}_i \mathbf{D}_{i+1} \rangle$, and $\pi_2' = \langle \mathbf{D}_{i+2} \cdots \mathbf{D}_n \rangle$.

Each inference rule consists of two sequents, and has the following interpretation: if the upper sequent (G_{i+1}) can be inferred, then the lower sequent (G_i) can also be inferred.

The inference rules above capture the roles of scene scripts (*i.e.*, serial-Horn rules), the state oracle and the transition oracle, as follows:

Rule 1 deals with serial-Horn rule definitions. Informally, this rule replaces an instance of the rule 'head' by an instance of its body. Notice that the path split does not change.

Rule 2 deals with state tests. It says that a condition b satisfied in $\mathbf{D}_i$ can be added to the front of the formula *rest*. Notice that the path split does not change.

Rule 3 deals with tests on a pair of states (*i.e.*, the last state of π_1 and the first state of π_2). Informally, b is attached to the front of *rest*, so that the first (leftmost) state of π_2 is removed from the path split and added to the (right-most) end of π_1, *i.e.*, the formula b can be proved from $\mathbf{D}_i$ and in the path split $\langle \mathbf{D}_{i+1} \cdots \mathbf{D}_n \rangle$.

Given the similarities between this inference system and $\mathcal{TR}$'s proof system, the proofs of correctness and completeness ([71], Appendices A and B) are parallel.

It is worth pointing out that, in a sequent, the leftmost atom of the serial-goal is always selected.

Example 4 (Proving formulas using the inference system) Suppose a scene consists of three sequential snapshots of objects a and b (whose images are represented by the terms p and q respectively); hence the path $\langle \mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3 \rangle$. Assume in the first state, $\mathbf{D}_1$, p is disconnected from q; in $\mathbf{D}_2$, they are also disconnected, but the distance separating them is smaller; and in $\mathbf{D}_3$, p and q are externally connected. That is,

$$\begin{aligned} &disC(p,q) \in \mathcal{O}^d(\mathbf{D}_1), \\ &disC(p,q) \in \mathcal{O}^d(\mathbf{D}_2), \\ &co(p,q) \in \mathcal{O}^d(\mathbf{D}_3), \\ &dist(p,q,\mathbf{D}_2) > dist(p,q,\mathbf{D}_1) \\ &tlc(p,\mathbf{D}_2) < tlc(q,\mathbf{D}_2). \end{aligned} \tag{4}$$

Let us then prove that $approaching(p,q) \otimes mergeL(p,q)$ is true in the path $\langle \mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3 \rangle$. That is, let us prove that initially p is approaching q, and p is merging q on the left:

$\mathbf{P}, \langle \mathbf{D}_1 \rangle \circ \langle \mathbf{D}_2, \mathbf{D}_3 \rangle \vdash approaching(p,q) \otimes mergeL(p,q)$

if $\mathbf{P}, \langle \mathbf{D}_1, \mathbf{D}_2 \rangle \circ \langle \mathbf{D}_3 \rangle \vdash mergeL(p,q)$

by Inference Rule 3 and (4)

if $\mathbf{P}, \langle \mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3 \rangle \circ \langle \rangle \vdash ( )$ by Inference Rule 3 and (4).

This deduction succeeds because the bottom-most sequent is an axiom.

3.2.2 The deductive algorithm

We now consider the problem of providing a formal algorithm for image sequence interpretation. More specifically, given a sequence of changes that took place in the world, such as those depicted in Fig. 1, our objective is to provide a logic-based method which allows us to find an explanation for those changes. Therefore, we want to obtain a $\mathcal{T}$-logic formula, say Δ, which explains the changes in the path (*i.e.*, sequence of snapshots) π.

The process which we use to obtain Δ can be informally described as follows: let us assume all we have initially is the background theory $\mathbf{P}$, consisting of a set of scene scripts specifying high level predicates representing commonsense concepts about space (see Example 2), and the given path $\pi = \langle \mathbf{D}_1, \mathbf{D}_2 \cdots \mathbf{D}_k \rangle$ of length k. The objective is then to find a formula Δ which is true in the path π. If Δ exists, then it can be represented as $\phi_1 \otimes \phi_2 \otimes \cdots \otimes \phi_{k-1}$, where each ϕ_i is a formula of the form $a_1 \wedge a_2 \wedge \cdots \wedge a_n$, where $a_i \in \mathcal{O}^t(\mathbf{D}_j, \mathbf{D}_{j+1})$, $(1 \leq j < k)$. Therefore, Δ can be seen as an explanation for the state transitions in π.

An intuitive approach to obtain the transition predicates $\phi_i, 1 \leq i < k$, is to input the first and second states to the transition oracle and obtain the set of predicates, say $\{a_1, a_2, \ldots, a_n\}$, representing the elementary transitions noted in the state change. In effect, we can denote this set using the ground formula $\phi_1 = a_1 \wedge a_2 \wedge \cdots \wedge a_n$. Then we would do the same for the second and third states in the sequence, thus obtaining ϕ_2, which represents the other state change. Notice that the $\mathcal{T}$-logic formula $\phi_1 \otimes \phi_2$ is an explanation for the changes that occurred in the first three states of the path. If we repeat this procedure with all states of the path, adding in each iteration the transition predicates output from the transition oracle to the tail of the serial conjunction $a_1 \otimes a_2 \otimes \cdots$, we would eventually build the serial conjunction $\phi_1 \otimes \phi_2 \otimes \cdots \otimes \phi_{k-1}$, which is a formula that explains all the changes in the path.

For instance, in Example 4, we have seen how to use the inference rules presented in Sect. 3.2.1 to prove that $approaching(p,q) \otimes mergeL(p,q)$ is true in $\pi = \langle \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \rangle$. Now suppose we were interested in obtaining this formula instead of the state changes it forces. We would first input $\mathbf{D}_1$ and $\mathbf{D}_2$ to the transition oracle and obtain as a result the ground predicate $approaching(p,q)$, which represents the elementary transition that took place in the path split (assuming the transition oracle does not include any other elementary transition in the returned set). Then we would do the same for $\mathbf{D}_2$ and $\mathbf{D}_3$, thus obtaining $mergeL(p,q)$, which represents the other state change. At this point, there are no more states in the path to be analyzed, and concatenating the two elementary transitions we

obtain: $approaching(p,q) \otimes mergeL(p,q)$. We have already proved that this formula is true in the given path; hence it is an explanation for the changes in the path.

Algorithm 1 formalizes the process described above.

Algorithm 1 (Deductive algorithm) *This algorithm is an extension of the SDL-style resolution presented in Definition 3.*

From a given path π, we can obtain ${}^{k}\otimes\texttt{state}$, which is a $\mathcal{T}$-logic constant formula[3] *also true in π. From π we can also obtain the goal clause $\leftarrow G_0$, where G_0 is the sequent*:

$\mathbf{P}, \pi \vdash {}^{k}\otimes\texttt{state}$, *where* $\pi = \langle \mathbf{D}_1 \rangle \circ \langle \mathbf{D}_2 \cdots \mathbf{D}_k \rangle$.

To find a serial-goal Δ_n which is true in the same path π, a refutation of the form $\leftarrow G_0, \Delta_0 \cdots \leftarrow G_n, \Delta_n$ is constructed, where:

- *each G_i is a sequent,*
- *each Δ_i is a serial-goal,*
- *G_n is the sequent $\mathbf{P}, \pi \vdash ()$, i.e., the axiom of the inference system presented in Sect. 3.2.1, and*
- *Δ_0 is the empty formula.*

Assume

$$G_i = \mathbf{P}, \langle \mathbf{D}_1 \cdots \mathbf{D}_j \rangle \circ \langle \mathbf{D}_{j+1} \cdots \mathbf{D}_n \rangle \vdash {}^{m}\otimes\texttt{state},$$
$$m \geq 1.$$

There are two inference rules to obtain $\leftarrow G_{i+1}, \Delta_{i+1}$ from $\leftarrow G_i, \Delta_i$:

Rule A: *If $\mathcal{O}^t(\mathbf{D}_j, \mathbf{D}_{j+1}) = \{a_1, a_2, \ldots a_n\}$, then $\Delta_{i+1} = \Delta_i \otimes a_1 \wedge a_2 \wedge \cdots \wedge a_n$. To obtain G_{i+1}, we use inference rule 3 on G_i:*

$$G_{i+1} = \mathbf{P}\langle \mathbf{D}_1 \cdots \mathbf{D}_j \mathbf{D}_{j+1} \rangle \circ \langle \mathbf{D}_{j+2} \cdots \mathbf{D}_n \rangle \vdash {}^{m-1}\otimes\texttt{state}.$$

Rule B: *If the background theory includes formula $d \leftarrow \phi$ and ϕ produces the same changes in the path split $\langle \mathbf{D}_1 \cdots \mathbf{D}_j \rangle$ as Δ_i, i.e., $\mathbf{P}, \langle \mathbf{D}_1 \cdots \mathbf{D}_j \rangle \vdash \Delta_i \wedge \phi$, then $\Delta_{i+1} = d$. In this case, since no state transition was detected, $G_{i+1} = G_i$.*

Example 5 (Obtaining the explanation for the changes in a path) In this example we show how the procedure introduced above works. Let the background theory $\mathbf{P}$ include rule (1) of Example 2, and the aforementioned pair of oracles. Notice that Rule (1) is a script for a scene in which an object is passing by another, as illustrated in Fig. 4.

[3] Recall that ${}^{k}\otimes\texttt{state} \equiv \underbrace{\texttt{state} \otimes \cdots \otimes \texttt{state}}_{k}$.

Let also $i(a)$ and $i(b)$ be two distinct images of objects a and b, respectively. The sequence in Fig. 4 can be interpreted as follows:

$\mathbf{P}, \langle \mathbf{D}_1 \rangle \circ \langle \mathbf{D}_2 \mathbf{D}_3 \mathbf{D}_4 \mathbf{D}_5 \rangle \vdash {}^{5}\otimes\texttt{state}, ()$

if $\mathbf{P}, \langle \mathbf{D}_1 \mathbf{D}_2 \rangle \circ \langle \mathbf{D}_3, \mathbf{D}_4 \mathbf{D}_5 \rangle \vdash$
$\quad {}^{4}\otimes\texttt{state}, approaching(i(a), i(b))$

from **Rule A**, assuming:

$\quad approaching(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2)$

if $\mathbf{P}, \langle \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \rangle \circ \langle \mathbf{D}_4 \mathbf{D}_5 \rangle \vdash$
$\quad {}^{3}\otimes\texttt{state}, approaching(i(a), i(b))$
$\quad \otimes mergeL(i(a), i(b))$

from **Rule A**, assuming:

$\quad mergeL(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_2, \mathbf{D}_3)$

if $\mathbf{P}, \langle \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{D}_4 \rangle \circ \langle \mathbf{D}_5 \rangle \vdash$
$\quad {}^{2}\otimes\texttt{state}, approaching(i(a), i(b))$
$\quad \otimes mergeL(i(a), i(b))$
$\quad \otimes emergeR(i(a), i(b))$

from **Rule A**, assuming:

$\quad mergeR(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_3, \mathbf{D}_4)$

if $\mathbf{P}, \langle \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{D}_4 \mathbf{D}_5 \rangle \circ \langle\ \rangle \vdash$
$\quad \texttt{state}, approaching(i(a), i(b))$
$\quad \otimes mergeL(i(a), i(b)) \otimes emergeR(i(a), i(b))$
$\quad \otimes receding(i(a), i(b))$

from **Rule A**, assuming:

$\quad receding(i(a), i(b)) \in \mathcal{O}^t(\mathbf{D}_4, \mathbf{D}_5)$

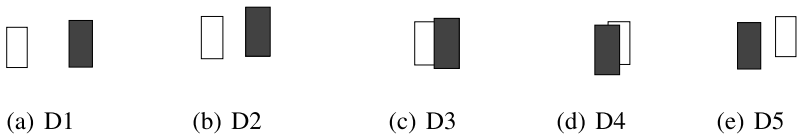
if $\mathbf{P}, \langle \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{D}_4 \mathbf{D}_5 \rangle \circ \langle\ \rangle \vdash (), approaching(i(a), i(b))$
$\quad \otimes mergeL(i(a), i(b)) \otimes emergeR(i(a), i(b))$
$\quad \otimes receding(i(a), i(b))$

from the definition of `state`

if $\mathbf{P}, \langle \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{D}_4 \mathbf{D}_5 \rangle \circ \langle\ \rangle \vdash (), passingBy(a, b)$

from **Rule B** and formula (1).

The following theorem shows that, given a path π and a background theory $\mathbf{P}$, if the method presented in Algorithm 1 provides an expression Δ which explains the changes in the path according to $\mathbf{P}$, then in every model of $\mathbf{P}, \pi$ satisfies formula Δ according to the semantic theory of Transaction Logic, the language upon which the theory of $\mathcal{T}$-logic is based. In order to provide a formal statement for this theorem we assign to Algorithm 1 a function symbol:



(a) D1 (b) D2 (c) D3 (d) D4 (e) D5

Fig. 4 A scene in which an object passes by another

$comp : P \times path \rightarrow \delta$, where P is a set of background theories, each consisting of a rulebase and a pair of oracles (state and transition oracles); *path* is a set of sequences of states, *i.e.*, a set of paths; and δ is a set of serial-Horn formulas. Therefore, if the algorithm is correct, then $comp(\pi, \mathbf{P})$ provides an explanation Δ for a given path π and background theory $\mathbf{P}$.

Theorem 1 (Correctness of the algorithm) *Let π be a path and $\mathbf{P}$ a background theory, then the following statement is true*:

if $comp(\pi, \mathbf{P}) = \Delta$ **then** $\mathbf{P}, \pi \models \Delta$.

In the expression above, the statement $\mathbf{P}, \pi \models \Delta$ expresses a form of logical entailment in $\mathcal{TR}$ called executional entailment [71]. Informally, this statement means that π is an execution path of Δ. That is, if one performs a refutation of Δ using $\mathcal{TR}$'s proof theory [71], then π can be derived from this refutation.

Proof We prove that Δ explains the changes on π by induction on the structure of π.

- Base case (one state transition): Assume π is represented by the path split $\langle \mathbf{D}_0 \rangle \circ \langle \mathbf{D}_1 \rangle$. If we apply inference rule **A**, then $comp(\pi, \mathbf{P}) = \Delta = a_1 \wedge \cdots \wedge a_n$, where a_i are the elementary perceptual changes that have occurred between the two states.[4] Moreover, since π can be derived from a (SDL-style) refutation of Δ, then Δ is true in π. Formally, $P, \pi \models \Delta$.
- General case (n state transitions): Assume π is represented by the path split $\langle \mathbf{D}_0 \mathbf{D}_1 \cdots \mathbf{D}_{n-1} \rangle \circ \langle \mathbf{D}_n \rangle$ and $comp(\langle \mathbf{D}_0 \mathbf{D}_1 \cdots \mathbf{D}_{n-1} \rangle, \mathbf{P}) = \Delta_{n-1}$. From the base case, if rule **A** can be applied, then $comp(\pi, \mathbf{P}) = \Delta_n = \Delta_{n-1} \otimes \phi$, where ϕ is a formula that explains the transition between $\mathbf{D}_{n-1}$ and $\mathbf{D}_n$; i.e., $P, \pi \models \Delta_{n-1} \otimes \phi$. Therefore, it follows from the soundness and completeness of the $\mathcal{TR}$ proof system [71] that the method presented in Algorithm 1 provides correct explanations for any path π. □

In the next section we present a prototype implementation of the framework introduced in this section.

[4] Notice that if we apply inference rule **B** instead of rule **A**, then no state transition takes place; only the sequence of predicates denoting the movement of objects in space represented by Δ gets replaced by another, logically implied, predicate.

3.3 A computer vision prototype system for image sequence interpretation

Figure 5 shows the major components of the computer vision prototype system which was implemented.

The prototype consists of two modules: Module 1 and 2 in Fig. 5. Module 1 performs information extraction. It takes raw image data as input, performs Blob Coloring on the data and creates a region map. This region map is then input to Module 2, which performs the high-level interpretation of the image sequence. In a nutshell, the region map is input to the State Oracle, which outputs the logical relations amongst spatial regions in the image. These relations are then input to the Transition Oracle, which determines the transitions that occurred between pairs of snapshots. The Inference Engine then uses these transitions, together with scene scripts (*i.e.*, serial-Horn rules) provided by the user, to output an interpretation for the geometrically observable changes in the scene.

3.3.1 Image segmentation

Image segmentation and object detection were implemented using the Blob Coloring technique [59], because this particular technique also facilitates the calculation of other image features, such as area of an object, perimeter, centroid, and the like.

The image segmentation process was designed to work as follows:

In the first step, Red Green Blue (RGB) pixel components of image data were loaded into a tridimensional matrix (*image height* × *image width* × *pixel RGB values*). To avoid using color as the basis for pixel comparison and thus provide a faster algorithm for region map creation, the tridimensional encoding was transformed into a more simple, bi-dimensional, gray level encoding (*image height* × *image width* × *pixel gray level*).

In the second step, a Blob coloring technique was used for labeling regions of the image. The result of this process was then stored in a third matrix. In this matrix, each entry (h, w) represents the region in which pixel (h, w) is located in the image.

In the next step, information about area, color, and location of the objects (regions) was extracted. The calculation of the area of an object was based on the summation of the number of pixels in the respective region. The color of each of the regions was determined by correlating the region with

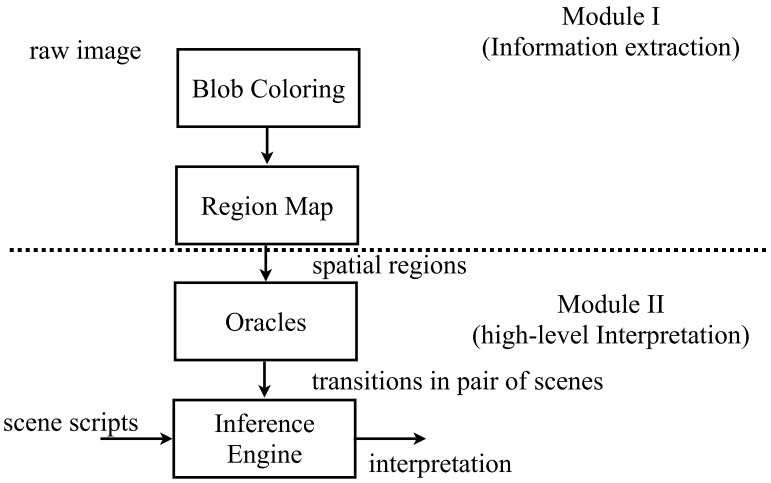


Fig. 5 The computer vision prototype system

the respective object in the original matrix. Each object's location in the image was determined by the first pixel of the region of the respective object. The information obtained in this last step was then stored in a table (henceforth called *InfoTable*). Each line in the InfoTable contains information about the area, color and location of a region. Table 2 shows an example of such table. Notice that small values for area represent noise in the image.

3.3.2 The oracles

The state oracle was implemented as a procedure that receives features of a given scene stored in TableInfo (see Table 2) and, based on these features, detects the objects in the scene by filtering out noise (objects with area less than 200 pixels) and ignoring scene background (objects with area greater than 5000 pixels), determines the distance between pairs of object images in the scene, compares it with a predefined δ threshold, and determines which of the relations mentioned in Definition 1 hold for the objects in the scene.

The transition oracle was implemented in a similar way. Based on Definition 2, a procedure was implemented that takes two states and two object images and finds which transition predicates hold for the objects in the states.

3.3.3 The inference engine

The Inference Engine consists of two modules: a *translator*, which converts scene scripts (*i.e.*, serial-Horn rules provided by the knowledge engineer describing composite moments of objects) into finite state machines; and a *controller*, that identifies when the state changes in the machine entails a particular scene script. States in such state machine represent state changes specified in the body of the respective serial-Horn rule. Once the controller receives a transition from the Transition Oracle, it searches for a state machine that has a state from which that transition is possible. If such state machine is found, the controller allows the state transition to occur in that machine. Then it waits for more transitions.

When a state machine reaches a final state, the controller then notifies that a given scene script has been identified.

3.3.4 Other considerations regarding the operation of the system

The implementation of the deductive algorithm introduced by Definition 1 is as follows.

Features extracted from a scene are stored in InfoTable (see Table 2) and supplied as input to the state oracle. This oracle then relates image spatial regions with concepts about space and connectivity amongst these regions.

To consistently identify regions across subsequent scenes, a comparison of object colors, areas and locations was used. As stated earlier, it was assumed that a notion of continuity should prevail in regards to the values of such parameters [51]. That is, an object would not perform a great jump or simply disappear in a sequence. Therefore, a minimal variance was allowed for the aforementioned parameters.

Once the second scene has been analyzed, the transition oracle verifies and stores which transitions took place in the pair of scenes. This analysis is based on the spatial relations returned from the state oracle and on the built-in functions that calculate distance between objects and the area of objects. These transitions are then passed to the controller of the inference engine which then operates as described above.

It is important to notice that, by using a state machine as the conceptual model for the inference engine, it is possible not only to process long sequences of scenes, but also to know in advance what will be the next transition between scenes, and consequently, what are the possible spatial relations which will occur in the next scene.

Table 2 Features extracted from an image

Region	Area	Color (R,G,B)	Location (row,column)
0	7663	(255, 255, 255)	(0, 0)
1	1	(139, 106, 215)	(76, 34)
2	3	(177, 172, 255)	(76, 31)
3	9	(105, 97, 255)	(76, 22)
⋮		⋮	

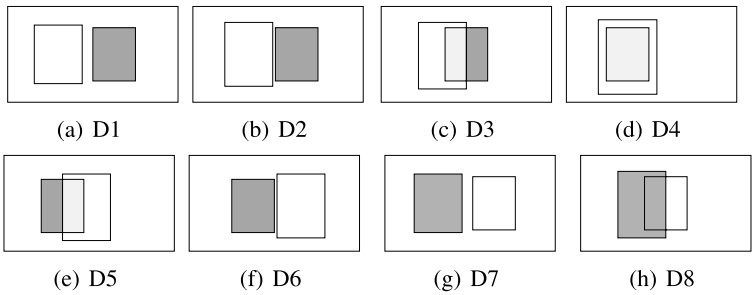


Fig. 6 Schematic figure of the rotation event

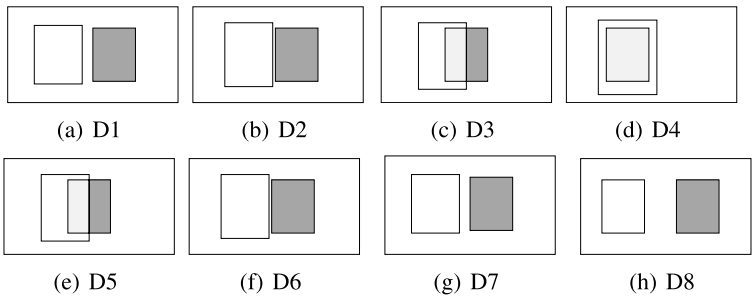


Fig. 7 Schematic figure of the anti-rotation event

3.4 The experiments

The objective of the experimental procedure was to verify if the system can properly interpret two types of movements: rotation and anti-rotation from any snapshot sequence where only one of this events was present in any particular sequence. In a rotation two objects rotate around a common fixed axis (Fig. 6). An anti-rotational movement is similar to a rotation up to the point where one of the objects gets occluded by the other (Fig. 7). At that point the de-occlusion of the object occurs at the same side where the occlusion took place. This particular type of rotation was used to verify how the system would behave when attempting to interpret ambiguous movements.

Below we provide the scene scripts (serial-Horn rules) for rotation and anti-rotation. Figure 8 shows the respective state machine(s):

$$
\begin{aligned}
&\mathit{rotation}(A, B)\\
&\quad \leftarrow \mathit{events}1(i(A), i(B)) \otimes \mathit{mergeR}(i(A), i(B))\\
&\qquad \otimes \mathit{emergeL}(i(B), i(A)) \otimes \mathit{events}2(i(A), i(B)),\\
&\mathit{rotation}(A, B)\\
&\quad \leftarrow \mathit{events}1(i(A), i(B)) \otimes \mathit{mergeL}(i(A), i(B))\\
&\qquad \otimes \mathit{emergeR}(i(B), i(A)) \otimes \mathit{events}2(i(A), i(B)),\\
&\mathit{antirotation}(A, B)\\
&\quad \leftarrow \mathit{events}1(i(A), i(B)) \otimes \mathit{mergeR}(i(B), i(A))\\
&\qquad \otimes \mathit{emergeR}(i(B), i(A)) \otimes \mathit{events}2(i(A), i(B)),\\
&\mathit{antirotation}(A, B)\\
&\quad \leftarrow \mathit{events}1(i(A), i(B)) \otimes \mathit{mergeL}(i(B), i(A))\\
&\qquad \otimes \mathit{emergeL}(i(B), i(A)) \otimes \mathit{events}2(i(A), i(B)),\\
&\mathit{events}1(i(A), i(B))\\
&\quad \leftarrow \mathit{approachobs}(i(A)) \wedge \mathit{recedeobs}(i(B)),\\
&\mathit{events}2(i(A), i(B))\\
&\quad \leftarrow \mathit{approachobs}(i(B)) \wedge \mathit{recedeobs}(i(A)).
\end{aligned}
$$

Based on these two types of movement six situations were analyzed:

Table 3 Results for pairs of objects with distinct colors, where *rot* and *antRot* stand for rotation and anti-rotation respectively

	Set 1		Set 2		Set 3		Set 4		Set 5	
	rot	*antRot*	*rot*	*antRot*	*rot*	*antRot*	*rot*	*antRot*	*rot*	*antRot*
Distant objects	+	+	+	+	+	+	+	+	+	+
Close objects	+	+	+	+	+	+	+	+	+	+

Table 4 Results for pairs of objects with the same color (perceptually indistinguishable)

	Set 1		Set 2		Set 3		Set 4		Set 5	
	rot	*antRot*	*rot*	*antRot*	*rot*	*antRot*	*rot*	*antRot*	*rot*	*antRot*
Distant objects	–	–	+	+	+	+	–	–	+	+
Close objects	–	–	–	–	–	–	–	–	–	–

1. two objects of different colors, and distant from each other;
2. two objects of same color, distant from each other;
3. two objects different colors, close to each other; and
4. two objects of same color, close to each other.

By "distant from each other" we mean that in the initial state the two objects are disconnected (*disC*, according to Definition 1), whereas "close to each other" means that the objects are externally connected (*extC*, according to Definition 1) at the initial state of the image sequence.

It is worth noting that in each of the experiments, both rules: for rotation and anti-rotation, were available to the system. Therefore, what we expect to show here is whether the system is capable of disambiguating between two possible explanations for a single image sequence. We also want to evaluate the extent to which our system, built with simple image processing techniques added to qualitative relations about space and continuity constraints, is capable of solving the challenging problem of anchoring symbols to sensor data from perceptually indistinguishable objects [72].

4 Results

Our empirical results are presented in Tables 3 and 4, and summarized in Table 5. In Tables 3 and 4 a "+" means "the rotational (anti-rotational) movement was interpreted correctly at the end of the image sequence".

Table 3 shows the results of the interpretation of rotation and anti-rotation from data-sets obtained from scenarios with two objects of distinct colors. In this case the system was capable of interpreting all data-sets correctly.

The results obtained from perceptually indistinguishable objects (objects of the same shape, size and color) are presented in Table 4. There we note that the system had a degradation in performance: it correctly interpreted 60% of the data-sets for distant objects and completely misinterpreted the data-sets from close objects. The reason for this loss in performance relates to the use of fixed thresholds for encoding the variances on area and position (cf. discussed at the end of Sect. 3.3). For close objects, the variance in the area was so minimal that it was not possible to find one set of thresholds to represent the change in area caused by the movement in depth, *e.g.*, when an object rotates around another. Similarly, the results relative to the distant-objects data-sets suffer from the rigidity of the thresholds on the variance of area and position; however, in this case, the thresholds seem to complement each other allowing 60% of the cases to be properly interpreted.

Table 5 Summary of the results

	Close objects	Distant objects
Same color	0%	60%
distinct color	100%	100%

Table 5 presents a summary of all the above results.

5 Discussion and open issues

This paper tackles the problem of image sequence evaluation [41] from a logic-based perspective following the lines proposed in [19, 24]. The context that motivates the development of this work is the investigation of inference methods that best operate on space perceptions and how space can be represented to facilitate spatial inference.

In particular, the present paper extends the spatial reasoning theories proposed in [19, 24] (which accounts for the interpretation of pairs of snapshots of the world) allowing the evaluation of dynamic scenes. More specifically, we take as starting point the work proposed in [24], that defines events based on pairs of snapshots of the world, such as *approaching*, *receding*, *splitting* and *merging*, and build events such as *passing by* and *rotating*, which involve sequences

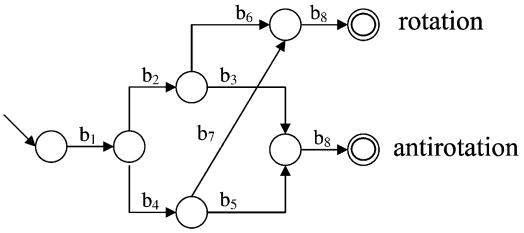


Fig. 8 State machines for rotation and anti-rotation

where:

$b_1 : events1(i(A), i(B))$
$b_2 : mergeR(i(B), i(A))$
$b_3 : emergeL(i(B), i(A))$
$b_4 : mergeL(i(B), i(A))$
$b_5 : emergeR(i(A), i(B))$
$b_6 : emergeR((i(B), i(A))$
$b_7 : emergeL((i(B), i(A))$
$b_8 : events2(i(A), i(B))$

of the former. In order to accommodate events of sequences of snapshots we use, as underlying formalism, Transaction Logic ($\mathcal{TR}$) [57, 58] whose semantics defines models based on *paths of states*, following the ideas proposed in process logic [65].

The use of transaction logic to account for the semantics of image sequence interpretation is justified by the fact that the language underlying this work makes assertions about a changing world. That is, formulas in the proposed language are interpreted along sequences of states, i.e., paths. This feature of the logic naturally relates to the problem of dynamic scene interpretation.

In the present paper we used the machinery of $\mathcal{TR}$ to define an algorithm for inferring changes that occurred in image sequences. The algorithm works by, first, proposing an arbitrary sequence of formulas (a serial conjunction of formulas) to explain a given sequence of images, that are considered as states in the algorithm. The algorithm, then, substitutes each formula in this sequence by a conjunction of atomic formulas that are explanations for each state change.

We show correctness of the algorithm by recurring to the soundness and completeness proof of $\mathcal{TR}$, as the algorithm uses $\mathcal{TR}$ proof system to build the explanations for the state changes.

The proposed approach was implemented as a four-stage process: first, a blob coloring algorithm provides region maps from a static camera observing rotation events. In a second stage, these region maps are input to the logical system that generates a description of each snapshot (provided by a state oracle). Third, the interpretation of pairs of snapshots is output by the transition oracle. Finally, the interpretation of the entire image sequence is provided.

We tested the implementation of the proposed framework on the task of interpreting sequences of images showing two contrasting events: some sequences show a pair of objects rotating around a fixed axis, and some others show a movement that we called "anti-rotation", in which two objects start rotating around each other but do not complete the rotation movement. We tested the system with objects of distinct colors and with objects of the same color. With the latter we wanted to verify the extent to which our system could disambiguate rotation and anti-rotation from two perceptually indistinguishable objects.

The results show that the interpretation of image sequences containing objects of different colors was 100% accurate. It has become evident that the correct interpretation of scenes with objects of the same color depends on the similarity of the sequences, since in this case variance in area and position are the two features used for interpreting a dynamic scene. Another factor that affects the interpretation of the movements of objects of the same color is the distance between objects. The closer the objects, the smaller are the variances on area and position. Consequently, the more difficult it is to identify such objects.

Taken in isolation, the experimental results presented in Sect. 4 could be interpreted as consequences of the shortcomings of the low-level vision processes, and thus overshadowing the impact of our approach on the scene interpretation process. However, it is important to place these results into perspective: on the one hand, we have assumed only the distance sub-theory of the approach developed in [19] (leaving aside theories about depth and size, for instance); on the other hand, we have used one of the simplest algorithms for segmenting images (blob coloring), without paying much attention to noise filtering. Nevertheless the results show that, even under these restrictions, the framework proposed is capable of interpreting a rotation event from sequences of images. Moreover, the possibility to represent more complex object behaviors, such as the ∞-shaped pattern of Example 3, suggests that these behaviors could also be interpreted from vision data using the ideas developed in this paper.

It may be argued that the information used is very limited in scope, as only distance information is assumed. However, this is a work about updates with respect to a spatial theory defined on sensor data, and not on the spatial theory itself (which has been investigated elsewhere [19]). Therefore, the use of distance information is just a case study for a broader work on spatial reasoning and image sequence evaluation. In this sense, everything developed here could be extended using the more complete theory defined in [19]. Having a

simple spatial theory allowed us to concentrate on the investigation of the update (*i.e.*, state change) phenomena underlying the reasoning process.

It may also be argued that our proposed formalism for logical inference is overly sophisticated with respect to the underlying computational model (implemented as a finite-state automata). However, it is important to notice that the finite-state automata represents the rules of the rotation domain, which serves as an example of application for the framework proposed. As put forth in [73]: for the logicist tradition in AI, it is not paramount to have a one-to-one correspondence between the logic description of the system and its implementation. Moreover, logic formalism forces the computational procedures, not the other way around. In other words, the encoding of the rules about rotation (our application example) into finite-state automata does not imply that every domain modeled using the approach proposed here should have the same computational realization. Thus, the assumption of a more complex spatial theory could lead to more complex computational procedures to be developed, without changing the underlying logical framework.

The present work does not take into account reasoning about actions and change (RAC) [56]; therefore, this work assumes perception as a passive process. The integration of the ideas developed above with a RAC formalism (the situation calculus) is subject of our current investigations. It is worth noting also that the knowledge needed in the interpretation process (i.e. the scene scripts discussed in Sect. 3.1) is hand-coded, we plan to investigate how the methods for inductive learning protocol behavior [74] and mathematical axioms [75] could be used to generate basic spatial axioms to support the image interpretation process discussed in this paper.

On the practical front, an important future extension of the research presented here is the replacement of the blob coloring algorithm with more robust image segmentation methods, such as [76, 77], in order to improve the input to the logic-based image interpretation method.

One issue for future investigation is how our method could operate on-the-fly, *e.g.*, as part of a computer vision system for a robot that roams around. In our prototype, the method has been implemented as an off-line process.

6 Conclusion

In this paper we introduced a logical formalism to account for the problem of interpreting sequences of images. This formalism brings together the notion of path semantics [57] and the spatial reasoning theory proposed in [24] providing, thus, a rigorous logical account for image sequence interpretation. There two main contributions of this work. First, a method was proposed for obtaining a logic formula that explains the geometrically observable changes that occurred in an image sequence. This method is an extension of the inference engine proposed [57] that provides a sound and complete account for the state update phenomenon. We show that the correctness of the proposed framework follows from these results. Second, the present work extends the qualitative spatial reasoning theory proposed in [19, 24, 31] by incorporating rules that account for the interpretation of long sequences of transitions, rather than only interpreting subsequent pairs of states. Our experiments show that the system correctly interprets rotational movements for objects of different colors and provides satisfactory results for interpreting such movements from perceptually indistinguishable objects.

Acknowledgements Marcus V. dos Santos was partially funded by the MIC (Ministry of Information and Communication), Korea, under the Foreign Professor Invitation Program of the IITA (Institute for Information Technology Advancement), and by NSERC (Natural Sciences and Engineering Research Council of Canada). Paulo Santos was partially funded by FAPESP, Sao Paulo, Brazil.

References

1. Randell D, Cui Z, Cohn A (1992) A spatial logic based on regions and connection. In: Proceedings of KR, Cambridge, pp 165–176
2. Moratz R, Renz J, Wolter D (2000) Qualitative spatial reasoning about line segments. In: ECAI, pp 234–238
3. Schlieder C (1996) Qualitative shape representation. In: Burrough PA, Frank AU (eds) Geographic objects with indeterminate boundaries. Taylor & Francis, London, pp 123–140
4. Freksa C (1992) Using orientation information for qualitative spatial reasoning. In: Theories and methods of spatial-temporal reasoning in geographic space. Lecture notes in computer science, vol 629. Springer, Berlin
5. Ligozat G (1998) Reasoning about cardinal directions. J Vis Lang Comput 9(1):23–44
6. Cohn AG, Hazarika SM (2001) Qualitative spatial representation and reasoning: an overview. Fundam Inform 46(1–2):1–29
7. Stock O (ed) (1997) Spatial and temporal reasoning. Kluwer Academic, Dordrecht
8. Randell DA, Cohn AG, Cui Z (1992) Computing transitivity tables: a challenge for automated theorem provers. In: Kapur D (ed) Proceedings of CADE, Saratoga Springs. Lecture notes in computer science. Springer, Berlin, pp 786–790
9. Cohn AG, Bennett B, Gooday J, Gotts N (1997) Representing and reasoning with qualitative spatial relations about regions. In: Stock O (ed) Spatial and temporal reasoning. Kluwer Academic, Dordrecht, pp 97–134
10. Cui Z, Cohn A, Randell D (1992) Qualitative simulation based on a logic of space and time. In: Proceedings of AAAI, California, pp 679–684
11. Gotts N (1994) How far can we 'C'? Defining a 'doughnut' using connection alone. In: Proceedings of KR, Bon, Germany, pp 246–257
12. Wolter F, Zakharyaschev M (2000) Spatio-temporal representation and reasoning based on RCC-8. In: Proceedings of KR, San Francisco, pp 3–14
13. Muller P (2002) Topological spatio-temporal reasoning and representation. Comput Intell 18(3):420–450
14. Randell D, Witkowski M (2002) Building large composition tables via axiomatic theories. In: Proceedings of KR, Toulouse, France, pp 26–35

15. Köhler C (2002) The occlusion calculus. In: Proceedings of cognitive vision workshop, Zürich, Switzerland
16. Reinz J, Nebel B (1999) On the complexity of qualitative spatial reasoning: a maximal tractable fragment of the region connection calculus. Artif Intell 108:69–123
17. Galton A (1994) Lines of sight. In: Proceedings of the seventh annual conference of AI and cognitive science, Dublin, Ireland, pp 103–113
18. Randell D, Witkowski M, Shanahan M (2001) From images to bodies: Modeling and exploiting spatial occlusion and motion parallax. In: Proceedings of IJCAI, Seattle, pp 57–63
19. Santos PE (2007) Reasoning about depth and motion from an observer's viewpoint. Spat Cogn Comput 7(2):133–178
20. Galton A (2000) Qualitative spatial change. Oxford University Press, Oxford
21. Galton A (1995) Towards a qualitative theory of movement. In: Spatial information theory, pp 377–396
22. Erwig M, Schneider M (2002) Spatio-temporal predicates. IEEE Trans Knowl Data Eng 14(4):881–901
23. Erwig M (2004) Toward Spatiotemporal Patterns. In: Spatio-temporal databases. Springer, Berlin, pp 29–54
24. Santos P, Shanahan M (2002) Hypothesising object relations from image transitions. In: van Harmelen F (ed) Proceedings of ECAI, Lyon, France, pp 292–296
25. Reiter R, Mackworth A (1989) A logical framework for depiction and image interpretation. Artif Intell 41(2):125–155
26. Matsuyama T, Hwang VS (1990) SIGMA: a knowledge-based image understanding system. Plenum, New York
27. Poole D, Goebel R, Aleliunas R (1987) Theorist: a logical reasoning system for defaults and diagnosis. In: Cercone N, McCalla G (eds) The knowledge frontier—essays in the representation of knowledge. Springer, Berlin, pp 331–352
28. Schroeder C, Neumann B (1996) On the logics of image interpretation: model construction in a formal knowledge representation framework. In: International conference on image processing, Switzerland, vol 2, pp 785–788
29. Neumann B, Möller R (2008) On scene interpretation with description logics. Image Vis Comput 26(1):82–101
30. Shanahan M (1996) Robotics and the common sense informatic situation. In: Proceedings of ECAI, Budapest, Hungary, pp 684–688
31. Santos P, Shanahan M (2003) A logic-based algorithm for image sequence interpretation and anchoring. In: Proceedings of IJCAI, Acapulco, Mexico, pp 1408–1410
32. Hazarika SM, Cohn AG (2002) Abducing qualitative spatio-temporal histories from partial observations. In: Proceedings of KR, Toulouse, France, pp 14–25
33. Fernyhough J, Cohn AG, Hogg DC (2000) Constructing qualitative event models automatically from video input. Image Vis Comput 18:81–103
34. Bennett B, Cohn A, Magee D (2005) Enforcing global spatio-temporal consistency to enhance reliability of moving object tracking and classification. Künstl Intell 2:32–35
35. Nagel H-H (1977) Analysing sequences of tv-frames: System design considerations. In: Proceedings of IJCAI, Cambridge, p 626
36. Tsotsos JK, Mylopoulos J, Covvey HD, Zucker SW (1980) A framework for visual motion understanding. IEEE Trans Pattern Anal Mach Intell 2(6):563–573, Special Issue on Computer Analysis of Time-Varying Imagery
37. Tsotsos JK (1985) Knowledge organization and its role in representation and interpretation for time-varying data: the ALVEN system. Comput Intell 1:16–32
38. Herzog G (1995) From visual input to verbal output in the visual translator. Technical Report 124, Universitat des Saarlandes
39. Herzog G, Wazinski P (1994) Visual Translator: linking perceptions and natural language descriptions. Artif Intell Rev 8(2–3):175–187
40. Gerber R, Nagel H-H, Schreiber H (2002) Deriving textual descriptions of road traffic queues from video sequences. In: Proceedings of ECAI, Lyon, France, pp 736–740
41. Nagel H-H (2000) Image sequence evaluation: 30 years and still going strong. In: Proceedings of ICPR, Barcelona, Spain, pp 1149–1158
42. Nagel H-H (1988) From image sequences towards conceptual descriptions. Image Vis Comput 6(2):59–74
43. Bouthemy P, François E (1993) Motion segmentation qualitative dynamic scene analysis from an image sequence. Int J Comput Vis 10(2):157–182
44. Mitiche A, Bouthemy P (1996) Computation and analysis of image motion: a synopsis of current problems and methods. Int J Comput Vis 19(1):29–55
45. Buxton H (2002) Learning and understanding dynamic scenes activity: a review. Image Vis Comput 21(1):125–136
46. Frank T, Haag M, Kollnig H, Nagel H-H (1996) Characterization of occlusion situations occurring in real-world traffic scenes. In: Proceedings of the workshop on conceptual descriptions from images, ECCV, Cambridge, UK, pp 43–57
47. Brand M (1997) Physics-based visual understanding. Comput Vis Image Underst 65(2):192–205
48. Brand M, Birnbaum L, Cooper P (1993) Sensible scenes: visual understanding of complex structures through causal analysis. In: Proceedings of AAAI, Washington, DC, pp 588–593
49. Brand M (1996) Understanding manipulation in video. In: Proceedings of the 2nd international conference on face and gesture recognition, pp 94–99
50. Siskind JM (1995) Grounding language in perception. Artif Intell Rev 8(5–6):371–391
51. Mann R, Jepson A, Siskind JM (1997) The computational perception of scene dynamics. Comput Vis Image Underst 65(2):113–128
52. Siskind JM (2000) Visual event classification via force dynamics. In: Proceedings of AAAI, Austin, pp 149–155
53. Hayes PJ (1984) The second naïve physics manifesto. In: Hobbs J, Moore RC (eds) Formal theories of the common sense world. Ablex, Norwood
54. Gärdenfors P (2000) Conceptual Spaces: the geometry of thought. MIT Press, Cambridge
55. Chella A, Frixione M, Gaglio S (2000) Understanding dynamic scenes. Artif Intell 123(1–2):89–132
56. Reiter R (2002) Knowledge in action. MIT Press, Cambridge
57. Bonner A, Kifer M (1993) Transaction logic programming. In: Proceedings of the tenth international conference on logic programming (ICLP). MIT Press, Cambridge, pp 257–279
58. Bonner A, Kifer M (1998) A logic for programming database transactions. In: Logics for databases and information systems. Kluwer Academic, Dordrecht
59. Ballard DH, Brown C (1982) Computer vision. Prentice Hall, Englewood Cliffs
60. Marr D (1982) Vision: a computational investigation into the human representation and processing of visual information. Freeman, San Francisco
61. Huang C (1990) Contour generation and shape restoration of the straight homogeneous generalized cylinder. Int Conf Pattern Recognit A 90:409–413
62. Cabalar P, Santos P (2006) Strings and holes: an exercise on spatial reasoning. In: Sichman J (ed) Proceedings of SBIA–IBERAMIA. Lecture notes in artificial intelligence, vol 4140. Springer, Berlin, pp 419–429
63. Santos P, Cabalar P (2007) Holes, knots and shapes: a spatial ontology of a puzzle. In: 8th international symposium on logical formalizations of commonsense reasoning (Commonsense'07), Stanford, CA
64. Anger F, Rodriguez R, Guesgen H, van Benthem J (1996) Space, time, and computation: trends and problems. Appl Intell 6:5–9

65. Harel D, Kozen D, Parikh R (1982) Process logic: expressiveness, decidability, completeness. J Comput Syst Sci 2(25):144–170
66. Guesgen H (2002) Reasoning about distance based on fuzzy sets. Appl Intell 17(3):265–270
67. Hernández D, Clementini E, di Felice P (1995) Qualitative distances. In: Spatial information theory. Lecture notes in computer science, vol 988. Springer, Berlin, pp 45–57
68. Newell A (1982) The knowledge level. Artif Intell 18(1):87–127
69. Freksa C (1991) Conceptual neighbourhood and its role in temporal and spatial reasoning. In: Decision support systems and qualitative reasoning. Elsevier Science, Amsterdam, pp 181–193
70. Kuipers B (1994) Qualitative reasoning: modelling and simulation with incomplete knowledge. MIT Press, Cambridge
71. Bonner A, Kifer M (1995) Transaction logic programming (or a logic of declarative and procedural knowledge). Tech. Rep. CSRI-323, University of Toronto, November 1995. http://www.cs.toronto.edu/~bonner/transaction-logic.html
72. Santore J, Shapiro S (2002) Identifying perceptually indistinguishable objects: Is that the same one you saw before? In: AAAI workshop on cognitive robotics, Edmonton, Canada, pp 96–102
73. Shanahan M (1999) What sort of computation mediates between perception and action? In: Logical foundations for cognitive agents: contributions in honor of Ray Reiter. Springer, Berlin, pp 352–369
74. Needham C, Santos P, Magee D, Devin V, Hogg D, Cohn A (2005) Protocols from perceptual observations. Artif Intell J 167:103–136
75. Santos P, Magee D, Cohn A, Hogg D (2004) Combining multiple answers for learning mathematical structures from visual observation. In: Proceedings of the 16th European conference on artificial intelligence (ECAI-04), Valencia, Spain
76. Dambreville S, Rathi Y, Tannen A (2006) Shape-based approach to robust image segmentation using kernel PCA. In: CVPR '06: proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition, Washington, DC. IEEE Computer Society, Los Alamitos, pp 977–984
77. Chen YB, Chen OT-C (2006) Robust image segmentation using modified edge-following scheme with automatically-determined thresholds. In: Proceedings of the first conference on innovative computing, information and control, vol 3, pp 292–295