Spatio-Temporal Dynamics

ECAI 2010 Workshop Proceedings

Mehul Bhatt, Hans W. Guesgen, Shyamanta M. Hazarika (Eds.)

SFB/TR 8 Report No. 023-06/2010
Report Series of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition
Universität Bremen / Universität Freiburg

Contact Address:

Dr. Thomas Barkowsky
SFB/TR 8
Universität Bremen
P.O.Box 330 440
28334 Bremen, Germany

Tel +49-421-218-64233
Fax +49-421-218-64239
barkowsky@sfbtr8.uni-bremen.de
www.sfbtr8.uni-bremen.de

ECAI 2010

Workshop proceedings

Spatio-Temporal Dynamics

European Conference on Artificial Intelligence
Lisbon, Portugal

16 August 2010

Organizing and Editorial Committee

Mehul Bhatt. SFB/TR 8 Spatial Cognition, University of Bremen
P.O. Box 330 440, 28334 Bremen, Germany
T +49 (421) 218 64 237, F +49 (421) 218 98 64 237
bhatt@informatik.uni-bremen.de

Hans W. Guesgen. School of Engineering and Advanced Technology Massey University, Private Bag 11222 Palmerston North 4442, New Zealand
T +64 (6) 356 9099 extn 7364, F +64 (6) 350 2259
h.w.guesgen@massey.ac.nz

Shyamanta M Hazarika, Computer Science and Engineering
School of Engineering, Tezpur University
Tezpur - 784028, Assam, India
smh@tezu.ernet.in

Advisory and Program Committee

Christophe Claramunt (Naval Academy Research Institute, France)
Debasis Mitra (Florida Institute of Technology, USA)
Ernest Davis (New York University, USA)
Hans Guesgen (Massey University, New Zealand)
Jochen Renz (Australian National University, Australia)
John Stell (University of Leeds, UK)
Kathleen-Stewart Hornsby (University of Iowa, USA)
Mehul Bhatt (University of Bremen, Germany)
Nico van de Weghe (Ghent University, Belgium)
Paulo E. Santos (Technical University FEI, Brazil)
Shyamanta Hazarika (Tezpur University, India)
Stefan Woelfl (University of Freiburg, Germany)
Thomas Bittner (SUNY at Buffalo, USA)

Invited Participation

Prof. Antony Galton
University of Exeter, United Kingdom

Organizing Institutions

SFB/TR 8 Spatial Cognition. University of Bremen, and University of Freiburg (Germany)

Massey University (New Zealand)

Tezpur University (India)

Contents

Preface

Welcome to the Workshop on Spatio-Temporal Dynamics at the European Conference on Artificial Intelligence 2010 in Lisbon, Portugal. The workshop marks a first point of departure from the founding workshops on Qualitative Spatial and Temporal Reasoning at ECAI, and is born out of a clearly recognised need within the community to pursue an application guided paradigm shift in fundamental research on formal methods in spatial information theory.

Knowledge Representation and Reasoning has been one of the key thrust areas within Artificial Intelligence research. Driven by the motivation for a qualitative approach for the embodiment of commonsense spatial knowledge in intelligent systems, Qualitative Spatial Information Theory has emerged as a discipline within Artificial Intelligence. Located within this discipline are specialisations concerned with the development of formal methods to represent and reason about Space, Time, Actions and Change.

Space, actions and change are inextricably linked: actions and events are a crucial connecting-link between space and spatial change, e.g., spatial configurations typically change as a result of interaction within the environment. Actions and events, both in a predictive as well as an explanatory sense, also constitute the mechanisms by which we establish and nurture commonsense knowledge about the world that we live in: our anticipations of spatial reality conform to our commonsense knowledge of the effects of actions and events in the real world. Similarly, our explanations of the perceived reality too are established on the basis of such apriori established commonsense notions. This view of Space, Actions and Change is general and applicable in a wide-range of application areas: qualitative spatial and temporal reasoning in general and formalising spatial change in particular is increasingly becoming a core issue within many application domains such as Robotics and Computer Vision, Ambient Intelligence and Smart Environments, Spatial / Architecture Design, GIS / Spatial Information Systems, Mobile and Location-based Computing.

The edited volume covers both theory and application-centric research in the area of spatio-temporal dynamics. Thrust is on research that focuses on formalising commonsense spatial knowledge and directs the integration of qualitative spatial reasoning with general approaches for reasoning about action and change. Applications that demonstrate the utility of well-established qualitative spatial and temporal calculi are also covered.

The proceedings of STeDy 2010 would be a contribution primarily to Spatio-Temporal Representation and Reasoning within Qualitative Spatial Information Theory. Additionally, it is envisaged that the results will also offer direct guidance to other AI Practitioners for the application of formal methods in spatio-temporal dynamics in their respective disciplines.

Mehul Bhatt, *Hans Guesgen*, *Shyamanta Hazarika*
(STeDy 2010 Co-Chairs)

Invited Talk

The Formalities of Affordance

It is an obvious truth that the possibilities for action and movement are conditioned by the physical spatial environment. In the terminology of J. J. Gibson, these possibilities are defined by the "affordances" of environmental features, and the key to being a successful agent in the physical world is being able to perceive and exploit these affordances. In this talk I want to explore to what extent it is possible to characterise different types of affordance in terms of familiar spatial and temporal logics such as the RCC systems and the interval calculus, e.g., to characterise formally such notions as "container", "passageway", "entrance", and "barrier", and the types of action or movement that are afforded (or "disafforded") by environmental features having these properties.

Antony Galton
University of Exeter
United Kingdom

The Formalities of Affordance

Antony Galton[1]

Abstract. It is an obvious truth that the possibilities for action and movement are conditioned by the physical spatial environment. In the terminology of J. J. Gibson, these possibilities are defined by the "affordances" of environmental features, and the key to being a successful agent in the physical world is being able to perceive and exploit these affordances. To what extent can affordances be characterised in terms of low-level environmental features using the methods of traditional logic-based commonsense knowledge representation? Following an introductory general discussion, this paper concentrates on a particular case, the affordance of containment, for which we give a sequence of successively more detailed and lower-level analyses.

1 Affordances and Image Schemas

The notion of *affordance* was introduced by J. J. Gibson as part of his radical "ecological" theory of perception. Whereas previous theories had held that an individual's perception of its environment must be mediated by percepts corresponding to the ever-shifting patterns of sensory stimulation to which the individual is subject, Gibson believed that the environment is perceived *directly*, in the form of the ambient array of surfaces constituting the environment within which the individual moves and acts. Although the patterns of sensory stimulation must clearly play a part in giving rise to the direct perception of the surrounding surfaces, they are not themselves perceived, but serve merely as conduits by which the information contained in those surfaces is brought to the attention of the perceiver. That we are not aware of the patterns of sensory stimulation themselves should be sufficiently obvious if we consider the case of the eye: if we were able somehow to observe the patterns of light falling on the retina, we would certainly not be able to discern from these the external world which, in practice, we perceive with such immediacy; instead, all we should see would be a "blooming buzzing confusion", as a result of the rapid movements of the eyeballs as well as the movements of the subject's own head and body.

A key feature of Gibson's theory is the further observation that the potentialities for movement and action afforded to an individual by its environment are inherent properties of the surface layout by which the environment is defined. As Gibson himself puts it,

> Perhaps the composition and layout of surfaces *constitute* what they afford. If so, to perceive them is to perceive what they afford. This is a radical hypothesis, for it implies that the "values" and "meanings" of things in the environment can be directly perceived. Moreover, it would explain the sense in which values and meanings are external to the observer. [11, p.127]

Thus we perceive directly that a firm, more or less horizontal surface supported about 50cm above the surrounding ground is suitable, if sufficiently wide and deep, for sitting on — it "affords sitting" — and a sufficiently high and wide aperture in a more or less vertical solid surface can be passed through (it "affords entering"). A well-known attempt to make explicit the physical properties that a surface layout must exhibit in order to be possessed of a certain affordance is that of Warren [25] who, amongst other things, shows experimentally that in order for a set of stairs to be climbable for a given subject, the ratio between the vertical height of each step and the subject's own leg-length should be not more than 0.88.

In investigating affordances we should distinguish between several distinct goals, all of which must be achieved before a complete theory can be obtained. We may refer to these as upper-, middle- and lower-level goals, and they may be formulated as follows:

1. The upper-level goal is to answer what may be called "ecological" or "environmental" questions concerning the role of affordances in the life of an individual, how they can be used to explain features of human and animal behaviour, and how they can be exploited for the better design of environments.
2. The middle-level goal is concerned with characterising exactly what affordances are: this may be called the "ontological" question. How is an affordance defined, and what is the logical relationship between statements about affordances and other statements about the world?
3. The lower-level goal is the answer the "aetiological" question of where affordances come from, exactly how the physical layout of surfaces determines the affordances it has for any given class of creatures.

As an example of the middle-level goal, Steedman [22, 23] considers the affordances associated with doors. He uses a *linear dynamic event calculus* to formalise such statements as that a door can be gone through if open, but not if shut; if it is pushed when shut, it becomes open, and vice versa; if one is inside, then the result of going through the door is to be outside, and vice versa. These capture the affordances of a door *qua* passageway as well as *qua* barrier. On the other hand, no consideration is given to the physical characteristics that something must have in order to be able to function as (i.e., possess all the relevant affordances of) a door.

As pointed out by Frank and Raubal [7] and elaborated further by Kuhn [17], affordances are closely related to *image schemas* [18], recurring patterns which we employ to structure our understanding of the world we live in, and which are presumed to play a fundamental role in human cognition and language. Examples of image schemas include CONTAINER and PATH: the link with affordances is obvious, since to be a container is precisely to *afford containment*, while to be a path is to *afford passage*. Thus at least in many cases image schemas may be characterised in terms of the affordances of actual exemplars of those schemas.

An example of the upper-level goal is Jordan *et al*'s sketch for an affordance-based model of *place* in GIS [16]. As is well recognised, the notion of place is complex, not to be reduced to some simplistic

[1] University of Exeter, UK, email: apgalton@ex.ac.uk

construct in terms of location. A place is a portion of the environment that can fulfil certain purposes of an agent or community of agents, e.g., "here is a good place to have our picnic". In order for X to be a good place for Y to Z it is necessary, at least, that X affords Zing to Y. Jordan *et al* provide a useful discussion of the many factors that need to be taken into account in giving an affordance-based model of place, though their claim to have presented a "methodology to model places with affordances" is perhaps overstated.

This paper is concerned less with the upper or middle level goals than with the comparatively neglected lower level: in virtue of what does a given surface layout possess a particular set of affordances? Some of the work of Warren cited above (e.g., concerning the physical requirements for steps to be climbable) falls under this category. This particular example is quantitative in nature: the climbability of steps is referred to numerical measurements of both steps and climber. This is obviously important, since the numerical measurements make all the difference. Nonetheless, the quantitive questions cannot really be asked unless certain qualitative conditions are satisfied first: to be a candidate for being a flight of stairs, for example, there must exist an appropriately configured sequence of alternating horizontal surfaces and vertical displacements, and in the absence of this, or a close enough approximation, there is nothing to measure!

In the spirit of qualitative reasoning in AI, our aim is to consider the low-level question from a qualitative point of view. In particular, we shall be concerned with the following question: To what extent can the affordance-generating features of surface layouts be specified in terms of simple qualitative calculi such as RCC [20]? The analysis will be very much in the spirit of the formalisations of commonsense knowledge exemplified by such works as [12, 13, 21, 5], in which the notion of affordance is certainly frequently implicit, even if not brought to the fore as the explicit focus of attention. In the space available, it will only be possible to look in detail at one particular type of affordance, that of *containment*.

2 A Few Preliminaries

An affordance is a potentiality for action offered by some environmental feature to an agent. Gibson stressed the *mutuality* implicit in this definition — it takes two to make an affordance, that which affords something, and that to which it affords it. Thus the formal expression of affordance must be relational in nature. The "action" which is afforded does not necessarily involve motion (e.g., a text affords reading, a bed affords sleeping) but in the most typical cases it does so. Hence the formal expression of affordance will often involve an analysis of some kind of motion. For this, we require an appropriate grounding in spatial and temporal representations.

2.1 Spatial regions

In this paper we use the well-known RCC system of [20], and specifically the following relations:

$P(x, y)$	x is part of y
$PP(x, y)$	x is a proper part of y
$TP(x, y)$	x is a tangential part of y
$TPP(x, y)$	x is a tangential proper part of y
$EC(x, y)$	x is externally connected to y
$O(x, y)$	x overlaps y
$PO(x, y)$	x partially overlaps y

We treat these as relations between spatial regions rather than objects (but see below, §2.2). Spatial relations between objects are expressed using RCC relations between their *positions*. The position of object o at time t, denoted $pos(o, t)$, is a spatial region which coincides with the spatial extent of o at t. Note that pos encodes both position and shape: if o changes shape, then $pos(o, t)$ changes. Use of this notation does not presuppose a "Newtonian" notion of absolute space: as in everyday life, positions are always specified in a relative way, by reference to some framework of objects which, for the purposes at hand, can be regarded as fixed (e.g., the walls of a room, the surface of the earth) even though from some wider perspective they may be regarded as moving. The use of pos only presupposes that we have some such framework implicitly to hand. For a detailed discussion of the related notion of "relative place", see [6].

We shall not attempt to define exactly what a spatial region is, but merely content ourselves with the observation that a spatial region is a possible position for an object. As such, a region is paradigmatically three-dimensional, since material objects are. However, we will also need to refer to the boundary (or surface) of a three-dimensional region, and this is of course two-dimensional. We write ∂x to denote the boundary of x. Other spatial notions needed are the *convex hull* of a region r, denoted $cvhull(r)$, and the relation of *congruence* between spatial regions, denoted $Congruent(r_1, r_2)$. This must be stipulated to be an equivalence relation, and in addition it should satisfy the rule that any part of a region congruent to a given region z must be congruent to part of z, i.e.,

$$P(x, y) \wedge Congruent(y, z) \rightarrow \exists u(Congruent(x, u) \wedge P(u, z)) \quad (1)$$

but no further attempt will be made to define it here.

Although we make no commitment to a representation of regions as sets of points, we will make use of the set-theoretic notations for union ($\cup$), intersection ($\cap$), and difference ($\setminus$) on the assumption that some suitable analogous operations would be definable in any "pointless" theory of regions.

2.2 Physical objects

Physical objects include both *material* objects, which are made of matter, and *non-material* objects such as holes (and in particular the inner spaces of containers) which are dependent on material objects but not themselves material. Both material and non-material pbjcts have positions. The predicate $Material(x)$ is used to say that an object is material.

We shall use a parallel series of RCC relations, notated P^*, PP^*, TP^*, etc, to apply to physical objects, where "connection" is now understood to mean physical attachment rather than spatial contiguity. Thus objects are EC^* if they are actually joined together, DC^* if not. If objects o_1 and o_2 are not joined together but are touching at time t, the relation between them can be expressed as $DC^*(o_1, o_2) \wedge EC(pos(o_1, t), pos(o_2, t))$.

Material objects are characterised by the *principle of non-interpenetrability*, which says that non-overlapping material objects cannot simultaneously occupy overlapping positions, i.e.,

$$Material(o_1) \wedge Material(o_2) \wedge \neg O^*(o_1, o_2) \rightarrow \forall t \neg O(pos(o_1, t), pos(o_2, t)) \quad (2)$$

2.3 Time

We shall have cause to refer to specific motion events. A number of different formalisms are available for this purpose, notably the

method of temporal arguments, event-type reification, and event-token reification; these methods are described and compared in [10]. In this paper, we use the method of temporal arguments, by which, to say that an event of type E occurs over an interval $[t_1, t_2]$, we write $E(t_1, t_2)$, using the terms t_1 and t_2 as temporal arguments to the predicate E expressing the event type. Conversion to the other formalisms is mostly straightforward, if it is desired to go on to exploit the greater expressivity of those formalisms.

2.4 Modality and possible futures

Since it refers to what *can* happen rather than to what *does* happen, affordance is a *modal* notion. Its formal expression must therefore use either modal logic or some other formalism capable of expressing an appropriate notion of modality. It is a non-trivial task to specify exactly the notion of modality we require, and some discussion of this is needed before we can proceed further.

Affordances are important because of their role in determining possible future actions: affordance is a potentiality, and what is now afforded, and therefore potential, may become actualised in the future. If we talk about the affordance that something had at some past time, we are implicitly referring to the possible futures running forward from that time. Thus the form of modality appropriate for describing affordances is future-directed: with reference to any time point, we are interested in its different possible futures, but regard its past as fixed. The possibility operator we will use may be characterised informally as follows: $\Diamond P$ is true at t if and only if there is some possible future of t such that, if that future is the actual future, then P is true at t. This can be described formally in terms of the *history structures* of [1], in which this operator is notated ⧫.[2] Under this interpretation, the logic of $\Diamond$ is given by the modal system S5.

This does not, however, fully characterise the meaning of the operator. What does "possible future" mean? To illustrate the problem, consider a slot machine which will accept 1 euro coins: this means (at least) that the slot affords entry by a 1 euro coin. Does the slot afford entry by a metal sphere of diameter 12mm? The practical commonsense answer is "no": the sphere is too wide to fit into the slot. But what if I take a hammer and flatten the sphere into a disk? Then it will surely fit into the slot (the volume of the sphere is a little less than that of a 1 euro coin). The natural reply to this is to say that this is "cheating": it is not what we meant when we asked if the slot could admit the metal sphere. But now consider this case: I have written a letter on an A4 sheet of paper; I have an A5-sized envelope: can I use it to post my letter? This time the answer is surely "yes": I can fold my letter in two and slip it into the envelope. The envelope affords containment for an A4 sheet. Somewhere between folding a sheet of paper in two and hammering a metal sphere flat lies the borderline between those possible histories which we wish to allow for the purpose of defining affordances and those which we do not. But where exactly? The paper-folding is more easily reversible than the sphere-flattening: but it is not completely reversible, since you can never get rid of a crease in a sheet of paper. Perhaps the key lies in the notion that folding a letter to fit it into an envelope is an entirely normal and expected procedure: it is what we do. Flattening a sphere to fit it into a slot intended for coins is highly unusual and only likely to be done under exceptional circumstances.

We therefore do not want our modal operators to range over all conceivable futures, or even all physically possible ones. Somehow we must restrict our attention to those futures in which exceptional, affordance-disrupting events do not occur except perhaps in exceptional circumstances. It is in terms of these futures that the modal operators $\Box$ and $\Diamond$ are to be interpreted. Of course, to say this is to say virtually nothing until we have characterised what "normal" or "expected" means. We acknowledge the ultimate necessity of doing this, but meanwhile proceed to the technicalities of characterising affordances on the assumption that some suitable definition of the modal operators can be given.

[2] This operator is also used — notated M — in [8, Ch.7].

2.5 Rigidity

Modality enters into the definition of another physical property, namely *rigidity*, which will be important in what follows. A material object is rigid if it *cannot* change shape. In reality, of course, absolute rigidity is a fiction, but in practice many objects can be treated as if they were rigid, and in particular for the logic of containment the distinction between rigid objects such as apples and boxes and non-rigid objects such as bags and scarves (and human bodies!) is important. An object is rigid if all of its possible positions are congruent:

$$\begin{aligned} &Rigid(o) =_{\mathrm{df}} \\ &\forall t \forall t' \forall r \forall r' (\Diamond(pos(o,t) = r) \wedge \Diamond(pos(o,t') = r') \rightarrow \\ &\qquad\qquad Congruent(r, r')) \end{aligned} \tag{3}$$

3 Case study: The Affordance of Containment

What is a container? It is hard to give a non-circular answer. A container is something which can contain other things. What does it mean to contain something? For A to contain B is for B to be in A. What does "in" mean? "A is in B" means that B contains A ... and we are back where we started.

We might say that for A to contain B is for A to constrain the position of B in a certain way. For example, the coins in my pocket go wherever my pocket goes, unless they are taken out. The water in a jug is held in place by the jug — without the jug, the water would spread out and find its way to the lowest accessible spaces. But if a man is in a house, in what sense is his position constrained by the house? What about a tree in a field?

3.1 Contained space

We do not attempt to *define* containers or the containment relation here; but we can at least try to say as much as we can about it that is clear, definite, and formalisable. To this end, we make use of the notion of the *contained space* of a container, introduced by Hayes [13] in the context of containers for liquids. A cup, for example, is a solid ceramic object used for containing liquids; its contained space is the space partially enclosed by the material of the cup, within which anything contained by the cup is located. As Hayes says, the contained space "is *not* a physical object but is characterized by a certain capacity and by being in a certain relation to a container". In our terminology, it is a non-material object dependent on the container. The contained space of a container is well-defined since there is a point beyond which, if more liquid is added, it will overflow; the surface of the liquid at this point defines the upper boundary of the contained space. For solid or granular matter, it is harder to specify that part of the boundary of the contained space which is not shared with the container itself. We will not attempt to address this problem here.[3]

[3] Note also that for solids, we often speak of containment even when only part of the contained object is actually in the contained space of the container, e.g., a vase containing flowers, where only the stalks of the flowers are actually inside the vase. Such examples have been discussed extensively in the literature on spatial prepositions [15, 24, 3].

The contained space $cs(x)$ of a container x belongs to x, but is not part of it. It is not a spatial region, since it may be located at different spatial regions at different times; as x moves around, $cs(x)$ moves with it.[4] In particular, $cs(x)$ is always contiguous with x, i.e.,

$$Container(x) \rightarrow \forall tEC^*(cs(x), x).$$

and it is always located within the convex hull of the region occupied by x:

$$Container(x) \rightarrow \forall t(P(pos(cs(x), t), cvhull(pos(x), t))).$$

A container is *closed* when the boundary of its contained space forms part of the boundary of the container itself:

$$Closed(x, t) =_{\text{df}} Container(x) \wedge P(\partial pos(cs(x), t), \partial pos(x, t)).$$

When a container is not closed, the connected components of $\partial pos(cs(x), t) \setminus \partial pos(x, t)$ are (again following [13]) called *portals* of the container. They are the entrances and exits by which objects can enter or leave the container. Many containers exhibit both open and closed states: a box with a hinged lid, for example, is open when the lid is raised, and closed when it is down.

3.2 Containment

To say that container c contains object o at a given time is to say that the spatial region occupied by o at that time is part of the spatial region occupied by the contained space of c, i.e., [5]

$$Contains(c, o, t) =_{\text{df}} P(pos(o, t), pos(cs(c), t)) \quad (4)$$

To formalise the notion that c affords containment for o, we need to say that c *can* contain o. At time t, this will be the case so long as it is possible for c to contain o at some time at or in the future of t:

$$CanContain(c, o, t) =_{\text{df}} \exists t'(t \leq t' \wedge \Diamond Contains(c, o, t')). \quad (5)$$

It is implicit in the use of $\Diamond$ here that o can be moved over to c and can enter it, with either or both undergoing changes of shape needed to allow this to happen — all such changes being of the kind we have called "normal" or "expected".

3.3 Containment and rigidity

In general, we do not wish to make any assumptions concerning the rigidity or otherwise of containers and what can be put in them. We have four distinct cases here, as shown in the following table:

	Rigid container	Non-rigid container
Rigid object	Apple in a box	Apple in a bag
Non-rigid object	Scarf in a box	Scarf in a bag

We consider the form taken by $CanContain$ in the case where c and o are both rigid. It seems obvious that if a container is rigid, then so is its contained space; so we postulate the rule

$$Container(c) \wedge Rigid(c) \rightarrow Rigid(cs(c)) \quad (6)$$

Hence $cs(c)$ is rigid as well. Assume $CanContain(c, o, t)$. From (5) this means there is a time t' not earlier than t such that $\Diamond Contains(c, o, t')$, i.e., from (4),

$$\Diamond P(pos(o, t'), pos(cs(c), t')).$$

What regions exist cannot vary between different possible futures, so this implies there are regions r_1 and r_2 such that

$$P(r_1, r_2) \wedge \Diamond(pos(o, t') = r_1) \wedge \Diamond(pos(cs(c), t') = r_2).$$

By (3), since o and $cs(c)$ are both rigid, this implies[6]

$$P(r_1, r_2) \wedge Congruent(r_1, pos(o, t)) \wedge Congruent(r_2, pos(cs(c), t)).$$

Finally, from (1), since congruence is an equivalence relation, we obtain $\exists u(Congruent(pos(o, t), u) \wedge P(u, pos(cs(c), t)))$. Thus, a rigid container can contain a rigid object only if the latter is congruent to part of the contained space of the latter:

$$CanContain(c, o, t) \wedge Rigid(c) \wedge Rigid(o) \rightarrow \exists u(Congruent(pos(o, t), u) \wedge P(u, pos(cs(c), t)))$$

3.4 Trajectories and continuity

Although we have seen how to say that a container can contain an object, we have not really addressed our original question, which is *by virtue of what* does this potentiality obtain? Consider the case in which object o is outside container c at time t_0, and inside it at later time t_1. Over the interval $[t_0, t_1]$ both o and c may change position and shape. We can track the values of $pos(o, t)$ and $pos(c, t)$ as t runs from t_0 through to t_1. These specify the *trajectories* of o and c. A condition for an object to come to be inside a container is that suitable trajectories for both the object and the container exist.

A *trajectory* is simply a continuous sequence of spatial regions. Formally, it may be represented by a continuous function

$$traj : [0, 1] \rightarrow \mathcal{R},$$

where $\mathcal{R}$ is the set of all spatial regions.[7] Exactly what is meant by "continuous" here needs discussion. A number of approaches to this have been suggested in the literature. One way is to adopt a four-dimensional view, and try to characterise continuity in terms of the shape of the spatio-temporal extent of the motion considered as a region in four dimensions [19, 14]. Another approach, closer in spirit to our current enterprise, is to characterise continuity in terms of some metric on the space of possible regions [9, Ch. 7][4, 5]; metrics considered include the Hausdorff distance and variations on that, and the volume of the symmetric difference. For each such metric Δ, a trajectory $traj$ may be characterised as continuous with respect to that metric in the usual way, i.e.,

$$\forall t \in [0, 1] \forall \epsilon > 0 \exists \delta > 0 \forall t' \in [0, 1] (|t - t'| < \delta \rightarrow |traj(t) - traj(t')| < \epsilon).$$

[4] The ontology of contained spaces is similar to the ontology of holes [2], sharing many of the same problems and difficulties.

[5] We do not need to specify that c is a container; if c is not a container, $cs(c)$ can be defined to be the null object, so $pos(cs(c), t)$ will also be null, and hence $Contains(c, o, t)$ is necessarily false.

[6] Here we are also using the trivial facts $\Diamond(pos(o, t) = pos(o, t))$ and $\Diamond(pos(cs(c), t) = pos(cs(c), t))$.

[7] Expressions of the form $traj(x)$ should be understood as "syntactic sugar" for something along the lines of $value(traj, x)$, where $traj$ is a term rather than a function symbol; this will allow us to quantify over trajectories without breaking the bounds of first-order logic. But for ease of reading we shall retain the notation $traj(x)$.

The particular trajectories we are interested in are sequences of possible positions of an object. The following formula says that o follows trajectory $traj$ over the interval $[t_0, t_1]$:

$$Follows(o, traj, t_0, t_1) =_{\mathrm{df}} \forall t \left(t_0 \leq t \leq t_1 \rightarrow pos(o,t) = traj\left(\tfrac{t-t_0}{t_1-t_0}\right)\right)$$

Of course, this trajectory cannot be followed if there are obstacles in the way; but this need not be specified explicitly, since given $Follows(o, traj, t_0, t_1)$, non-interpenetrability (2) already implies that when o is at any point in the trajectory, no other body overlaps the position it then occupies. It is not necessary for the whole trajectory to be unoccupied by other objects throughout $[t_0, t_1]$: an obstacle is fine so long as it is removed when you get to it.

Continuity of motion is now secured by means of the rule

$$\begin{array}{l}\forall t \forall t' \forall r \forall r' (t < t' \wedge pos(o,t) = r \wedge pos(o,t') = r' \rightarrow \\ \quad \exists traj (traj(0) = r \wedge traj(1) = r' \wedge \\ \qquad Follows(o, traj, t_0, t_1)))\end{array} \tag{7}$$

3.5 Entry into a container

In order for o to come to be inside c during the interval $[t_0, t_1]$, o and c must follow trajectories which begin with o and c in positions such that o is outside c, and end with them in positions such that c contains o, and which are such that at no time do the positions of o and c overlap. This motion can be divided into three parts: first, o and c get into a position where o is "just outside" c; then o actually enters c, arriving at a position where it is "just inside" c; and finally, it may proceed to a resting position inside c. The middle phase is the crucial one: this is what we will call the *entering* event.

We must now formally characterise the actual process of entering. This begins at the latest time when o is "just outside" c, i.e., $EC(pos(o,t), pos(cs(c),t))$, and ends at the first time when o is "just inside" c, i.e., $TPP(pos(o,t), pos(cs(c),t))$. We can put

$$\begin{array}{l}Enters(o, c, t_0, t_1) =_{\mathrm{df}} \\ \quad \exists traj_o \exists traj_c (\\ \qquad Follows(o, traj_o, t_0, t_1) \wedge \\ \qquad Follows(c, traj_c, t_0, t_1) \wedge \\ \qquad \forall t (t_0 \leq t \leq t_1 \rightarrow \\ \qquad\quad EC(pos(o,t), pos(cs(c),t)) \leftrightarrow t = t_0 \wedge \\ \qquad\quad TPP(pos(o,t), pos(cs(c),t)) \leftrightarrow t = t_1))\end{array} \tag{8}$$

It follows from non-interpenetrability (2) that the positions of o and c never overlap, i.e., we have $\neg O(pos(o,t), pos(c,t)))$ at all times t. Hence we do not need to specify this explicitly in (8).

For $t_0 < t < t_1$, we have $PO(pos(o,t), pos(cs(c),t))$, and if we assume that o is a one-piece object, this means that o must intersect a portal p of c. We will return to the implications of this later.

It should be emphasised that in order to get o inside c, either or both bodies may need to change shape (cf. the table above). This is allowed for in (8), since there is no reason why the values of $traj_o$ should all be congruent, and similarly for the values of $traj_c$. In particular, c may be closed initially; but this does not matter so long as a portal has opened at the time o needs to enter it.

From continuity, it seems plausible that

$$\begin{array}{l}\neg O(pos(o,t), pos(cs(c),t)) \wedge Contains(c, o, t') \rightarrow \\ \quad \exists t_0 \exists t_1 (t \leq t_0 < t_1 \leq t' \wedge Enters(o, c, t_0, t_1))\end{array} \tag{9}$$

Can we prove this from the rules and definitions we have given so far? If not, what further rules are needed? These are currently unanswered questions.

The definition (8) tells us what it is for o to enter c; but if we are interested in affordances, we want rather to specify what it means to say that o *can* enter c. The obvious definition is

$$CanEnter(o, c, t) =_{\mathrm{df}} \exists t' \Diamond Enters(o, c, t, t') \tag{10}$$

In particular, we would like to prove the modalised version of (9):

$$\begin{array}{l}\neg O(pos(o,t), pos(cs(c),t)) \rightarrow \\ \quad (CanContain(o, c, t) \leftrightarrow \\ \qquad \exists t' (t \leq t' \wedge CanEnter(o, c, t')))\end{array} \tag{11}$$

This deceptively simple formula is none the less highly significant. The predicate $CanContain$ expresses the bare affordance of containment, which (4) defines as the potentiality for actual containment. This is, as we noted, a rather high-level view of the affordance, abstracting away from the features of the world in virtue of which containment is afforded in any particular situation. By invoking the principle of continuity of movement, we were able to express a more detailed precondition for the affordance of containment, namely, not just that an object can be situated inside a container, but that it can come to be there, in other words that there is a trajectory by which it can enter it. This is what is expressed by $CanEnter$; it gives us a somewhat lower-level view of the affordance. Formula (11) links these two views, the higher-level to the lower, by asserting in effect that they refer to the same underlying reality.

3.6 Entry at a portal

A still lower-level view is possible. Let us return to the observation that during the entering event, o must intersect a portal of c. Consider informally the preconditions for o to be able to enter c via the portal p. One is that c can contain o, i.e., its interior can be so located that o lies wholly inside it. Another is that there is a continuous series of cross-sections of o, each of which fits inside p. Here p and the cross-sections of o are two-dimensional entities. These conditions are not so far sufficient, as can be seen from Figure 1, where the vase is large enough to contain the ball, and every cross-section of the ball fits into the entrance portal of the vase (shown by the dotted line), but still the ball cannot enter the vase assuming both are rigid. (Of course a rubber ball could be squeezed past the constriction in the neck of the vase.) We need an additional condition, that there is a possible position of o inside c that is tangential to p.

How do we express these conditions formally? We assume here that o is a one-piece object (as opposed to, for instance, a two-piece object such as a teapot which consists of a body and a separate lid). Then $pos(o,t)$ is always a connected spatial region, which means that any two points within it can be joined by a one-dimensional

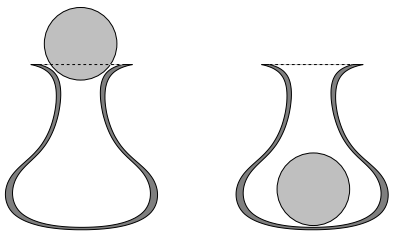


Figure 1. The vase could contain the ball if only the ball could get into it.

path lying wholly within the region. If $PO(pos(o,t), pos(cs(c),t))$, then from the definition of PO, part of o lies outside $cs(c)$ and part of o lies inside $cs(c)$. For each pair of points in $pos(o,t)$, one of which lies outside $cs(c)$ and the other inside, there is a path between them which (i) lies wholly within $pos(o,t)$ and (ii) intersects the boundary $\partial pos(cs(c),t)$. Let x be the sum of all the intersections of such paths with $\partial pos(cs(c),t)$, so we have $P(x, pos(o,t)) \wedge P(x, \partial pos(cs(c),t))$. By non-interpenetrability (2), o does not overlap c, and hence no part of o overlaps the boundary of c. In particular, $\neg O(x, \partial pos(c,t))$. This means that we have $P(x, \partial pos(cs(c),t) \setminus \partial pos(c,t))$, i.e., x is part of the portals of c.

By formalisation of this argument, one might hope to prove, from the principles enunciated so far, the following formula:

$$\begin{array}{l} Enters(o,c,t_0,t_1) \rightarrow \\ \quad \forall t(t_0 < t < t_1 \rightarrow \\ \quad\quad \exists r_1 \exists r_2 (pos(o,t) = r_1 \cup r_2 \wedge \\ \quad\quad\quad\quad \neg O(r_1, pos(cs(c),t)) \wedge \\ \quad\quad\quad\quad P(r_2, pos(cs(c),t)) \wedge \\ \quad\quad\quad\quad P(\partial r_1 \cap \partial r_2, \partial pos(cs(c),t) \setminus \partial pos(c,t)))) \end{array} \tag{12}$$

Note that this does not imply that $\partial r_1 \cap \partial r_2$ (our earlier x) is part of just one portal of c. A horseshoe-shaped object could have its two ends inserted into different entrances of a container with more than one entrance; but if the object is to enter the container, then all of it must pass through just one portal eventually. No doubt an argument based on continuity should enable us to establish this, but again, the details are at present unclear.

4 Concluding remarks

To summarise what we have done, we began with the goal of characterising in formal terms the conditions under which it can be said that a certain affordance exists, namely the affordance of containment which a container has in relation to an object. We began with a very high level characterisation which amounted to little more than a definition of what it means to say that one thing can contain another. This was the definition (5). By combining this definition with the condition of continuity (7), we were able to spell out a lower-level condition for the container to afford containment to the object, namely that it is possible for the object to enter the container; this is expressed in the definition of $CanEnter$ (10), the full details of which are contained in the definition of the $Enters$ predicate (8). Finally, by invoking the principle of non-interpenetrability (2), we were able, at least informally, to tease out a still lower-level condition for the affordance, relating the portals of the container to the sequence of cross-sections of the object which must intersect the portal as it enters the container. This was expressed, in part, by (12).

The general approach may be summarised as follows. To define what it means for some object or collection of objects to afford some action A to an object o, we begin by defining what it means for o actually to perform A, and then use a modalised form of this definition to provide a high-level definition of the affordance itself. Then, by invoking general principles such as continuity and non-interpenetrability, we tease out successively lower-level conditions for the affordance to exist. In this way, we gradually approach the goal of specifying just what it is about any particular physical layout that results in its having the affordances that it does. To relate this back to the original source of the affordance idea in Gibson's theories of perception, we can now say that we are able to perceive affordances by perceiving these lower-level conditions, which, we must assume, are more directly accessible to our perceptual apparatus.

It has to be admitted that so far much of this is programmatic. Even to handle fully the one case considered in this paper, namely containment, requires more detailed formal work than it has been possible to present here. Then there is whole field of enquiry ripe for investigation: affordance of shifting, lifting, hiding, opening, closing, and all the other potentialities offered by our environment which define the scope and limits of human action in the world.

REFERENCES

[1] B. Bennett and A. P. Galton, 'A unifying semantics for time and events', *Artificial Intelligence*, **153**, 13–48, (2004).

[2] Roberto Casati and Achille Varzi, *Holes and Other Superficialities*, MIT Press, Cambridge, MA and London, 1994.

[3] K. Coventry and S. Garrod, *Saying, Seeing and Acting: The Psychological Semantics of Spatial Prepositions*, Psychology Press, 2004.

[4] Ernest Davis, 'Continuous shape transformations and metrics of regions', *Fundamenta Informaticae*, **46**, 31–54, (2001).

[5] Ernest Davis, 'Pouring liquids: A study in commonsense physical reasoning', *Artificial Intelligence*, **172**, 1540–1578, (2008).

[6] M. Donnelly, 'Relative places', *Applied Ontology*, **1**(1), 55–75, (2005).

[7] Andrew U. Frank and Martin Raubal, 'Formal specification of image schemata — a step towards interoperability of geographic information systems', *Spatial Cognition and Computation*, **1**(1), 67–101, (1999).

[8] A. P. Galton, *The Logic of Aspect: an Axiomatic Approach*, Clarendon Press, Oxford, 1984.

[9] A. P. Galton, *Qualitative Spatial Change*, Oxford University Press, Oxford, 2000.

[10] A. P. Galton, 'Operators vs arguments: The ins and outs of reification', *Synthese*, **150**, 415–441, (2006).

[11] James J. Gibson, *The Ecological Approach to Visual Perception*, Houghton Mifflin, Boston, 1979.

[12] Patrick J. Hayes, 'The naïve physics manifesto', in *Expert Systems in the Micro-Electronic Age*, ed., D. Michie, 242–70, Edinburgh University Press, (1979).

[13] Patrick J. Hayes, 'Naive physics I: Ontology for liquids', in *Formal Theories of the Commonsense World*, eds., J. R. Hobbs and R. C. Moore, 71–107, Ablex, Norwood, New Jersey, (1985).

[14] S. M. Hazarika and A. G. Cohn, 'Qualitative spatio-temporal continuity', in *Spatial Information Theory: Foundations of Geographic Information Science*, ed., D. R. Montello, 92–107, Springer, Berlin, (2001).

[15] Annette Herskovits, *Language and Spatial Cognition*, Cambridge University Press, Cambridge, 1986.

[16] T. Jordan, M. Raubal, B. Gartrell, and M. J. Egenhofer, 'An affordance-based model of place in GIS', in *Eighth International Symposium on Spatial Data Handling, Vancouver, Canada*, eds., T. Poiker and N. Chrisman, pp. 98–109, (1998).

[17] Werner Kuhn, 'An image-schematic account of spatial categories', in *Spatial Information Theory, Proceedings of the 8th International Conference, COSIT 2007, Melbourne, Australia*, eds., S. Winter, M. Duckham, L. Kulik, and B. Kuipers, pp. 152–168. Springer, (2007).

[18] G. Lakoff and M. Johnson, *Metaphors We Live By*, University of Chicago Press, Chicago, 1980.

[19] Philippe Muller, 'A qualitative theory of motion based on spatio-temporal primitives', in *Principles of Knowledge Representation and Reasoning: Proceedings of KR'98*, eds., A.G. Cohn, L.K. Schubert, and S.C. Shapiro, San Mateo, Calif., (1998). Morgan Kaufmann.

[20] D. A. Randell, Z. Cui, and A. G. Cohn, 'A spatial logic based on regions and connection', in *Principles of Knowledge Representation and Reasoning: Proceedings of KR'92*, eds., B. Nebel, C. Rich, and W. Swartout, pp. 165–76, San Mateo, CA, (1992). Morgan Kaufmann.

[21] Murray Shanahan, 'An attempt to formalise a non-trivial benchmark problem in common sense reasoning', *Artificial Intelligence*, **153**, 141–165, (2004).

[22] Mark Steedman, 'Formalizing affordance', in *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*, pp. 834–839, (2002).

[23] Mark Steedman, 'Plans, affordances, and combinatory grammar', *Linguistics and Philosophy*, **25**(5), 723–753, (2002).

[24] Claude Vandeloise, *L'Espace en français*, Editions du Seuil, Paris, 1986.

[25] W. H. Warren, 'Constructing an econiche', in *The Ecology of Human-Machine Systems*, eds., J. Flach, P. Hancock, J. Caird, and K. Vicente, 210–237, Erlbaum, Hilldsale, NJ, (1995).

The Use of Change Identifiers to Update Footprints of Dot Patterns in Real Time

Maximillian Dupenois[1] and **Antony Galton**[2]

Abstract. Commonly, in the field of spatial knowledge representation, there is a need to assign to a group of individual entities, considered as an aggregate, a spatial location known as its 'footprint'. The problem of finding an appropriate footprint for an aggregate in a static context has been fairly thoroughly researched, but little thought has been given to possible changes of the footprint over time resulting from the movement of individuals into, out of, or within the aggregate. For many practical applications, it is required to track the footprint of a 'live' dynamic system such as a crowd or flock. This paper looks at the problems involved in maintaining footprints over non-static dot patterns and how to negotiate the trade-offs between efficiency of computation and accuracy of result. The key notion is to use 'change identifiers' to determine when and how often the footprint of a moving aggregate should be updated. Preliminary results from an implemented system are presented.

1 Introduction

In spatial information theory one often encounters the problem of representing groups or aggregates, which at a fine level of granularity appear as pluralities with a scattered distribution but at a coarser granularity may be treated as single coherent individuals with their own behaviour and properties. Familiar examples from the everyday world include forests (i.e., aggregates of trees), flocks and crowds (aggregates of animals or people), and conurbations (aggregates of buildings).

In recent research, attention has been paid to the problem of assigning a spatial location to an aggregate considered as a unit, given as inputs the spatial locations of the individual components [5, 9]. In abstract form, the two-dimensional problem is, given a set of *dots* (i.e., objects sufficiently compact to be idealised as points) in the plane, to determine a *footprint* representative of the spatial distribution of the collection of dots taken as a whole. The footprint will be a two-dimensional region, which, depending on one's purposes, may be required to satisfy various contraints such as polygonality, connectedness, topological regularity, convexity, etc [6]. The problem generalises to three dimensions in the obvious way, but for simplicity the discussion in this paper will be restricted to the two-dimensional version.

It cannot be too strongly emphasised that there does not exist a uniquely "best" or "correct" footprint for a given dot pattern. In [8] it was shown experimentally that footprints selected as "good" by human subjects represent optimal trade-offs between the conflicting goals of minimising the area and minimising the perimeter, but this certainly does not tell the whole story. For human purposes, an important feature of a good footprint is that it "looks" right, that is, it represents a shape that we "see" in the dot pattern itself. But for some purposes, one might prefer to use a footprint that is very easily computed (e.g., the minimal axis-aligned bounding rectangle) or which has well-known mathematical properties (e.g., the convex hull), even though in many cases these do not provide a close visual match to the dot pattern.

Many different algorithms for generating footprints from dot sets have been proposed, in contexts such as geographical information theory [1, 9], pattern recognition [13, 5], computer vision [11], and computational geometry [7], to cite only a few representative examples. In all these cases, however, the assumption is that the dot patterns are *static*. In reality, many examples of collectives or aggregates are dynamic, with either the location or the membership, or both, varying over time [16]. Of our examples above, flocks and crowds vary in both these respects over a short time scale; forests and conurbations also vary, but the time-scale is typically several orders of magnitude greater. There has been some work on tracking aggregates but most of this has been centred around object tracking within video (where the pixels are the dots), [4] and [2]. However, there is a difference between tracking a fixed shape amongst background noise and maintaining a footprint of a possibly changing dot pattern. There is also the work reported in [12] concerning tracking herds, but this is less interested in the shape of the footprint, focusing instead on higher level features of its evolution, represented by four major 'evolvements': *expand, join, shrink* and *leave*.

The problem we address in this paper is how to track the footprint of dynamically changing aggregates of dots. In the case of fast-moving aggregates, an added constraint is that the tracking should take place in real time. Footprint algorithms typically run in time $O(n \log n)$ or worse (sometimes much worse), where n is the number of dots. Hence recomputing the footprint *ab initio* every time there is a change in the dot pattern will be computationally costly, making real-time recomputation infeasible in many cases. Importantly we have limited the data we expect to simply the coordinate positions of the dots, this is to keep the system as general as possible, although information such as identity may allow for extensions to the change identifiers.

One possible approach would be to look for a way to *update* the footprint incrementally rather than recompute it entirely. In an ideal world, one could do this in such a way that the footprint assigned to the dots at any time is always identical to the footprint that would be obtained if it were recomputed. In general, for most types of footprint it is unlikely that such exact tracking can be accomplished with significantly less cost than recomputing the footprint every time.

Instead, we propose a method by which the position of the dots

[1] University of Exeter, England, email: m.p.dupenois@ex.ac.uk
[2] University of Exeter, England, email: a.p.galton@ex.ac.uk

in relation to the most-recently computed footprint is continuously monitored, and the footprint is only recomputed when the mismatch between the dot positions and the current footprint exceeds some preassigned threshold of accuracy. Clearly there will be a trade-off between the level at which the accuracy threshold is set and the resultant frequency of recomputation, and we investigate the nature of this trade-off with a view to optimising it.

Our approach is to use a suite of easily computable *change identifiers*, each with its own threshold. Recomputation of the footprint is triggered when some aggregate value computed from the values returned by the change identifiers exceeds a given threshold. In the simplest form this aggregate value could be a count of how many of the change identifiers individually exceed their thresholds, amounting in effect to a vote amongst the change identifiers. Alternatively, the change identifiers could be ranked in order of importance and a weighted combination of their values compared with some threshold. We investigate the effect of using different sets of change identifiers, and different ways of combining the results returned by them.

The plan for the remainder of this paper is as follows. In §2 we fix some terminology and formalise the approach described above in the form of an algorithm presented in pseudocode. In §3 we discuss a range of possible change identifiers, evaluating them in terms of their ease of computation and informativeness in relation to the task in hand. In §4 we consider combinations of change identifiers, and discuss the computation of aggregate values and thresholds. In §5, we provide a theoretical analysis of the kinds of results to be expected from running the algorithm, and in §6 we describe the current state of our implementation and some preliminary results. Finally, in §7 we summarise the results obtained so far and outline our plans for future work.

2 Process

The basic process we implement is shown as Algorithm 1, which works as follows. The incoming data consists of a sequence of dot patterns (which might come from, e.g., observations relayed by sensor arrays). The dot pattern associated with time step i is denoted DP_i, and is referred to as the **current dot pattern** when i is the current time. An algorithm for generating footprints from dot patterns is assumed given (we shall refer to this as the **footprint algorithm**), and at the beginning of the sequence a footprint $f(DP_0)$ is generated for dot pattern DP_0 and saved as the **stored footprint** SFP_0. The dot pattern DP_0 from which it is generated is stored as the **stored dot pattern** (SDP_0). At subsequent time steps, the change identifiers are used to determine whether a new footprint should be computed; this is done by evaluating the extent to which the current dot pattern DP_i differs from the previously stored dot pattern SDP_{i-1}. If this value, $eval(DP_i, SDP_{i-1}, SFP_{i-1})$, exceeds some pre-set threshold, then a new footprint $f(DP_i)$ is generated as the new stored footprint SFP_i, and the current dot pattern is used as the new stored dot pattern DP_i. Otherwise, the stored dot pattern and footprint are retained from the previous time step. For any dot pattern DP_i, the footprint $f(DP_i)$ that would be computed from it (whether or not this computation actually takes place) will be referred to (admittedly somewhat tendentiously, bearing in mind the non-uniqueness of the footprint) as the **true footprint** for that dot pattern.

3 Change Identifiers

Each change identifier returns a value representing some measure of change. To produce this value it has access to the stored dot pattern, the current dot pattern and the stored footprint. Most of the identifiers listed below do not use the stored footprint; this enables them to be used in conjunction with a wide range of footprint algorithms, since they make no assumptions concerning the nature of the footprint (e.g., whether it must be polygonal, can have holes or multiple components, etc.). To assess whether the value it returns should force a footprint update, a threshold is associated with each change identifier; and if change identifiers are to be combined, a method to normalise their values is required. These ideas are discussed in the section on change identifier sets (§4). We describe the case where a change identifier exceeds its threshold as a 'failure' since in this case the stored footprint is deemed to have failed to represent the current dot pattern accurately.

Algorithm 1 Main Process

```
1: i = 0
2: Input first dot pattern DP_0
3: SFP_0 = f(DP_0)
4: SDP_0 = DP_0
5: repeat
6:    i = i + 1
7:    Input DP_i
8:    if eval(DP_i, SDP_{i-1}, SFP_{i-1}) > threshold then
9:       SDP_i = DP_i
10:      SFP_i = f(DP_i)
11:   else
12:      SDP_i = SDP_{i-1}
13:      SFP_i = SFP_{i-1}
14:   end if
15: until No more input available
```

The identifiers listed below are not exhaustive and we are not presenting them as a definitive final set; they do, however, cover a range of possible transformation types the dot pattern could undergo, e.g., changes in position, changes in distribution, and changes in membership of the dot pattern. For ease of reference we assign to each change identifier a label in SMALL CAPITALS.

3.1 Change in centroid scaled by the bounding box: CENTROID

This change value is given by the distance between the centroids of the current dot pattern and the stored dot pattern. It is normalised by dividing it by the diagonal of the bounding box of the stored dot pattern. If the dot pattern has n dots, the total computation time is $O(n)$ (If the dots are held in a suitable tree data structure, the bounding boxes can be found in time $O(\log n)$, but this does not reduce the overall order-of-magnitude complexity.)

3.2 Change in variance from the centroid: VARIANCE

The difference between the variances of the current and stored dot patterns.[3] We use variance rather than standard deviation so as to avoid the processing time involved in computing the square root. This measure can also be computed in time $O(n)$.

3.3 Change in axis-aligned medians: MEDIAN

This is given by the distance between the 'medians' of the current and stored dot patterns, where the (axis-aligned) median of a dot pat-

[3] The variance is the mean squared distance of the dots from the centroid.

tern is defined as the point whose coordinates are the medians of the x-coordinates of the dots and the y-coordinates of the dots respectively. This is analogous to the centroid but computed using the median rather than the mean. It has the advantage of not being sensitive to outliers, however, unlike the centroid, it is not rotation-invariant.

3.4 Percentage change in number of dots: DOTS

This is the difference in number of dots between the current dot pattern and the stored dot pattern as a percentage of the number of dots in the stored dot pattern. This can be computed in $O(n + i)$ time, where i is the number of dots from the previous pattern.

3.5 Change in bounding box: BOUNDINGBOX

This is given by the area of the symmetric difference between the bounding boxes of the current and stored dot patterns; it can be computed as the sum of the areas of the bounding boxes, less twice the area of their intersection. For purposes of normalisation, it is expressed as a fraction of the area of the bounding box of the stored dot pattern. If the dots are held in a two-dimensional tree data structure, this can be computed in time $O(\log n)$.

3.6 Proportion of points outside the boundary of the stored footprint: OUTSIDE

The fraction of dots outside the current footprint. By using the ray-casting method [15] we can find this in $O(nm)$ time, where m is the number of edges of the footprint. This is only sensibly applied if the footprint algorithm does not allow outliers, i.e. dots present in the dot pattern but not in the completed footprint. It should be noted that this is the only change identifier on our list which makes use of the stored footprint.

4 Change Identifier Sets

The change identifiers are used to compute the term $eval(DP_i, SDP_{i-1}, SFP_{i-1})$ used in Algorithm 1. While it is certainly possible to use any one of the change identifiers on its own for this purpose, it seems likely, given the relatively undiscriminating nature of each of them considered individually, that better results will be achieved when a group of two or more identifiers is used, with their values combined in some way to give $eval(DP_i, SDP_{i-1}, SFP_{i-1})$. In our implementation to be described below, we use an xml file to collect together change identifiers to this end.

There are two important decisions to make when using more than one identifier: Are they evaulated in order? and how are their values combined? We wanted the system to allow for different value choices such that we could run multiple setups and compare how effective they are, so the xml has element types for various parameters. The identifiers all have a priority associated with them, they are then evaluated in ascending order. The set can be run concurrently, but it was found that for small dot sets the time taken to start the threads would be slower than the time taken to run the footprint algorithm. The xml has elements for giving different thresholds to each individual identifier, a change identifier is considered to have failed if the amount of change it returns exceeds the value of the threshold parameter. There is a total threshold parameter that is attached to the set, the identifiers' values are summed and if the result is greater than the total threshold value the set is considered to have failed and the footprint is redrawn.

However, the identifiers have different scales of measurement, so that, for example, to add BOUNDINGBOX directly to VARIANCE would be to combine two very different units together and therefore may give undue importance to one identifier over another. A multiplier parameter is applied to the change value of the identifier before it is added to the total value to mitigate the effect of such inequalities. If the bias can not be handled by the multiplier, the set can also have an integer parameter setting a threshold of how many individual identifiers are allowed to fail.

5 Analysis

The purpose of using change identifiers is to enable the evolution of a footprint to be tracked more efficiently than by recomputing the footprint at each time step. The footprint is only recomputed when the change identifiers indicate that the dot pattern has changed sufficiently to make the mismatch with the current stored footprint unacceptably great. The number of footprint recomputations, and hence the total time taken to process a given sequence of dot patterns, will depend on the change identifiers used, and the threshold settings. We define variables as follows:

- $t_{FP}(i)$ is the time taken to compute the footprint from the dot pattern at step i.
- $t_{CI}(i)$ is the time taken to evaluate the change identifiers at step i.
- $r(i)$ is a Boolean variable, set to 1 if the footprint is in fact recomputed at step i, and zero otherwise.

The total computation time over a run of n dot patterns is thus

$$T_{CI} = t_{FP}(0) + \sum_{i=1}^{n} (t_{CI}(i) + r(i)t_{FP}(i)).$$

The value of T_{CI} is minimum when the change identifier threshold is set so high that the footprint is never recomputed after the start of the sequence (so $r(i) = 0$ for $1 \leq i \leq n$):

$$T_{\min} = t_{FP}(0) + \sum_{i=1}^{n} t_{CI}(i).$$

It is maximum when the change identifier threshold is set so low that the footprint is recomputed at every time step (so $r(i) = 1$ for all i):

$$T_{\max} = t_{FP}(0) + \sum_{i=1}^{n} (t_{CI}(i) + t_{FP}(i)).$$

If change identifiers are not used at all, and the footprint recomputed at every time step, then the total time taken is

$$T_{NCI} = \sum_{i=0}^{n} t_{FP}(i) = t_{FP}(0) + T_{\max} - T_{\min}.$$

If it is assumed that always $t_{CI}(i) < t_{FP}(i)$ (for if not, there would be little point in using change identifiers) then $T_{\min} < T_{NCI} < T_{\max}$, so the relative size of T_{CI} and T_{NCI} — which provides a measure of the time advantage, if any, gained by using change identifiers — depends on the threshold settings.

This time advantage must be set against the accuracy with which the footprint is tracked. The cost of using change identifiers comes from the fact that, most of the time, the stored footprint differs from

the true footprint. To measure this cost, we need a way of quantifying the extent of this mismatch. The difference between two footprints can be measured in various ways, e.g., using Hausdorff distance, or symmetric area difference (see [10, Ch. 7] for a discussion). Here we will use only the symmetric area difference, but the principles described below would apply equally to other measures.

The symmetric difference between two regions comprises the parts of each region that do not overlap the other; it is given by

$$R_1 \Delta R_2 = (R_1 \setminus R_2) \cup (R_2 \setminus R_1) = (R_1 \cup R_2) \setminus (R_1 \cap R_2).$$

We use the area of this as a measure of the dissimilarity between two footprints; and since we are only interested in comparisons, not absolute values, we normalise this area by expressing it as a fraction of the area of the 'true' footprint ($FP_i = f(DP_i)$). Thus the aggregate mismatch between the stored footprint and the true footprint over a dot-pattern sequence of length n is given by

$$mismatch = \sum_{i=0}^{n} \frac{||FP_i \Delta SFP_i||}{||FP_i||},$$

where $||X||$ denotes the area of region X.

If the footprint is recomputed every time, corresponding to total computation time $T_{\max}$, we have $SFP_i = FP_i$ for every i, so $mismatch = 0$. At the other extreme, the maximum value of $mismatch$ is obtained when the footprint is never recomputed, corresponding to $T_{\min}$. There is thus a trade-off between accuracy and computation time, as indicated in Figure 1, where different choices of change identifier thresholds correspond to different positions on the curve. The optimal setting for the change identifier threshold depends on the relative importance attached to the conflicting goals of minimizing both computation time and accumulated footprint error; but in any case no time advantage can be obtained for mismatches below the value m at which $T_{CI} = T_{NCI}$.

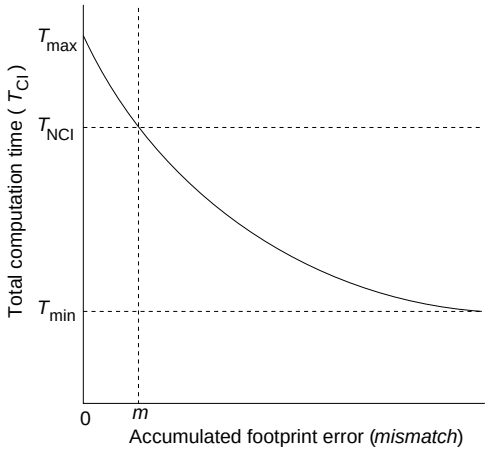


Figure 1. Total computation time against aggregate footprint error

6 Implementation

6.1 System

We have implemented a system to test the ideas presented in this paper. The system is split into modular parts: the engine, the change identifiers, the application and the footprint algorithms. The application initialises an instance of the engine to which it passes the footprint algorithm to use and the identifier set to process, then it starts the instance. The engine sits in a waiting state checking an internal queue to see if it has dot patterns to process. The application passes dot patterns to the engine; as in a live system it does not wait for a response but sends them continuously. The engine processes the dot patterns and notifies the application each time it generates a new stored footprint. Once the application has sent all the patterns to the engine it sends a command to stop.

If we are running this instance as a test then the engine also processes the dot patterns without using the change identifiers, recomputing the footprint every time, and records the following data:

- Time taken to run the engine over the entire set of dot patterns.
- Time taken to run the footprint algorithm.
- The state of the change identifier set at each timestep:
  - How long each change identifier took to evaluate.
  - Which change identifiers failed.
  - The value each change identifier returned.
  - What the total change was.
  - If the change identifier set enforced recomputation of the footprint, then which change identifier(s) caused the set to fail.
- Time taken to run the set.
- The current dot pattern at each time step.
- The stored footprint at each time step.
- The 'true' footprint at each time step.

By running this control we can see how much time is saved using the change identifier sets and draw similarity comparisons, giving us quantitative data to see how far the stored footprint deviates from the 'true' footprint at any time step. We use the methods described in §5 to produce two graphs: the first plots the symmetric area difference against time step, and the second plots the computation time for each time step. Results from some preliminary tests using this method are described below.

The other component of note in the system is a properties holder linked to the dot patterns. The change identifiers typically compare values computed from the current and stored dot patterns. But of course, any stored dot pattern was once current, so its value for each identifier will have been computed already. It is inefficient to compute it again so the pattern stores it in a mapping table once it is first computed.

6.2 Current Results

We have run tests on streams of 500 dot patterns containing up to 1000 dots each. We have implemented a collective motion pattern generator which can use different methods to produce streams of dot patterns. The method that generated the patterns for the current tests makes use of the principles of separation, cohesion and aggregation used to define behaviours in the Boids system of [14]. The footprint algorithm used is the upper and lower convex hull algorithm as given in [3]. A separate program has been written to showcase the two footprints for each timestep (one with change identifiers the other without) and time details from the test (See Figure 2).[4]

Currently the only two change identifiers for which full tests have been run are BOUNDINGBOX (§3.5) and DOTS (§3.4). Both of these

[4] The screenshot is from a smaller test than the one mentioned above so that the footprints are clearly visible on the small image

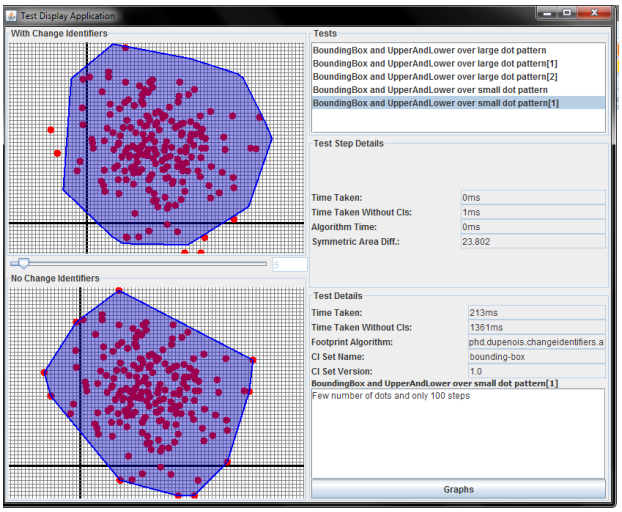


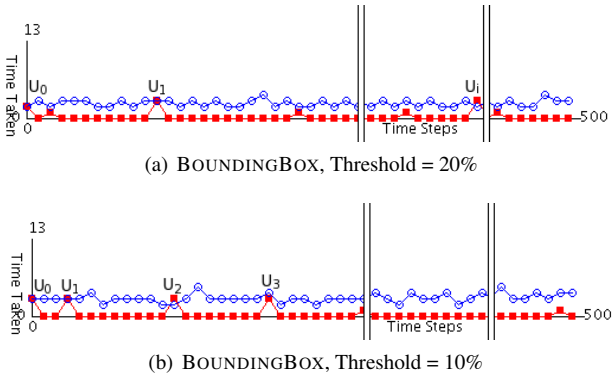
Figure 2. Screenshot of Result Display App.

consistently show better run times for when using change identifiers than when not, for a range of threshold settings.

The results display application produces the two graphs described in §6.1. The computation time graph (Figure 3) has two lines: The squares are on the line representing the run with change identifiers, and the circles are on the line representing the run without. The vertical double bars represent where the graph has been cut and stitched — with 500 steps the graph is too long to display in its entirety. As would be expected, the computation times when change identifiers are used is consistently less than when they are not; in fact it generally takes less than 1ms to run and therefore is less than 1ms over the footprint algorithm time when it updates. The time steps at which it updates can clearly be seen on the graph at U_0–U_i. Figure 3 shows the case where the change identifier is BOUNDINGBOX and the threshold is set at 20% (Figure 3(a)) and 10% (Figure 3(b)). The 10% threshold updates more often and we have a total time (T_{CI} in Figure 1) of 90ms for the run compared to 61ms for the 20%, but both are far below the times for the comparison runs which update at each timestep, with total times (T_{NCI}) of 1331ms for the 20% run and 1342ms on the 10%.[5]

The symmetric area difference graph (Figure 4) also clearly shows the update times (U_0–U_i). More interesting is what it can tell us about the change of the dot pattern. The frequency with which these updates occur shows us how static the dot pattern is and, if we know the change identifier(s) used, how it changed. The area difference graph for threshold 20%(Figure 4(a)) levels out towards the end, although the cropping obscures this. This levelling out indicates that the bounding box of the dot pattern did not change by over 20% for these time steps. The area difference during this static period is around 16%; if this is within allowed footprint error then we are saving large amounts of time across the period by not updating. If, however, 16% is considered too great a footprint difference then we need to change the threshold values on the identifier set to update earlier. Figure 4(b) shows a run with the BOUNDINGBOX threshold set at 10%, and as mentioned above, this causes many more updates. Significantly, Figure 4(b) does not level out as Figure 4(a) does, showing that lowering the threshold picked up changes ignored by the larger value. The accumulated errors (as described in §5) for Figure 4(b) and Figure 4(b) are 4545.5 and 2826 respectively; these may seem large but are accumulated over 500 time steps and give us an average error of 9.091 and 5.652 per time step. Whether or not these are acceptable will depend on specific application requirements.

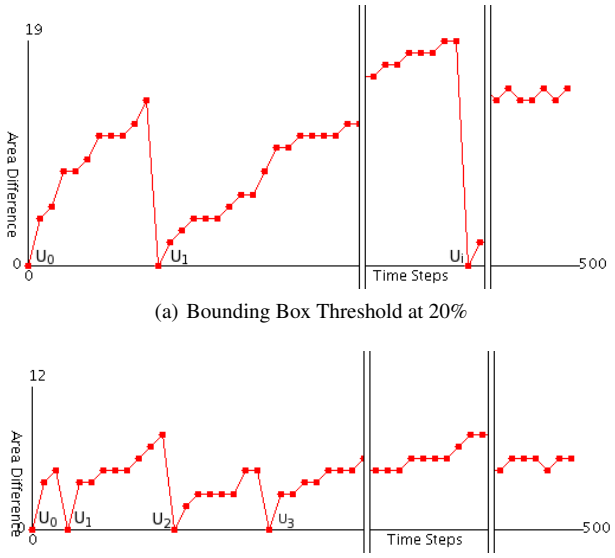
[5] Note that, in an ideal world, these values for T_{NCI} would be equal, since if no change identifiers are used the difference in threshold is irrelevant; the small difference actually found merely reflects the fact that in a real-world computing environment there will always be some variation in computation times even for identical computations.



(a) BOUNDINGBOX, Threshold = 20%

(b) BOUNDINGBOX, Threshold = 10%

Figure 3. Graph of Time Taken against Time Steps



(a) Bounding Box Threshold at 20%

(b) Bounding Box Threshold at 10%

Figure 4. Graph of Footprint Area Difference against Time Steps

7 Conclusions and Further Work

The principles behind the change identifiers appear to be sound. The graphs show a consistent saving of 5ms per time step using only the BOUNDINGBOX change identifier. There have not yet been enough tests performed to allow a systematic comparison of the usefulness

of different change identifiers, but the change in bounding box has shown itself to be able to identify dot pattern changes and update accordingly.

The continuation of this work includes implementing the rest of the change identifiers and running basic tests on them, as with BOUNDINGBOX, to see if they affect the update times with any regularity. Once done, an application using the principles of optimisation will be created to sort through the variations of change identifier sets over a particular dot pattern stream with a particular footprint algorithm. The results of this will be plotted on a graph of accumulated footprint error against time taken, as described in §6.1. This application will need to be run over several footprint algorithms and dot pattern streams.

With regard to the different types of footprint algorithm; the χ-hull algorithm from [5], the α-shape from [7] and the swinging-arm algorithm from [9] will be implemented. The majority of non-convex footprint algorithms require some external parameter (α in the α-shape, line length in the χ-hull and arm length in the swinging-arm), but fortunately the selection of this parameter does not greatly concern us. Our immediate concern is with how well we can track the footprint, not how appropriate the footprint is for the dot pattern.

In [16], several types of collective movement are described, and having sets of dot pattern streams that replicate these movements would lend weight to the accuracy rating of the change identifiers. It could show that the identifier in question was accurate over all types, accurate only for some, or for none. As well as this archetypal data, we want to apply the system to real-world examples.

Another, purely qualitative, method of judging the performance of the system is to appeal to human intuition. We can record the streams as 'movies' of the footprint evolving with the dot pattern. These movies can be played to a group of experimental subjects who are asked to rate how well they felt the footprint kept up with the dot pattern. Importantly the test should be set up in such a way that the notion of a good footprint is disentangled from how well it can be tracked. Results from this experiment would indicate just how important people think accuracy is. Data from the experiment may enable us to state which change identifier sets give acceptable accuracy for high efficiency and may help us say something about what properties of the dot pattern are most important when generating a footprint.

Also of interest will be the comparison between the quantitative and the qualitative data. Comparing the accuracy assessments from the human-subject study with the results from the quantitative testing may tell us which change identifiers are most important to human intuition.

Other accuracy measures, e.g., Hausdorff distance, will also be implemented, and it will be interesting to see how they relate to each other. A side interest will be to see how they relate to the accuracy ratings from the human study, as it may be that one of the measures is, implicitly, more used by the human mind than others.

REFERENCES

[1] A. Arampatzis, M. van Kreveld, I. Reinbacher, C. B. Jones, S. Vaid, P. Clough, H. Joho, and M. Sanderson, 'Web-based delineation of imprecise regions', *Computers, Environment and Urban Systems*, **30**, 436–459, (2006).

[2] Shai Avidan, 'Ensemble tracking', in *In CVPR*, pp. 494–501, (2005).

[3] Mark Berg, Otfried Cheong, Marc Kreveld, and Mark Overmars, *Computational Geometry: Algorithms and Applications*, Springer, 3rd edn., April 2008.

[4] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer, 'Kernel-based object tracking', *IEEE Transactions On Pattern Analysis and Machine Intelligence*, **25**(5), 564–577, (2003).

[5] M. Duckham, L. Kulik, M. Worboys, and A. Galton, 'Efficient generation of simple polygons for characterizing the shape of a set of points in the plane', *Pattern Recognition*, **41**(10), 3224–3236, (2008).

[6] Max Dupenois and Antony Galton, 'Assigning footprints to dot sets: An analytical survey', in *Spatial Information Theory: Proceedings of the 9th International Conference COSIT 2009*, eds., K. S. Hornsby, C. Claramunt, M. Denis, and G. Ligozat, pp. 227–244, Berlin, (2009). Springer.

[7] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, 'On the shape of a set of points in the plane', *IEEE Transactions on Information Theory*, **IT-29**(4), 551–559, (1983).

[8] A. P. Galton, 'Pareto-optimality of cognitively preferred polygonal hulls for dot patterns', in *Spatial Cognition VI: Learning, Reasoning and Talking about Space*, eds., C. Freksa, N. S. Newcombe, P. Gärdenfors, and S. Wölfl, pp. 409–425. Springer, (2008).

[9] A. P. Galton and M. Duckham, 'What is the region occupied by a set of points?', in *Geographic Information Science: Proceedings of the 4th International Conference, GIScience 2006*, eds., M. Raubal, H. J. Miller, A. U. Frank, and M. F. Goodchild, pp. 81–98. Springer, (2006).

[10] Antony Galton, *Qualitative Spatial Change*, Oxford University Press, 2000.

[11] Gautam Garai and B. B. Chaudhuri, 'A split and merge procedure for polygonal border detection of dot pattern', *Image and Vision Computing*, **17**, 75–82, (1999).

[12] Yan Huang, Cai Chen, and Pinliang Dong, 'Modeling herds and their evolvements from trajectory data', in *GIScience '08: Proceedings of the 5th international conference on Geographic Information Science*, pp. 90–105, Berlin, Heidelberg, (2008). Springer-Verlag.

[13] M. Melkemi and M. Djebali, 'Computing the shape of a planar points set', *Pattern Recognition*, **33**, 1423–1436, (2000).

[14] Craig W. Reynolds, 'Flocks, herds and schools: A distributed behavioral model', in *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 25–34, New York, NY, USA, (1987). ACM.

[15] Paul S. Heckbert and Eric Haines, *Berg, Mark and Cheong, Otfried*, chapter A Ray Tracing Bibliography, Morgan Kaufmann, 2002.

[16] Zena M. Wood and Antony P. Galton, 'A taxonomy of collective phenomena', *Applied Ontology*, **4**, 267–292, (2009).

Qualitative Spatio-Temporal Data Integration and Conflict Resolution

Jan Oliver Wallgrün[1] and **Frank Dylla**[2]

Abstract. We describe an approach for integrating qualitative spatio-temporal information from different knowledge bases. The method is able to resolve conflicts ranging from inconsistencies within a single knowledge base to contradictions between the different knowledge bases and to violations of spatial integrity constraints stemming from background knowledge. The input information consists of temporally delimited qualitative constraint networks representing temporal snapshots of spatial configurations, for instance stemming from a dynamic GIS. The developed merging operator extends distance-based approaches to qualitative merging based on the notion of conceptual neighborhood to spatio-temporal information by formalizing the notion of a continuous sequence of constraint networks with a minimal deviation from the input information. In addition to proposing the theoretical merging operator, we address computational and algorithmic aspects of implementing the operator.

1 INTRODUCTION

With today's trend towards decentralized data storage and ad hoc cooperation, the problem of adequately combining information from different sources and resolving potential conflicts between different knowledge bases is becoming more and more important [12]. While the need to represent and process spatio-temporal information about events and processes in an integrated way has been widely recognized [3, 14, 26], we still need methods able to deal with spatio-temporal information coming in the form of a simple snapshot-like representation [13], for instance in order to derive the integrated representations from mainly unprocessed data. In this paper, we therefore address the problem of how spatio-temporal information from different sources can be integrated in the most suitable way. The approach we develop is tailored to spatial representation formalisms developed in the AI subfield of qualitative spatial reasoning (QSR) [4, 23], suitable for tasks in which precise quantitative information is not available or not desirable. More precisely, we are considering representations based on so-called qualitative calculi which not only describe a set of spatial or temporal relations suitable for representation but also provide algebraic operations that allow for reasoning about the relations. Performing data integration on a qualitative level can also be advantageous when the input information is given quantitatively, but the result is supposed to satisfy a set of integrity constraints provided in qualitative form.

Approaches for integrating static qualitative spatial information and resolve potential conflicts have been proposed during the last years [7, 5, 6, 24] under the label of merging approaches for qualitative constraint networks, which essentially are qualitative relational descriptions of spatial configurations based on a single qualitative calculus. These approaches are based on the idea of using conceptual neighborhood between the relations of a qualitative calculus [9] to define a distance over constraint networks [2, 18]. Condotta et al. [5, 6] relate this approach to work on logic-based information fusion [12, 16]. In contrast to these operators which are *model-based* in the sense that the result only depends on the consistent scenarios of the input networks, we propose a *relation-based* approach in [24].

We now extend this idea from static spatial information to qualitative spatio-temporal information. The resulting approach uses the idea of describing spatial change on a qualitative level via a path through the conceptual neighborhood graph [8] for configurations of more than two objects [20, 21]. The input information contained in the knowledge bases consists of temporally delimited qualitative constraint networks representing temporal snapshots of spatial configurations, for instance stemming from a dynamic GIS (cf. Figure 5 for a concrete example). The developed spatio-temporal merging operator Γ formalizes and yields a continuous sequence of constraint networks with a minimal deviation from the input information. It is able to resolve several kinds of conflicts: (1) inconsistencies within a single knowledge base, (2) contradictions between different knowledge bases, and (3) violations of spatial integrity constraints stemming from background knowledge. We discuss computational and algorithmic aspects of our theoretical merging operator, pointing out that general implementations of the operator will be too expensive to be applicable in practice and, hence, good approximations or solutions for specific subproblems are needed.

We start by laying down the required background knowledge from the area of qualitative spatial and temporal reasoning and introduce our notations. In Section 3, we then develop our theoretical spatio-temporal integration approach and define the merging operator. Section 4 discusses implementational issues and describes an approximate algorithm for computing the merging result.

2 BACKGROUND

In the following, we recapitulate fundamental concepts from the area of qualitative temporal and spatial reasoning focusing on qualitative calculi, the notion of conceptual neighborhood, and distance-based merging approaches for static qualitative spatial information.

2.1 Qualitative calculi

A *qualitative calculus* $\mathcal{C}$ defines a set $\mathcal{B}_\mathcal{C}$ of spatial or temporal relations over a domain of objects $\mathcal{D}_\mathcal{C}$, i.e. points, lines, regions, etc. in the case of a spatial calculus or time points or intervals in the case of a temporal calculus.[3] For every pair of objects from the domain,

[1] IRTG GRK 1498 Semantic Integration of Geospatial Information, University of Bremen, Germany, wallgruen@informatik.uni-bremen.de

[2] SFB/TR 8 Spatial Cognition, University of Bremen, Germany, dylla@sfbtr8.uni-bremen.de

[3] In this work, we restrict ourselves to calculi over binary relations. However, our approach can also be adapted for calculi over relations of higher arity.

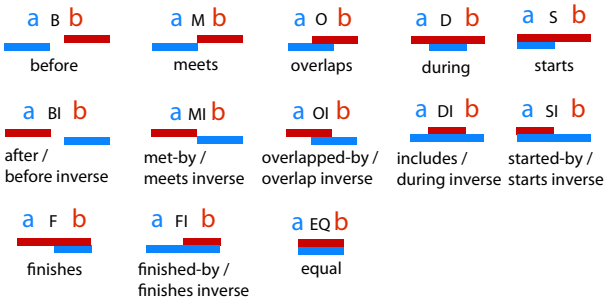


Figure 1. The thirteen base relations of Allen's Interval Algebra.

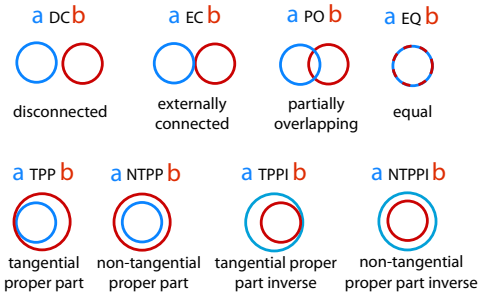


Figure 2. The eight base relations of the RCC-8 calculus.

exactly one relation from this set of so-called *base relations* holds (i.e., $\mathcal{B}_\mathcal{C}$ is jointly exhaustive and pairwise disjoint). Figures 1 and 2 illustrate the base relations of two well-known qualitative calculi: Allen's Interval Algebra (AIA) [1], a temporal calculus dealing with relations between time intervals, and the Region Connection Calculus (RCC-8) [22], a spatial calculus dealing with topological relations between regions.

To be able to express incomplete or imprecise spatial knowledge, a qualitative calculus actually is concerned with the so-called set of general relations $\mathcal{R}_\mathcal{C}$ containing all possible unions of base relations. For instance using RCC-8, given $r = \text{EC} \cup \text{DC} \cup \text{PO}$, $A\ r\ B$ would express that A is either *disconnected from*, *externally connected to*, or *partially overlapping* B. Complete ignorance is expressed by the universal relation $U = \bigcup_{b \in \mathcal{B}_\mathcal{C}} b$. We here adopt the often used way of notating general relations as sets of base relations instead of unions, meaning that $\mathcal{R}_\mathcal{C} = 2^{\mathcal{B}_\mathcal{C}}$ and that the relation above will be denoted as $A\ \{\text{EC}, \text{DC}, \text{PO}\}\ B$. Another special relation is the empty relation $\varnothing$ which cannot be realized by any pair of objects.

In addition to defining relations, a qualitative calculus also defines a set $\mathcal{O}_\mathcal{C} = \{\cap, \cup, ^-, ^\smile, \circ\}$ of operations over $\mathcal{R}_\mathcal{C}$. $\cap$, $\cup$, and $^-$ are the operations of intersection, union, and complement which keep their set-theoretic meaning. The unary operation $^\smile$ is the converse operation which tells us the relation holding between B and A from the relation holding between A and B, e.g. $\{\text{NTPP}\}^\smile = \{\text{NTTPI}\}$. The binary composition operation $\circ$ yields the relation that has to hold between A and C when we know the relation holding between A and B as well as between B and C, e.g., $\{\text{TPP}\} \circ \{\text{EC}\} = \{\text{DC}, \text{EC}\}$.

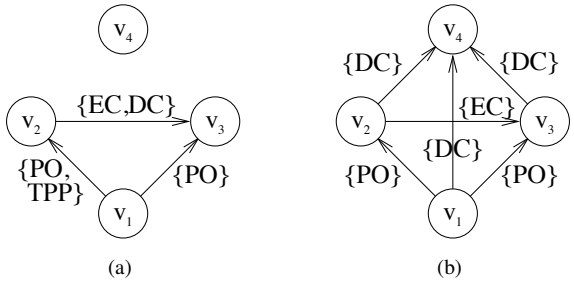


Figure 3. (a) A general RCC-8 constraint network and (b) a consistent scenario of this network.

2.2 Qualitative constraint networks

A spatial arrangement of objects O_i can be described qualitatively based on a qualitative calculus $\mathcal{C}$ by providing a set of relational facts using relations from $\mathcal{R}_\mathcal{C}$, e.g., $O_1\{\text{TPP}\}O_2, O_2\{\text{EC}\}O_3, O_1\{\text{DC}, \text{EC}\}O_3$, etc. The relations can be seen as constraints that restrict which values of $\mathcal{D}_\mathcal{C}$ can be assigned to the objects. We formally define such a qualitative spatial representation as a *qualitative constraint network* (QCN) in which the objects correspond to variables and the spatial relations correspond to constraints.

Definition 1 (Qualitative Constraint Network)**.** *A qualitative constraint network over a qualitative calculus $\mathcal{C}$ is a pair (V, C) where:*

- *$V = \{v_1, v_2, ...v_m\}$ is a set of variables*
- *$C : V^2 \rightarrow \mathcal{R}_\mathcal{C}$ is a function mapping each pair of variables from V to a relation from $\mathcal{R}_\mathcal{C}$ where $C(v_i, v_j) = r \in \mathcal{R}_\mathcal{C}$ means that relation $v_i\ r\ v_j$ has to hold for the values assigned to v_i and v_j*
- *for all $1 \leqslant i, j \leqslant m$, $C(v_i, v_i) = id$ and $C(v_i, v_j) = C(v_j, v_i)^\smile$ holds (id is the identity relation of $\mathcal{C}$).*

In the remainder of this text we will also use the abbreviation C_{ij} instead of $C(v_i, v_j)$. One way to illustrate such a qualitative constraint network is by a directed graph as shown in Figure 3 containing a vertex for every variable v_i and one directed edge for every pair of variables v_i, v_j with $i < j$ which is labeled by the corresponding relation. By convention, edges labeled with the universal relation U are omitted.

Naturally, the scene description provided by a QCN can be *consistent* or not. The QCN, hence, can be seen as a constraint satisfaction problem (CSP) in which the domain is typically infinite (e.g., points in $\mathbb{R}^2$). An assignment of values from $\mathcal{D}_\mathcal{C}$ to the variables v_i is a *solution* if it satisfies all constraints C_{ij}. A QCN N is *consistent* if it has at least one solution. A QCN s is called *atomic* or a *scenario* if any C_{ij} consists of a single base relation. We say that a scenario $s = (V, C')$ *is a scenario of QCN* $N = (V, C)$ if all $C'_{ij} \subseteq C_{ij}$. The scenario in Figure 3(b) is a consistent scenario of the network in Figure 3(a).

We will use a predicate *consistent(N)* to state that QCN N is consistent. We denote the set of all scenarios of N as $\langle\!\langle N \rangle\!\rangle$ and the set of all *consistent* scenarios as $[\![N]\!]$. QCN will refer to a QCN in which all constraints C_{ij} are set to U and, hence, $[\![QCN]\!]$ stands for the set of all consistent scenarios for an implicitly given number of variables.

There exist two main approaches for deciding consistency of QCNs, both based on techniques developed for discrete CSPs. The so-called algebraic closure algorithm enforces a local consistency called path-consistency [17] and runs in $O(n^3)$ time for n variables. If algebraic closure is not sufficient to decide consistency for the relations occurring in the network, a backtracking search is performed

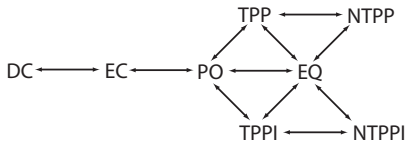


Figure 4. Conceptual neighborhood graph for RCC-8 as given in [11] (assuming that objects may change their size).

[19] that recursively splits the constraints. The resulting procedure has exponential worst-case time complexity.

2.3 Conceptual neighborhood

Our merging approach will be based on a notion of similarity or distance between QCNs. Similarity is related to how the relations of the QCN can change, an aspect which is described by the notion of *conceptual neighborhood* introduced in [9]. Two relations of a spatial calculus are *conceptually neighbored*, if they can be continuously transformed into each other without resulting in a third relation in between. For instance, DC is conceptually neighbored to EC but not to PO as one would have to pass through EC to get to PO. The concrete conceptual neighbor relation depends on the concrete set of continuous transformations we assume [9, 7] which in turn need to be grounded in spatial change over time [10]. For this work, it is sufficient to assume that a suitable conceptual neighborhood relation has been defined which is irreflexive and symmetric. It can be represented by the so-called *conceptual neighborhood graph* $\mathcal{CNG}$ as illustrated in Figure 4 for RCC-8.

2.4 Merging qualitative constraint networks

We just introduced the conceptual neighborhood graph as a way to measure the distance or similarity of the base relations of a calculus. Seeing the conceptual neighborhood graph $\mathcal{CNG}_\mathcal{C}$ of a calculus $\mathcal{C}$ as an undirected weighted graph[4], one can define the distance $d_{B\leftrightarrow B}$ between two base relations $b_i, b_j \in \mathcal{B}_\mathcal{C}$ as the shortest path distance between the corresponding nodes in the graph:

$$d_{B\leftrightarrow B}(b_i, b_j) \quad = \quad \text{shortest path distance between } b_i \text{ and } b_j \text{ in } \mathcal{CNG}_\mathcal{C} \tag{1}$$

The distance function $d_{B\leftrightarrow B}$ forms the basis for formulating distance measures over complete QCNs which in turn has been used to define distance-based operators for merging the information from an input set $\mathcal{N} = \{N_1, ..., N_n\}$ where each knowledge base N_i is a single QCN [7, 24, 5].[5] First, the distance between two atomic qualitative constraint networks $s = (V, C)$ and $s' = (V, C')$ over the same set of m variables and the same calculus is defined using an aggregation operator that determines how the distances between corresponding constraints in s and s' given by $d_{B\leftrightarrow B}$ are combined. Candidates for this aggregation operator, which we will denote as $\circledcirc$, are the sum or the max operator. In all examples later on, we will use sum. The distance itself is defined as:

$$d_{S\leftrightarrow S}(s, s') = \mathop{\circledcirc}_{1\leqslant i<j\leqslant m} d_{B\leftrightarrow B}(C_{ij}, C'_{ij}) \tag{2}$$

[4] We here assume uniform weights but the weights can in principal be used to express a cognitive distance as described in [15].

[5] We here mainly follow our relation-based version [24] of the operators defined by Condotta et al. [5].

The notion behind distance-based merging operators $\Delta(\mathcal{N})$ for QCNs is that the result should be based on the consistent scenarios that are closest to the input networks. The distance between a scenario and a general constraint network and based on that, the distance between a scenario and the set of input networks $\mathcal{N}$ are defined as follows. For determining how close a scenario s is to a constraint network N, all scenarios of N are considered and the distance to the closest one is taken. The resulting distance $d_{S\leftrightarrow N}(s, N)$ is then given by

$$d_{S\leftrightarrow N}(s, N) = \min_{s'\in\langle\!\langle N\rangle\!\rangle} d_{S\leftrightarrow S}(s, s') \tag{3}$$

To measure the distance between a scenario s and the set $\mathcal{N}$ of all input networks N_i, one aggregates over the individual distances $d_{S\leftrightarrow N}(s, N_i)$ using another aggregation operator $\circledast$ (again we will use sum in all examples). The resulting distance is given by:

$$d_{S\leftrightarrow\mathcal{N}}(s, \mathcal{N}) = \mathop{\circledast}_{1\leqslant k\leqslant n} d_{S\leftrightarrow N}(s, N_k) \tag{4}$$

The final merging result $\Delta(\mathcal{N})$ can then be defined as the set of all consistent scenarios that have a minimal distance $d_{S\leftrightarrow\mathcal{N}}$ to $\mathcal{N}$:[6]

$$\Delta(\mathcal{N}) = \{s \in [\![QCN]\!] \mid \forall s' \in [\![QCN]\!] : d_{S\leftrightarrow\mathcal{N}}(s', \mathcal{N}) \geqslant d_{S\leftrightarrow\mathcal{N}}(s, \mathcal{N})\} \tag{5}$$

2.5 Conceptual neighborhood of configurations

In addition to facilitating distance-based merging of QCNs including the resolution of conflicts, the distance measure $d_{S\leftrightarrow S}(s, s')$ between two scenarios can also be used to define a complex conceptual neighborhood graph ($\mathcal{CCNG}$) extending the notion of the conceptual neighborhood graph to more than two objects [20]. Each node in the $\mathcal{CCNG}$ corresponds to a scenario of n objects for a given calculus and the nodes for s and s' are connected in accordance with $d_{S\leftrightarrow S}$, capturing the notion of continuous spatial change on the level of qualitative configurations (see [21] for a more formal discussion).

Figure 6 shows a small section of the complex neighborhood graph of RCC-8 for configurations of three objects, which will play an important role later on. The relations A to B, B to C, and A to C for each node are written below each other. The scenarios marked by the flash symbol are inconsistent, while all other are consistent. The idea that the complex neighborhood graph represents continuous change on a qualitative level and, hence, each continuous spatio-temporal development should correspond to a path through the complex neighborhood graph (consisting of consistent scenarios only) will form the basis for our integration and conflict resolution approach for incomplete and potentially contradicting spatio-temporal information.

3 SPATIO-TEMPORAL MERGING

In the following, we develop an approach to integrate qualitative information from n knowledge bases (KB) into a single consistent spatio-temporal description. Together, the knowledge bases KB_i form the input set $\mathcal{N} = \{KB_1, KB_2, ... KB_n\}$ of our integration problem. We assume that each KB_i consists of a set $\{QCN_1^{[i]}, QCN_2^{[i]}, ... QCN_{m_i}^{[i]}\}$ of QCNs over a given qualitative spatial calculus. Each QCN describes a static snapshot of the configuration of a set of objects represented by its variables. Without loss of generality, we assume that all QCNs are about the same set of objects which can be achieved by simply adding non-occuring objects

[6] An alternative is to define it as a single QCN built from these consistent scenarios. However, this may lead to additional solutions contained in the resulting network (cf. [24]).

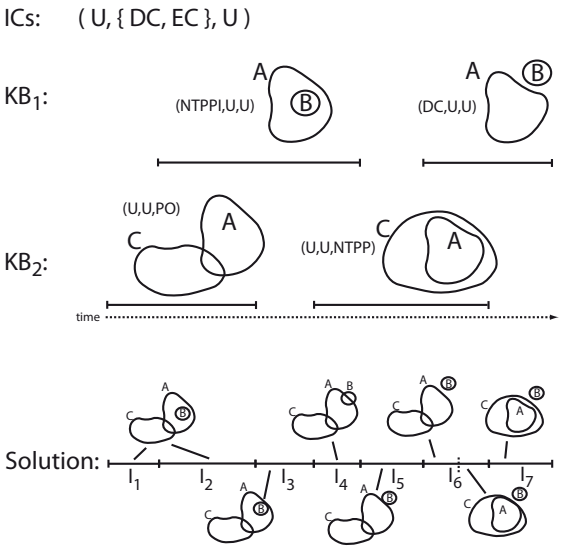


Figure 5. Example of a qualitative spatio-temporal integration problem consisting of an integrity constraint (IC) and two knowledge bases. One optimal solution is shown at the bottom of the figure.

and setting all their relations to the universal relation U. In Figure 5, we show an illustration of such an input set consisting of two KBs KB_1 and KB_2 of which the first one contains information about objects A and B, while the second one contains information about the relation between A and C both using information from the RCC-8 calculus. The actual QCNs over all three objects are abbreviated by triples (r_1, r_2, r_3) where r_1 is the relation between A and B, r_2 is the relation between B and C, and r_3 is the relation between A and C. For instance, the first QCN in KB_1 is the QCN $(\text{NTPPI}, \text{U}, \text{U})$ stating that B is a non-tangential proper part of A, while saying nothing about C. The figure also provides *exemplary* illustrations of the specified configurations.

Each $QCN_j^{[i]}$ is associated with a time interval denoted as $ti(QCN_j^{[i]})$ which states when this particular snapshots is supposed to hold (the interval can also be a single time point). While we allow for gaps between the intervals to represent incomplete information, we assume that for any given time at most one snapshot is given in a single KB. Hence, we demand that the AIA relations between the intervals of consecutive QCNs in a KB are either *meets* or *before*: $ti(QCN_k^{[i]})$ $\{\text{meets}, \text{before}\}$ $ti(QCN_{k+1}^{[i]})$. Furthermore, we assume that we know the AIA base relations holding between each pair of intervals from different KBs (as would be the case if the time intervals are specified quantitatively). For instance, it is known that the relation between the interval of the first QCN in KB_1 and the second QCN in KB_2 is *overlap*: $ti(QCN_1^{[1]})$ $\{\text{overlaps}\}$ $ti(QCN_2^{[2]})$. In principle, the approach described here could also be generalized to the case where these AIA relations are not known precisely (e.g., to allow for representing uncertainty in the temporal information), as long as the overall set of AIA relations would be consistent.

The described input still allows for inconsistencies on different levels when considering the combined information: First of all, a single QCN in a KB might be inconsistent. Second, if there exist QCNs in a KB with meeting intervals, the QCNs might not respect the continuity constraint [11] stating that change is continuous and, hence, only elementary changes with regard to the conceptual neighborhood are possible. Third, if time intervals for QCNs in different KBs intersect, the combined spatial information may not be consistent. In addition, we introduce another source of potential inconsistencies by taking into account that in data integration tasks it is common to assume that the result has to satisfy a set of integrity constraints (IC). We here restrict us to sets of ICs which can be translated into relational constraints between specific pairs of objects and, hence, can be represented by a single QCN R_{IC}. For instance, in the example in Figure 5 we assume that B and C are two regions representing different kinds of land uses and, hence, should not be allowed to overlap. As a consequence, their topological relation has to be EC or DC at every point in time and the resulting QCN R_{IC} is $(\text{U}, \{\text{DC}, \text{EC}\}, \text{U})$. This means that at any given time the actual configuration has to be a scenario of this network.

While the approach developed in this section is able to deal with and resolve all types of inconsistencies just described, the actual example from Figure 5 only contains an inconsistency of the last type, a violation of the integrity constraint. As the intervals for the first QCN of KB_1 and the second QCN of KB_2 intersect, the combined network $(\text{NTTPI}, \text{U}, \text{NTPP})$ generated by taking the intersections of the corresponding constraints does not have a consistent scenario that does not violate the ICs: given that A is a non-tangential proper part of C and B is a non-tangential proper part of A, the relation between B and C can only be NTPP and never EC or DC.

While the merging operators described in Section 2.4 provide a means to resolve such contradictions in static scene descriptions, we cannot simply apply them to every time interval in order to derive a consistent and reasonable merging result of the overall spatio-temporal information. A significant extension is needed to make sure that the result (1) obeys the continuity constraint, meaning it corresponds to a path through the $\mathcal{CCNG}$, and (2) the result minimizes the overall deviation from the given input information.

Point (1) means that a solution S should be a sequence $\langle QCN_1^{[S]}, QCN_2^{[S]}, ...QCN_l^{[S]} \rangle$ of QCNs with associated time intervals $ti(QCN_i^{[S]})$ satisfying the following criteria:

- each $QCN_i^{[S]}$ is a consistent scenario compliant with R_{IC}
- $ti(QCN_1^{[S]})$ starts together with the first interval of $ti(QCN_j^{[i]})$
- $ti(QCN_l^{[S]})$ ends together with the last interval of $ti(QCN_j^{[i]})$
- $ti(QCN_i^{[S]})$ $\{\text{meets}\}$ $ti(QCN_{i+1}^{[S]})$ for all i
- $adjacent(QCN_i^{[S]}, QCN_{i+1}^{[S]})$ where $adjacent$ is the adjacency relation of scenarios in the $\mathcal{CCNG}$ of the underlying calculus.

At the bottom of Figure 5, we see one such possible solution. It corresponds to the path in the $\mathcal{CCNG}$ from Figure 6 given by the arrow labeled with (II), showing that it indeed represents a contiguous sequence of changing configurations.

We now proceed by formalizing the notion of a solution that minimizes the overall deviation from the input information. First, we observe that the set of intervals $ti(QCN_j^{[i]})$ from $\mathcal{N}$ induces a partition of the considered part of the time line into intervals I_i so that for each interval the same set of snapshots from $\mathcal{N}$ holds over the complete interval (in the example from Figure 5 we get seven intervals I_1 to I_7). The boundaries of these intervals correspond to changes in the input information.

We now restrict us to potential solutions S' for which the sequence $\langle ti(QCN_1^{[S']}), ..., ti(QCN_l^{[S']}) \rangle$ is a subdivision of this partition, meaning that the intervals $ti(QCN_j^{[S']})$ do not cross the boundaries of the intervals I_i. This can be achieved by splitting intervals in the original solution S. We introduce a function $qcn(...)$ that yields the subset of all QCNs $QCN_j^{[i]}$ from $\mathcal{N}$

that hold during interval $ti(QCN_j^{[S']})$ of S'. If for a KB KB_i no QCN holds in $ti(QCN_j^{[S']})$, both the previous and the following QCN are added to the resulting set in order to not completely ignore this KB for this interval but rather use a qualitative interpolation when assessing the deviation. For instance, $qcn(I_3)$ for I_3 in Figure 5 would yield the following set of three QCNs: $\{(\text{NTPPI}, \text{U}, \text{U}), (\text{U}, \text{U}, \text{PO}), (\text{U}, \text{U}, \text{NTPP})\}$ where the first QCN stems from KB_1, while the other two are the previous and following one from KB_2.

The deviation of a scenario $QCN_j^{[S']}$ in S' from the input set $\mathcal{N}$ is then given by the following distance function $d_{Q\leftrightarrow\mathcal{N}}$:

$$d_{Q\leftrightarrow\mathcal{N}}(QCN_j^{[S']}, \mathcal{N}) = d_{S\leftrightarrow\mathcal{N}}(QCN_j^{[S']}, qcn(ti(QCN_j^{[S']}))) \tag{6}$$

Essentially, what we do here is applying the distance measure defined in Eq. 4 to the set of QCNs holding during the given interval. In a final step, we sum up over all intervals of S' to derive its overall deviation $d(S', \mathcal{N})$:

$$d(S', \mathcal{N}) = \sum_{j=1}^{l} d_{Q\leftrightarrow\mathcal{N}}(QCN_j^{[S']}, \mathcal{N}) \tag{7}$$

Given this formalization of the deviation of what is essentially a temporally delimited path through the $\mathcal{CCNG}$ and assuming that $\mathcal{S}$ stands for the set of all potential solutions S', we define our qualitative spatio-temporal merging operator $\Gamma(\mathcal{N})$ to yield all elements of $\mathcal{S}$ that have a minimum deviation:

$$\Gamma(\mathcal{N}) = \{S' \in \mathcal{S} \mid \forall S'' \in \mathcal{S} : d(S'', \mathcal{N}) \geqslant d(S', \mathcal{N})\} \tag{8}$$

The solution at the bottom of Figure 5 is also an minimal one. Its intervals correspond to the intervals I_1 to I_7 with the exception that I_6 is refined into two subintervals. Overall, Γ in this example yields four minimal solutions. The corresponding paths through the neighborhood graph are shown by the arrows in Figure 6.

4 IMPLEMENTING THE OPERATOR

The qualitative spatio-temporal merging operator Γ captures what one would consider an optimal merging approach for the considered kind of integration problem. However, given the high-level of complexity involved (consistency checking for RCC-8 is already NP-hard and merging adds another dimension of complexity), implementations of the general operator described here will most likely remain restricted to small problem instances. The following considerations give some idea about the size of the problem: If b is the number of base relations and f the average branching factor in the conceptual neighborhood graph (e.g., 8 and 2.75, respectively, for RCC-8), the number of nodes in the complex conceptual neighborhood graph for v variables is of order $O(v^{2\times b})$ and the average branching factor is $v^2 \times f$. Under these circumstances, the best one can hope for is that subclasses of problems for specific assumptions exist which will lead to more efficient implementations. We therefore see the described approach more as a theoretical framework to compare concrete merging algorithms against, which for instance might only approximate the results of the theoretical operator.

One first step in this direction is the implementation that we will briefly describe in the following. It uses consistency checking and the computation of $d_{S\leftrightarrow\mathcal{N}}(s, \mathcal{N})$ and $\Delta(\mathcal{N})$ already provided by our own QSR toolbox SparQ [25]. The search space (including the $\mathcal{CCNG}$) is expanded during the search. In addition, the algorithm is based on two main underlying restrictions:

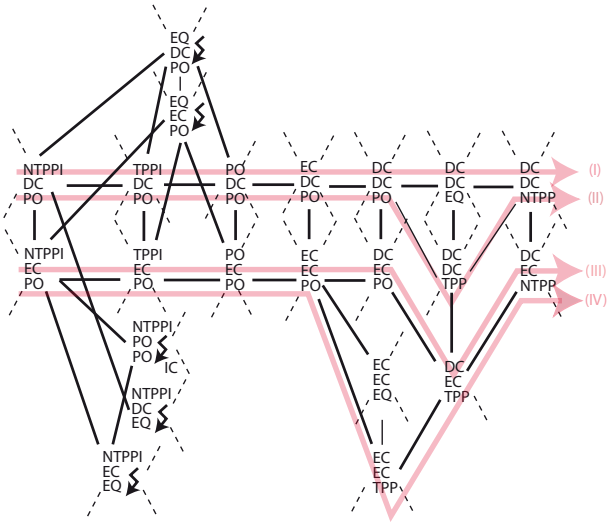


Figure 6. Section of the complex neighborhood graph for RCC-8 for three variables including several inconsistent scenarios (the scenario with the subscript 'IC' is consistent but violates the integrity constraints). The four arrows depict the sequences of scenarios corresponding to the four optimal solutions of the example from Figure 5.

- It only considers solutions S' in which $QCN_1^{[S']}$ is a scenario from $\Delta(qcn(I_1))$ and $QCN_l^{[S']}$ is a scenario from $\Delta(qcn(I_m))$ where I_1 and I_m are the first and last of the intervals induced by $\mathcal{N}$ (cmp. Eq. 4).
- It only computes *one* optimal solution not *all* as in the definition of Γ.

The first restriction means that we have a fixed (and typically small) set of start and end scenarios. Applying this approach to the example from Figure 5 results in exactly the two leftmost scenarios in Figure 6 as start scenarios and the two rightmost ones as end scenarios. Overall, the restrictions allow us to simultaneously expand the search space from the start and end scenarios and to make efficient use of branch and bound techniques to prune large parts of the search space, once a first good solution has been found.

An expansion step during the search in general looks as follows (in the case of forward search): The current state is essentially described by the scenario s we are currently in, the index i of the induced interval I_i that we are currently considering, and the deviation d accumulated so far. There now exist two options: (1) staying in s and moving i forward to the next interval I_i or (2) moving to a neighbored scenario in the $\mathcal{CCNG}$ that is consistent and satisfies the ICs. The second option can be performed with or without moving i forward. New search states are generated for each of the described possibilities which involves the computation of a new $d_{Q\leftrightarrow\mathcal{N}}(QCN_i^{[S']}, \mathcal{N})$ value that is then used to update the deviation d. When d becomes larger than the deviation of the best solution known so far, expansion can be stopped and the corresponding branch of the search space is discarded. A new solution is found, when an end scenario is reached and the index i is that of the last induced interval I_m.

Figure 7 shows the sequence of expansion steps that lead to the optimal solution from Figure 5. The start scenario $(\text{NTPPI}, \text{DC}, \text{PO})$ has zero deviation for interval I_1. Then the decision is made to stay at this scenario in the complex neighborhood graph but increase i so that we are now considering interval I_2. $d_{Q\leftrightarrow\mathcal{N}}$ again results in zero

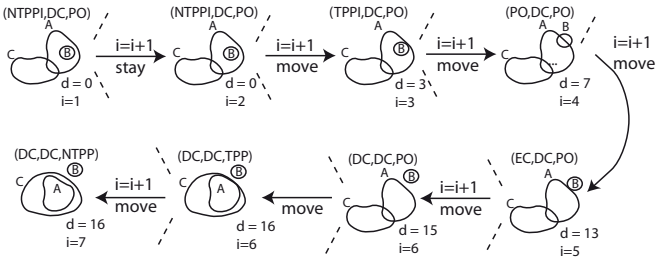


Figure 7. Expansion steps that leads to the optimal solution from Figure 5.

costs and, hence, the total deviation d stays at zero. In the next step, we move to the neighbored consistent scenario $(\text{NTPPI}, \text{DC}, \text{PO})$, while simultaneously increasing i. $d_{Q \leftrightarrow \mathcal{N}}$ for this new scenario and I_3 yields a deviation of 3 which is added to d. Later on, we are in scenario $(\text{DC}, \text{DC}, \text{PO})$ and interval I_6. Here the decision leading to the best solution is to move to the neighbored scenario $(\text{DC}, \text{DC}, \text{TPP})$ without increasing i resulting in the mentioned subdivision of I_6.

For the example from Figure 5, we already know that all optimal solutions have one of these as start and end scenarios and, hence, the algorithm will find an optimal solution. However, a theoretical investigation under which assumptions this is the case as well as an empirical evaluation of the general performance of the algorithm sketched here is still outstanding. In addition, more sophisticated search approaches and restrictions still need to be investigated. In particular, the consequences of introducing intermediate goals which could increase the efficiency significantly but, on the other hand, might lead to suboptimal solutions need to be examined.

5 CONCLUSIONS

We described a theoretical model for integrating potentially conflicting qualitative spatio-temporal information stemming from different sources into a single consistent description. The input information consists of spatial snapshots delimited in time and the result is optimal in the sense that it constitutes the smallest deviation from the input information as measured by distance functions derived from the notion of conceptual neighborhood. While being hampered by a high complexity, the described model should be a valuable means for the development of more efficient approaches, either based on approximations or for subclasses of problems of the general problem described here. A first approach in this direction has already been described and now needs to be further investigated as part of future research. In addition, the described framework can be generalized further by allowing for underspecificity in the temporal information.

ACKNOWLEDGEMENTS

The authors would like to thank Diedrich Wolter and the anonymous reviewers for helpful comments and suggestions. Financial support by the Deutsche Forschungsgemeinschaft under grants IRTG GRK 1498 Semantic Integration of Geospatial Information and SFB/TR 8 Spatial Cognition is gratefully acknowledged.

REFERENCES

[1] James F. Allen, 'Maintaining knowledge about temporal intervals', *Communications of the ACM*, **26**(11), 832–843, (1983).

[2] H. Tom Bruns and Max J. Egenhofer, 'Similarity of spatial scenes', in *Seventh International Symposium on Spatial Data Handling*, pp. 173–184. Taylor & Francis, (1996).

[3] C. Claramunt and M. Thériault, 'Managing time in GIS: An event-oriented approach', in *Recent Advances on Temporal Databases*, eds., J. Clifford and A. Tuzhilin, pp. 23–42. Springer, (1995).

[4] A. G. Cohn and S. M. Hazarika, 'Qualitative spatial representation and reasoning: An overview', *Fundamenta Informaticae*, **46**(1-2), 1–29, (2001).

[5] Jean-François Condotta, Souhila Kaci, and Nicolas Schwind, 'A framework for merging qualitative constraints networks', in *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference*, eds., David Wilson and H. Chad Lane, pp. 586–591. AAAI Press, (2008).

[6] Jean-François Condotta, Souhila Kaci, Pierre Marquis, and Nicolas Schwind, 'Merging qualitative constraints networks using propositional logic', in *10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty(ECSQARU'09)*, (2009).

[7] Frank Dylla and Jan Oliver Wallgrün, 'Qualitative spatial reasoning with conceptual neighborhoods for agent control', *Journal of Intelligent and Robotic Systems*, **48**(1), 55–78, (2007).

[8] Max J. Egenhofer and Khaled K. Al-Taha, 'Reasoning about gradual changes of topological relationships', in *Proceedings of the International Conference GIS - From Space to Territory*, pp. 196–219, London, UK, (1992). Springer-Verlag.

[9] Christian Freksa, 'Conceptual neighborhood and its role in temporal and spatial reasoning', in *Decision Support Systems and Qualitative Reasoning*, eds., M. Singh and L. Travé-Massuyès, 181 – 187, (1991).

[10] Antony Galton, *Qualitative Spatial Change*, Oxford University Press, 2000.

[11] Alfonso Gerevini and Bernhard Nebel, 'Qualitative spatio-temporal reasoning with RCC-8 and Allen's Interval Calculus: Computational complexity', in *ECAI'02*, ed., Frank van Harmelen, pp. 312–316. IOS Press, (2002).

[12] Éric Grégoire and Sébastien Konieczny, 'Logic-based approaches to information fusion', *Information Fusion*, **7**(1), 4–18, (2006).

[13] P. Grenon and B. Smith, 'Snap and span: Towards dynamic spatial ontology', *Spatial Cognition and Computation*, **4**(1), 69–104, (2004).

[14] Kathleen Hornsby and Max J. Egenhofer, 'Qualitative representation of change', in *COSIT'97*, eds., Stephen C. Hirtle and Andrew U. Frank, pp. 15–33. Springer, (1997).

[15] Alexander Klippel and Rui Li, 'The endpoint hypothesis: A topological-cognitive assessment of geographic scale movement patterns', in *Spatial Information Theory*, pp. 177–194. Springer, (2009).

[16] Jinxin Lin and Alberto O. Mendelzon, 'Merging databases under constraints', *International Journal of Cooperative Information Systems*, **7**, 55–76, (1996).

[17] A. K Mackworth, 'Consistency in networks of relations', *Artificial Intelligence*, **8**, 99–118, (1977).

[18] R. Moratz and C. Freksa, 'Spatial reasoning with uncertain data using stochastic relaxation', in *Fuzzy-Neuro Systems 98*, ed., W. Brauer, 106–112, Infix; Sankt Augustin, (1998).

[19] Bernhard Nebel, 'Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ORD-Horn class', *CONSTRAINTS*, **3**(1), 175–190, (1997).

[20] Marco Ragni and Stefan Wölfl, 'Temporalizing spatial calculi: On generalized neighborhood graphs', in *KI'05*, ed., Ulrich Furbach, pp. 64–78. Springer, (2005).

[21] Marco Ragni and Stefan Wölfl, 'Temporalizing cardinal directions: From constraint satisfaction to planning', in *KR*, eds., Patrick Doherty, John Mylopoulos, and Christopher A. Welty, pp. 472–480. AAAI Press, (2006).

[22] David A. Randell, Zhan Cui, and Anthony Cohn, 'A spatial logic based on regions and connection', in *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pp. 165–176. Morgan Kaufmann, (1992).

[23] J. Renz and B. Nebel, 'Qualitative spatial reasoning using constraint calculi', in *Handbook of Spatial Logics*, eds., M. Aiello, I. E. Pratt-Hartmann, and J. F.A.K. van Benthem, 161–215, Springer, (2007).

[24] Jan Oliver Wallgrün and Frank Dylla, 'A relation-based merging operator for qualitative spatial data integration and conflict resolution', Technical Report 022-06/2010, SFB/TR 8 Spatial Cognition, (2010).

[25] Jan Oliver Wallgrün, Lutz Frommberger, Diedrich Wolter, Frank Dylla, and Christian Freksa, 'Qualitative spatial representation and reasoning in the SparQ-toolbox', in *Spatial Cognition V*, eds., T. Barkowsky, M. Knauff, G. Ligozat, and D. Montello, 39–58, Springer, (2007).

[26] Michael F. Worboys and Kathleen Hornsby, 'From objects to events: Gem, the geospatial event model', in *GIScience'04*, eds., M. J. Egenhofer, C. Freksa, and H. J. Miller, pp. 327–344. Springer, (2004).

Knowledge-based adaptive thresholding from qualitative robot localisation using cast shadows

Paulo E. Santos[1] and **Valquiria Fenelon**[2] and **Hannah M. Dee**[3]

Abstract. This paper presents results of a mobile robot qualitative self-localisation experiment using information from cast shadows. Shadow detection was accomplished by mapping the images from the robot's monocular colour camera into a HSV colour space and then thresholding on V. We present results of self-localisation using two methods for obtaining the threshold automatically: in one method the images are segmented according to their grey-scale histograms, in the other the threshold is set according to a prediction about the robot's location, given a shadow-based map defined upon a qualitative spatial reasoning theory. This map-related threshold search is the main contribution of the present work, and to the best of our knowledge this is the first work that uses qualitative spatial representations both to perform egolocation and to calibrate a robot's interpretation of its perceptual input.

1 Introduction

Cast shadows as cues for depth perception have been used to enhance depictions of natural scenes since the Renaissance [11]. Recent research within psychology suggests that the human perceptual system gives preferential treatment to information from shadows when inferring motion in depth and perceiving 3D scene layout. These studies suggest that information coming from shadows can override such basic notions as conservation of object size, rather than discard or distrust shadow information [18, 6, 20]. Casati in [3] points out that cast shadows also contain information that are not used during passive perception, for instance, information about the presence and location of the light source and the caster; the intensity of the source; the caster's shape; the screen texture; and the distance between the caster and the screen.

Whilst psychologists have demonstrated the centrality of shadows to our own perception of depth, size and motion, much work in computer vision and robotics starts from the premise that shadows are sources of noise rather than information. The present work falls within the small but growing area of research which aims to use shadows not as sources of noise, but as sources of information. This requires not only a model of the kinds of information that shadows can purvey, but also a robust and accurate shadow detection system. Researchers within both computer vision and robotics have been working in this area – many engaged in shadow suppression in videos from fixed cameras, but some engaged in the more challenging task of shadow identification, localisation and use.

The contribution of this paper is the investigation of a qualitative self-localisation method using information from cast shadows. We discuss the experimental evaluation of this method using two techniques for obtaining the threshold automatically for segmenting each image picked out by a robot's camera: in one method the images are segmented according to its grey-scale histogram, in the other method the threshold is searched according to a prediction about the robot's location, given a shadow-based qualitative map.

This paper is organised as follows. Section 2 outlines related research from within both computer vision and robotics. Section 3 describes the theory upon which the work is based - the Perceptual Qualitative Relations about Shadows (PQRS), which formalises the problem of shadow reasoning and egolocation within a qualitative spatial reasoning context. The adaptive thresholding methods considered in this work are presented in Section 4, and the experiments are described in Section 5. Discussions are drawn on Section 6 and Section 7 concludes this paper.

Throughout this paper, constants are written in upper-case letters and variables in lower case, unless explicitly stated otherwise.

2 Related research

When considering the task of segmentation of moving objects from a static background, shadows are a frequent source of false positives [10, 21] and therefore shadow *suppression* is a major research area. In this context, shadow detection in computer vision almost always involves some model of the colour of the *screen* or in computer vision terminology *background*, and detection is performed using a model of shadows characterising them as *roughly the same colour as the background, but darker'*. Perhaps the simplest shadow detection method proposed is that of [34], in which a grey-scale image is simply thresholded and the darker pixels are labelled *shadow*; however this approach fails on complex images and in situations where lighting changes due to either environmental effects or egomotion. Prati in [27] provides an overview and a taxonomy of early shadow-detection techniques, dividing them into *model-based* and *non-model-based*, however this categorisation does not apply well to more recent works, many of which can be thought of as ensemble methods [21, 26].

Cucchiara et al. in [10] take as their starting point detected moving objects (and a background model). The pixel values of moving objects are converted to the HSV (Hue, Saturation and Value) colour space, and then these are investigated to determine whether they are real moving objects or merely shadow pixels. This is accomplished by considering observed and background values of all three HSV components, considering the difference between foreground and background values for H and S, and the ratio of the two V values. This captures the intuitive observations that shadows are about the same hue as the same part of the scene unshadowed, slightly more saturated, and darker. A similar approach based upon the observation of colour changes in cast shadows is presented in

[1] Paulo Santos is with Electrical Engineering Dep., FEI, S. Paulo, Brazil, email: psantos@fei.edu.br
[2] Valquiria Fenelon is with Escola Politécnica, USP, S. Paulo, Brazil
[3] Hannah Dee is with School of Computing, University of Leeds, UK

[30]. Stauder et al. in [32] use assumptions about the background (*it will dominate the scene*), the nature of shadows and luminance (*shadows are darker and tend to have uniform shading*) and the presence of moving and static edges. Other methods for shadow filtering are described in [24, 35, 14], an overview of such approaches is left for future work.

Within the last two or three years, models inspired by the physics of light have become more prominent. These systems, rather than simply observe that "shadows are a bit darker", consider the nature of reflectance and the effect of lighting changes on perceived colour. Martel-Brisson and Zaccarin [22] take a simplified reflectance model and use it to learn the way in which colours change when shaded, and Huang and Chen [15] have also incorporated a richer, physics-based colour model for shadow detection based upon the work of Maxwell et al. [23]. Maxwell presents a bi-illuminant dichromatic reflection model, which enables the separation of the effects of lighting (direct and ambient) from the effects of surface reflectance. Huang and Chen [15] simplify this model in several ways, such as assuming that the ambient illumination is constant, which enables them to implement shadow detection based upon the simplified model in video analysis.

There are a few systems within computer vision that use cast shadows as sources of information rather than noise. [2] use known 3D locations and their cast shadows to perform camera calibration and light location (using known casters and screen to tell about the light source); [4] uses the moving shadows cast by known vertical objects (flagpoles, the side of buildings) to determine the 3D shape of objects on the ground (using the shadow to tell about the shape of the screen). Balan et al. [1] use shadows as a source of information for detailed human pose recognition: they show that using a single shadow from a fixed light source can provide a similar disambiguation effect as using additional cameras.

In robotics, the story is similar. Fitzpatrick and Torres-Jara in [13], inspired by the research reported in [5], track the position of a robotic arm and its shadow cast on a table to derive an estimate of the time of contact between the arm and the table. Shadows are detected in this work using a combination of two methods: in the first, a background model of the workspace is built without the arm and then used to determine light changes when the arm is within the camera view. The second method compares subsequent frames in order to detect moving regions of light change. The authors motivate their work pointing out that depth from shadows and stereopsis may work as complementary cues for robot perception, while the latter is limited to surfaces rich in textures, the former works well in smooth (or even reflective) surfaces. Cheah et al. [7] present a novel controller for a robot manipulator, providing a solution to the problem of trajectory control in the presence of kinematic and dynamic uncertainty. In order to evaluate their results, an industrial robot arm was controlled using the visual observation of the trajectory of its own shadow. Lee and colleagues [17] use cast shadows inside pipes to detect landmarks: by fitting bright lights to the front of their pipe inspection robot, they can determine when a pipe bends by detecting cast shadows.

Information from shadows are also considered in unmanned autonomous planetary exploration. Tompkins et al. [33] describe an autonomous path planning system that takes into account various conditions of the robot's state, including particularities of the terrain and lighting. In this context, the information about shadows cast by terrain irregularities allows the rover to plan a trajectory that maximises the trade-off between the exposure of the solar cells to sun light and the limited resources in planetary missions. Kunii and Gotoh [16] propose a *Shadow Range Finder* system that uses the shadow cast by a robot arm on the surface of a terrain in order to obtain depth information around target objects. In planetary explorations this type of system may provide low-cost, energy-saving, sensors for the analysis of the terrain surrounding rock samples of interest.

More recently, Santos et al. [31] describe an initial representation of cast shadows in terms of a spatial logic formalising occlusion relations. This representation, called Perceptual Qualitative relations about Shadows (PQRS), is used in a mobile robot self-localisation procedure in office-like environments. The present paper builds upon this representation and, therefore, the next section describes it in more detail.

3 Perceptual qualitative relations about shadows (PQRS)

Perceptual Qualitative Relations about Shadows (PQRS) [31] is a theory inspired by the idea that shadows provide the observer with the viewpoint of the light source, as they are a projection of the caster from it. Equivalently, we can say that every point in the shadow region is totally occluded by the caster from the viewpoint of the light source. This idea is developed by representing relations of occlusion and shadows within the scope of Qualitative Spatial Reasoning (QSR) field of research, which is part of the artificial intelligence sub-area known as Knowledge Representation and Reasoning [12].

The goal of QSR is to provide appropriate formalisms for representing and reasoning about spatial entities, such as part-whole relations, connectivity, orientation, line segments, size and distance, amongst others [9, 8].

PQRS assumes a static light source, denoted by L, situated above the observer (in agreement to recent research on the psychophysics of perception [19]). It is also assumed that the scenes are observed from an egocentric point of view (v), and that shadows are cast on a single screen Scr which does not need to be flat.

The basic part of PQRS is based on one particular QSR theory: the Region Occlusion Calculus (ROC) [29], which is itself built upon one of the best known QSR approaches: the Region Connection Calculus (RCC) [28]. RCC is a first-order axiomatisation of spatial relations based on a reflexive, symmetric and non-transitive 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 at least one point in common.

Assuming the $C/2$ relation, some mereotopological relations between two spatial regions can be defined, such as *disconnected from* (DC), *equal to* (EQ), *overlaps* (O); *part of* (P) ; *partially overlaps* (PO); *proper part of* (PP); *externally connected* (EC) and *tangential* or *non-tangential proper part* (resp. (TTP) and ($NTTP$)).

Using RCC relations, along with the primitive relation $TotallyOccludes(x, y, v)$ (which stands for "x totally occludes y with respect to the viewpoint v"), the Region Occlusion Calculus (ROC) represents the various possibilities of interposition relations between two arbitrary-shaped objects. In particular, with RCC and the primitive $TotallyOccludes/3$, it is possible to define occlusion relations for *non occlusion* ($NonOccludes/3$), *partial occlusion* ($PartiallyOccludes/3$) and *mutual occlusion* ($MutuallyOccludes/3$). In fact, [29] defines 20 such relations. However, considering the ROC relations between a caster o and its shadow s, from a viewpoint v, only the following relations have models in PQRS: $\{NonOccludesDC(o, s, v)$, $NonOccludesEC(o, s, v)$, $PartiallyOccludesPO(o, s, v)$, $PartiallyOccludesTPP(o, s, v)$, $TotallyOccludesTPPI(o, s, v)$, $TotallyOccludesEQ(o, s, v)$ and $Totally$-

$OccludesNTPPI(o, s, v)\}$. Figure 1 represents these relations, where the dashed object is the caster and the blank is its shadow.

The Region Occlusion Calculus makes a distinction between the occupancy regions of bodies and their images (or projections) from the viewpoint of an observer by assuming the function $region(x)$, which maps a body x to its occupancy region, and the function $image(x, \nu)$ that maps a body x (and the viewpoint ν) to the body's image. Therefore, given two bodies X and Y and a viewpoint ν, the statement $PartiallyOccludesTPP(X, Y, v)$ is defined as $PartiallyOccludes(X, Y)$ *and* $TPP(image(X), image(Y))$.

It is worth pointing out also that the "I" in the relations $TotallyOccludesTPPI(o, s, v)$ and $TotallyOccludesNTPPI(o, s, v)$ represents the inverse of TPP and PP, resp.; so, for instance, $TotallyOccludesTPPI(o, s, v)$, means that the caster o totally occludes its shadow s, but s is the tangential proper part of o.

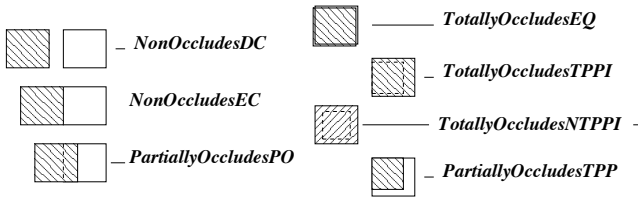


Figure 1. The ROC relations that are part of PQRS.

Apart from the ROC relations inherited in PQRS, it assumes the primitive $Shadow(s, o, Scr, L)$ that represents that a shadow s is cast by a caster o, from the light source L. The axiom constraining the $Shadow/4$ relation is represented by Formula 1 below.

$$\begin{aligned} Shadow(s, o, Scr, L) \leftrightarrow\ & PO(r(s), r(Scr)) \wedge \\ & TotallyOccludes(o, s, L) \wedge \\ & \neg\exists o' TotallyOccludes(o', o, L). \end{aligned} \quad (1)$$

The axiom represented in Formula 1 states that the shadow of a caster o is the region in a screen Scr that is totally occluded by o from the light source viewpoint L.

3.1 Relative location

The formalism summarised above can be used to reason about shadows from arbitrary viewpoints: relating shadows with occlusion suggests the distinction of five regions defined from the lines of sight between the light source, the caster and its shadow (or the top-half part of the latter if it is cast on the floor), as represented in Figure 2. Therefore, any viewpoint v located on Region 1 will observe the shadow s and the object o as $NonOccludesDC(o, s, v)$; similarly, if v observes o and s from Region 3 it should see that $PartiallyOccludesPO(o, s, v)$ and from Region 5 that $TotallyOccludesNTPPI(o, s, v)$. Region 4 is the surface defined by the lines of sight from l tangential to o and s, from where v would observe $TotallyOccludesTPPI(o, s, v)$. In Region 2, v perceives object and shadow as $NonOccludesEC(o, s, v)$. Regions 2 and 4 are in fact *boundaries* separating regions 1 and 3, and between 3 and 5 respectively. Therefore, it is virtually impossible for a robot to locate itself on them. In the real robot environment, however, regions 2 and 4 are extended assuming an interval of uncertainty around these boundaries. Figure 3 represents the regions used in the experiments of this paper, where L is the light source, O is the object (caster) and S is its shadow.

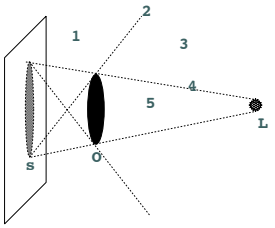


Figure 2. Distinct regions implied by the observation of a shadow and its caster. It is worth noting that, in this figure, regions 2 and 4 are zero-width boundaries.

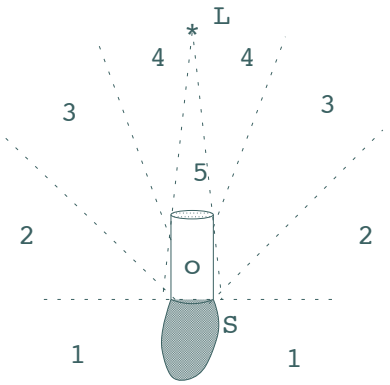
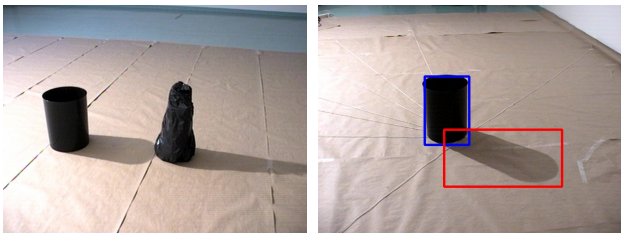


Figure 3. Regions implied by the observation of a shadow and its caster.

This idea for qualitative robot self-localisation using cast shadows was implemented on our Pioneer PeopleBot mobile robot using its monocular colour camera to obtain snapshots of objects and their shadows in an office-like environment (following the guidelines presented in [31]). Shadow detection was accomplished by first mapping the images captured by the camera into a HSV colour space. These images were then segmented by thresholding on V, whereby high values (light objects) are filtered out and low values (dark objects) are casters. Shadows are located within a value range in between light and dark objects. Morphological operators and the saturation value were used to filter noise (such as reflections of the light source on the object or background shadows). The robot was set to navigate through the room, stopping after a certain time interval to analyse its position with respect to the object-shadow locations according to the diagram shown in Figure 3. One example of the snapshots used in this work is shown in Figure 4(b). Shadow correspondence, which is the problem of matching each shadow to its caster [18, 20], is solved in this work by assuming a simple heuristic: the shadow that is connected to an object's base is the shadow of this object. When there are various shadows connected to the object's base, the caster is associated with the shadow that is further away from the light source (Fig. 4(a) shows an example of such situation).

Given a threshold Th, a $Scene$ and a viewpoint ν, Algorithm 1 summarises the method for self-localisation described in this section.

In Algorithm 1 the ROC relations between a caster O and its shadow S are evaluated according to a threshold on the distance between the (top part of) the shadow when *Non Occlusion* holds. If the shadow is in some degree occluded by its caster, from the observer's viewpoint, the ROC relation is evaluated according to a percentage of the shadow that can be observed from behind the caster:



(a) Two shadows (b) Example segmented image

Figure 4. (a) two shadows in one object's base and (b) example of a segmented image

Algorithm 1 PERCEPTION_ACTION(Th, Scene, ν)

1: segment $Scene$ using the threshold Th to obtain a caster O and its shadow S
2: **if** $NonOccludesDC(O, S, \nu)$ **then**
3: robot is on region 1
4: **else if** $NonOccludesEC(O, S, \nu)$ **then**
5: robot is on region 2
6: **else if** $PartiallyOccludesPO(O, S, \nu)$ **then**
7: robot is on region 3
8: **else if** $TotallyOccludesTPPI(O, S, \nu)$ **then**
9: robot is on region 4
10: **else if** $TotallyOccludesNTPPI(O, S, \nu)$ **then**
11: robot is on region 5
12: **else**
13: FAIL
14: **end if**

$PartiallyOccludesPO(O, S, \nu)$ is interpreted when more than 10% of the shadow is observed; $TotallyOccludesTPPI(O, S, \nu)$ is assumed when less than (or equal to) 10% is still observed; and, $TotallyOccludesNTPPI(O, S, \nu)$ is concluded when no part of the shadow is seen from behind the caster.

The material presented up to this point is discussed in greater details in [31]. The remainder of this paper is completely original.

4 Adaptive thresholds for foreground/background segmentation

In this work we investigate the use of two distinct methods for automatically finding the best threshold for each given image: the traditional Otsu's method [25] and a threshold search related to the robot's prediction. The latter is the main contribution of the present paper.

Otsu's method [25] works by finding the threshold (t) that maximises the inter-class variance σ between two groups of pixels. Formula 2 expresses σ in terms of the threshold-dependent class probabilities ($\omega_1(t)$ and $\omega_2(t)$) and class means ($\mu_1(t)$ and $\mu_2(t)$) of groups 1 and 2.

$$\sigma^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (2)$$

The second method for finding the best threshold uses the knowledge about the robot's previous location in order to make a prediction about it's current location. This procedure works as follows. The robot has to start in a known region. From this position the robot moves to another region (according to the diagram in Figure 3) in a moving action that is currently preprogrammed, but that still suffers from actuator noise. In this new position the robot captures a snapshot of the target object and uses it to decide on its location. If the location interpreted matches the prediction of its current position, then the robot moves on. If not, the robot varies the threshold until it finds a match between its predicted and interpreted positions, or fails otherwise. This method is summarised in Algorithm 2 below.

In the pseudocode THRESHOLD AND POSITION (Algorithm 2), the function MOVING_ACTION(s_0, v, dir, I) gives the prediction of the robot's position s_i after its motion from the position s_0, with speed v, direction dir and for a time interval I; the function PERCEPTION_ACTION($th, Scene, \nu$) outputs the perceived robot's position according to the observed PQRS relation, for a threshold th, a $Scene$ and a viewpoint ν (as discussed in Section 3.1), th_{aux} is an auxiliary variable for threshold and s_0, s_i and sj are variables for the robot's position. The constant $Step$ is used to update the threshold from its minimum (Th_{min}) to its maximum (Th_{max}) values (in this work these constants were set at $Step = 5, Th_{min} = 40$ and $Th_{max} = 230$).

Algorithm 2 THRESHOLD AND POSITION(th_0, t_0, $scene$)

1: s_i = MOVING_ACTION(s_0, v, dir, I)
2: s_j = PERCEPTION_ACTION($th_0, Scene, \nu$)
3: **if** $(s_i == s_j)$ **then**
4: return (th_0, s_i)
5: **else**
6: $th_{aux} = Th_{min}$
7: **while** $((si \neq sj)$ and $(th_{aux} < Th_{max}))$ **do**
8: $th_{aux} = th_{aux} + Step$
9: s_j =PERCEPTION_ACTION($th_{aux}, Scene, \nu$)
10: **end while**
11: **if** $(th_{aux} > Th_{max})$ **then**
12: return FAIL
13: **else**
14: return (th_{aux}, s_j)
15: **end if**
16: **end if**

To the best of our knowledge this is the first paper where the segmentation threshold is obtained as a result of the robot's prediction of its location according to a qualitative map. In this way, we use the PQRS theory not only for robot self-localisation based upon shadow perception, but also for the refinement of the shadow perception itself. The next section presents an empirical evaluation of this technique.

5 Experiments

This section describes the results of the experiments on robot localisation with respect to the map in Figure 3. In these experiments the robot collected 1361 snapshots around the target object, which provides the frame of reference (e.g. the black bucket in Figure 4(b)). This target was always within camera view, but not necessarily at its centre. We allowed up to three objects within the robot's field of view. Due to the narrow view of the robot, and the use of a single dominant light source, localisation estimates with respect to each object do not contradict one another.

The baseline experiment uses fixed thresholds for image analysis chosen by experimentation within one of the camera views. These results are represented in Table 1, which shows a poor global performance of the system (47%) on localising the robot in every region. A high accuracy was obtained in the specific region used to calibrate the threshold (above 70% with respect to region 1), but within

other regions the results where lower or equal to 50%. The poor performance outside of region 1 is because the foreground/background segmentation is not optimal for images obtained under other light conditions (i.e., the distinct position configurations between robot, caster and light produced by the agent's motion). In fact, by tweaking the thresholds, the system improved its performance in locating the robot on other regions, however this improvement came at the expense of losing accuracy on region 1.

Table 1. Fixed thresholds

Region	n. of images	correct answers	correct answers (%)
1	320	235	73
2	436	119	27
3	438	222	50
4	111	44	40
5	56	21	38
Global	1361	641	47

The obvious approach for improving the poor results obtained by fixed-thresholding is to *adjust the thresholds for each snapshot taken*. The technique we have used to perform this adjustment is the Otsu method [25] (cf. Section 4). This should be able to automatically find the threshold for segmenting objects of interest (i.e. casters and their shadows) from background. The results obtained are represented in Table 2.

Table 2. Adaptive threshold using the Otsu method

Region	n. of images	correct answers	correct answers (%)
1	320	190	59
2	436	195	45
3	438	157	36
4	111	27	24
5	56	14	25
Global	1361	583	43

Table 2 shows that the results with a variable threshold method, surprisingly, were slightly worse than those obtained with a fixed threshold (Table 1). For global localisation, the method answered correctly on 43% of the total 1361 snapshots. The localisation at region 1 was correct in 59% of the trials (decreasing from the 70% obtained with a fixed threshold), and the localisation accuracy on the other regions was below 50%. Investigation of the pixel value distributions indicated that the problem is that these distributions are not in general bi-modal, which increases the difficulty of searching for an appropriate threshold from the image histogram.

In our third set of results, the robot was set to vary the threshold until the interpretation of the target object and its shadow matches a robot's prediction of its location (using Algorithm 2, as explained in Section 4). The results obtained are represented in Table 3, which shows that the system achieved an accuracy of around 90% in all regions. Thus the incorporation of knowledge about shadow appearance, and reasoning based upon past location, can greatly assist in the refinement of a simple shadow-detection algorithm, outperforming also a traditional algorithm for adaptive thresholding.

6 Discussion and open issues

In this work we investigated robot self-localisation using qualitatively distinct regions defined of a visual observation of cast shadows. Central to this problem is the segmentation of casters and shadows from the background, which was accomplished here by thresholding on value (V) on the HSV colour space. In the present paper we proposed a new strategy for calibrating this threshold, where the prediction about the robot's location is used to search for a match between the interpreted position (as given from visual observation) and its predicted location. In order to evaluate this method, we presented three sets of experiments whereby different ways of defining the threshold were tried. In the first set of experiments a hand-coded fixed threshold was used, in the second set of experiments we used Otsu's method [25] in order to find the threshold values from the image histogram. Finally, the third set of experiments presents the results of applying our proposed method for matching the prediction with the observation.

Table 3. Knowledge-based adaptive threshold

Region	n. of images	correct answers	correct answers (%)
1	320	297	93
2	436	385	88
3	438	410	94
4	111	102	92
5	56	48	86
Global	1361	1242	91

The intuition behind the experiments with fixed thresholds was to provide a lower-bound for the evaluation of our idea, since (as we hypothesised) nothing could perform worse than a hand-coded threshold. Experiments with Otsu's method were then to set the standard, as this is one of the most traditional methods for adaptive thresholding. However, it turned out that Otsu's performance was in fact approximately as accurate as that of using the fixed threshold. This is due to the fact that we chose for the first set of experiments the best threshold we could find, after a number of trials where the value was changed by hand. Otsu's method, however, had to deal with arbitrary images, where it had to maximise a value that is dependent on an *a priori* hypothesis of bi-modal pixel distribution. This was not the case in some of the snapshots taken by the robot: a great number of them suffered from the effect of reflections of the light source on the caster; moreover, from some angles, there was a negative gradient of luminosity just behind the object. These problems caused Otsu's method to perform worse than using a fixed threshold.

In contrast, the method for calibrating the threshold using the prediction about the robot's location performed as well as could be expected, obtaining an accuracy of around 90% with respect to our dataset containing 1361 snapshots of the robot's environment. However, this method is totally dependent on the capability of the robot's actuators on generating accurate predictions for the robot's future location, given a moving action. In this paper, the robot's motion was completely pre-programmed in order to minimise the actuator's noise, this gave us the guarantee that we were only evaluating the localisation procedures. We leave for a future work applying this framework on a system that has a path planning module, so that it can be verified how the qualitative localisation procedure proposed in this paper is affected by errors in the planning-acting-sensing cycle. Evaluating the ideas put forward in this paper on a more complex scenario is also a desirable future goal.

Also subject for our long term investigations is the complete exploration of the knowledge content of shadows, as described in [3], in order to create a robotic system that is capable of perceiving (and interpreting) shadows in a similar fashion to humans. The reason for pursuing this goal resides in our hypothesis that the human percep-

tual system, by preferring shadow information over other depth cues (even when these cues contradict each other [20]), is in fact saving processing time. Investigating how this could be accomplished in robotic systems is a major motivation of this work.

Although this work explores only a qualitative theory about space, this choice does not preclude the use of quantitative or statistical methods. Rather, we believe that qualitative methods in robotics should complement the traditional numerical algorithms, providing another processing level where it is possible to extract information from the knowledge level.

7 Conclusion

This paper has demonstrated how the incorporation of qualitative spatial representation and *a priori* knowledge about shadow regions can be combined to enhance a simple shadow-detection algorithm based upon thresholding. Future work will involve the incorporation of more sophisticated shadow detection algorithms, and the extension of the current snapshot-based system to one which incorporates continuous video, and the inclusion of shadow reasoning within the perception-planning-action loop.

A number of questions have been raised by this work, and we consider these questions in themselves to be a useful contribution. For example, how can shadows improve object localisation when contrasted to object-based methods? Under what conditions can shadows be effectively exploited? How can we combine predictive shadow-based localisation with predictive localisation based upon object pose? These are all questions which we hope to consider in more depth in future work.

REFERENCES

[1] A. O. Balan, M. J. Black, H. Haussecker, and L. Sigal, 'Shining a light on human pose: On shadows, shading and the estimation of pose and shape', *Proc. International Conference on Computer Vision (ICCV)*, (2007).

[2] Xiaochun Cao and Hassan Foroosh, 'Camera calibration and light source orientation from solar shadows', *Proc. Computer Vision and Pattern Recognition (CVPR)*, **105**(1), 60–72, (2007).

[3] R. Casati, 'The shadow knows: a primer on the informational structure of cast shadows', *Perception*, **33**(11), 1385–1396, (2004).

[4] Y. Caspi and M. Werman, 'Vertical parallax from moving shadows', in *Proc. Computer Vision and Pattern Recognition (CVPR)*, New York, USA, (2006).

[5] U. Castiello, D. Lusher, C. Burton, and P. Disler, 'Shadows in the brain', *Journal of Cognitive Neuroscience*, **15:6**, 862–872, (2003).

[6] P. Cavanagh, 'The artist as neuroscientist', *Nature*, **434**, 301–307, (2005).

[7] C.-C Cheah, C. Liu, and J.-J. E. Slotine, 'Adaptive tracking control for robots with unknown kinematic and dynamic properties', *I. J. Robotic Res.*, **25**(3), 283–296, (2006).

[8] A. G. Cohn and S. M. Hazarika, 'Qualitative spatial representation and reasoning: An overview', *Fundamenta Informaticae*, **46**(1-2), 1–29, (2001).

[9] A. G. Cohn and J. Renz, 'Qualitative spatial representation and reasoning', in *Handbook of Knowledge Representation*, 551–596, Elsevier, (2008).

[10] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, and A. Prati, 'The sakbot system for moving object detection and tracking', in *Video-based Surveillance Systems: Computer Vision and Distributed Processing (Part II - Detection and Tracking)*, 145–158, Kluwer Academic Publishers, (2001).

[11] T. da Costa Kauffmann, 'The perspective of shadows: The history of the theory of shadow projection', *Journal of the Warburg and Courtauld Institutes*, **38**, 258–287, (1979).

[12] R. Davis, H. Shrobe, and P. Szolovits, 'What is a knowledge representation?', *AI Magazine*, **14**(1), 17–33, (1993).

[13] P.M. Fitzpatrick and E.R. Torres-Jara, 'The power of the dark side: using cast shadows for visually-guided touching', in *Proc. of the 4th IEEE/RAS International Conference on Humanoid Robots*, pp. 437–449, (2004).

[14] J.-W. Hsieh, W.-F. Hu, C.-J. Chang, and J.-S. Chen, 'Shadow elimination for effective moving object detection by Gaussian shadow modeling', *Image and Vision Computing*, **21(6)**, 505–516, (2003).

[15] J. B. Huang and C. S. Chen, 'Moving cast shadow detection using physics-based features', in *Proc. Computer Vision and Pattern Recognition*, (2009).

[16] Y. Kunii and T. Gotoh, 'Evaluation of Shadow Range Finder: SRF for Planetary Surface Exploration', in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2573–2578, (2003).

[17] J-S. Lee, S-G. Roh, D. W. Kim, H. Moon, and H. R. Choi, 'In-pipe robot navigation based upon the landmark recognition system using shadow images', in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1857–1862, (2009).

[18] P. Mamassian, 'Impossible shadows and the shadow correspondence problem', *Perception*, **33**, 1279–1290, (2004).

[19] P. Mamassian and R. Goutcher, 'Prior knowledge on the illumination position', *Cognition*, **81**(1), B1–B9, (2001).

[20] P. Mamassian, D. C. Knill, and D. Kersten, 'The perception of cast shadows', *Trends in cognitive sciences*, **2(8)**, 288–295, (1998).

[21] N. Martel-Brisson and A. Zaccarin, 'Learning and removing cast shadows through a multidistribution approach', *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **29 (7)**, 1134–1146, (2007).

[22] N. Martel-Brisson and A. Zaccarin, 'Kernel-based learning of cast shadows from a physical model of light sources and surfaces for low-level segmentation', in *Proc. Computer Vision and Pattern Recognition*, (2008).

[23] B. Maxwell, R. Friedhoff, and C. Smith, 'A bi-illuminant dichromatic reflection model for understanding images', in *Proc. Computer Vision and Pattern Recognition*, (2008).

[24] S. Nadimi and B. Bhanu, 'Physical models for moving shadow and object detection in video', *IEEE transactions on Pattern Analysis and Machine Intelligence*, **26(8)**, 1079– 1087, (2004).

[25] N. Otsu, 'A threshold selection method from gray-level histograms', *IEEE Trans. Sys., Man., Cyber.*, (9), 62–66, (1979).

[26] F. Porikli and J. Thornton, 'Shadow flow: A recursive method to learn moving cast shadows', in *Proc. International Conference on Computer Vision (ICCV)*, (2005).

[27] A. Prati, I. Mikic, M.M. Trivedi, and R. Cucchiara, 'Detecting moving shadows: algorithms and evaluation', *IEEE transactions on Pattern Analysis and Machine Intelligence*, **25(7)**, 918–923, (2003).

[28] D. Randell, Z. Cui, and A. Cohn, 'A spatial logic based on regions and connection', in *International Conference on Knowledge Representation and Reasoning*, pp. 165–176, Cambridge, U.S., (1992).

[29] D. Randell, M. Witkowski, and M. Shanahan, 'From images to bodies: Modeling and exploiting spatial occlusion and motion parallax', in *International Joint Conference on Artificial Intelligence*, pp. 57–63, Seattle, U.S., (2001).

[30] Elena Salvador, Andrea Cavallaro, and Touradj Ebrahimi, 'Cast shadow segmentation using invariant color features', *Computer Vision and Image Understanding (CVIU)*, **95**(2), 238–259, (2004).

[31] P. Santos, H.M. Dee, and V. Fenelon, 'Qualitative robot localisation using information from cast shadows', in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 220–225, (2009).

[32] J. Stauder, R. Mech, and J. Ostermann, 'Detection of moving cast shadows for object segmentation', *IEEE Transactions on multimedia*, **1(1)**, 65–76, (1999).

[33] P. Tompkins, A. Stentz, and W. L. Whittaker, 'Automated surface mission planning considering terrain, shadows, resources and time', in *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '01), Montreal, Canada.*, (June 2001).

[34] A. Troccoli and P. K. Allen, 'A shadow-based method for model registration', in *Computer Vision and Pattern Recognition Workshops*, (2004).

[35] Y. Wang, K.-F. Loe, T. Tan, and J.-K. Wu, 'A dynamic Hidden Markov Random Field Model for foreground and shadow segmentation', in *Proc. IEEE Workshop on Applications of Computer Vision*, (2005).

Zooming in on Collective Motion

Zena Wood[1] and **Antony Galton**[2]

Abstract.

The movement patterns of collectives are probably one of the most important properties of a collective that we may wish to qualitatively reason about. However, despite their ubiquity in our everyday lives, it would appear that we do not currently possess the necessary tools. Granularity plays a crucial role in the description of motion. Therefore, if a framework is to be developed that allows the sufficient representation and qualitative reasoning power about collective motion, it must be possible to consider the framework over different spatial and temporal granularities. Such a framework is presented and explored through the use of examples.

1 Introduction

Collectives are ubiquitous in everyday life, with examples including traffic jams, flocks of birds and crowds. Since some collectives are defined by their movement patterns it is probably one of the most important properties that we may wish to reason about, especially qualitatively. Consider the movement patterns detected by an intelligent traffic monitoring system. The detection of a cluster of stationary vehicles could be interpreted as a traffic jam. Movement patterns could also be used to detect when an accident has occurred and, therefore, the need for officials at the scene or a suitable diversion route to be put in place. Despite the importance of being able to reason about the motion exhibited by the collective (i.e., collective motion), it would appear that we do not currently possess the necessary tools.

Much research has been carried out into the movement patterns that are exhibited by individuals and the important features of such movement identified [2, 3, 8]. However, there are many additional features that must be taken into consideration when dealing with collective motion. A key difference is the crucial role that is played by granularity in the description of motion. Hornsby and Egenhofer [7] note that the use of multiple granularities allows a user to uncover much more information about the movement in the data and they present a model which allows the movement of individuals or objects to be represented over multiple granularities both in space and time. Due to the structure of a collective and its motion, collective motion forces us to consider the notions of both spatial and temporal granularity. If a framework is to be developed that allows sufficient representation and qualitative reasoning power about collective motion, it must be possible to consider the framework over different spatial and temporal granularities. In this paper we present such a framework.

The ultimate goal of the research reported in this paper is to develop an ontology of collectives and collective behaviour that can adequately support the representation of and reasoning about such phenomena for the purposes of modelling, reasoning and prediction. As a prerequisite we believe it is necessary to survey the range of collectives and their associated movement patterns in order to classify them for the purpose of the ontology. The proposed framework examines the types of motion that become apparent when a collective is described at different levels of granularity. This approach could be considered as a method of 'zooming' from a coarse spatial granularity to a fine granularity and stopping at each level where an important aspect of the collective becomes apparent. A similar approach is used to take account of the effects of temporal granularity on the way in which motion is described.

An explanation of why the notions of spatial and temporal granularity must be considered is given in section 2 followed by a brief discussion of the relevant existing research (section 3). The proposed framework is set out in section 4 with examples given throughout to illustrate how it may used; section 6 indicates the relationships that exist between the different aspects of the framework. The paper concludes with a list of research questions which must be addressed in order to complete the development of the framework (section 7).

2 The Role of Granularity

We have stated that the structure of a collective and its motion forces us to consider the notions of spatial and temporal granularities, but what exactly do we mean by this?

The way in which motion is described is dependent on the temporal granularity at which it is observed. Consider walking as an example. At a coarse level of temporal granularity this can be seen as the motion from one location to another but at a fine level, walking can be considered as comprising the repeated movement of one leg after another. Except in the case of a very simple application, any framework that represents motion should take account of the importance of temporal granularity and allow the motion to be analysed over multiple levels of temporal granularity. Some of the existing research decomposes complex patterns into simpler ones [8, 2, 3]. Usually referred to as *primitives*, these simpler movement patterns are ones in which only one movement parameter changes. The approach of using primitives as 'building blocks' to form more complex movement patterns could be seen as one way in which the granularity problem can be overcome. However, such primitives must be chosen with care (see section 4).

Each collective can be observed from at least two points of view: at a lower-level where all that can be seen are the individual members of the collective; or, at a higher-level where the collective itself can be considered as a single entity. We propose that these two levels can be conceived as two distinct levels of spatial granularity (i.e., a fine level and a coarse one). However, why is this distinction important when it is the motion of the collective that we are concerned with?

At both of these levels, different aspects of the collective's movement can be observed. One might argue that the motion of the collec-

[1] University of Exeter, UK, email: Z.M.Wood@ex.ac.uk
[2] University of Exeter, UK, email: A.P.Galton@ex.ac.uk

tive is simply the aggregated motions of the individual members and therefore only the motion of the individuals or that of the collective needs to be analysed. In one sense this is true since the movement of any collective arises from the addition of the movement of the individuals (i.e., as a vector sum). However, the motion of the individuals may be qualitatively different from that of the collective that they are members of and it is the qualitative nature of the motion that we wish to reason about.

Consider the movement patterns exhibited by two collectives: a procession, and a crowd which slowly drifts west. In the former example the motion of the collective when considered as a single unit will be qualitatively similar to that of its individual members; in [11] these two motions are said to be *co-ordinated*. The latter example would exhibit two *uncoordinated motions* [11]: the individual members will be moving around randomly whereas a linear motion would be observed in a westerly direction when considering the collective as a single unit. These two examples illustrate why both the motion of the collective and that of its members must be considered, but it also indicates that the relationship that exists between the two types of movement must also be represented. If we focused on the motion of the individuals, important information about the motion of the collective, when considered as a single unit, may be lost, and vice versa.

As well as a fine and a coarse level of spatial granularity needing to be considered, a third may also be needed. At a given level of spatial granularity a collective can be observed as occupying a two or three dimensional region of space. This region of space could be referred to as the 'footprint' of the collective.[3] It is important that we distinguish the coarsest level, at which the collective is treated as a point, from an intermediate level, at which the extension of the footprint is seen. The ways in which a footprint evolves is a key feature of the collective and could impact on the movement pattern that is being observed. Motion patterns such as splitting, merging, expansion and contraction are only some of the possible salient features of the evolution of the footprint; others are discussed in section 4.

3 Existing Research

This section will focus on existing research into motion which considers the importance of granularity. A detailed investigation into existing research into motion in general can be found in [10].

Although little work has been done on the effects of granularity on collective motion, there have been some investigations into its effects on the description of an individual's motion. [3, 8, 2] all propose the breaking down of complex movement patterns into simpler primitives where only one movement parameter changes and, as already noted, this is one approach in which the granularity problem could be overcome. However, these ideas do not seem to be explicitly explored in their research. Much of the existing research specifies that a suitable level of granularity should be chosen in relation to what is being observed; switching between different levels of granularities is not possible. The ability to move between levels of granularity is a difficult problem to overcome but has been examined by Hornsby and Egenhofer [7].

Hornsby and Egenhofer propose a model which allows the movement of individuals or objects to be represented over multiple granularities both in space and time. Inspired by Hägerstrand's time geography [6], the proposed model represents each individual's movement as a geospatial lifeline. Similarly to that of Laube et al [8], a geospatial lifeline records the locations visited by the individual over a period of time, but Hornsby and Egenhofer also allow the user to choose the level of granularity at which to describe the geospatial lifeline. Depending on the level chosen, the lifeline will be modelled as a either a lifeline bead, necklace, thread, tube or trace — a lifeline bead occurs at the finest level of granularity and a lifeline trace at the coarsest.

Calculated according to an individual's given start and end points in space-time and maximum speed, the set of possible locations that an individual may have visited or passed through is given by a lifeline bead. If the granularity that the lifeline is observed from is refined, a lifeline necklace can be observed. Consisting of a sequence of beads, the necklace arises from additional sample points being introduced into the model. It is important to note that the end point of one bead will also be the starting point of the next bead. Many methods are presented within [7] to view the movement at a coarser level of granularity. One could move from viewing a necklace to a lifeline thread, a lifeline tube or a lifeline trace; this series allows the model to be viewed at an increasingly coarse level of granularity.

4 The Proposed System

Owing to the crucial role that granularity (both spatial and temporal) plays in the description and therefore possible reasoning power of a framework that represents collective motion, we have decided to base our proposed framework on these ideas.

Each example of collective motion that is being considered will be examined at three different levels of spatial granularity: a coarse level where the collective can be observed as a point, an intermediate level where a possible footprint for the collective can be determined and a fine level where the individual members are apparent. Each of these levels of spatial granularity will be taken in turn and the type of motion exhibited at this level considered.

To describe the temporal structure of motion we will make use of the distinction between processes and events. This is, to be sure, notoriously slippery, and the terms have been used in different ways by different authors (cf. [4]). By a process we shall here understand a type of activity that is *homogeneous* and *open-ended*. The former means that a process does not break down into distinct phases that are qualitatively distinct; the latter, that it does not have intrinsic necessary start and end points but can in principle be continued indefinitely. A clear example that exhibits both the characteristics is the rotation of the earth about its axis. However, for both the characteristics mentioned, a caveat is required. First, homogeneity must be understood in relation to the temporal granularity under which the phenomenon is being described. As mentioned above, we may describe the activity of walking either at a fine temporal granularity as a succession of discrete overlapping leg movements or at a coarse granularity as a smooth movement along a trajectory.

Regarding open-endedness, what matters is continuation *in principle*. The process of falling, for example, is typically terminated when the falling object strikes the ground or other surface obstructing its downward path, and such a surface will always be present in an earth-bound setting; however, in any particular case one can imagine that the obstructing surface is not there, in which case the object could continue falling. The termination of any particular falling by striking the ground is not, in other words, intrinsic to the falling itself. Contrast this with a structured activity such as baking a cake, which in principle cannot be continued beyond the point at which the cake is made (any subsequent cake-making activity relating to a different cake and therefore not part of the same process). Hence baking a cake is not a process in our sense, although both baking, considered gener-

[3] There is no unique footprint for each collective; many possible footprints could be identified for any one collective [5].

ically without reference to what is being baked, and baking cakes (where an indefinite plurality of cakes is understood) are.

We contrast processes with events, which are characterised as occurrences with definite beginning and endpoints. The inclusion of a beginning and ending as parts of the event already make it minimally structured (and thus non-homogeneous), and events typically exhibit other, internal, structure as well. Moreover, inclusion of the end-point means that the event is not open ended, since once it has stopped, that particular event cannot continue. The simplest forms of event are

1. Transitions, which occur when a process becomes either active (i.e., starts) or inactive (stops), or when one process is replaced by another. Examples are: starting to walk, stopping walking, and breaking from a walk into a run, respectively.
2. "Chunks" of homogeneous process. A chunk of walking happens when someone starts walking, keeps walking for a while, and then stops walking (or breaks into a run). A chunk is by nature bounded by transitions at either end.

Our central contention is that complex movements can be analysed as sequences of *episodes*, each of which is characterised as a maximal chunk of homogeneous process, with contiguous episodes linked by transitions. As noted, homogeneity is granularity-relative, and hence the episodes one uses to describe a particular movement will also be dependent on granularity. Thus what at a coarse granularity may be described as a single episode consisting of a person walking from A to B, at a very fine granularity may be resolved into a succession of overlapping leg-movement episodes (see figure 1).

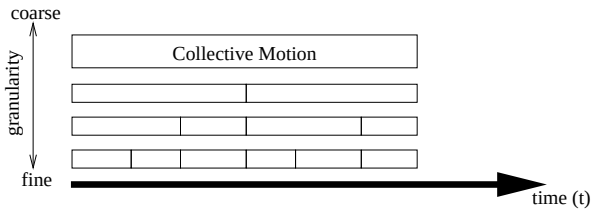


Figure 1. Decomposition of motion into episodes at different granularity levels.

To determine the necessary episode types for each aspect of the collective's motion (i.e., at the three levels of spatial granularity that we have previously identified), the key qualitative differences relative to the aspect of the collective that is being observed will be identified. Consider the simple example illustrated at the left of figure 2. At a temporal granularity where this movement pattern is observed, qualitative difference in the direction of the movement results in the identification of two episodes: the first episode consists of continuous movement northward and the second continuous movement eastwards. The change in direction signifies the end of the first episode and the beginning of the second. However, when this movement pattern is observed at a finer temporal granularity (the right of figure 2), this abrupt change of direction appears as gradual. An episode of *turning* is here revealed as interposed between northward and eastward sections of movement.

5 Identifying the Necessary Episodes

For each aspect of the collective's motion that is to be described (i.e., the collective, its footprint and its individual members), we will define a basic set of episode types which will serve as the building blocks for constructing the complex movement patterns that are discernible at that granularity. To do this, the salient qualitative features of each type of motion must be identified. It is important to remember that the framework that we are proposing is aimed at providing a means of qualitatively reasoning about collective motion. We have tried not to restrict ourselves to a specific domain and therefore some applications may not require all of the episode types that are set out within this section; some may also require more.

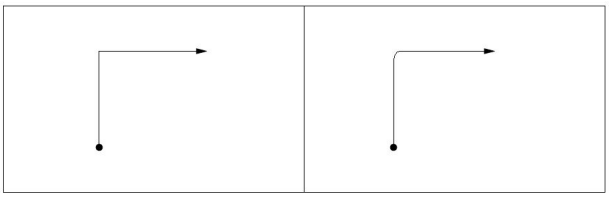


Figure 2. A simple example of motion considered at two levels of temporal granularity.

5.1 Observing the Collective at a Coarse Granularity

Each collective that is being considered will first be described at a level of spatial granularity where it can be treated as a point. This will allow us to identify the motion that is exhibited by the collective when considered as a single compact entity. Much research has been carried out into the features of this type of movement (i.e., point-to-point movement) [1, 2, 3] but not in terms of episodes or homogeneous processes. Therefore, we must identify the key spatial characteristics of such movement.[4]

Initially consider whether or not the point is moving. Although often ignored we believe a period when the collective is stationary to be an important type of movement pattern. Consider the movement pattern exhibited by a commuter or an elderly person living at home. A sustained stationary period could indicate a traffic jam or problem respectively. STATIONARY and NON-STATIONARY episodes are differentiated by looking at the speed of the movement; this will or will not be respectively zero. Since we are qualitatively reasoning about the motion, three more salient episode types regarding the speed of the motion can be defined: UNIFORM MOTION, ACCELERATED MOTION and DECELERATED MOTION. The first indicates a period of movement where the speed is constant but not zero. One may wish to identify the points at which motion begins and ends by including ACCELERATION FROM REST and DECELERATION TO HALT. This distinction seems sensible: for example, although at one level of granularity a point may seem to go from stationary to uniform motion instantaneously, there must be some acceleration at a lower level of granularity to obtain uniform motion from a stationary state.

Point-to-point motion cannot be sufficiently modelled solely by looking at its speed. If the speed is not equal to zero (i.e., a non-stationary episode), the direction of the motion could be considered another salient qualitative feature. The direction of movement could be constant (i.e., LINEAR MOTION); or vary, resulting in a CURVING MOTION. Further episode types could be defined to represent the different possible directions of linear motion. For example, the direction could be characterised as north, south, east or west. However, these are rather arbitrary; some users may require further detail regarding

[4] The corresponding episode types are given in SMALL CAPS.

the direction and wish to include north and north-east. Where curving motion occurs in a setting where an appropriate reference frame is identified (e.g., an object moving over a surface, or an object with intrinsic 'upper' or 'lower' sides) we can further distinguish between CURVING LEFT and CURVING RIGHT. However, in a more general setting this distinction may be inapplicable.

The remaining episode types at this level of spatial granularity come from further refinement of the description of the motion's curvature. The curvature of a motion pattern could be constant, given CIRCULAR MOTION. A special case of this type of motion can be defined where the start and end point of the motion are equal (CIRCULAR MOTION WITH RETURN TO START). However the curvature of a motion pattern could also increase or decrease. Consider the motions depicted in 3. The motion patterns depicted on the right illustrate two episodes where the curvature increases (SPIRALLINGOUT) whereas those shown on the left has decreasing curvature (SPIRALLINGIN). The use of the word 'spiral' could be confusing since it day-to-day language this usually refers to a motion that has gone one or more full turns. However, we believe it remains the best term to denote these episode types.

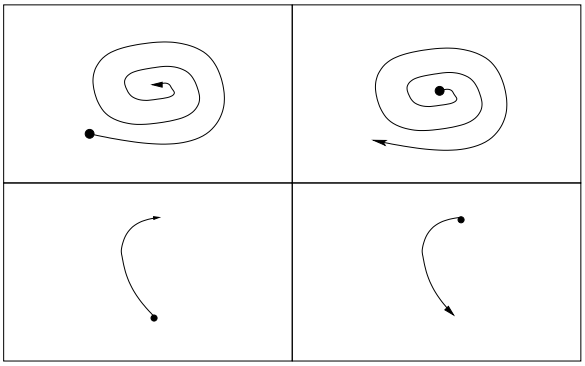


Figure 3. Four episodes of spiralling motion.

The motion patterns shown in figure 4 illustrate how we can use some of the episode types that we have defined. It is important to note, that only one parameter of the motion needs to change in order to define a new episode. For example, if the point is moving at a constant speed in a linear direction but then starts to accelerate in the same direction, the change in speed will indicate a new episode type of accelerated motion. However, we are not viewing the movement pattern in real-time and therefore, cannot include episode types relating to speed. Going from left to right, the first motion pattern shows LINEAR MOTION followed by SPIRALLINGOUT; the second CIRCULAR MOTION WITH RETURN TO START followed by LINEAR MOTION; and, the third motion indicates CURVING LEFT and CURVING RIGHT with a point of inflection in the middle.

5.2 Observing the Collective at an Intermediate Granularity

At this level of granularity a possible footprint for the collective and its evolution are examined. The ways in which a footprint can be derived is an active field of research [5] but it lies outside the scope of this paper; for our purposes it is assumed that a suitable footprint has already been chosen.

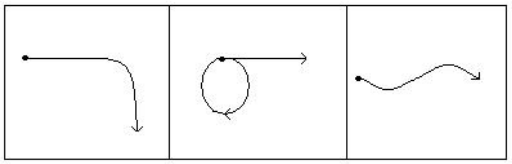


Figure 4. Example of point-to-point motion that comprises more than one episode.

The ways in which a footprint can evolve can be characterised by changes in its size, shape and orientation. However the number of these possible changes, especially those relating to shape, is vast and it would be very difficult to systematically and clearly define all of the necessary episode types at this level of granularity as done in the previous section. Therefore, in this section we present only some of the possible episode types that could be used in our framework — it is definitively not considered exhaustive. In certain contexts only some of the episode types presented may be relevant but in other contexts more may need to be defined.

With regard to the size of a footprint, a salient difference is whether or not it changes. It may not (i.e., CONSTANT SIZE) but if the size does vary episode types can be defined according to whether the size is increasing (EXPANSION) or decreasing (CONTRACTION). For those footprints whose size changes, the rate of growth of this change could be considered (i.e.,UNIFORM GROWTH, INCREASING GROWTH or DECREASING GROWTH). In order to define these episode types, the way in which size is measured must be determined. One possibility is to use the area (or volume if three-dimensional).

The way in which a shape can change is a complex topic and a substantial research field within itself. Many measures of shape have been identified [9]; it would not be possible to include all of these. Instead we have identified a few possible examples that we believe establish some good qualitative measures of shape.

Abler *et al* look at the structure of a dynamic phenomenon at its origin and destination points [1]. Four spatial characteristics are highlighted as being capable of representing a wide range of phenomena: points, lines, areas and volumes. They use these four characteristics to form a 4×4 matrix that they believe could account for all types of movement. We can use this concept to characterise the different transitions that a footprint can undergo in terms of its dimensionality. For example, consider a flock of birds all initially perched along a telephone wire. At some point, the flock moves off the wire and flies off to a nearby piece of land. Whilst on the cable, the flock occupies an elongated one-dimensional region. As soon as some of the birds begin to take flight, the flock's footprint is three-dimensional; once landed, the representative footprint decreases in dimensionality to two dimensions. The dimensionality of a footprint forms a discrete series but the change in dimension results in a new episode type. Therefore, the following episode types can be defined: 1D-TO-2D, 1D-TO-3D, 2D-TO-3D, 2D-TO-1D and 3D-TO-1D. Of course the dimensionality of the footprint may not change and this is represented by CONSTANT DIMENSIONALITY.

As already noted, collectives can split or merge; this feature of a collective can only be observed at the level of granularity where the evolution of the footprint is being considered. Both of these features could be represented by suitable episode types. The two episode

types SPLITTING and MERGING would link episodes in which the number of components do not change (CONSTANT NUMBER OF COMPONENTS).

Other possible qualitative differences in terms of a footprint's shape could be its CIRCULARITY represented by the ratio of the object to its least bounding circle; its COMPACTNESS represented by the relation between its perimeter and its area; or its ELONGATION which in two-dimensions will be represented by the aspect ratio of its minimal bounded rectangle (in any orientation). The question arises as to how to qualitatively measure these three latter properties. One could simply define episode types which considered whether or not the feature is maintained, increased or lost. For example, consider the evolution of the footprint in figure 5. Between t_1 and t_2 the footprint can be considered as MAINTAINED CIRCULARITY. However, during t_2 and t_3 the collective loses some of its circularity and therefore can be considered of episode type DECREASED CIRCULARITY. Depending on the context, compactness or elongation may be preferred to describe the evolution of this footprint instead of circularity, yielding the episode types CONSTANT COMPACTNESS, INCREASING COMPACTNESS, DECREASING COMPACTNESS, CONSTANT ELONGATION, INCREASING ELONGATION and DECREASING ELONGATION.

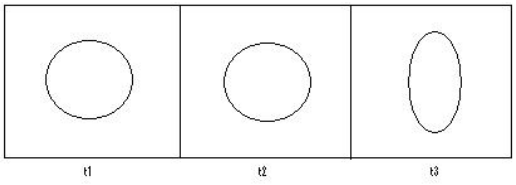


Figure 5. An example of how the shape of a footprint may change

The final qualitative feature that we consider here is the orientation of the footprint and whether or not it is possible to consider a footprint as rotating. Consider the footprints depicted in figure 6. Out of these two footprints, it would only be possible to identify whether or not the footprint on the right is rotating[5]. Since the footprint represents the region occupied by a collective it seems odd to consider the footprint as rotating. If a footprint could be considered as rotating, it could rotate in either a clockwise or anticlockwise direction (i.e., ROTATE RIGHT and ROTATE LEFT).

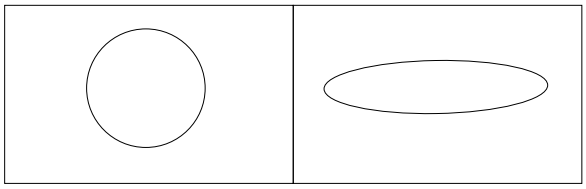


Figure 6. Identifying rotation

[5] This is a perfect example of where the necessity of an episode type depends on context.

5.3 Observing the Collective at a Fine Granularity

As regards to the motion of any one individual — which may be considered pointlike — the available motion patterns are precisely those already discussed in relation to the motion of the collective as a whole. In this section we are only concerned with how the motion of all the individual members of the collective are related to each other.

The possible salient features of the movement of the individuals could be found by examining how their movement patterns relate to each other. In this respect an initial distinction could be made according to whether or not the movements exhibited by the individuals were qualitatively similar or not. If they are, then two episode types could be defined: STATIONARY, where none of the individuals are moving and CONCURRENT MOTION, where all the individuals are moving and their movement patterns are qualitatively similar. However, a group of individuals could exhibit various types of CORRELATED MOTION (figure 7): they could all be moving towards a central point CONVERGING, from a central point DIVERGING, or in parallel PARALLEL MOTION, in FORMATION MOVING, towards a central point CONVERGING or from a central point DIVERGING.

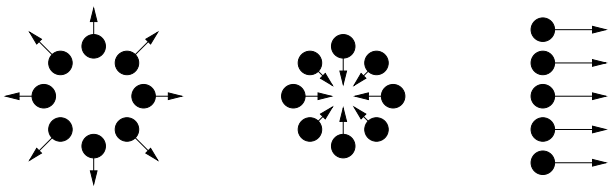


Figure 7. Examples of correlated motion.

At the opposite extreme to fully correlated motion is totally uncoordinated CHAOTIC MOTION but these are only extremes, and many intermediate gradation can exist. Currently, this episode type (i.e., CHAOTIC MOTION) has not been distinguished further. However, among these intermediate levels other notions may exist such as 'full' or 'lagged co-incidence in space and time' as defined by Dodge *et al* [3] but a systematic taxonomy of this remains problematic.

6 Discussion

Although not investigated exhaustively, it is clear that relationships exist between episodes that have been identified for different levels of granularity. Here a brief discussion is given on these relationships and what it means in terms of the reasoning about collective motion.

If all the individuals are stationary then the collective itself must be stationary. The converse, however is false. Within a stationary collective, the individuals can exhibit all manner of movement patterns, either correlated or chaotic. As an example of the former consider a collective where all of the individuals are moving around a central point (figure 8). At a coarse level of granularity where the collective can be observed as a point, the collective will appear stationary. Similarly the footprint will appear to have a constant shape and size. Therefore, it is at the level of the individuals where all of the information regarding the collective's motion is concentrated.

An episode of expansion or contraction of the footprint could be caused in two ways: the individual members of the collective could be converging or diverging; or, members may be leaving or joining

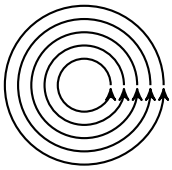


Figure 8. Individuals all move around a central point.

the collective. Variable membership is an important feature of collective phenomena [11] and episodes such as expansion and contraction when the individuals are not converging and diverging respectively could be a good indication of this. In addition to variable membership, the movement of the individuals could reveal the presence of sub-collectives. Consider the traffic travelling along a road. The movement patterns exhibited by the traffic identify two sub-collectives: one consisting of the traffic travelling in one direction along the road and another whose members are traveling in the opposite direction.

7 Further Work

In order to complete the development of the proposed framework the following research questions must be answered.

- Can a sufficient collection of episode types for representing the evolution of the footprint and the movements of the individuals be more systematically defined?
- It seems unlikely that simply stating that the individuals' motions are chaotic would be sufficient but can anything more be said about this type of movement. How does their chaotic movement relate to that of the collective? More examples of this type of collective motion must be examined to determine the answers to these questions.
- It would be useful to consider in more detail the way in which we describe movements in our day-to-day lives. Although this has been done to a degree in determining the current list of episodes; a more thorough investigation may lead to more episodes being defined.
- The proposed framework currently only represents each example of motion as a set of episodes. Can we include temporal characteristics to represent whether a motion pattern comprises a set of repeated episodes or a sequence of episodes?
- Must our framework examine what happens between episodes (i.e., the transitions)?
- The illustrative examples give an indication of how much information can be obtained using our framework but what is the best way to represent this information?
- We could consider whether the three levels that have been identified in our framework are sufficient for more complex cases. It is crucial that the levels are identified where the important features of a collective occur. There is at least one case where more than three levels are necessary (e.g., a collective of collectives). However, any new levels appearing in this case would be a combination of the basic three that we have already identified. It may be worth considering if there are cases where more levels exist than those generated by the basic three.
- Currently this work is only theoretical and in order to establish the efficiency of our proposed framework, it must be tested on movement patterns that are being observed in real time. We are currently in the process of collecting running data from various races from which we hope to be able to extract examples of movement patterns that can be adequately described using our framework.

8 Conclusion

Granularity, both spatial and temporal, plays a crucial role in the way in which collective motion is described, and therefore affects the reasoning power of any framework that represents collective motion. A framework has been proposed that takes into account the importance of granularity by allowing the motion of the collective to be observed over different levels of spatial and temporal granularity. By combining the information gathered over these levels, more detailed knowledge about the different types of collective motion can be obtained.

REFERENCES

[1] R. Abler, S. Adams, and P. Gould, *Spatial Organization: The Geographer's view of the World.*, Prentice-Hall International, 1970.

[2] N. Andrienko and G. Andrienko, 'Desgining visual analystics methods for massive collections of movement data.', *Cartographica*, **42**(2), 117–138, (2007).

[3] S. Dodge, R. Weibel, and A.K. Lautenschütz, 'Towards a taxonomy of movement patterns.', *Information Visualization*, **7**, 240–252, (2008).

[4] A. P. Galton, 'Experience and history: Processes and tehir relation to events', *Journal of Logic and Computation*, **18**(3), 323–340, (2007).

[5] A.P. Galton and M. Duckham, 'What is the region occupied by a set of points?', in *Geographic Information Science: Proceedings of the 4th International Conference, GIScience 2006*, eds., M. Raubal, H. J. Miller, A.U. Frank, and M.F. Goodchild, (2006).

[6] T. Hägerstrand., 'What about people in regional science?', *Papers in Regional Science*, **24**(1), 621, (1970).

[7] Kathleen Hornsby and Max J. Egenhofer, 'Modeling moving objects over multiple granularities', *Annals of Mathematics and Artificial Intelligence*, **36**(1-4), 177–194, (2002).

[8] P. Laube, S. Imfeld, and R. Weibel, 'Discovering relative motion patterns in groups of moving point objects', *International Journal of Geographical Information Science,*, **19**(6), 639668, (2005).

[9] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision (3^{rd} Edition)*, Thomson Learning, (2007).

[10] Z. Wood and A P. Galton, *Classifying Collective Motion*, chapter Behaviour Monitoring and Interpretation - BMI: Smart Environments, 129–155, IOS Press,, Amsterdam, 2009. Gottfried B, Aghajan H (eds).

[11] Z. Wood and A. P. Galton, 'A taxonomy of collectives', *Applied Ontology*, (2009).

Spatio-Temporal Abduction for Scenario and Narrative Completion

(A Preliminary Statement)

Mehul Bhatt[1] and **Gregory Flanagan**[2]

Abstract. Hypothetical reasoning is a form of inference that is useful in many application domains within the purview of *dynamic spatial systems*. Within a spatial context, this form of inference necessitates the ability to model (abductive) explanatory reasoning capabilities in the *integrated* context of formal *spatial calculi* on the one hand, and high-level logics of *action and change* on the other.

We present preliminary results by demonstrating the manner in which this form of explanatory reasoning may be implemented within the framework of the Event Calculus, which is a high-level formalism for representing and reasoning about actions and their effects. We use an example from the domain of automatic (virtual) cinematography / story-visualization and story-boarding, where the objective is to control camera / perspectives and animate a scene on the basis of apriori known *film-heuristics* and partial scene descriptions available from discourse material. Underlying the example domain lies the ability to perform spatio-temporal abduction in a generic context.

1 Introduction

Hypothetical reasoning about '*what could be*' or '*what could have been*' on the basis of, and possibly instead of, '*what is*' is a form of inference that is useful in many applications involving representing and reasoning about dynamic spatial knowledge. Many application instances within fields such as cognitive robotics, dynamic / event-based GIS, ambient / smart environments, and spatial design may all be considered to be within the scope of a class of *dynamic spatial systems* that require hypothetical reasoning capabilities in specific, and the ability to reason about space, actions and change in an integrated manner in general [5]. Both from a representational as well as a computational viewpoint, the basic set of requirements in all these application domains remains the same. Primarily, the following may be deemed important:

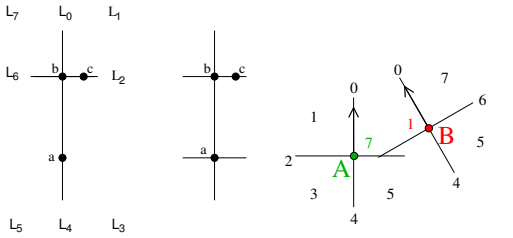
1. Qualitative scene modeling: the capability to abstract from precise geometric modeling of scenes and agent perspectives (e.g., of robots, avatars) by the use of qualitative spatial representation calculi pertaining to various aspects of space such as topology, orientation, directions, distance, size
2. Static scenario inference: given partial scene descriptions consisting of sets of spatial relationships between domain entities, the derivation of complete scene models by the application of constraint reasoning algorithms that infer the implicit spatial relationships from the explicit ones by exploiting the relational properties of the spatial calculi being utilized
3. Scenario and narrative completion: this is the most general case, where given partial narratives that describe the evolution of a system (e.g., by way of temporally ordered scene observations of a robot, event-based GIS datasets) in terms of high-level positional and occurrence information, the ability to derive completions that bridge the narrative by interpolating the missing spatial and action / event information in a manner that is consistent with domain-specific and domain-independent rules / dynamics

Whereas (1) and (2) involve static scenario inference in a strictly spatial sense, (3) necessitates commonsense reasoning about space, actions / events and change in an integrated manner [5]. The underlying intuition here being that spatial configurations in both real and virtual setups change as a result of interaction (i.e., actions and events) in the environment and that the formulation of hypothesis about perceived phenomena is closely connected to the commonsensical notions of interaction (e.g., manipulation, movement) in the real world. For instance, within a GIS, spatial changes could denote (environmental) changes in the geographic sphere at a certain temporal granularity and could bear a significant relationship to natural events and human actions, e.g., changes in land-usage, vegetation, cluster variations among aggregates of demographic features, and wild-life migration patterns. Here, event-based and object-level reasoning at the spatial level could serve as a basis of explanatory analyses, for instance by abduction, within a GIS [11, 16, 36]. Similarly, within a *behavior monitoring* and/or security system for a smart environment (e.g., home, office), *recognition of dynamic scenes* from changes in pre-designated configurations of qualified spatial configurations could be used as a basis of behaviour monitoring, activity recognition, alert generation and so forth [6, 8, 15].

From a computational viewpoint, hypothetical reasoning within the class of aforediscussed *dynamic spatial systems* requires a form of abductive explanation capability that may be implemented with a formal logic of action and change. This in turn necessitates the *embedding* of qualitative spatial calculi— i.e., the high-level axiomatic constitution of qualitative calculi —within a particular logic of action and change that is intended to be utilized [3, 4, 5].

In this paper, we demonstrate the manner in which explanatory reasoning may be implemented within the framework of the Event Calculus, which is a high-level formalism for representing and reasoning about actions and their effects. Specifically, we illustrate how spatio-temporal abduction may be directly realized with the discrete version of the Event Calculus (available as a reasoning engine) in

[1] SFB/TR 8 Spatial Cognition, University of Bremen, Germany. **email**: bhatt@informatik.uni-bremen.de
[2] Department of Computer Science, CalPoly, USA



(a) $\mathcal{SCC}$ and $\mathcal{DCC}$ Partitioning (b) The $\mathcal{OPRA}_2$ relation $\vec{A}\ 2\angle\frac{1}{7}\ \vec{B}$

Figure 1: Orientation Systems

the context of qualitative spatial calculi. We use an example from the domain of automatic-cinematography / story-visualization and story-boarding, where the objective is to control a camera and / or animate a scene on the basis of apriori known *film-heuristics* and partial *scene descriptions* available from discourse material.

2 Space and Change: Preliminaries

2.1 Qualitative Spatial Reasoning

The field of Qualitative Spatial Reasoning (QSR) investigates abstraction mechanisms and the technical computational apparatus for representing and reasoning about space within a formal, non-metrical framework [10]. Logical formalizations of space and tools for efficiently reasoning with them are now well-established [29]. Formal methods in spatial representation, referred to as spatial calculi, may be classified into two groups: topological and positional calculi. When a topological calculus such as the Region Connection Calculus (RCC) [28] is being modeled, the primitive entities are spatially extended and could possibly even be 4D spatio-temporal histories (e.g., in a domain involving the analyses of motion-patterns). Alternately, within a dynamic domain involving translational motion in a plane, a point-based (e.g., Double Cross Calculus [14], $\mathcal{OPRA}_m$ [22]) or line-segment based (e.g., Dipole Calculus [30]) abstraction with orientation calculi suffices.

In addition to ontological differences with respect to the nature of the primitive spatial entities, i.e., points, line-segments or spatially extended bodies, multiple viewpoints also exists with respect to the conceptualization of orientation relations themselves. In general, orientation either refers to the direction in which one object is situated relative to another, or the direction in which an object is pointing. The family of point-based orientation calculi define the direction of a located point to a reference point with respect to a perspective point [14]. Within this approach, three axes are used, one is specified by the perspective point and the reference point, the other two axes are orthogonal to the first one and are specified by the reference point and the perspective point respectively. These axes define 15 different ternary base relations.

The Single and the Double Cross Calculi [14] use points and orthogonal lines to partition the space in a finite number of locations, which could take the form of points, lines and extended regions (See Fig. 1). Similarly, Fig. 1(b) illustrates one primitive relationship for the Oriented Point Relation Algebra (OPRA) [22], which is a spatial calculus consisting of oriented points (i.e., points with a direction parameter) as primitive entities. The granularity parameter m determines the number of angular sectors, i.e., the number of base relations. Applying a granularity of $m = 2$ results in 4 planar and 4 linear regions (Fig. 1(b)), numbered from 0 to 7, where region 0 coincides with the orientation of the point. The family of $\mathcal{OPRA}_m$ calculi are designed for reasoning about the relative orientation relations between oriented points and are well-suited for dealing with objects that have an intrinsic front or move in a particular direction.

2.2 The Event Calculus

The Event Calculus is a logical language for reasoning about actions and change [20, 33]. It defines a set of predicates and axioms that provide an inference mechanism to decided "*what is true when*". In this paper we use a version of the Event Calculus, which is restricted to the domain of discrete time, referred to as the Discrete Event Calculus [23].

The (discrete) Event Calculus defines a fluent as a boolean property that represents a value that can change by direct or indirect effects of an event. Fluents can represent a numerical value of a quality, such as the temperature of a room, or a boolean proposition, such as *it is cold*. Additionally, fluents adhere to the commonsense notion of inertia, which states that a fluents value can only be changed by the effects of an event. Therefore, if some fluent is true at time-point t_0, it will necessarily be true at some later time-point t_1, unless an event occurs between t_0 and t_1 such that it effects the value of the fluent.

The Event Calculus provides a set of predicates and axioms to represent problem domains. The *HoldsAt* predicate defines when a fluents holds for a certain value; the *Happens* predicate defines when an event happen; and the *Initiates*, *Releases*, *Terminates* predicates express the effects events have on fluents. The following axioms relate the Discrete Event Calculus predicates to the properties of the calculus as described above.

$$HoldsAt(f,t+1) \Leftarrow HoldsAt(f,t), \wedge \neg ReleasedAt(f,t+1) \wedge \exists e.(Happens(e,t) \wedge Terminates(e,f,t))) \quad (1a)$$

$$HoldsAt(f,t+1) \Leftarrow Happens(e,t), \wedge Initiates(e,f,t) \quad (1b)$$

$$\neg HoldsAt(f,t+1) \Leftarrow \neg HoldsAt(f,t), \wedge \neg ReleasedAt(f,t+1) \neg \wedge \exists e.(Happens(e,t) \wedge Initiates(e,f,t))) \quad (1c)$$

$$\neg HoldsAt(f,t+1) \Leftarrow Happens(f,t), \wedge Terminates(e,f,t) \quad (1d)$$

Axioms (1a–1b) ensure the common-sense notion of inertia is enforced over fluents. (1a) states that a fluent f holds true at a time-point t + 1 if it held true at time-point t and that it was not released from it's inertia from the effect of an event. 1a additionally states that a fluent is true at a time-point t + 1 if an event occurs at t, which causes the fluent to be true. 1c and 1d describe the conditions when a fluents does not hold. A detailed discussion of the Event Calculus formalism is not necessary for this paper, but may be consulted in authoritative sources [23, 33].

3 The Domain of Automatic Cinematography

Automatic cinematography aims to derive a sequence of camera shots (i.e. the camera's perspective / orientation to the actors, camera's focus, and angle of view, etc.) from descriptions provided in a script [18] [9] [12]. In practice, most automatic cinematography involves using a knowledge-base of filming heuristics to control the perspective / placement of a camera based on contextual cues of the scene. In this context, a film can be viewed as a hierarchy [18]; the top of the film hierarchy is the script, which consists of a sequence of time-ordered narrative descriptions, referred to as scenes. Each scene, in turn, provides contextual information, in the form of actions

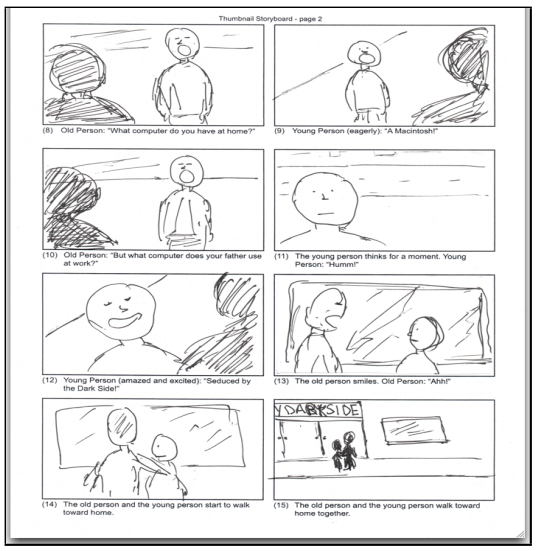


Figure 2: Artistic Impression of a Storyboard.

and events that can be used to derive a specific camera shot. The objective of each camera shot is to capture the sequence of events in a manner that is cinematically pleasing.

As an example, lets look at the simple, but common, scene in Fig. 2 depicting a group of two actors in a conversation. A storyboard, such as in Fig. 2[3], is typically an artist's impression on the basis of a film/drama script/screenplay. Within this scenario, the context of each scene is based on the current state of each actor with regards to their participation in the conversation, i.e. *talking, listening*, or *reacting*. Below is a sample script that involves two actors, Kendra and Annika, engaged in a conversation. In the example, contextual cues are provided as key words that indicate the current state of each actor, i.e. "Kendra starts to *talk*" and "Annika *reacts* sheepishly" and so forth:

ACT 1: Kendra and Annika
[Establishing-shot] -- Kendra and Annika
Kendra starts talking to Annika--[''*dialogue*'']
[Cut: mid-shot] -- Annika reacts anxiously to Kendra
Kendra continues talking to Annika
[Cut: Close-up] Annika responds to
Kendra--[''*astonish*'']
End.

As the scenes progress the states of each actor change as the conversation develops. From this information, it is the job of the (automatic) cinematographer to decide on an appropriate sequence of camera shots to properly depict the conversation. The result of this process is similar to the storyboard found in Figure 2., which shows the the perspective of the camera throughout the key moments of the scene. Because this scenario is so common in film, cinematic patterns have emerged that defined heuristics to capture this particular type of situation, referred to by cinematographers as a *film idiom* [2]. These idioms have been defined for many typical cinematic situations, such as groups of actors in a conversation, or an action sequence. In general, a film idiom can be seen as a set of declarative rules that specify a mapping between the use of camera shots to a situational context. We formally build-up on these aspects in the sections to follow.

[3] Source: http://upload.wikimedia.org/wikiversity/en/e/e4/Mok_Thumbnail_storyboards_tiny2.png

4 Explanation by Spatio-Temporal Abduction

Diametrically opposite to projection and planning is the task of postdictum or explanation [25, 27], where given a set of time-stamped observations or snap-shots, the objective is to explain which events and/or actions may have caused the observed state-of-affairs. Explanation, in general, is regarded as a converse operation to temporal projection essentially involving reasoning from effects to causes, i.e., reasoning about the past [31].

4.1 Narratives

Explanation problems demand the inclusion of a narrative description, which is essentially a distinguished course of actual events about which we may have incomplete information [21, 26]. Narrative descriptions are typically available as *observations* from the real / imagined execution of a system or process. Since narratives inherently pertain to actual observations, i.e., they are temporalized, the objective is often to assimilate / explain them with respect to an underlying process model and an approach to derive explanations.

In the automatic cinematography domain set-out in Section 3, narrative descriptions are (implicitly) available from linguistic descriptions about *acts* and *scenes* within a drama or film script.[4] Here, with the understanding that the progression of the script can be thought of as an imaginary evolution of the system, the symbolically grounded information from the script is equivalent to temporally ordered observations that constitute the available narrative.

$$\begin{bmatrix} \Phi_1 \equiv \neg HoldsAt(Talking(Kendra), t_1) \wedge \neg HoldsAt(Talking(Annika), t_1) \wedge \\ \quad \neg HoldsAt(Reacting(Annika), t_1) \wedge \neg HoldsAt(Reacting(Kendra), t_1) \\ \Phi_2 \equiv HoldsAt(Talking(Kendra), t_2) \wedge \neg HoldsAt(Talking(Annika), t_2) \wedge \\ \quad \neg HoldsAt(Reacting(Annika), t_2) \wedge \neg HoldsAt(Reacting(Kendra), t_2) \\ \Phi_3 \equiv HoldsAt(Talking(Kendra), t_3) \wedge \neg HoldsAt(Talking(Annika), t_3) \wedge \\ \quad HoldsAt(Reacting(Annika), t_3) \wedge \neg HoldsAt(Reacting(Kendra), t_3) \end{bmatrix} \tag{2a}$$

$$[t_1 < t_2 < t_3] \tag{2b}$$

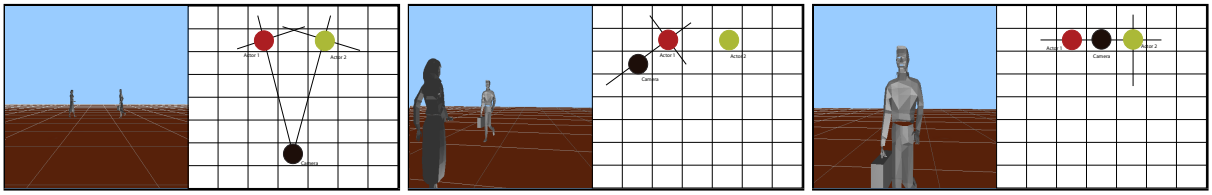
(Spatial) Narratives in the Film Domain

One of the key creative roles of a cinematographer / director or a story-boarding artist is to anticipate / visualize the scene on the basis of applicable film-idioms / heuristics (Section 3) that are suited to filming a particular scenario / narrative, such as the one exemplified in (4). For instance, in the context of the ongoing example, the applicable idioms are *EstablishingShot*, *ExternalShot* and *ReactionShot*.

$$\begin{bmatrix} \Phi_1 \rightarrow EstablishingShot(actor_1, actor_2) \\ \Phi_2 \rightarrow ExternalShot(actor_1, actor_2) \\ \Phi_3 \rightarrow ReactionShot(actor) \end{bmatrix} \tag{3}$$

Each of these film heuristics have a specific *structural form* that is identifiable with respect to relative orientation of the camera and the

[4] We ignore the translation from a linguistic to a symbolic/predicated form; this is beyond the scope of the objective of this paper.



(a) Establishing Shot (b) External Shot (c) Reaction Shot

Figure 3: Structural Form of Film Idioms (Automatic Cinematography: *2 Avatars and 1 Virtual Camera*)

actors involved in a scene or the applicable film idiom[5]:

$$\left[\begin{array}{l}(\forall\, t).\ HoldsAt(EstablishingShot(actor_1, actor_2),\ t) \rightarrow \\ \qquad HoldsAt(\phi_{scc}(camera_1, actor_1, actor_2, scc_3),\ t) \wedge \\ \qquad HoldsAt(\phi_{scc}(camera_1, actor_2, actor_1, scc_5),\ t)\end{array}\right] \quad (4a)$$

$$\left[\begin{array}{l}(\forall\, t).\ HoldsAt(ExternalShot(actor_1, actor_2),\ t) \rightarrow \\ \qquad HoldsAt(\phi_{scc}(camera_1, actor_1, actor_2, scc_1),\ t) \wedge \\ \qquad HoldsAt(\phi_{scc}(camera_1, actor_2, actor_1, scc_5),\ t)\end{array}\right] \quad (4b)$$

$$\left[\begin{array}{l}(\forall\, t).\ HoldsAt(ReactionShot(actor_1, actor_2),\ t) \rightarrow \\ \qquad HoldsAt(\phi_{scc}(camera_1, actor_1, actor_2, scc_4),\ t) \wedge \\ \qquad HoldsAt(\phi_{scc}(camera_1, actor_2, actor_1, scc_4),\ t)\end{array}\right] \quad (4c)$$

Consider the illustration in Fig. 3 for the present film domain: the world consists of three point-abstracted entities— 2 *avatars* and 1 virtual *camera*.[6] Further, suppose that container space is modeled a discrete grid world together with relative orientation relationships among the entities as per the partitioning scheme of the Single-Cross Calculus (see Section 2.1, Fig. 1). For this ongoing "*Kendra and Annika*" script, further suppose that the *camera* is the only entity that is able to move, i.e., change location from one grid-cell to another. For a scenario such as this, explanation by spatio-temporal abduction could serve as a basis of *scenario and narrative completion*, and for this particular example, the derivation of ideal *camera placements* as a side-effect of the abduction process. The general structure of such as derivation is explained next, and the ongoing example is further continued in Section 4.2.

Structure of (Abductive) Explanation

It is easy to intuitively infer the general structure of narrative completion (by abductive explanation). Consider the illustration in Fig. 4 for a branching / hypothetical situation space that characterizes the complete evolution of a system. In Fig. 4 – the situation-based history $< s_0,\ s_1, \ldots,\ s_n >$ represents one path, corresponding to an actual time-line $< t_0,\ t_1, \ldots,\ t_n >$, within the overall branching-tree structured situation space. Given incomplete narrative descriptions, e.g., corresponding to only some ordered time-points (such as in Fig. 3) in terms of high-level spatial (e.g., topological, orientation) and occurrence information, the objective of explanation is to derive one or more paths from the branching situation space, that could best-fit the available narrative information. Of course, the completions that bridge the narrative by interpolating the missing spatial and action/event information have to be consistent with both domain-specific and domain-independent rules/dynamics.

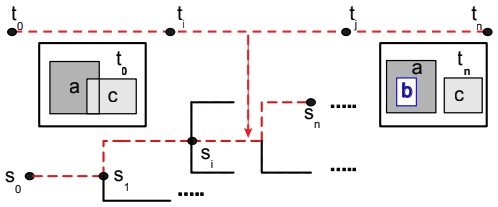


Figure 4: Branching / Hypothetical Situation Space

Many different formalizations of spatio-temporal explanation are possible, such as within a belief revision framework [1], nonmonotonic causal formalizations in the manner of [17], Situation Calculus [3, 4, 32] and so forth; this paper is a pragmatic illustration of the manner in which this may be achieved in the context of the discrete Event Calculus [20, 24].

4.2 Scenario and Narrative Completion

Figure 5 consists of a *narrative* (completion), starting with time-points t_1 and ending at time-point t_{12}: this denotes an *abduced* evolution of the system, as represented by the sequence of qualitative state descriptions for 2 stationery and 1 moving entity. For clarity, images from a 3D simulation are included together with the relational / graph-based illustrations for each of the time-points.

The narrative completion is abduced from an initial narrative description consisting of observations only of time-points $[t_1,\ t_4,\ t_6, t_8,\ \ t_{12}]$. Precisely, from the available observations, the narrative completion has been abduced on the basis of available *camera actions – pan, zoom, move* – and pre-specified knowledge or heuristics / film-idioms about desired camera placements, e.g., *establishing shot*, *external shot*, *mid-shot*, *close-up* and so forth. Needless to say, we have excluded one key representational aspect: the description so far has only focussed on the domain-specific representational aspects, and details about encoding the semantics of the spatial calculus, in this case the single-cross relations, have been excluded so far. The spatio-temporal abduction actually works on the basis of the embedding of all the high-level axiomatic aspects of the spatial calculus, together with the domain-theory of the "film world'. Without the embedding, spatial calculi— composition theorems, continuity constraints — have no semantic interpretation within the Event Calculus reasoner.

We further discuss the embedding of the spatial calculus within the Discrete Event Calculus in Section 4.3. Here, we conclude the

[5] This may be easily generalized to n actors/entities in the scene. Further, note that the formal interpretation of the spatial / structural form of an idiom is open-ended, and subject to the richness of the spatial calculi and other aspects (e.g., scene illumination) being modeled. For the purposes of this example, we restrict to an interpretation strictly in terms of orientation relationships.

[6] The third entity in the simulation is a virtual camera that records the other two entities in the scene, and hence is not visible within the 3D illustration of Fig. 3(c).

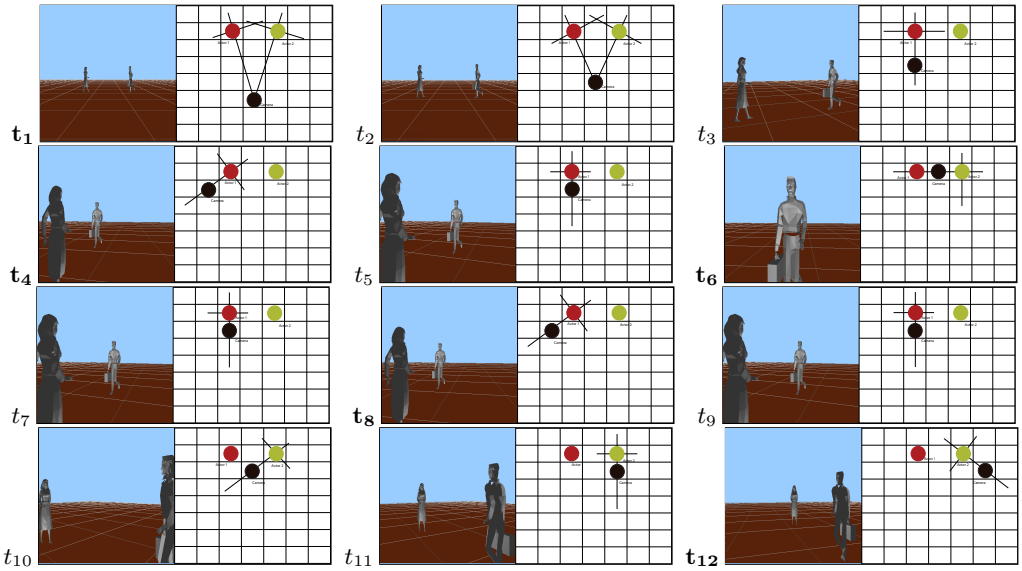


Figure 5: Scenario and Narrative Completion (by Abduction).

demonstration of the application with the remark that for this example, the resulting narrative completion is usable by a virtual reality and/or an automatic cinematography system to automatically generate animations and / or perspective visualizations for automatic storyboarding.

4.3 Embedding Spatial Calculi

In order to derive a spatio-temporal narrative completion directly on the basis of the semantics of the Discrete Event Calculus (DEC) reasoner, it is necessary to encode the high-level axiomatic aspects that determine the constitution of a qualitative spatial calculus. For the example under consideration in this paper, one only needs to encode certain key aspects of the Single/Double-Cross calculus (in addition to the domain-theory). Precisely, the following aspects need to be modelled explicitly: jointly exhaustive and pair-wise disjointness (JEPD) of the base relations, the composition theorems and the continuity constraints of the relationship space.

4.3.1 JEPD Property

Orientational spatial relations, such as those defined by the Single-Cross calculus, are jointly exhaustive, mutually disjoint and pairwise disjoint. The property of jointly exhaustive and mutually disjoint can be sufficiently expressed in the DEC using *n* state constraints of the form (5), where *n* is the number of total base relations defined by the calculus.

$$\left[\begin{array}{l}(\forall\, t).\, \neg[HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_0),\ t) \vee \\ \quad HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_1),\ t) \vee \\ \quad HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_2),\ t) \vee \\ \quad HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_3),\ t) \vee \\ \quad HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_4),\ t) \vee \\ \quad HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_5),\ t) \vee \\ \quad HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_6),\ t)] \equiv_{def} \\ \quad HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_7),\ t))\end{array}\right] \tag{5a}$$

$$\left[\begin{array}{l}(\forall\, t).\, \neg[HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_0),\ t) \wedge \\ \quad HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_1),\ t)])\end{array}\right] \tag{6a}$$

$$\left[\begin{array}{l}(\forall\, t).\, \neg[HoldsAt(\phi_{scc}(o_1,o_2,o_3,1),\ t) \wedge \\ \quad HoldsAt(\phi_{scc}(o_1,o_2,o_3,2),\ t)])\end{array}\right] \tag{6b}$$

Similarly, the property of pairwise disjointness can be expressed using [*n*(*n* - 1)/2] ordinary constraints of the form in (6).

4.3.2 Composition Theorems

The composition theorems of the Single-Cross calculus have to represented in the DEC using state constraints. For Single-Cross, this formulation requires 8 X 8 state constraints of the form in (7).

$$\left[\begin{array}{l}(\forall\, t).\, [HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_0),\ t) \wedge HoldsAt(\phi_{scc}(o_2,o_3,o_4,scc_0),\ t)] \\ \quad \rightarrow HoldsAt(\phi_{scc}(o_1,o_2,o_4,scc_4),\ t) \\ (\forall\, t).\, [HoldsAt(\phi_{scc}(o_1,o_2,o_3,scc_1),\ t) \wedge HoldsAt(\phi_{scc}(o_2,o_3,o_4,scc_1),\ t)] \\ \quad \rightarrow HoldsAt(\phi_{scc}(o_1,o_2,o_4,scc_5),\ t) \vee HoldsAt(\phi_{scc}(o_1,o_2,o_4,scc_6),\ t) \\ \quad \vee\ HoldsAt(\phi_{scc}(o_1,o_2,o_4,scc_7),\ t)\end{array}\right] \tag{7a}$$

Global compositional consistency of scenario descriptions is a key contributing factor determining the crucial commonsensical notion of the *physically realizability* [3, 4] (for model elimination) of narrative descriptions during the abduction process [7].

4.3.3 Conceptual Neighborhood

The conceptual neighborhood graph [13] for a set of spatial relations reflects their corresponding continuity structure, namely the the direct, continuous transformations that are possible among the relations as a result of motion by translation and / or deformation. Lets assume that the binary (reflexive) predicate $neighbour(\gamma_1,\ \gamma_2)$ denotes the presence of a continuity relation between qualitative relations γ_1 and γ_2.

$$\left[\begin{array}{l}ConceptualNeighbor(scc_0,scc_1) \\ ConceptualNeighbor(scc_1,scc_2) \\ ConceptualNeighbor(scc_6,scc_5)\end{array}\right] \tag{8}$$

Given this, one only needs to represent one predicate per continuity link of the form in (8) as reflected by a particular partitioning of the spatial calculus being utilized, which in this case, is the Single-Cross calculus.

[7] Other factors, not relevant for the example of this paper, are *physical* and *existential* consistency [3, 4].

5 Discussion and Outlook

The logic-based integration of specialized *spatial representation formalisms* on the one hand and general logics for *reasoning about actions and change* is an initiative that defines the broad agenda of the work described in this paper. The preliminary results presented herein are guided by the proposition of integrated *Reasoning about Space, Actions and Change* (RSAC) [5], which we regard to be a useful paradigm for applications of formal methods in Qualitative Spatial Representation and Reasoning (QSR) within realistic *Dynamic Spatial Systems*.

The specific aim of this paper has been to demonstrate the manner in which *spatio-temporal abduction* may be performed with an off-the-shelf logic of action and change, namely the Event Calculus, that is available a reasoning tool. We demonstrate this in the context of an example "*automatic cinematography*" domain, which we regard to be intuitively appealing in order to communicate the often contrived idea of *logical explanation by abduction*. We emphasize that automatic cinematography is not merely a toy domain: in the entertainment industry, a wide-range of applications require the to ability to visualize complex— single agent and multi-perspective —scenes in a dynamic and real-time context. Major applications include automatic (virtual) cinematography or in general automatic story visualization and story-boarding (as addressed here), real-time perspective modeling for games, the modeling of interaction in hybrid or real–virtual environments, e.g., for e-learning, and so forth. As another emerging application, consider the domain of *spatial computing for design* [7]. Here, abductive reasoning in general plays a significant role in reasoning about hypothetical spatial structures.

From a theoretical perspective, we are presently pursuing the integration of specialized (infinite domain) spatial reasoning tools such as *SparQ* [34]and *GQR* [35] within the Discrete Event Calculus reasoner; the approach being adopted here is to ensure a separation of concerns during the satisfiability checking process (based on the *relsat solver*[8]) underlying the DEC reasoner. Experiments are also in progress to achieve a similar sort of integration of spatial reasoners with not only the *relsat solver*, but also other extensions into the answer-set programming (ASP) framework[9] [19]. The challenge here lies in maintaining the separation during the constraint solving stage between the native *relsat / ASP* solvers that underly the Event Calculus reasoner and specialized spatial reasoning tools, i.e., the objective here is not to replace SAT or ASP solvers, but rather to complement their constraint solving capabilities with the aforestated specialized spatial reasoning tools. From an application perspective, we are pursuing the application of the proposed spatio-temporal abduction mechanism in other application areas of interest, e.g., design creativity, dynamic GIS, smart environments.

Acknowledgments

We gratefully acknowledge the funding and support by the Alexander von Humboldt Stiftung (Germany) and the German Research Foundation (DFG). Gregory Flanagan was supported by a Research Scholarship (May 2009) by the SFB/TR 8 Spatial Cognition Center.

REFERENCES

[1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson, 'On the logic of theory change: Partial meet contraction and revision functions', *J. Symb. Log.*, **50**(2), 510–530, (1985).
[2] Daniel Arijon, *Grammar of the Film Language*, Hastings House, New York, 1976.
[3] Mehul Bhatt, 'Commonsense inference in dynamic spatial systems: Phenomenal and reasoning requirements', in *23rd International Workshop on Qualitative Reasoning (QR 09), Ljubljana, Slovenia, June 2009*, eds., Ivan Bratko and Jure Žabkar, pp. 1–6, (2009). (Part 1 of 2).
[4] Mehul Bhatt, 'Commonsense inference in dynamic spatial systems: Epistemological requirements', in *FLAIRS Conference: Special Track on Spatial and Temporal Reasoning*. AAAI Press, (2010). (Part 2 of 2): to appear.
[5] Mehul Bhatt, 'Reasoning about space, actions and change: A paradigm for applications of spatial reasoning', in *Qualitative Spatial Representation and Reasoning: Trends and Future Directions*. IGI Global, USA, (2010).
[6] Mehul Bhatt and Frank Dylla, 'A qualitative model of dynamic scene analysis and interpretation in ambient intelligence systems', *International Journal of Robotics and Automation*, **24**, (2009). (to appear, url as of 14.11.2009).
[7] Mehul Bhatt and Christian Freksa, 'Spatial computing for design: An artificial intelligence perspective', in *NSF International Workshop on Studying Visual and Spatial Reasoning for Design Creativity (SDC'10)*, (2010). (to appear).
[8] Mehul Bhatt and Hans Guesgen, eds. *Spatial and Temporal Reasoning for Ambient Intelligence Systems (STAMI 09)*. SFB/TR 8 Spatial Cognition Report Series, No. 020-08/2009, September 2009.
[9] David B. Christianson, Sean E. Anderson, Li wei He, David H. Salesin, Daniel S. Weld, and Michael F. Cohen, 'Declarative camera control for automatic cinematography', (1996).
[10] A G Cohn and J Renz, 'Qualitative spatial reasoning', in *Handbook of Knowledge Representation*, eds., Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, Elsevier, (2007).
[11] Helen Couclelis, 'The abduction of geographic information science: Transporting spatial reasoning to the realm of purpose and design', in *COSIT*, eds., Kathleen Stewart Hornsby, Christophe Claramunt, Michel Denis, and Gérard Ligozat, volume 5756 of *Lecture Notes in Computer Science*, pp. 342–356. Springer, (2009).
[12] Steven M. Drucker and David Zeltzer, 'Camdroid: a system for implementing intelligent camera control', in *I3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pp. 139–144, New York, NY, USA, (1995). ACM.
[13] Christian Freksa, 'Conceptual neighborhood and its role in temporal and spatial reasoning', in *Decision Support Systems and Qualitative Reasoning*, eds., M. Singh and L. Travé-Massuyès, pp. 181–187. North-Holland, Amsterdam, (1991).
[14] Christian Freksa, 'Using orientation information for qualitative spatial reasoning', in *Proceedings of the Intl. Conf. GIS, From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, pp. 162–178, Berlin, (1992). Springer-Verlag.
[15] Antony Galton, 'Causal reasoning for alert generation in smart homes.', in *Designing Smart Homes*, pp. 57–70, (2006).
[16] Antony Galton and James Hood, 'Qualitative interpolation for environmental knowledge representation', in *ECAI*, eds., Ramon López de Mántaras and Lorenza Saitta, pp. 1017–1018. IOS Press, (2004).
[17] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner, 'Nonmonotonic causal theories', *Artif. Intell.*, **153**(1-2), 49–104, (2004).
[18] Li-wei He, Michael F. Cohen, and David H. Salesin, 'The virtual cinematographer: a paradigm for automatic real-time camera control and directing', in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 217–224, New York, NY, USA, (1996). ACM.
[19] Tae-Won Kim, Joohyung Lee, and Ravi Palla, 'Circumscriptive event calculus as answer set programming', in *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*, pp. 823–829, San Francisco, CA, USA, (2009). Morgan Kaufmann Publishers Inc.
[20] Robert Kowalski and Marek Sergot, 'A logic-based calculus of events', *New Gen. Comput.*, **4**(1), 67–95, (1986).
[21] Rob Miller and Murray Shanahan, 'Narratives in the situation calculus', *J. Log. Comput.*, **4**(5), 513–530, (1994).
[22] Reinhard Moratz, 'Representing relative direction as a binary relation of oriented points', in *ECAI*, pp. 407–411, (2006).
[23] Erik T. Mueller, *Commonsense Reasoning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.
[24] Erik T. Mueller, 'Automating commonsense reasoning using the event calculus', *Commun. ACM*, **52**(1), 113–117, (2009).
[25] C. S. Pierce, *The Collected Papers of Charles Sanders Peirce*, Harvard University Press, 1935.
[26] Javier Pinto, 'Occurrences and narratives as constraints in the branching structure of the situation calculus.', *J. Log. Comput.*, **8**(6), 777–808, (1998).
[27] D. Poole, R. Goebel, and R. Aleliunas, 'Theorist: A logical reasoning system for defaults and diagnosis', in *The Knowledge Frontier*, eds., N. Cercone and G. McCalla, 331–352, Springer, New York, (1987).
[28] David A. Randell, Zhan Cui, and Anthony Cohn, 'A spatial logic based on regions and connection', in *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, 165–176, Morgan Kaufmann, San Mateo, California, (1992).
[29] Jochen Renz and Bernhard Nebel, 'Qualitative spatial reasoning using constraint calculi', in *Handbook of Spatial Logics*, 161–215, (2007).
[30] C. Schlieder, 'Reasoning about ordering', in *Proc. of COSIT'95*, volume 988 of *Lecture Notes in Computer Science*, 341–349, Springer, Berlin, Heidelberg, (1995).
[31] Murray Shanahan, 'Prediction is deduction but explanation is abduction.', in *IJCAI*, pp. 1055–1060, (1989).
[32] Murray Shanahan, 'Explanation in the situation calculus.', in *IJCAI*, pp. 160–165, (1993).
[33] Murray Shanahan. The event calculus explained, 1999.
[34] Jan Oliver Wallgrün, Lutz Frommberger, Diedrich Wolter, Frank Dylla, and Christian Freksa, 'Qualitative spatial representation and reasoning in the sparq-toolbox', in *Spatial Cognition V: Reasoning, Action, Interaction: International Conference Spatial Cognition 2006*, eds., Thomas Barkowsky, Markus Knauff, Gerard Ligozat, and Dan Montello, volume 4387 of *LNCS*, 39–58, Springer-Verlag Berlin Heidelberg, (2007).
[35] Matthias Westphal, Stefan Woelfl, and Zeno Gantner, 'GQR: A fast solver for binary qualitative constraint networks', in *AAAI Spring Symposium on Benchmarking of Qualitative Spatial and Temporal Reasoning Systems*, (2009).
[36] Michael F. Worboys, 'Event-oriented approaches to geographic phenomena.', *Int. Journal of Geographical Inf. Science*, **19**(1), 1–28, (2005).

[8] *relsat*: http://www.bayardo.org/resources.html
[9] *ecasp, f2lp*: http://reasoning.eas.asu.edu/f2lp/

Improving Solutions of Problems of Motion on Graphs by Redundancy Elimination

Pavel Surynek[1] and **Petr Koupý**[2]

Abstract. Problems of motion on graphs are addressed in this paper. These problems represent an abstraction for a variety of tasks which goal is to construct a spatial-temporal plan for a set of entities that move in a certain environment and need to reach given goal positions. Specifically, the quality (length) of solutions of these problems is studied. Existing state-of-the-art algorithms for generating solutions are suspected of producing solutions containing redundancies of a priori unknown nature. A visualization tool has been developed to discover such redundancies. Knowledge about solutions acquired by the tool served as a basis for the formal description of redundancies and for the development of methods for their detection and elimination. A performed experimental evaluation showed that the elimination of described redundancies improved existing solutions significantly.

Keywords: path planning, multiple robots, tractable class, graphs

1. INTRODUCTION AND CONTEXT

Problems of motion on a graph as they are introduced in [5, 8, 12] represent a basic abstraction for many real-life and theoretical tasks. The classical task that can be abstracted as a problem of motion on a graph takes place in a certain physical environment where mobile *entities* are moving (for example *mobile robots*). Each entity is given its initial and goal position in the environment. The task is to build a spatial-temporal plan for all the entities such that they reach goal positions following this plan while the plan satisfies certain natural constraints. These constraints are typically constituted by a requirement that entities must avoid obstacles in the environment and must not collide with each other.

The standard abstraction that is adopted throughout this work uses an undirected graph to model the environment. The vertices of this graph represent positions in the environment and the edges represent an unblocked way between two positions. An arrangement of entities in the environment is abstracted as a simple assignment of entities to vertices. At least **one** vertex remains **unoccupied** in order to make the movement of entities possible. The time is discrete; it is an ordered set of *time steps* isomorphic to the structure of *natural numbers*. A way how an arrangement of entities can be transformed into another can slightly differ in variants of the problem.

1.1 Motivation by Practice

The abstract problems of motion on a graph are motivated by many real-life problems. The most typical motivating example is a motion planning of a group of mobile robots that are moving in *2-dimensional space* [8]. Generally, if there is enough free space in the environment, algorithms based on search for shortest paths in a graph can be used [12]. However, if there is little free space, different methods must be used [5, 9, 10].

Many well known puzzles can be formulated as the problem of motion on a graph. The best known is so called Lloyd's 15-puzzle and its generalizations [7, 12]. In practice, various mobile or movable objects may represent the entities – for example, a rearrangement of containers in a storage area can be interpreted as a problem of motion on a graph where entities are represented by containers. Indeed, this approach has been used for planning motions of automated straddle carriers in a storage area in Patrick port facility at Port Brisbane in Queensland [8]. Although the approach suggested in [8] does not scale for larger number of entities, it clearly demonstrates the usefulness of discussed abstractions. Entities do not necessarily have to be physical objects. Virtual spaces of computer simulations and games convey many situations where motions of certain entities must be planned.

It is necessary to stress that contrary to multi-agent motion planning [4], the **centralized approach** is adopted in this work. That is, the environment is fully observable for the central planning mechanism and the individual entities merely execute the submitted centrally created plan.

1.2 Specific Open Questions

There exist several relatively efficient methods for solving problems of motion on a graph. This work is particularly targeted on solution generation methods described in [9, 10]. These methods represent state-of-the-art algorithms for the class of problems where the graph modeling the environment is *bi-connected* and where there are **many entities** placed in the graph (the graph is relatively full with **small unoccupied space**). Despite the qualities of these methods, the generated solution are suspected of containing certain *redundancies*. This is a **conjecture** whose examination is the main contribution of this paper. If it is the case that generated solutions contain redundancies, then a question how they can be removed to improve the solution arises.

The task is thus to analyze solutions of non-trivial size which is manually infeasible. Moreover, it is necessary to emphasize that searched redundancies are of a priori unknown nature. Therefore a comfortable software tool ***GraphRec*** [6] has been developed to allow visual analysis of solutions of problems of motion on a graph. The GraphRec software solves two issues that are difficult to be handled manually. First, the tool draws the graph modeling the environment of the problem on the screen. An embedding of the graph into two dimensions with few edge crossings is preferred to enable **comfortable observation**. Second, motions of entities on the graph are visualized by the tool in time.

Several types of redundancies were discovered by the GraphRec software in solutions. They are formally described in this paper. Further, methods for automated discovery and elimination of these redundancies are suggested and analyzed theoretically as well as experimentally.

The top level **organization** of the paper has two parts. The first part explains a variant of the problem of motion on a graph

[1,2] Charles University in Prague, Faculty of Mathematics and Physics, Malostranské náměstí 25, 118 00 Praha 1, Czech Republic, pavel.surynek@mff.cuni.cz, petr.koupy@gmail.com.

This work is supported by The Czech Science Foundation under the contract number 201/09/P318 and by The Ministry of Education, Youth and Sports, Czech Republic under the contract number MSM 0021620838.

(section 2) and the basic solving algorithm (section 3); this part merely recalls existing concepts. The second part contains the main contribution of this work; the GraphRec visualization tool is introduced (section 4), redundancy elimination methods are described (section 5), and the benefit of suggested methods is justified in the experimental section (section 6).

2. PEBBLE MOTION ON A GRAPH

The basic variant of the motion problem is known as *pebble motion on a graph* [5, 12]. The role of an entity is represented by a pebble here. The task is given by an undirected graph with an *initial* and a *goal arrangement* of pebbles in the vertices of this graph. Each vertex of the graph contains at most one pebble and at least one vertex remains unoccupied. The task is to find a sequence of moves for each pebble such that all the pebbles reach their goal vertices. A pebble can move into a **neighboring unoccupied** vertex while no other pebble is entering the target vertex at the same time. The following definition formalizes the problem. An illustrative instance of the problem is shown in figure 1.

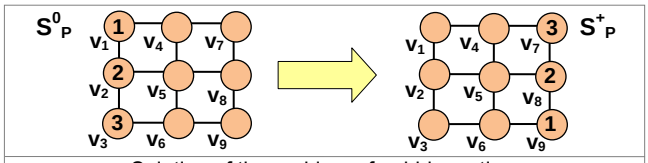
Definition 1 ***(pebble motion on a graph).*** Let $G = (V, E)$ be an undirected graph and let $P = \{p_1, p, \dots, p_\mu\}$ be a set of pebbles where $\mu < |V|$. The ***initial arrangement*** of pebbles is defined by a simple function $S_P^0 : P \longrightarrow V$ (that is $S_P^0(p_i) \neq S_P^0(p_j)$ for $i, j = 1,2, \dots, \mu$ with $i \neq j$); the ***goal arrangement*** of pebbles is defined by another simple function $S_P^+ : P \longrightarrow V$. A problem of ***pebble motion on a graph*** is the task to find a number ξ and a sequence $S_P = [S_P^0, S_P^1, \dots, S_P^\xi]$ where $S_P^k : P \longrightarrow V$ is a simple function for every $k = 1,2, \dots, \xi$. The following constraints must hold:

(i) $S_P^\zeta = S_P^+$, that is, pebbles finally reach their destinations.

(ii) Either $S_P^k(p) = S_P^{k+1}(p)$ or $\{S_P^k(p), S_P^{k+1}(p)\} \in E$ for every $p \in P$ and $k = 1,2, \dots, \xi - 1$.

(iii) If $S_P^k(p) \neq S_P^{k+1}(p)$ then $S_P^k(q) \neq S_P^{k+1}(p)$ for $\forall q \in P$ such that $q \neq p$ must hold for every $p \in P$ and $k = 1,2, \dots, \xi - 1$, that is, a pebble can move to a currently unoccupied vertex.

The problem described above is formally a quadruple $\Pi = (G = (V, E), P, S_P^0, S_P^+)$. □



Solution of the problem of pebble motion on a graph with P={1,2,3}

ξ=6	P	S^0_P	S^1_P	S^2_P	S^3_P	S^4_P	S^5_P	$S^6_P=S^+_P$
	1	v_1	v_4	v_7	v_8	v_9	v_9	v_9
	2	v_2	v_2	v_1	v_4	v_7	v_8	v_8
	3	v_3	v_3	v_3	v_2	v_1	v_4	v_7

Figure 1. *An illustration of a problem of* ***pebble motion on a graph***. The task is to move pebbles from their initial positions specified by S_P^0 to the goal positions specified by S_P^+. A solution of length 6 is shown.

In practice, the *quality of solution* matters. The typical measures of the quality of solution are its **length** (the total number of moves) and **duration** (which corresponds to the number ξ). These numbers are required to be small. Unfortunately, requiring either the length of the solution or its duration to be as small as possible makes the problem **intractable** [7] (the decision variant of the problem is *NP*-complete). This fact is the main reason why existing methods for generating optimal solutions do not scale for larger number of entities [8] (the problem is called *multi-robot path planning* in these works). On the other hand, if there is **no requirement** on the quality, the question whether there exists a solution is in the ***P* class** [5]. However, methods giving evidence that the problem belongs to the *P* class described in [5] generate solutions that are too long and unsuitable for practice. Therefore it is necessary to find a **compromise** between the quality of solution and computational cost of its construction. Methods following this compromise are described in [9, 10]. Solutions produced by these methods will be submitted to analysis by the visualization tool in order to find out how they can be further improved.

3. SOLVING MOTION PROBLEMS

This section is devoted to a brief recall of algorithms described in [9, 10]. An insight into the structure of solutions produced by these algorithms is crucial to understand their quality.

The most important class of pebble motion problems is formed by those whose graph is *bi-connected* which intuitively means that each pair of vertices is connected by two disjoint paths. The following definition specifies bi-connectivity formally.

Definition 2 ***(connectivity, bi-connectivity).*** An undirected graph $G = (V, E)$ is ***connected*** if $|V| \geq 2$ and for every pair of distinct vertices $u, v \in V$ there exists a path connecting u and v in G. An undirected graph $G = (V, E)$ is ***bi-connected*** if $|V| \geq 3$ and for every vertex $u \in V$ the graph $G' = (V - \{u\}, E \cap \{\{v, w\} | v, w \in V \wedge v \neq u \wedge w \neq u\}$ is connected. □

The importance of this class of problems is assessed by the fact that they are **almost always solvable**. Moreover, spatial environments in real tasks are often abstracted as two dimensional **grids** which are bi-connected in most cases.

If the bi-connected graph contains at least **two unoccupied** vertices and it is not isomorphic to a cycle, then every goal arrangement of pebbles is reachable from every initial arrangement [9]. If the graph contains just **one unoccupied** vertex which can be without loss of generality fixed, then any arrangement of pebbles can be regarded as a *permutation* with respect to the initial arrangement. A permutation is *even* if it can be composed of the even number of transpositions; otherwise it is *odd*. If the goal arrangement represents an **even** permutation, then the problem is **always solvable**. In case of an odd permutation, the problem is solvable if and only if the graph contains a cycle of odd length [12]. A treatment of instances containing more than two unoccupied vertices will be discussed further.

For the sake of completeness, it is adequate to mention the case of pebble motion problems on **general graphs**. This case can be solved using methods for bi-connected case. Every undirected graph can be decomposed into a tree of bi-connected components [11]. Having such a decomposition, the pebbles need to be moved into their target bi-connected components first (this may not always be possible). Then the method for the bi-connected case is applied within individual bi-connected components.

An inductive construction of bi-connected graphs by adding *loops* is a pivotal concept in developing solving algorithms. Let $G = (V, E)$ be a graph, a *loop* with respect to G is a sequence of vertices $L = [u, x_1, x_2, \dots, x_l, v]$, where $u, v \in V$ and $x_i \notin V$ for $i = 1,2, \dots, l$ (it is allowed that $l = 0$). The result of *addition* of the loop L to the graph G is a new graph $G' = (V', E')$, where $V' = V \cup \{x_1, x_2, \dots, x_l\}$ and either $E' = E \cup \{\{u, v\}\}$ if $l = 0$ or $E' = E \cup \{\{u, x_1\}, \{x_1, x_2\}, \dots, \{x_{l-1}, x_l\}, \{x_l, v\}\}$ if $l \geq 1$. Every bi-connected graph $G = (V, E)$ can be constructed from a cycle by a sequence of loop additions. Such *loop decomposition* can be effectively determined in time $O(|V| + |E|)$ [11].

3.1 The *BIBOX-θ* Solving Algorithm

The *BIBOX-θ* algorithm [10] solves a case of the problem of pebble motion on a graph when the graph is bi-connected and there is single unoccupied vertex. The BIBOX-θ algorithm represents state-of-the-art for the described class of problems in terms of speed and quality of generated solutions. This is the main reason why solutions produced by this algorithm are studied here.

In the first phase of the algorithm, a loop decomposition is found; that is, a cycle - called *initial cycle* - and a sequence of loops is determined. Without loss of generality it is required that the unoccupied vertex within the goal arrangement of pebbles is in the initial cycle. The algorithm then proceeds inductively according to the loop decomposition from the last loop to the initial cycle with the first loop.

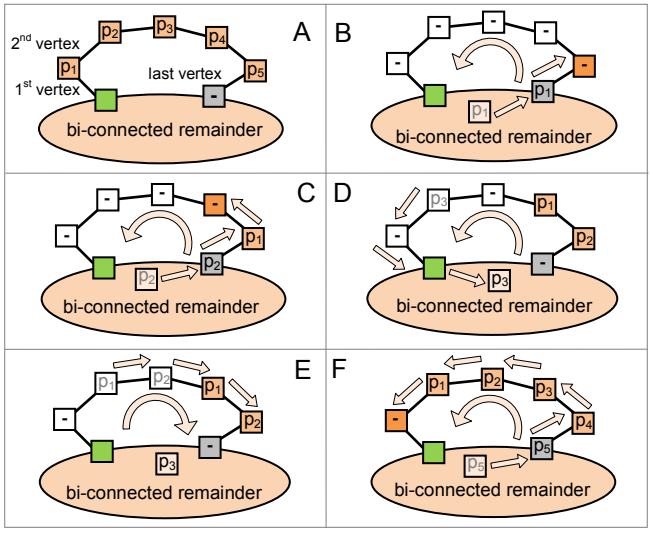


Figure 2. *The **process of placing pebbles** into a loop in the stack manner*. The goal arrangement of pebbles is shown in part **A**. Parts **B** and **C** show a process of ordering new pebbles into the loop in case when they are outside the loop. Part **D** and **E** show ordering process for a pebble when it is already inside the loop. Part **F** shows the final step in which pebbles reach their target vertices. The **green** vertex is unoccupied.

Two properties of bi-connected graphs with at least one unoccupied vertex are exploited while pebbles are placed within loops: **(i)** every vertex can be made unoccupied (this is even true for a connected graph), **(ii)** every pebble can be moved to an arbitrary vertex [9]. A loop is processed in the following way. An orientation of the loop is chosen first – this orientation determines ordering of vertices within the loop. The first and the last vertex of the loop are the connection points to the remainder graph. Then pebbles starting with the pebble whose goal position is in the second vertex of the loop are placed into the loop in the **stack manner**.

The current pebble is moved to the last vertex of the loop. Two cases must be distinguished here. If the pebble is already somewhere in the loop it must be moved outside first. If the current pebble is outside the loop, then it can be moved into the last vertex of the loop using property (ii) (only pebbles within the subgraph without the loop are moved). After placing the pebble into the last vertex of the loop, the loop is rotated once in the direction to the first vertex. The process is illustrated in figure 2.

When all the pebbles within the loop are processed the task is to solve the problem of the same type on a smaller graph – the finished loop is not considered anymore; a bi-connected graph without the last loop is bi-connected again. Nevertheless, the stack manner of placing pebbles cannot be applied for the initial cycle and the first loop of the decomposition. Therefore the algorithm uses a database containing **pre-calculated optimal solutions** for transpositions and rotation of pebbles along 3-cycles in graphs consisting of a cycle and a loop. A solution to any solvable instance on the initial cycle with the first loop is then composed of solutions from the database [10].

If it is the task to solve an instance of the problem with a bi-connected graph where there are **more than one** unoccupied vertices, then all the vertices except one are filled with **dummy pebbles**. The modified problem is then solved by the BIBOX-θ algorithm. Motions of dummy pebbles are finally filtered out of the resulting solution [9]. Such a process of producing solutions of problems with **many unoccupied vertice**s is suspected of generating *redundant moves* that may prolong the solution unnecessarily. However, this statement should be understood as a **conjecture** that has to be verified first.

4. VISUALIZATION TOOL

The examination and reviewing of the solution quality appeared to be difficult without certain automation. Therefore, a visualization tool *GraphRec* [6] has been developed (http://www.koupy.net/graphrec.php). The tool provides an animation engine for the entity movement together with features designed to support the observation of the solution time line. Any similar tool has not been available up until now. With the existing graph visualization software (e.g. *Graphviz* [1]) it is neither possible to represent entities nor move them among graph nodes.

4.1 Functional Requirements

Before the visualization can even occur, the graph on which the movement will be animated have to be embedded on the screen. Since we are dealing with bi-connected graphs, which are not necessarily *planar*, the embedding algorithm should **reduce** the amount of **crossing edges** while maintaining *Euclidean* distances between nodes proportional to the corresponding *shortest paths*.

The **animation of moving entities** is the core feature of the application. Since the solution is built over discrete time steps, these should be possible to play through or even step through in order to increase controllability of the observation. When examining certain part of the solution it is also necessary to provide adjustable speed of the animation and the possibility to jump quickly between various time steps. The clearness of the animation must be taken into attention as well. It appears that *highlighting* of moving entities greatly improves the overall perception of where the motion actually occurs. The demand for user vigilance might be further reduced by *distinguishing* between entities that are already in their final positions and that are not.

4.2 Tool Overview

GraphRec implements two *force-directed* planar embedding algorithms described in [2, 3]. Both methods are based on the simulation of a certain physical model. Whereas the model introduced in [2] considers nodes as repulsive particles and edges as contracting springs, another interpretation where chosen free node is connected by springs to the rest of anchored nodes is proposed in [3]. Owing to their physical background, force-directed algorithms often produce **expected and intuitive layouts** (figure 3).

The tool enables all graph elements to be assigned with various colors. This is especially important in scenarios such as observa-

tion of the movement of one particular entity or even group of entities, where **color differentiation** greatly improves their traceability. Colors are also utilized to distinguish entities in goal positions and to highlight moving entities as shown in figure 4.

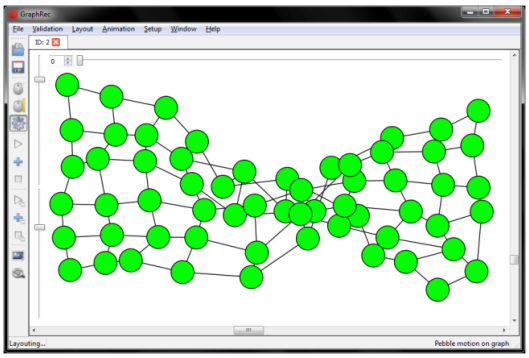


Figure 3. ***Graph layout*** gradually evolving into the regular grid.

Animation of the solution can be controlled in a similar way as playing a movie on a video recorder. Firstly, user adjusts the animation speed and specifies the starting time step. Then, it is possible to play or step through the animation time line. GraphRec supports the synchronized animation of more than one solution at once, which is for example useful when comparing differently optimized solutions for the same problem.

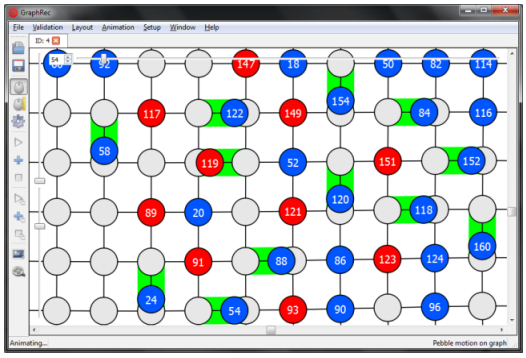


Figure 4. ***Moving entities*** emphasized by highlighted edges.

4.3 Discovering Redundancies

The presented visualization proved itself an effective way for discovering the nature of expected redundancies in solutions. Since the automatic detection of redundancies with unknown characteristics is not possible, the analysis by a human is essential. Because humans are mainly visual-oriented, the visualization of the problem seems to be suitable approach. Acquired knowledge was later used to formalize redundancies and to design methods for their removal.

4.4 Additional Features

GraphRec can find inconsistencies in solution by verifying its movements against constraints specified in the definition of the variant of motion problem. **Solution validation** is necessary to prevent the corruption of the animation. However, the validation can also be utilized for debugging of algorithms.

Moreover, GraphRec might be used as a presentation tool either in real time or to **produce media files**. The animation can be captured into raster and vector images or even into popular video formats. These files can be used within web presentations.

5. ELIMINATION OF REDUNDANCIES

Several types of redundancies were **discovered** using the GraphRec software within the generated solutions. A formal description of these redundancies and algorithms for their elimination are provided in the following sections. The process of transformation of a perception gained by the observation of the visualized solution to a formal description of a redundancy is a creative process. It is currently an open question whether some automation of this process is possible.

When reasoning about redundancies, it is convenient to assume solutions with just one move between consecutive time steps. The BIBOX-θ algorithm produces solutions in this form. A solution of this form can be viewed as a sequence of moves where the position of a move in the sequence corresponds to its time step of commencement. The notation $k_i: u_i \to v_i$ will denote a move of a pebble k_i from a vertex u_i to a vertex v_i commenced at time step i. The move is called *non-trivial* if $u_i \neq v_i$. From the formal point of view, the solution is a sequence of non-trivial moves $\Phi = [k_i: u_i \to v_i | i = 1,2,\dots,\xi - 1]$ (consistency with definition 1 is also assumed).

5.1 Inverse Moves

Definition 3 ***(inverse moves).*** A pair of consecutive moves $k_i: u_i \to v_i$ and $k_{i+1}: u_{i+1} \to v_{i+1}$ with $i \in \{1,2,\dots,\xi - 2\}$ are called *inverse* if $k_i = k_{i+1}$, $u_i = v_{i+1}$, and $v_i = u_{i+1}$. □

Observe that a pair of inverse moves can be left out of the solution without affecting its *validity* - it still solves the problem. However, elimination of an inverse pair may cause that another pair of inverse moves arises. Hence, it is necessary to remove inverse moves from the solution repeatedly until there are none.

The process of elimination of inverse moves is expressed below as algorithm 1. The worst case time complexity of the algorithm is $O(|\Phi|^2)$, space complexity is $O(|\Phi|)$.

Algorithm 1. Elimination of inverse moves.

```
function EraseInverseMoves (Φ): sequence
1:  do
2:      η ← ∅
3:      let [k_1: u_1 → v_1, k_2: u_2 → v_2, ..., k_{ξ-1}: u_{ξ-1} → v_{ξ-1}] = Φ
4:      for i = 1,2,...,ξ − 1 do
5:          if k_i: u_i → v_i and k_{i+1}: u_{i+1} → v_{i+1} are inverse then
6:              η ← η ∪ {k_i: u_i → v_i, k_{i+1}: u_{i+1} → v_{i+1}}
7:      Φ ← Φ − η
8:  while η ≠ ∅
9:  return Φ
```

5.2 Redundant Moves

Definition 4 ***(redundant moves).*** A sequence of moves $[k_{i_j}: u_{i_j} \to v_{i_j} | j = 1,2,\dots,l]$, where $I = [i_j \in \{1,2,\dots,\xi - 2\} | j = 1,2,\dots,l]$ is an increasing sequence of indices, is called *redundant* if $|\{k_{i_j} | j = 1,2,\dots,l\}| = 1$, $u_{i1} = v_{il}$, and for each move $k_\iota: u_\iota \to v_\iota$ with $i_1 < \iota < i_l \wedge \iota \notin I$ it holds that $k_\iota \neq k_{i_1} \Rightarrow u_{i_1} \notin \{u_\iota, v_\iota\}$. □

Redundant moves represent **generalization** of inverse moves (a pair of inverse moves form a redundant sequence). It is a sequence of moves which relocates a pebble into some vertex for the second time while other pebbles do not enter this vertex at any time step between the beginning and the end of the sequence. Eliminating a redundant sequence of moves preserves validity of the solution. Again, it is necessary to remove redundant sequences repeatedly since its removal may cause that another redundant sequence arises.

Algorithm 2 formalizes the process of removing redundant moves in the pseudo-code. The worst case time complexity is $O(|\Phi|^4)$, the space complexity is $O(|\Phi|)$.

Algorithm 2. Elimination of redundant moves.

function *EraseRedundantMoves* (Φ): **sequence**
do
  $\eta \leftarrow$*FindRedundantMoves*(Φ)
  $\Phi \leftarrow \Phi - \eta$
while $\eta \neq \emptyset$
return Φ

function *FindRedundantMoves* (Φ): **sequence**
let $[k_1: u_1 \rightarrow v_1, \ldots, k_{\xi-1}: u_{\xi-1} \rightarrow v_{\xi-1}] = \Phi$
for $i = 1,2,\ldots,\xi-2$ **do** {beginning of redundant sequence}
  for $j = \xi-1, \xi-2, \ldots, i+1$ **do** {end of redundant sequence}
    if $k_i = k_j \wedge u_i = v_j$ **then**
      $\eta \leftarrow \emptyset$ {redundant sequence}
      for $\tau = i, i+1, \ldots, j$ **do**
        if $k_i = k_\tau$ **then** $\eta \leftarrow \eta \cup \{k_\tau: u_\tau \rightarrow v_\tau\}$
      if *CheckRedundantMoves*(Φ, i, j) **then return** η
return $\emptyset$

function *CheckRedundantMoves* (Φ, i, j): **boolean**
let $[k_1: u_1 \rightarrow v_1, \ldots, k_{\xi-1}: u_{\xi-1} \rightarrow v_{\xi-1}] = \Phi$
for $\iota = i+1, i+2, \ldots, j-1$ **do**
  if $k_\iota \neq k_i \wedge u_i \in \{u_\iota, v_\iota\}$ **then return** $False$
return $True$

5.3 Long Sequences

Definition 5 ***(long sequence).*** Let $O^t(P)$ be a set of vertices occupied by a set of pebbles P at a time step t. A sequence of moves $[k_{i_j}: u_{i_j} \rightarrow v_{i_j} | j = 1,2,\ldots,l]$, where $I = [i_j \in \{1,2,\ldots,\xi-2 | j = 1,2,\ldots,l\}$ is an increasing sequence of indices, is called *long* if $|\{k_{i_j} | j = 1,2,\ldots,l\}| = 1$ and there exists a path $C = [c_1 = ui1, c2, \ldots, cn = vil]$ in G such that $n<l$, $C \cap Oi1(P-\{ki1\}) = \emptyset$, and for all the moves $k_\iota: u_\iota \rightarrow v_\iota$ with $i_1 < \iota < i_l \wedge \iota \notin I$ it holds that $k_\iota \neq k_{i_1} \Rightarrow \{u_\iota, v_\iota\} \cap C = \emptyset$. □

The concept of long sequence is a **generalization** of redundant sequence (the path C is empty in the case of redundant sequence). Intuitively, the long sequence can be replaced by a sequence of moves along a shorter path (cutoff path) into which other pebbles do not enter between the beginning and the end of the sequence. Replacing a long sequence of moves by a sequence of moves along the path C again preserves validity of the solution. The replacement of long sequences must be performed repeatedly since new long sequences may arise.

The process of replacement is formally expressed below as algorithm 3. The worst case time complexity is $O(|\Phi|^4 + \Phi 3V2$; the space complexity is $O\Phi + V + E$.

5.4 Summary of Redundancy Elimination

Redundancies described above were discovered using the GraphRec software. Notice that the gradual generalization was adopted in the description. Although long sequences subsume both less general redundancies, it is not advisable to apply their replacement directly. It is better to apply elimination of redundancies stepwise from the less general one to more general ones. The reason for this practice is the increasing time complexity of redundancy elimination algorithms. A sequence of moves submitted to the more complex algorithm is potentially shortened by eliminating less general redundancies using this practice.

It is possible to reason about the implementation of a certain level of automation in the search for other types of redundancies. The common requirement shared by all the definitions is that the resulting solution must be shorter.

Algorithm 3. Replacement of long sequences.

function *ReplaceLongMoves* (Φ, G): **sequence**
do
  $(\eta, \pi) \leftarrow$*FindLongMoves*(Φ, G)
  $\Phi \leftarrow \Phi - \eta$; $\Phi \leftarrow \Phi \cup \pi$
while $(\eta, \pi) \neq (\emptyset, [])$
return Φ

function *FindLongMoves* (Φ, G): **pair**
let $[k_1: u_1 \rightarrow v_1, \ldots, k_{\xi-1}: u_{\xi-1} \rightarrow v_{\xi-1}] = \Phi$
for $i = 1,2,\ldots,\xi-2$ **do**
  for $j = \xi-1, \xi-2, \ldots, i+1$ **do**
    if $k_i = k_j$ **then**
      $\eta \leftarrow \emptyset$
      for $\tau = i, i+1, \ldots, j$ **do**
        if $k_i = k_\tau$ **then** $\eta \leftarrow \eta \cup \{k_\tau: u_\tau \rightarrow v_\tau\}$
      $C \leftarrow$*CheckLongMoves*$(\Phi, i, j, |\eta|, G)$
      if $C \neq []$ **then**
        let $[c_1, c_2, \ldots, c_n] = C$
        $\pi \leftarrow [k_i: c_1 \rightarrow c_2, \ldots, k_i: c_{n-1} \rightarrow c_n]$
        return (η, π)
return $(\emptyset, [])$

function *CheckLongMoves* $(\Phi, i, j, l, G = (V, E))$: **sequence**
let $[k_1: u_1 \rightarrow v_1, \ldots, k_{\xi-1}: u_{\xi-1} \rightarrow v_{\xi-1}] = \Phi$
$(V', E') \leftarrow G$; $V' \leftarrow V' - O^i(P - \{k_i\})$; $E' \leftarrow E' \cap \{\{u, v\} | u, v \in V'\}$
for $\iota = i+1, i+2, \ldots, j-1$ **do**
  if $k_\iota \neq k_i$ **then**
    $V' \leftarrow V' - \{u_\iota, v_\iota\}$; $E' \leftarrow E' \cap \{\{u, v\} | u, v \in V'\}$
let C be a shortest path between u_i and v_j in $G' = (V', E')$
if C is defined **and** $|C| < l$ **then return** C
return []

6. EXPERIMENTAL EVALUATION

An experimental evaluation was made with the suggested methods for redundancy elimination. Algorithms 1, 2, and 3 were implemented in C++ and were tested on a set of benchmark instances of the problem of pebble motion.

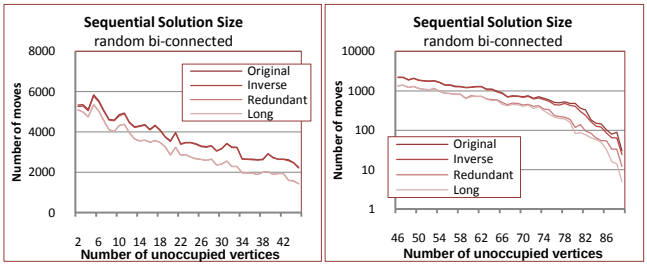


Figure 5. ***Solution length improvement*** *in random bi-connected graph.* Notice that the right part uses the logarithmic scale. The dependence on the increasing number of unoccupied vertices is shown. Up to 50% smaller solution can be obtained by eliminating redundant or long sequences.

Solutions found by the BIBOX-θ algorithm were submitted to redundancy elimination methods. The reduction of the length of the solution and runtime were measured. The implementation of redundancy elimination algorithms directly follows the pseudo-code given in section 5. It was always the case that the solution was processed by the less general redundancy elimination before it was submitted to more general one. In order to allow reproducibility of experiments the complete source code together with raw experimental data is provided at the web: http://ktiml.mff.cuni.cz/~surynek/research/ecaiw2010.

The first set of problems consists of **randomly generated bi-connected** graph with 90 vertices. The graph was constructed by

adding loops of random length (uniform distribution from 2..10) to the cycle of length 7 (actually tests were done with many random bi-connected graphs, indeed only one was selected for presentation here). The initial and the goal arrangement of pebbles were generated as random permutations.

The second set of testing instances consists of a **grid** of size 8×8 where the initial and the goal arrangement of pebbles were again random permutations. In both cases, the random permutation was generated by applying quadratic number of random transpositions of individual pebbles.

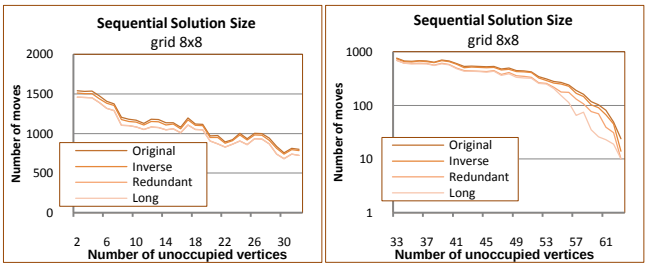


Figure 6. ***Solution length improvement*** *in the grid 8×8.* The right part uses logarithmic scale. Up to 10% smaller solution can be obtained by eliminating redundant or long sequences.

The reduction of the length of the solution depending on the increasing number of unoccupied vertices is shown in figures 5 and 6. Runtime of the individual methods is not presented due to space limitations. However, it can be summarized that the long sequence replacement is the most time consuming method. It consumed approximately 2 minutes (measured on a 2.4GHz machine) on instances with many pebbles.

It is possible to conclude that the solution can be reduced by up to 50% of the original size for problem on **random bi-connected** graph while better results are achieved when there is higher number of unoccupied vertices. For the **grid** 8×8, the reduction is not that large; the original size of the solution can be reduced by up to about 10%. Again, problems with higher number of unoccupied vertices render the possibility for better improvements.

Removal of redundant sequences represents the **best trade-off** between detection cost and solution improvement according to performed experiments. Whereas eliminating inverse moves or long sequences features utmost situations; the former brings almost no improvement; the latter is computationally too costly.

An expectable result is that the better improvement of solutions is gained when there are more unoccupied vertices in the input graph. Notice that definitions of redundancies are based on the **mutual non-interfering** motions of pebbles. The more unoccupied space is available in the graph the less interference between moves of pebbles is possible. The difference in the improvement for random bi-connected graphs and grids is partially caused by the difference of the average length of loops of the loop decomposition. The smaller these loops are the higher the interaction among pebbles is.

The most prohibitive aspect of the redundancy elimination methods with respect to their eventual practical application is quite high runtime. In additional experiments with larger graphs the runtime of removal of redundant sequences as well as the runtime of long sequence replacement was too high. However, this issue may be resolved by using better redundancy detection algorithms with lower asymptotic time complexity. This can be done by exploiting advanced data structures or by a so called opportunistic redundancy elimination which does not eliminate all the redundancies but only those that are encountered.

7. SUMMARY AND CONCLUSIONS

This work addressed the quality (length) of solutions of problems of pebble motion on a graph. Particularly, solutions generated by the existing state-of-the-art algorithm [9, 10] were analyzed with respect to presence of certain type of redundancies. If such redundancies really exist, which proved to be the case, their formal description and elimination was the next goal of this work. The new **visualization tool** *GraphRec* **has been developed** to enable comfortable analysis of solutions.

Several types of **redundancies were discovered** using the *GraphRec* software in generated solutions. **Methods for elimination** of described redundancies **were suggested** and experimentally evaluated. The performed experimental evaluation showed that solutions can be improved by up to 50% using the suggested methods. Another finding is that the better improvement can be gained for problems with higher number of unoccupied vertices.

REFERENCES

[1] Bilgin, A., Ellson, J., Gansner, E., Hu, Y., Koren, Y., North, S. *Graphviz - Graph Visualization Software*. Project web page, http://www.graphviz.org, 2009, (September 2009).

[2] Fruchterman, T. M. J. and Reingold, E. M. *Graph Drawing by Force-Directed Placement*. Software: Practice and Experience, Volume 21(November 1991), 1129-1164, John Wiley & Sons, 1991.

[3] Kamada, T. and Kawai, S. *An algorithm for drawing general undirected graphs.* Information Processing Letters, Volume 31 (January 1989), pp. 7-15, Elsevier, 1989.

[4] Kishimoto, A., Sturtevant, N. R. *Optimized algorithms for multi-agent routing.* Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Volume 3, pp. 1585-1588, IFAAMAS 2008.

[5] Kornhauser, D., Miller, G. L., Spirakis, P. G. *Coordinating Pebble Motion on Graphs, the Diameter of Permutation Groups, and Applications.* Proceedings of the 25th Annual Symposium on Foundations of Computer Science (FOCS 1984), pp. 241-250, IEEE Press, 1984.

[6] Koupý, P. *GraphRec - a visualization tool for entity movement on graph.* Student project web page, http://www.koupy.net/graphrec.php, 2010, (January 2010).

[7] Ratner, D. and Warmuth, M. K. *Finding a Shortest Solution for the N×N Extension of the 15-PUZZLE Is Intractable.* Proceedings of the 5th National Conference on Artificial Intelligence (AAAI 1986), pp. 168-172, Morgan Kaufmann Publishers, 1986.

[8] Ryan, M. R. K. *Exploiting subgraph structure in multi-robot path planning.* Journal of Artificial Intelligence Research (JAIR), Volume 31, (January 2008), pp. 497-542, AAAI Press, 2008.

[9] Surynek, P. *A Novel Approach to Path Planning for Multiple Robots in Bi-connected Graphs.* Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009), pp. 3613-3619, IEEE Press, 2009.

[10] Surynek, P. *An Application of Pebble Motion on Graphs to Abstract Multi-robot Path Planning.* Proceedings of the 21st International Conference on Tools with Artificial Intelligence (ICTAI 2009), pp. 151-158, IEEE Press, 2009.

[11] Tarjan, R. E. *Depth-First Search and Linear Graph Algorithms.* SIAM Journal on Computing, Volume 1 (2), pp. 146-160, Society for Industrial and Applied Mathematics, 1972.

[12] Wang, K. C., Botea, A. *Tractable Multi-Agent Path Planning on Grid Maps.* Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009), pp. 1870-1875, IJCAI Conference, 2009.

[13] Wilson, R. M. *Graph Puzzles, Homotopy, and the Alternating Group*. Journal of Combinatorial Theory, Ser. B 16, pp. 86-96, Elsevier, 1974.

On First-Order Propositional Neighborhood Logics: a First Attempt

D. Della Monica[1] and **G. Sciavicco**[2, 3]

Abstract. Propositional Neighborhood Logic (PNL) is the decidable interval-based temporal logic that features the modal operators corresponding to the Allen's relations *meets* and *met by*. Right PNL (RPNL) is the fragment of PNL featuring only one of the two modality allowed in PNL. In this paper, we introduce a new extension of RPNL, whose propositional letters are generalized into first-order formulas. In contrast with recent results on the decidability of first-order point-based temporal logics with only one variable, we show that the interval-based case yields undecidability. In particular, in this paper we prove that the first order version of RPNL, allowing first-order formulas with only one (possibly reused) variable, is undecidable with respect to most meaningful choices for temporal and first-order domains.

1 Introduction

Interval temporal logics are based on temporal structures over (usually) linearly ordered domains, where time intervals, rather than time instants, are the primitive ontological entities. The problem of representing and reasoning about time intervals arises naturally in various fields of computer science, artificial intelligence, and temporal databases, such as theories of action and change, natural language processing, and constraint satisfaction problems. In particular, temporal logics with interval-based semantics have been proposed as a useful formalism for the specification and verification of hardware [19] and of real-time systems [11].

A systematic analysis of the variety of relations between two intervals in a linear order was performed by Allen [1], who proposed the use of interval reasoning in systems for time management and planning. Allen identified the thirteen different binary relations between intervals on linear orders, hereafter referred to as Allen's relations. In [14], Halpern and Shoham introduced a multi-modal logic, hereafter called HS, involving modal operators corresponding to all Allen's interval relations and showed that such a logic is undecidable under very weak assumptions on the class of interval structures in which it is interpreted. One of the few known cases of decidable interval logics with truly interval semantics (not reducible to point-based semantics) is the *Propositional Neighborhood Interval Logic* (*PNL*) [5, 13]. PNL is a fragment of HS with only two modal operators, corresponding to the Allen's relations *meets* and its inverse *met by*. Its satisfiability problem has been shown to be decidable (NEXPTIME-complete) when interpreted over various classes of linearly ordered sets and, in particular, over domains based on natural numbers [6]; the results presented in the same paper and in [18] showed that all possible extensions of PNL with Allen's modal operator make the logic undecidable, which means that PNL is maximal in terms of decidability (as a matter of fact, there are extensions of PNL that are non-elementary decidable only if interpreted over finite prefixes of $\mathbb{N}$ and undecidable in most of the other cases), with respect to modal operators corresponding to Allen's relations. In [7, 8], authors proposed a 'metric' extension of PNL, called *Metric* PNL (MPNL, for short), which involves special propositional letters expressing equality or inequality constraints on the length of the current interval with respect to fixed integer constants. The satisfiability problem for MPNL interpreted in the interval structure over natural numbers is proved decidable in [8], with complexity between EXPSPACE and 2NEXPTIME when the integer constraints in formulae are represented in binary (and NEXPTIME-complete when the integer constraints in formulae are constant or represented in unary). In [17], the authors analyzed extensions of PNL and MPNL with binders and variables that allow one to store the length of the current interval with respect to decidability and showed that even the weakest natural extensions become undecidable, which in some cases is somewhat surprising, being in sharp contrast with the decidability of MPNL. Finally, (R)PNL and its metric version have been generalized to the spatial case [9, 4]. It is therefore natural to ask whether it is possible to generalize these logics by means of classical machinery, such as first order constructs, still keeping their good computational properties.

In this paper, we focus on a different extension of PNL, called *FORPNL* (*First Order RPNL*), obtained by generalizing propositional variables into first-order formulas. In the point-based case, the most prominent work concerning first-order temporal languages is the one by Hodkinson, Wolter and Zakharyaschev [15]. The authors show that first-order Linear Time Temporal Logic (LTL) with Since and Until, interpreted over discrete structures is already undecidable when only two distinct variables are allowed. The proof also applies for LTL with Next and Future only. But, unexpectedly, when one extends LTL with monadic first-order formulas (only one variable), the logic becomes decidable with temporal domains based on $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$, and $\mathbb{R}$ (in the last case the result holds only with finite first-order domains). We show here that for interval logics the situation is way worse. To this end, we consider the fragment of PNL, called *Right PNL* (*RPNL*), featuring only the modal operator corresponding to the Allen's relation *meets*; we prove that, independently from the properties of the underlying temporal order, the first-order extension of RPNL with only one variable over finite first-order domains is undecidable. This paper can be considered a first attempt of extending an interval-based temporal logic with truly first-order features (over the *first-order domain*), since previous work, such as ITL [19], only deal with first-order characteristics for the temporal domain. This

[1] University of Udine, Italy, *dario.dellamonica@uniud.it*
[2] University of Murcia, Spain, *guido@um.es*
[3] This author has been partially supported by the Spanish/South-African Project HS2008-0006, and by the Spanish project TIN2009-14372-C03-01.

also justify the choice for the name FORPNL: we want to keep the modal characteristics of the propositional logic, which allow one to move along the time domain only by means of the modal operators, and generalize the assertion over interval from propositional to first order. On the contrary, the cases of first order ITL [19] and NL [2] are different in this sense, since those languages include quantification over the temporal domain.

The paper is structured as follows. Section 2 introduces syntax and semantics of the logic we are interested in, namely FORPNL. Section 3 briefly reviews the state of the art on first order temporal logics. Next, in Section 4, we give the undecidability proof of FORPNL, before concluding.

2 First Order RPNL

At the propositional level, RPNL is built from a set $\mathcal{AP} = \{p, q, \ldots\}$ of propositional letters, the classical connectives $\vee, \neg$ (the remaining ones can be considered as abbreviations), and a modal operator $\Diamond$ which allows one to capture any right neighboring interval from the current one. Formulas are obtained from the grammar:

$$\varphi ::= \pi \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond\varphi.$$

where π is a pre-interpreted propositional letter that is true over all and only intervals of the type $[i, i]$, called point-intervals.

Given a linearly ordered domain $\mathbb{D} = \langle D, < \rangle$, a (*non-strict*) *interval* over $\mathbb{D}$ is any ordered pair $[i, j]$ such that $i \leq j$. An *interval structure* is a pair $\langle \mathbb{D}, \mathbb{I}(\mathbb{D}) \rangle$, where $\mathbb{I}(\mathbb{D})$ is the set of all intervals over $\mathbb{D}$. An *interval model* is a tuple $M = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), V \rangle$, where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D}) \rangle$ is an interval structure and $V : \mathbb{I}(\mathbb{D}) \to 2^{\mathcal{AP}}$ is a valuation function assigning to every interval the set of propositional letters that hold over it. Given an interval model $M = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), V \rangle$ and an interval $[i, j]$ over it, the semantics of RPNL-formulae is given by the clauses:

- $M, [i, j] \Vdash \pi$ iff $i = j$;
- $M, [i, j] \Vdash p$ iff $p \in V([i, j])$, for any $p \in \mathcal{AP}$;
- $M, [i, j] \Vdash \neg\psi$ iff it is not the case that $M, [i, j] \Vdash \psi$;
- $M, [i, j] \Vdash \psi \vee \tau$ iff $M, [i, j] \Vdash \psi$ or $M, [i, j] \Vdash \tau$;
- $M, [i, j] \Vdash \Diamond\psi$ iff there exists $h \geq j$ such that $M, [j, h] \Vdash \psi$.

A RPNL-formula φ is *satisfiable* if there exists a model M and an interval $[i, j]$ over it such that $M, [i, j] \Vdash \varphi$. The satisfiability problem for RPNL has been shown to be NEXPTIME-complete in [10].

We introduce now a first-order version of the logic RPNL, hereafter called *First Order RPNL* (*FORPNL*, for short). At the first-order level, propositional variables are generalized into *predicate symbols* $P, Q, \ldots$, each one of which has fixed arity. In addition, the language features a set of *individual variables* $x, y, \ldots$, a set of *individual constants* $a, b, \ldots$, and the *universal quantifier* $\forall x$ for each individual variable. Propositional variables can be viewed as 0-ary predicates. *Terms* $\tau_1, \tau_2, \ldots$ are either individual variables or individual constants. As standard, we have that $\exists x\varphi \equiv \neg\forall x\neg\varphi$. A *First Order Interval Model* is of the type $M = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathfrak{D}, \mathcal{I} \rangle$, where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D}) \rangle$ is an interval structure as before, $\mathfrak{D}$ is the *first-order domain* of M, and $\mathcal{I}$ is a function associating each interval of $\mathbb{I}(\mathbb{D})$ with a first-order structure

$$\mathcal{I}([i, j]) = \langle \mathfrak{D}, P^{\mathcal{I}([i,j])}, Q^{\mathcal{I}([i,j])}, \ldots \rangle.$$

At each interval $[i, j]$, a predicate $P^{\mathcal{I}([i,j])}$ is a relation on $\mathfrak{D}$ of the same arity as P (for propositional variable, it is simply true or false). Finally, λ is an *assignment* function mapping terms into elements in $\mathfrak{D}$. Notice that we are assuming that constants are *rigid*, that is, a constant a refers to the same element of the first-order domain $\mathfrak{D}$ regardless of which is the current interval. The semantics of FORPNL is the following:

- $M, [i, j], \lambda \Vdash \pi$ iff $i = j$;
- $M, [i, j], \lambda \Vdash P(\tau_1, \ldots, \tau_n)$ iff $P^{\mathcal{I}([i,j])}(\lambda(\tau_1), \ldots, \lambda(\tau_n))$;
- $M, [i, j], \lambda \Vdash \neg\psi$ iff it is not the case that $M, [i, j], \lambda \Vdash \psi$;
- $M, [i, j], \lambda \Vdash \psi \vee \phi$ iff $M, [i, j], \lambda \Vdash \psi$ or $M, [i, j], \lambda \Vdash \phi$;
- $M, [i, j], \lambda \Vdash \forall x\psi$ iff $M, [i, j], \lambda' \Vdash \psi$ for any assignment λ' that differs from λ at most for the value of x;
- $M, [i, j], \lambda \Vdash \Diamond\psi$ iff there exists $h \geq j$ such that $M, [j, h], \lambda \Vdash \psi$.

Therefore, FORPNL is a *partial* first order generalization of the propositional logic RPNL: one is allowed to move along the time domain by using only the modal operator, and to assert over a specific interval by using first-order construct. Moreover, it can be considered as the *product* of First-Order Logic and RPNL [12], since the first-order part and the modal part may interact freely.

3 Is FORPNL Without Hopes?

In this section, we recall some well-known results in the literature, that makes the result presented in this paper somehow surprising. First of all, we know that among the maximal first order logic fragments that have been shown to be decidable we can find:

- two-variable first order logic [3];
- two-variable first order logic over ordered domains (specifically, the class of all linear orders, and all linear orders over $\mathbb{N}$) [20].

In the framework of temporal logics, as already mentioned above, it has been shown in [15] that extending LTL (with Since and Until, but the result also applies to the fragment with Future and Next only) with a first-order machinery with two distinct variables yields undecidability. To retrieve decidability one must restrict the language by allowing only one variable.

We want to prove here that in the interval-based case, the situation is way worse. RPNL represents one of the first, and most studied, case of decidable interval logics. It has been shown to be decidable [6]:

- in the class of all linearly ordered sets;
- in the class of all discrete linearly ordered sets;
- in the class of all dense linearly ordered sets;
- in the class of all finite linearly ordered sets;
- in the class of all linearly ordered sets based on $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$.

In despite of the generally good behaviour of RPNL (w.r.t. the problem of satisfiability) and of the possibility of extending the temporal (point-based) logic LTL with first-order constructs, as we will prove below, the combination of almost any first-order ingredient and of the interval-based frame results in undecidability.

4 Undecidability

As it becomes clear from the above, there are a number of possible parameters here. Beside the usual possible choices for the temporal domain, that is, discrete, dense, finite, bounded, unbounded, and so on, we can vary on the first order component by assuming that the first-order domain is finite, infinite, constant, variable, expanding, or assuming other specific properties for it (linearity, discreteness, denseness, and so on), and also by limiting the number of distinct

variables in formulas. Since we are interested in tight undecidability results, in contrast with decidability results for first order point-based temporal logic, we focus our attention on very restrictive assumptions. In particular, assuming the temporal domain to be finite, the decidability result becomes really simple (although the complexity is the same as in the other cases, NEXPTIME, the constants hidden in the complexity function are low, and the idea under the model theoretic argument is easy to understand [10]). For these reasons, from now on, we assume that both D and $\mathfrak{D}$ are finite, and that our language has only one variable. Nevertheless, the results presented in this paper hold even over the class of all (resp., all dense, all discrete) linearly ordered sets, independently from the assumption on the first-order domain (infinite, expanding, dense, discrete, and so on). Moreover, in our construction there are neither free variables nor constants, so we omit the variable assignment λ.

We make use of the undecidability of the *Finite Tiling Problem* [16]. It is the problem of establishing whether, for a given set of tile types $\mathcal{T} = \{t_1, \ldots, t_k\}$, there exists a finite rectangle $\mathcal{R} = [1, X] \times [1, Y] = \{(i, j) : i, j \in \mathbb{N}, 1 \leq i \leq X, \text{and } 1 \leq j \leq Y\}$ for some $X, Y \in \mathbb{N}$, such that $\mathcal{T}$ can correctly tile $\mathcal{R}$ with the entire border colored by the same designated color \$, also called *side color*. To be more precise, for every tile type $t_i \in \mathcal{T}$, let $right(t_i)$, $left(t_i)$, $up(t_i)$, and $down(t_i)$ be the colors of the corresponding sides of t_i. To solve the Finite Tiling Problem for $\mathcal{T}$ one must find two natural numbers X and Y, and a mapping $f : \mathcal{R} \to \mathcal{T}$ such that:

$$\begin{array}{ll} right(f(i,j)) = left(f(i+1,j)), & 0 \leq i < X, 0 \leq j \leq Y, \\ up(f(i,j)) = down(f(i,j+1)), & 0 \leq i \leq X, 0 \leq j < Y, \end{array}$$

and that satisfies, in addition, the following constraints:

$$\begin{array}{llll} left(f(0,j)) = \$ & \text{and} & right(f(X,j)) = \$, & 0 \leq j \leq Y, \\ down(f(i,0)) = \$ & \text{and} & up(f(i,Y)) = \$, & 0 \leq i \leq X. \end{array}$$

where \$ is the side color of $\mathcal{R}$.

In order to perform the reduction from the Finite Tiling Problem for the set of tiles $\mathcal{T} = \{t_1, \ldots, t_k\}$ to the satisfiability problem for FORPNL, we will make use of some special 0-ary predicate symbols, namely u, Id, $\mathsf{up\_rel}$, final, $\mathtt{t}_1, \mathtt{t}_2, \ldots, \mathtt{t}_k$. The reduction consists of three main steps:

1. the encoding of the rectangle by means of a suitable finite chain of so-called 'unit' intervals (u-*intervals*, for short);
2. the encoding of the 'above-neighbor' relation by means of a suitable family of so-called $\mathsf{up\_rel}$-*intervals*; and
3. the encoding of the 'right-neighbor' relation.

Here is a sketch of the encoding. First, we set our framework by forcing the existence of a unique finite chain of u-intervals on the linear ordering (u-*chain*, for short). The u-intervals are used as cells to arrange the tiling. In other words, they represent the parts of the plane that must be covered by tiles. Next, we define a chain of Id-intervals (Id-*chain*, for short), each of them representing a row of the rectangle. Any Id-interval consists of a sequence of u-intervals; each Id will contain exactly the same number of u-intervals. Then, we use $\mathsf{up\_rel}$ to encode the relation that connects each tile with its above neighbor in $\mathcal{R}$. Finally, we introduce a set of 0-ary predicate symbols $\mathtt{T} = \{\mathtt{t}_1, \mathtt{t}_2, \ldots, \mathtt{t}_k\}$ corresponding to the set of tile types $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ and define a formula $\Phi_{\mathcal{T}}$ which is satisfiable if and only if there exists a finite rectangle $\mathcal{R}$ for some $X, Y \in \mathbb{N}$ and a proper tiling of $\mathcal{R}$ by $\mathcal{T}$, i.e., a tiling that satisfies the color constraints on the border tiles and between vertically- and horizontally-adjacent tiles.

The proof exploits the fact that introducing first order constructs makes it possible to express properties of the type: "*if an interval satisfyes φ, then all its beginning intervals (resp., ending intervals, strict sub-intervals) do not stisfy ψ*", where the strict sub-intervals of an interval $[a, b]$ are all intervals $[c, d]$ such that $a < c < d < b$. In order to express such properties, we firstly define some kind of 'nominals' for each point of the temporal domain. Intuitively, we univocally identify each point i of the temporal domain with a non-empty set of constants that make a special predicate true in intervals starting from i. More formally, we force a predicate of the type $P(x)$ in such a way that if $P(x)$ is true, for some x, over an interval $[i, j]$, then it can be possibly true (for the same x) only over interval starting from i and it must be false over all intervals starting from some different point $h \neq i$. For example, given an interval $[i, j]$ that satisfies $P^{\mathcal{I}([i,j])}(a)$, for some constant a, we force $\neg P^{\mathcal{I}([h,k])}(a)$ to hold over each interval $[h, k]$, with $h \neq i$. To this end, we exploit the following formula:

$$\Box\Box(\exists x \Diamond P(x) \wedge \forall x(\Diamond P(x) \rightarrow \Box(\neg\pi \rightarrow \Box\neg P(x)))) \quad (1)$$

It is easy to verify the following lemma:

Lemma 1 *Let M be a FORPNL model and $[i, j]$ an interval over it. If $M, [i, j] \Vdash (1)$, then for each $h \in D$:*

1. *there exists a point $k > h$ such that $P^{\mathcal{I}([h,k])}(a)$ holds for some a,*
2. *for each a such that $P^{\mathcal{I}([h,k])}(a)$ holds, then $\neg P^{\mathcal{I}([l,m])}(a)$ holds, for each $l \neq h$.*

At this step, we can express properties about beginning intervals, ending intervals, or strict sub-intervals of a given interval, by exploiting such a notion of nominals, formalized in the above lemma. For example, it is easy to see that the following formula correctly defines the operator $[B^{\varphi}_{\psi}]$ (resp., $[E^{\varphi}_{\psi}]$, $[D^{\varphi}_{\psi}]$), expressing the property: "*if an interval satisfies the property φ, then each beginning interval (resp., ending interval, strict sub-interval) satisfies the property ψ*", thus 'simulating' the modal operator $[B]$ (resp., $[E]$, $[D]$) of the logic HS, corresponding to the Allen's relation *begins* (resp., *ends*, *during*):

$$[B^{\varphi}_{\psi}] \equiv \Box\Box\forall x(\Diamond(\varphi \wedge \Diamond P(x)) \rightarrow \Box(\Diamond(\neg\pi \wedge \Diamond P(x)) \rightarrow \psi))$$
$$[E^{\varphi}_{\psi}] \equiv \Box\Box\forall x(\Diamond(\varphi \wedge \Diamond P(x)) \rightarrow \Box(\neg\pi \rightarrow \Box(\Diamond P(x) \rightarrow \psi)))$$
$$[D^{\varphi}_{\psi}] \equiv \left\{ \begin{array}{l} \Box\Box\forall x(\Diamond(\varphi \wedge \Diamond P(x)) \rightarrow \\ \Box(\neg\pi \rightarrow \Box(\Diamond(\neg\pi \wedge \Diamond P(x)) \rightarrow \psi))) \end{array} \right.$$

Notice that we are not able to properly define the HS operators $[B]$, $[E]$, and $[D]$, since we cannot capture beginning, ending, and during intervals of the current one.

To define the u-chain we use the following formulae:

$$\Diamond(\neg\pi \wedge \mathsf{u}) \quad (2)$$
$$\Box\Box(\mathsf{u} \rightarrow (\neg\pi \wedge (\Diamond\mathsf{u} \vee \Box\pi))) \quad (3)$$
$$[B^{\mathsf{u}}_{\neg\mathsf{u}}] \wedge [B^{\mathsf{u}}_{\neg\pi \rightarrow \neg\Diamond\mathsf{u}}] \quad (4)$$
$$(1) \wedge (2) \wedge (3) \wedge (4) \quad (5)$$

Lemma 2 *Let $M = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathfrak{D}, \mathcal{I} \rangle$ be a FORPNL model based on a finite linearly ordered temporal domain and with a finite first-order domain, such that*

$$M, [i_0, j_0] \Vdash (5).$$

Then, there exists a finite sequence of points $j_0 < j_1 < \ldots < j_n$, with $n > 0$, such that:

1. $M, [j_l, j_{l+1}] \Vdash \mathsf{u}$ *for each* $0 \leq l \leq n-1$*;*
2. $M, [j', j''] \Vdash \mathsf{u}$ *holds for no other interval* $[i', j']$.

Proof. If $M, [i_0, j_0] \Vdash (5)$, then, by (2), for some $j_1 > j_0$ the interval $[j_0, j_1]$ is a u-interval. By (3), j_1 starts a finite chain of u-intervals $[j_l, j_{l+1}]$, with $l \geq 0$. The satisfiability of (3) over finite temporal domains follows from the fact that the last point of the temporal domain satisfies $\Box\pi$. Now suppose, by contradiction, that for some interval $[j', j'']$, it is the case that $[j', j'']$ is a u-interval but $[j', j''] \neq [j_l, j_{l+1}]$ for all $l > 0$. Then either $j' = j_l$ for some l, contradicting the first conjunct of (4), or $j_l < j' < j_{l+1}$, contradicting the second conjunct of (4). ■

We now define the ld-chain with the following formulae:

$$\Diamond\mathsf{ld} \wedge \Box\Box((\Diamond\mathsf{ld} \rightarrow \Diamond\mathsf{u}) \wedge (\mathsf{ld} \rightarrow \neg\pi \wedge \neg\mathsf{u} \wedge (\Diamond\mathsf{ld} \vee \Box\pi))) \quad (6)$$

$$[B^{\mathsf{ld}}_{\neg\mathsf{ld}}] \wedge [B^{\mathsf{ld}}_{\neg\pi\rightarrow\neg\Diamond\mathsf{ld}}] \quad (7)$$

$$(6) \wedge (7) \quad (8)$$

Lemma 3 *Let* $M = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathfrak{D}, \mathcal{I} \rangle$ *be a FORPNL model based on a finite linearly ordered temporal domain and with a finite first-order domain, such that*

$$M, [i_0, j_0] \Vdash (5) \wedge (8).$$

Then, there exist a positive integer v *and a finite sequence of positive integers* $m_1, m_2, \ldots, m_v$ *and a finite sequence of points* $j_0^1 < j_1^1 < \ldots < j_{m_1}^1 = j_0^2 < \ldots < j_{m_2}^2 = \ldots = j_0^{v-1} < \ldots < j_{m_{v-1}}^{v-1} = j_0^v < \ldots < j_{m_v}^v$ *such that, for each* $1 \leq s \leq v$*, we have* $M, [j_0^s, j_{m_s}^s] \Vdash \mathsf{ld}$*, and no other interval satisfies* ld.

Proof. First of all, by Lemma 2, there is a finite sequence of points $j_0 < j_1 < \ldots < j_n$, with $n > 0$, defining a finite chain of u-intervals. By (6), j_0 starts a ld-interval, which must end at some $j_l > j_1$. By (6), each ld-interval is followed by another ld-interval, and each ld-interval must end at some j_l. Thus, every ld-interval spans several u-intervals, and there are finitely many ld-intervals. Let their number be v. Hence, the sequence $j_0 < j_1 < \ldots < j_n$ can be written as $j_0^1 < j_1^1 < \ldots < j_{m_1}^1 = j_0^2 < \ldots < j_{m_2}^2 = \ldots = j_0^{v-1} < \ldots < j_{m_{v-1}}^{v-1} = j_0^v < \ldots < j_{m_v}^v$, as required. We want to show that there are no other ld-interval beside those of the type $[j_0^s, j_{m_s}^s]$. This can be shown exactly as in Lemma 2, by using (7), joined with (1). ■

The above lemma guarantees the existence of an ld-chain. Now, we want to force the propositional letter up_rel to correctly encode the relation that connects pairs of tiles of the rectangle $\mathcal{R}$ that are vertically adjacent. Formally, we define two u-intervals $[j_l, j_{l+1}]$ and $[j_{l'}, j_{l'+1}]$ to be *above-connected* if and only if $[j_{l+1}, j_{l'}]$ is a up_rel-interval. At the same time, we want to make sure that each ld-interval spans the same number of tile-intervals. Intuitively, these two properties can be guaranteed by assuring that each u-interval of a ld-interval is connected with exactly one u-interval of the next ld-interval and with exactly one ld-interval of the previous level. To this end, firstly we suitably label u-intervals belonging to the last ld-interval with the propositional letter final. Then, we constraint each u-interval not belonging to the last ld-interval to be connected to at least one u-interval in the future (formula (10)) and at least one interval in the past (formula (16)) by means of a up_rel-interval. In order to guarantee the correct correspondence between u-intervals of consecutive ld-intervals and to guarantee that each u-interval is connected with at most one u-interval in the future and at most one u-interval in the past, we force the condition that no up_rel-interval is a biginning interval (resp., ending interval, strict sub-interval) of any other up_rel-interval. Finally, to guarantee that up_rel-intervals connect u-intervals belonging to consecutive ld-intervals, we have to make sure that no ld-interval is a biginning interval (resp., ending interval, strict sub-interval, strict super-interval) of a up_rel-interval.

$$\Box\Box(\mathsf{u} \wedge \Box\Box\neg\mathsf{ld} \leftrightarrow \mathsf{final}) \quad (9)$$

$$\Box\Box(\mathsf{u} \rightarrow (\neg\mathsf{final} \leftrightarrow \Diamond\mathsf{up\_rel}) \quad (10)$$

$$\Box\Box(\mathsf{up\_rel} \rightarrow \neg\mathsf{ld} \wedge \neg\pi \wedge \neg\mathsf{u} \wedge \Diamond\mathsf{u})) \quad (11)$$

$$\neg\Diamond\mathsf{up\_rel} \wedge \Box\Box(\Diamond\mathsf{up\_rel} \rightarrow \Diamond\mathsf{u}) \quad (12)$$

$$[B^{\mathsf{up\_rel}}_{\neg\mathsf{up\_rel}}] \wedge [E^{\mathsf{up\_rel}}_{\neg\mathsf{up\_rel}}] \wedge [D^{\mathsf{up\_rel}}_{\neg\mathsf{up\_rel}}] \quad (13)$$

$$[B^{\mathsf{up\_rel}}_{\neg\mathsf{ld}}] \wedge [E^{\mathsf{up\_rel}}_{\neg\mathsf{ld}}] \wedge [D^{\mathsf{up\_rel}}_{\neg\mathsf{ld}}] \quad (14)$$

$$[D^{\mathsf{ld}}_{\neg\mathsf{up\_rel}}] \quad (15)$$

$$\forall x(\Diamond(\mathsf{ld} \wedge \Diamond(\Diamond\mathsf{u} \wedge \Diamond P(x))) \rightarrow \Diamond\Diamond(\mathsf{up\_rel} \wedge \Diamond P(x))) \quad (16)$$

$$(9) \wedge (10) \wedge (11) \wedge (12) \wedge (13) \wedge (14) \wedge (15) \wedge (16) \quad (17)$$

Lemma 4 *Let* $M = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathfrak{D}, \mathcal{I} \rangle$ *be a FORPNL model based on a finite linearly ordered temporal domain and with a finite first-order domain, such that*

$$M, [i_0, j_0] \Vdash (5) \wedge (8) \wedge (17).$$

Then, we have that, for each $0 < s < v$ *and each* $0 \leq l < m_s$*,* $M, [j_{l+1}^s, j_l^{s+1}] \Vdash$ up_rel*, and no other interval satisfies* up_rel*. Moreover, we have that for each* $1 \leq s, s' \leq v$*,* $m_s = m_{s'}$.

Proof. Consider any u-interval $[j_l^s, j_{l+1}^s]$ not belonging to the last ld-interval. Formula (10) makes sure that j_{l+1}^s starts a up_rel-interval, which cannot be point-interval and must end at some point of the type $j_{l'}^{s'} > j_{l+2}^s$. First of all, observe that $j_{l'}^{s'} \geq j_0^{s+1}$, otherwise we would have a contradiction with (15). Similarly, we have that $j_{l'}^{s'} < j_{m_{s+1}}^{s+1}$, in order to avoid a contradiction with (14). Now, suppose by contradiction that $[j_0^s, j_1^s]$ is above-connected with $[j_l^{s+1}, j_{l+1}^{s+1}]$, with $l > 0$, for some s. By (16), there must be an up_rel-interval ending in j_0^{s+1} and starting from a point $j_{l'}^s$, with $l' > 0$. It must also be $l' > 1$, otherwise there would be two different up_rel-intervals starting at the same point j_1^s, contradicting the first conjunct of (13). So, it is the case that the up_rel-interval $[j_{l'}^s, j_0^{s+1}]$ is a strict sub-interval of the up_rel-interval $[j_1^s, j_l^{s+1}]$, contradicting the third conjunt of (13). By applying a similar argument, and assuming that up to a given l, $[j_l^s, j_{l+1}^s]$ is above-connected to $[j_l^{s+1}, j_{l+1}^{s+1}]$, it is easy to show also that $[j_{l+1}^s, j_{l+2}^s]$ (if any) is above-connected to $[j_{l+1}^{s+1}, j_{l+2}^{s+2}]$. From (13) it follows that each u-interval can be connected with at most one u-interval in the future and at most one in the past, so we can conclude that for each $0 \leq s, s' \leq v$, $m_s = m_{s'}$. ■

Finally, we can force all tile-matching conditions to be respected, by using the following formulae, where $\mathtt{T_r}$ (resp., $\mathtt{T_l}$, $\mathtt{T_u}$, $\mathtt{T_d}$) is the subset of T containing all tiles having the right (resp., left, up, down) side colored with \$.

$$\Box\Box\left(\mathsf{u} \rightarrow \bigvee_{\mathtt{t_q}\in\mathtt{T}} \mathtt{t_q} \wedge \bigwedge_{\mathtt{t_q}\neq\mathtt{t_{q'}}} \neg(\mathtt{t_q} \wedge \mathtt{t_{q'}})\right) \tag{18}$$

$$\Box\Box\left(\bigvee_{\mathtt{t_q}\in\mathtt{T}} \mathtt{t_q} \rightarrow \mathsf{u}\right) \tag{19}$$

$$\Box\Box\left(\bigvee_{\mathtt{t_q}\in\mathtt{T}} \mathtt{t_q} \rightarrow \left(\neg(\Diamond\mathsf{Id} \vee \Box\pi) \rightarrow \bigvee_{right(t_q)=left(t_{q'})} \Diamond\mathtt{t'_q}\right)\right) \tag{20}$$

$$\Box\Box\left(\bigvee_{\mathtt{t_q}\in\mathtt{T}} \mathtt{t_q} \rightarrow \left(\Diamond\mathsf{up\_rel} \rightarrow \bigvee_{up(t_q)=down(t_{q'})} \Diamond(\mathsf{up\_rel} \wedge \Diamond\mathtt{t'_q})\right)\right) \tag{21}$$

$$\Box\Box\left(\Diamond\mathsf{Id} \rightarrow \left(\Diamond \bigvee_{\mathtt{t_q}\in\mathtt{T_l}} \mathtt{t_q}\right) \wedge \left(\mathsf{u} \rightarrow \bigvee_{\mathtt{t_q}\in\mathtt{T_r}} \mathtt{t_q}\right)\right) \tag{22}$$

$$\exists x\left(\Diamond(\mathsf{Id} \wedge \Diamond P(x)) \rightarrow \Box\Box\left(\mathsf{u} \wedge \Diamond\Diamond P(x) \rightarrow \bigvee_{\mathtt{t_q}\in\mathtt{T_d}} \mathtt{t_q}\right)\right) \tag{23}$$

$$\Box\Box\left(\mathsf{u} \wedge \mathsf{final} \rightarrow \bigvee_{\mathtt{t_q}\in\mathtt{T_u}} \mathtt{t_q}\right) \tag{24}$$

$$(18) \wedge (19) \wedge (20) \wedge (21) \wedge (22) \wedge (23) \wedge (24) \tag{25}$$

Theorem 5 *Given any finite set of tiles $\mathcal{T}$ and a side color \$, the formula*

$$\Phi_{\mathcal{T}} := (5) \wedge (8) \wedge (17) \wedge (25)$$

is satisfiable in a finite linearly ordered temporal domain and finite first-order domain if and only if $\mathcal{T}$ can tile a finite rectangle $\mathcal{R}$, for some $X, Y \in \mathbb{N}$, with side color \$.

Proof. (Only if:): Suppose that $M, [i_0, j_0] \Vdash \Phi_{\mathcal{T}}$. Then, by Lemma 3, there is a sequence of points $j_0 = j_0^1 < j_1^1 < \ldots < j_{m_1}^1 = j_0^2 < \ldots < j_{m_2}^2 = \ldots = j_0^{v-1} < \ldots < j_{m_{v-1}}^{v-1} = j_0^v < \ldots < j_{m_v}^v = j_n$, and by Lemma 4, for each $1 \leq s, s' \leq v$, $m_s = m_{s'}$. We put $X = m_s$ and $Y = v$. For all l, s, where $0 \leq l \leq X - 1$, $1 \leq s \leq Y$, define $f(l, s) = t_q$ if and only if $M, [j_l^s, j_{l+1}^s] \Vdash \mathtt{t_q}$. From Lemma 2, 3, and 4 it follows that the function $f : \mathcal{R} \rightarrow \mathcal{T}$ defines a correct tiling of $\mathcal{R}$, where X and Y are defined as above.

(If:) Let $f : \mathcal{R} \mapsto \mathcal{T}$ be a correct tiling function of the rectangle $\mathcal{R} = [1, X] \times [1, Y]$ for some X, Y, and a given border color \$. For convenience, we will identify the tile-variables $\mathtt{t}_1, \mathtt{t}_2, \ldots \in \mathtt{T}$ with their corresponding tiles $t_1, t_2, \ldots \in \mathcal{T}$. We will show that there exist a model M and an interval $[i_0, j_0]$ such that $M, [i_0, j_0] \Vdash \Phi_{\mathcal{T}}$. Let $D = \mathfrak{D} = \mathbb{N}\,|_{X \cdot Y + 1}$, and let M the FORPNL model built over these two domains. We want to build an interpretation $\mathcal{I}$ in such a way that $M, [0, 1] \Vdash \Phi_{\mathcal{T}}$. Then, we put

$$\mathsf{u}^{\mathcal{I}([i,i+1])} \quad \forall i. 0 < i < X \cdot Y,$$

to guarantee that (5) is satisfied. Now, in order to satisfy the remaining part of $\Phi_{\mathcal{T}}$ on $[0, 1]$, it suffices to define the valuation for the remaining propositional letters and the predicate symbol P:

$$\begin{aligned} P^{\mathcal{I}([i,j])}(i) &\quad \forall i, j > 0 \\ \mathsf{Id}^{\mathcal{I}([i\cdot X+1,(i+1)\cdot X+1])} &\quad \forall i. 0 \leq i \leq Y - 1 \\ \mathsf{up\_rel}^{\mathcal{I}([i,i+X-1])} &\quad \forall i. 2 \leq i \leq X \cdot (Y - 1) + 1 \\ \mathsf{final}^{\mathcal{I}([i,i+1])} &\quad \forall i. X \cdot (Y - 1) + 1 \leq i \leq X \cdot Y \end{aligned}$$

Finally, we evaluate the tile-variables as follows. For each $\mathtt{t} \in \mathtt{T}$:

$$\mathtt{t_q}^{\mathcal{I}([i,i+1])} \Leftrightarrow f(l, s) = t_q \quad \forall i = X \cdot (s - 1) + l.$$

■

5 Conclusions

Temporal logic has found numerous applications in computer science, ranging from the traditional and well-developed fields of program specification and verification, temporal databases, and distributed multi-agent systems, to more recent uses in knowledge representation and reasoning. This is true both at the propositional and first-order level. In the interval-based temporal logic world, undecidability is the rule and decidability the exception. Propositional Neighborhood Logic is one of the first examples of properly interval-based temporal logics shown to be decidable. Recently, it has also been extended with a sort of metric features that allow one to constrain the length of an interval (over natural numbers), without losing decidability. On the line of [17], here we have shown that yet another classical extension for temporal logics, obtained by generalizing propositional letters into first-order formulas, oversteps the barrier of decidability, even in a very restrictive case such as that of monadic first order formulas with finite domains. At a first glance this result may appear discouraging, concerning our aim of finding decidable first-order interval temporal logics. Nevertheless, it should be pointed out that the modal constant π plays an important role in the reduction. Thus, it could be worth considering the satisfiability problem for the language devoid of such an operator, as well as the satisfiability problem for FORPNL restricted with some natural syntactic rule that constrain the relationship between the modal and the first-order components.

REFERENCES

[1] J. F. Allen, 'Maintaining knowledge about temporal intervals', *Communications of the ACM*, **26**(11), 832–843, (1983).

[2] R. Barua and Z. Chaochen, 'Neighbourhood logics: Nl and nl 2', Technical Report 120, UNU/IIST, Macau, (1997).

[3] E. Börger, E. Grädel, and Y. Gurevich, *The Classical Decision Problem*, Perspectives of Mathematical Logic, Springer, 1997.

[4] D. Bresolin, D. Della Monica, A. Montanari, P. Sala, and G. Sciavicco, 'A decidable spatial generalization of metric interval temporal logic', in *Proc. of 17th International Symposium on Temporal Representation and Reasoning (in press)*. IEEE Computer Society Press, (2010).

[5] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco, 'On Decidability and Expressiveness of Propositional Interval Neighborhood Logics', in *Proc. of the International Symposium on Logical Foundations of Computer Science*, volume 4514 of *LNCS*, pp. 84–99. Springer, (2007).

[6] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco, 'Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions', *Annals of Pure and Applied Logic*, **161**(3), 289 – 304, (2009).

[7] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco, 'Right propositional neighborhood logic over natural numbers with integer constraints for interval lengths', in *Proc. of the 7th IEEE Conf. on Software Engeneering and Formal Methods*, pp. 240–249, (2009).

[8] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco, 'Metric propositional neighborhood logic: Expressiveness, decidability, and undecidability', in *Accepted for publication in the Proc. of the 19th European Conference on Artificial Intelligence (ECAI)*, (2010).

[9] D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco, 'A tableau-based system for spatial reasoning about directional relations', in *Proc. of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2009)*, volume 5607 of *LNAI*, pp. 123–137. Springer, (2009).

[10] D. Bresolin, A. Montanari, and G. Sciavicco, 'An optimal decision procedure for Right Propositional Neighborhood Logic', *Journal of Automated Reasoning*, **38**(1-3), 173–199, (2007).
[11] Z. Chaochen, C. A. R. Hoare, and A. P. Ravn, 'A calculus of durations', *Information Processing Letters*, **40**(5), 269–276, (1991).
[12] D.M. Gabbay and V.B. Shehtman, 'Products of modal logics, part 1', *Logic Journal of the IGPL*, **6**(1), 6–1, (1998).
[13] V. Goranko, A. Montanari, and G. Sciavicco, 'Propositional interval neighborhood temporal logics', *Journal of Universal Computer Science*, **9**(9), 1137–1167, (2003).
[14] J. Halpern and Y. Shoham, 'A propositional modal logic of time intervals', *Journal of the ACM*, **38**(4), 935–962, (1991).
[15] I.M. Hodkinson, F. Wolter, and M. Zakharyaschev, 'Decidable fragment of first-order temporal logics', *Ann. Pure Appl. Logic*, **106**(1-3), 85–134, (2000).
[16] J. Kari, 'Reversibility and surjectivity problems of cellular automata', *Journal of Computer Systems and Science*, **48**, 149–182, (1994).
[17] D. Della Monica, V. Goranko, and G. Sciavicco, 'Hybrid metric propositional neighborhood logics with interval length binders', in *Accepted for publication in the Proc. of the International Workshop on Hybrid Logic and Applications (HyLo 2010)*, ENTCS, (2010).
[18] A. Montanari, G. Puppis, and P. Sala, 'Maximal decidable fragments of halpern and shoham's modal logic of intervals', Technical report, University of Udine, (2010).
[19] B. Moszkowski, *Reasoning about digital circuits*, Tech. rep. stan-cs-83-970, Dept. of Computer Science, Stanford University, Stanford, CA, 1983.
[20] M. Otto, 'Two variable first-order logic over ordered domains', *Journal of Symbolic Logic*, **66**(2), 685–702, (2001).

Encoding Spatial Domains with Relational Bayesian Networks

Valquiria Fenelon[1] and **Britta Hummel**[2] and **Paulo E. Santos**[3] and **Fabio G. Cozman**[4]

Abstract. This paper investigates an encoding of a probabilistic TBox using relational Bayesian networks that are specified through a probabilistic description logic. The probabilistic description logic extends the popular $\mathcal{ALC}$ language; on top of this language we add a few operations that are needed to represent the cardinal direction calculus. Using such resources we model roads containing lanes, and vehicles containing digital maps, GPS and video cameras.

1 Introduction

In this paper we study the combination of description logics and relational Bayesian networks as a language to encode qualitative spatial reasoning (QSR). Bayesian networks can be used to represent uncertainty in propositional domains [26], while relational Bayesian networks lift the representation to first-order. Our strategy is to restrict the full generality (and full complexity) of relational Bayesian networks by focusing on a class of relational Bayesian networks that can be specified using a probabilistic description logic called CR$\mathcal{ALC}$ [7, 6]. In a recent publication [30] we have discussed the use of CR$\mathcal{ALC}$ to encode a subset of a cardinal direction calculus [14, 19], but we did so by restricting some features of this calculus. In this paper we remove some of these restrictions by adding a few elements to our relational Bayesian networks (elements that cannot be directly expressed by CR$\mathcal{ALC}$ but that do not introduce substantial complexity). We then investigate the use of the resulting qualitative spatial reasoning formalism to handle queries about a traffic scenario.

We focus on *lane recognition* tasks. Lane recognition research has traditionally focused on estimating the geometric properties of the lane in front of the vehicle using on-board imaging devices (McCall and Trivedi [22] provide an overview of this area). Only a few attempts have been made at inferring *functional* properties of lanes [17], such as the permitted driving directions (e.g. *going up / going down* the road), the permitted turning directions (e.g. *right turn/straight ahead/left turn*), or the permitted traffic participants (e.g. *motor vehicles/emergency vehicles/cyclists*). Existing on-board sensors can only provide a highly incomplete picture of the functional road configuration, with substantial uncertainty.

This paper is organized as follows. After a literature review in the next section, Sections 3 and 4 present, respectively, a formalization of the chosen application scenario and its implementation, with emphasis on features that we have added to the CR$\mathcal{ALC}$ specification and that go beyond our previous publication [30]. We note that the present paper revisits material from this previous publication, contributing mainly on the extensions of CR$\mathcal{ALC}$ that handle role hierarchies and disjoint concepts needed in the cardinal direction calculus. Conclusions are left to Section 5.

[1] Valquiria is with Escola Politecnica, USP, S. Paulo, Brazil
[2] Britta is with Karlsruhe University, Germany
[3] Paulo is with FEI, S. Paulo, Brazil
[4] Fabio is with Escola Politecnica, USP, S. Paulo, Brazil

2 Literature Overview

This section reviews relevant literature on QSR, cardinal direction calculus, probabilistic description logics, and CR$\mathcal{ALC}$; this material is mostly lifted from our previous publication [30].

2.1 Qualitative spatial reasoning

The aim of Qualitative Spatial Reasoning (QSR) is the logical formalisation of knowledge from elementary spatial entities, such as spatial regions, line segments, cardinal directions, and so forth, as surveyed in [4, 5]. Relevant to the present work are the developments of spatial formalisms for computer vision and robotics. The first proposal for a logic-based interpretation of images is described in [28], where image interpretation is reduced to a constraint satisfaction problem on a set of axioms. Inspired by these ideas, [21] proposes a system that generates descriptions of aerial images, which more recently received a descriptive logic enhancement [24].

A spatial system based on spatio-temporal histories for scene interpretation was investigated in [15], which was inspired on an earlier proposal for learning event models from visual information [13]. More recently, [3] proposes a system that uses multiple spatio-temporal histories in order to evaluate an image sequence. A logic formalisation of the viewpoint of a mobile agent was presented in [27], and was further explored in the interpretation of scenes within a mobile robotics scenario in [29]. In [17], functional and geometric properties of roads and intersections could be inferred using an expressive, yet deterministic, description logic in combination with on-board vehicle sensors.

These approaches do not handle uncertainty, which is either left for the low-level processing [3] or simply ignored [29].

2.2 Cardinal direction calculus

The cardinal direction calculus (CDC) [14] is a formalism for reasoning about cardinal directions between spatial objects. The major reasoning task that CDC is concerned with is to infer the direction between two objects A and C, from the known directions between A and (another object) B and between B and C. The basic part of the calculus has nine relations: equal (eq), north (n), east (e), west (w), south (s), northwest (nw), northeast (ne), southeast (se) and southwest (sw).

This paper defines a CDC inspired on the formulation given in [19], where spatial objects are points in a two-dimensional space and

the cardinal directions between two objects A and B are defined as the two projections of the straight line from A to B: one on the axis South-North and the other on the axis West-East.

In this paper we assume that each road defines its local cardinal direction system, whereby the direction *"South-North"* goes from the origin of the road towards its end, following the road's centre line. In other words, the *"South-North"* direction between two objects A and B *on the road* are defined as the projection of the line from A to B on the road's centre line. The *"East-West"* direction is defined at every point of the road as the continuous orthogonal line to the tangent of the centre line at that point. Figure 1 shows an example of this local CDC.

In order to make clear that we are not dealing with global cardinal directions (while also taking inspiration of the dynamic nature of a traffic scenes), this paper refers to the directions *going down* and *going up* (the road), instead of resp. *"South"* and *"North"*, and *right–left* instead of *"East"–"West"*.

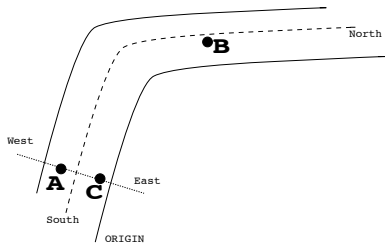


Figure 1. The local cardinal system for roads: A is to the south of B and to the west of C.

2.3 Probabilistic description logics

Description logics (DLs) are fragments of first-order logics originated in the 1970s as a means to provide a formal account of frames and semantic networks. In general terms, description logics are based on *concepts* , which represent sets of individuals (such as Plant or Animal); and *roles*, which denote binary relations between individuals, such as fatherOf or friendOf. Set intersection, union and complement are usual operators found in DLs, as well as some constrained forms of quantification. A key feature of most description logics is that their inference is decidable [1].

In recent years there have been an increasing interest in the combination of probabilistic reasoning and logics (and with description logics in particular) [25, 11, 23]. This combination is not only motivated by pure theoretical interest, but it is very relevant from an application standpoint in order to equip a reasoning system with relational inferences capable of making also probabilistic assessments.

In [18] a number of distinct probabilistic logics were proposed where probabilities were defined over subsets of domain elements. These logics, however, have difficulties to handle probabilistic assertions over individuals, as statistical information over the domain does not preclude information about individuals (this is known as the *direct inference* problem [2]). This problem is also present in various formalisms, as summarised in [11].

The direct inference problem is solved in [12] by adopting probabilities only on assertions. An alternative way around the direct inference problem is to assign probabilities to subsets of interpretations, rather than subsets of the domain. This solution was assumed in [9, 10] and also in CR$\mathcal{ALC}$.

2.4 CR$\mathcal{ALC}$: credal $\mathcal{ALC}$

This section summarizes CR$\mathcal{ALC}$, a probabilistic extension of the popular $\mathcal{ALC}$ description logic [31]. CR$\mathcal{ALC}$ inherits all the constructs of $\mathcal{ALC}$, summarised as follows. The basic vocabulary of $\mathcal{ALC}$ contains individuals, concepts (sets of individuals) and roles (binary relations of individuals). Given two concepts C and D, they can be combined to form new concepts from *conjunction* ($C \sqcap D$), *disjunction* ($C \sqcup D$), *negation* ($\neg C$), *existential* restriction ($\exists r.C$) and *value restriction* ($\forall r.C$).

A concept *inclusion*, $C \sqsubseteq D$, indicates that the concept D contains the concept C and a *definition*, $C \equiv D$, indicates that the concepts C and D are identical. The set of inclusions and definitions constitute a *terminology*. In general, a terminology is constrained to be acyclic, i.e., no concept can refer to itself in inclusions or definitions.

The semantics of $\mathcal{ALC}$ is defined by a domain $\mathcal{D}$ and an interpretation function $\mathcal{I}$, which maps: each individual to a domain element; each concept to a sub-set of $\mathcal{D}$; and, each role to a binary relation $\mathcal{D} \times \mathcal{D}$, such that the following holds: $\mathcal{I}(C \sqcap D) = \mathcal{I}(C) \cap \mathcal{I}(D)$; $\mathcal{I}(C \sqcup D) = \mathcal{I}(C) \cup \mathcal{I}(D)$; $\mathcal{I}(\neg C) = \mathcal{D} \backslash \mathcal{I}(C)$; $\mathcal{I}(\exists r.C) = \{x \in \mathcal{D} | \exists y : (x, y) \in \mathcal{I}(r) \wedge y \in \mathcal{I}(C)\}$; $\mathcal{I}(\forall r.C) = \{x \in \mathcal{D} | \forall y : (x, y) \in \mathcal{I}(r) \rightarrow y \in \mathcal{I}(C)\}$.

An inclusion $C \sqsubseteq D$ holds if and only if $\mathcal{I}(C) \subseteq \mathcal{I}(D)$, and a definition $C \equiv D$ holds if and only if $\mathcal{I}(D) = \mathcal{I}(D)$. For instance, $\mathsf{C} \sqsubseteq (\exists\ \mathsf{hasSibling.Woman}) \sqcap (\forall \mathsf{buys.(Fish} \sqcup \mathsf{Fruit))}$ indicates that C contains only individuals who have sisters and buy fruits or fishes.

In the probabilistic version of $\mathcal{ALC}$ (CR$\mathcal{ALC}$), on the left hand side of inclusions/definitions only concepts may appear. Given a concept name C, a concept D and a role name r, the following probabilistic assessments are possible:

$$P(C) \in [\underline{\alpha}, \overline{\alpha}], \quad (1)$$

$$P(C|D) \in [\underline{\alpha}, \overline{\alpha}], \quad (2)$$

$$P(r) \in [\underline{\beta}, \overline{\beta}]. \quad (3)$$

We write $P(C|D) = \underline{\alpha}$ when $\underline{\alpha} = \overline{\alpha}$, $P(C|D) \geq \underline{\alpha}$ when $\underline{\alpha} < \overline{\alpha} = 1$, and so on.

In order to guarantee acyclicity, no concept is allowed to use itself in deterministic (or probabilistic) inclusions and definitions.

The semantics of CR$\mathcal{ALC}$ is based on probabilities over interpretations so that the direct inference problem can be avoided. In other words, probabilistic values are assigned to the set of all interpretations. The semantics of Formula (1) is, thus: for any $x \in \mathcal{D}$, the probability that x belongs to the interpretation of C is in $[\underline{\alpha}, \overline{\alpha}]$. That is,

$$\forall x \in \mathcal{D} : P\Big(\Big\{ \mathcal{I} : x \in \mathcal{I}(C) \Big\} \Big) \in [\underline{\alpha}, \overline{\alpha}].$$

Informally, the semantics can be represented as:

$$\forall x \in \mathcal{D} : P(C(x)) \in [\underline{\alpha}, \overline{\alpha}].$$

The semantics of Expressions (2) and (3) is then:

$$\forall\ x \in \mathcal{D}\ :\ P(C(x)|D(x)) \in [\underline{\alpha}, \overline{\alpha}],$$
$$\forall\ (x, y) \in \mathcal{D} \times \mathcal{D}\ :\ P(r(x, y)) \in [\underline{\beta}, \overline{\beta}].$$

Given a finite domain, a set of sentences in CR$\mathcal{ALC}$ specifies probabilities over all instantiated concepts and roles. In general, a set of probabilities is specified by a set of sentences (for example, one

may specify $P(C) \in [0.2, 0.3]$, allowing all probability values in an interval). A few assumptions guarantee that a single probability distribution is specified by a set of sentences: unique-names, point-probabilities on assessments, rigidity of names [6]. Under these assumptions, a finite domain and a set of sentences specify a unique Bayesian network over the instantiated concepts and roles. To compute the probability of a particular instantiated concept or role, one can generate this Bayesian network and then perform probabilistic inference in the network. Because the domains we deal with in this paper are relatively small, we follow this propositionalisation strategy in our examples. For large domains it may be impractical to explicitly generate a Bayesian network. In this case, approximate algorithms can be used and, in particular, algorithms based on variational methods have been developed with success [6].

3 CR$\mathcal{ALC}$ encoding of a traffic scenario

This section presents a formalisation in CR$\mathcal{ALC}$ of a road traffic domain. Incomplete sensor data and domain knowledge in the form of road building regulations are used to solve functional lane recognition tasks. Let *ego-road* and *ego-lane* denote, respectively, the road and the particular lane on which a vehicle is driving.

The scenario we represent consists of a road where each lane goes either *up* or *down*. Dividing every pair of adjacent lanes is either a *dashed divider* or a *solid divider*. The scenario also contains an experimental vehicle equipped with three on-board sensors: a digital map, a GPS and a video camera. The task of the formalism is to estimate two functional properties of the ego-road using on-board vehicle sensors. First, which lane corresponds to the ego-lane? The answer is one element of the set: $\{1, \ldots, n\}$, where n is the number of lanes in the road. This task is derived from the fact that current differential GPS receivers are able to reliably determine a vehicle's ego-road, but not its ego-lane (e.g. [16]). Second, which driving direction does each lane permit? The answer is either up or down the road, relative to the ego-road's coordinate system.

The functional properties of lanes that are adjacent to the ego-lane are poorly picked out by state-of-the-art vehicle sensors. One reason for that is the narrow field of view of cameras pointing in the driving direction, which causes blind spots over a large portion of the adjacent lanes. Besides, other vehicles frequently occlude relevant image cues, such as divider markings, arrow markings, and traffic signs. Finally, some properties are not explicitly given in the form of symbols but need to be derived from the context by the human driver (e.g. right-handed traffic rules, as assumed in this work).

These observations are reflected by the sensor input available to solve this task.

First, a video-based divider marking recognition is available. Such a sensor recognises lane divider markings on the right of the vehicle and classifies them into either dashed or solid divider lines. Hit and false alarm rate of the recognition task, and the confusion table of the classification task, are given in Tables 1(a) and 1(b), respectively.

Second, a vehicle has a map-matched GPS position that retrieves the current road from a digital map and provides the vehicle's driving direction on that road segment, discretised into "going up" or "going down" relative to the road's coordinate system [16]

Third, a digital navigation map is available, providing classification of the road into either one-way or two-way traffic and an estimate for the number of lanes. Table 1(c) is a confusion table for this classification task.

It is worth pointing out that tables 1(a) and 1(c) are based on comparing the algorithm's outcomes with ground truth [17], whereas the data in Table 1(b) was estimated.

Table 1. Sensor model. In the confusion tables (b) and (c), columns denote ground truth and rows denote estimates.

(a) Video: Divider Recognition

Hit rate	.51
FA rate	.23

(b) Video: Divider Classification

	Solid	Dashed
Solid	.80	.067
Dashed	.20	.933

(c) Digital map: Road Classification

	Oneway	Twoway
Oneway	.99	.01
Twoway	.01	.99

A taxonomy of concepts and roles relevant to the traffic task is now presented (mostly from our previous publication [30]). The concept Lane is defined using two primitives, Up and Down; the concept Divider is defined as the union of DashedDivider and SolidDivider, and Vehicle is either going up (GoingUp) or going down (GoingUp):

$$\mathsf{Lane} \equiv \mathsf{Up} \sqcup \mathsf{Down} \tag{4}$$

$$\mathsf{Divider} \equiv \mathsf{DashedDivider} \sqcup \mathsf{SolidDivider} \tag{5}$$

$$\mathsf{Vehicle} \equiv \mathsf{GoingUp} \sqcup \mathsf{GoingDown} \tag{6}$$

In Formulae (7)–(11) and (13) below we use the abbreviation $\mathsf{disjoint}(\mathsf{t}_1, \mathsf{t}_2, \ldots, \mathsf{t}_n)$ to represent the set of statements about pairwise disjoint terms, i.e., $\mathsf{t}_i \sqsubseteq \neg \mathsf{t}_j \; \forall i, j \in 1, ..., n, i \neq j$.

$$\mathsf{disjoint}(\mathsf{Vehicle}, \mathsf{Divider}, \mathsf{Lane}) \tag{7}$$

$$\mathsf{disjoint}(\mathsf{Up}, \mathsf{Down}) \tag{8}$$

$$\mathsf{disjoint}(\mathsf{DashedDivider}, \mathsf{SolidDivider}) \tag{9}$$

$$\mathsf{disjoint}(\mathsf{GoingUp}, \mathsf{GoingDown}) \tag{10}$$

$$\mathsf{disjoint}(\mathsf{OnOneWayRoad}, \mathsf{OnTwoWayRoad}) \tag{11}$$

The taxonomy of roles consists of CDC relations only. Out of the nine cardinal directions, only three are relevant to the task at hand east (e), west (w) and equal (eq), since the domain does not have cross-roads.

In this paper the cardinal direction "west" is implicit, as it is not directly defined but it is used in some restrictions such as DashedDivider. (this is more efficient than the representation strategy used in our previous publication [30]). Another change from our previous work is that we use a global point of view (bird-eye) with fixed coordinates (north, south, east, west). This simplifies inference through Bayesian networks, as discussed later.

The relation eq represents the fact that a vehicle being located on a particular lane. We have:

$$\mathsf{cdc} \equiv \mathsf{e} \sqcup \mathsf{w} \sqcup \mathsf{eq} \tag{12}$$

$$\mathsf{disjoint}(\mathsf{e}, \mathsf{w}, \mathsf{eq}) \tag{13}$$

A set of hard constraints is now defined on road building regulations. The first two constraints (Formulae (14) and (15)) formalise the semantics of right-handed traffic: to the right of a lane allowing for traffic *going up* the road (with respect to the road's egocentric coordinate system) there must only be lanes allowing for "going up" traffic, and to the left of traffic *going down* the road there must only be "down" lanes. When a vehicle is "going up" in a lane with direction *up*, to its east there is a "solid divider" or to its west there is a "dashed divider"and there is also a lane to its east that is *up*. Similarly, when a vehicle is "going down" in a lane with direction *down*,

to its west there is a "solid divider" or to its west there is a "dashed divider" and there is also a lane to its west that is *down*.

$$\mathsf{GoingUp} \sqsubseteq \exists \mathsf{e}.(\mathsf{SolidDivider} \sqcap \neg \mathsf{Lane}) \sqcup \exists \mathsf{e}.(\mathsf{DashedDivider} \sqcap \mathsf{Up}) \quad (14)$$

$$\mathsf{GoingDown} \sqsubseteq \forall \mathsf{w}.(\mathsf{SolidDivider} \sqcap \neg \mathsf{Lane}) \sqcup \exists \mathsf{w}.(\mathsf{DashedDivider} \sqcap \mathsf{Down}) \quad (15)$$

Formulae (16) and (17) refer to the dividers function, which may be distinct in different countries. A dashed divider divides two lanes with same driving direction, whereas a solid divider either marks the road border or it separates roads with opposing driving directions:

$$\mathsf{DashedDivider} \sqsubseteq (\exists \mathsf{e}.\mathsf{Up} \sqcap \exists \mathsf{w}.\mathsf{Up}) \sqcup (\exists \mathsf{e}.\mathsf{Down} \sqcap \exists \mathsf{w}.\mathsf{Down}) \quad (16)$$

$$\mathsf{SolidDivider} \sqsubseteq (\neg \exists \mathsf{e}.\mathsf{Lane} \sqcap \exists \mathsf{w}.\mathsf{Lane}) \sqcup (\neg \exists \mathsf{w}.\mathsf{Lane} \sqcap \exists \mathsf{e}.\mathsf{Lane}) \sqcup (\exists \mathsf{e}.\mathsf{Up} \sqcap \exists \mathsf{w}.\mathsf{Down}) \sqcup (\exists \mathsf{e}.\mathsf{Down} \sqcap \exists \mathsf{w}.\mathsf{Down}) \quad (17)$$

Finally, the following axiom states that a two-way road has traffic in both directions (Formula (18)).

$$\mathsf{OnTwoWayRoad} \sqsubseteq \exists \mathsf{cdc}.\mathsf{Up} \sqcap \exists \mathsf{cdc}.\mathsf{Down} \quad (18)$$

The probabilities given in Tables 1a-1c can be justified as follows. First, new concepts (prefixed by Sensed) are added as subclasses of Sensor for all probabilistic inputs:

$$\mathsf{Sensor} \equiv \mathsf{SensedOnOneWayRoad} \sqcup \mathsf{SensedOnTwoWayRoad} \sqcup \mathsf{SensedDivider} \quad (19)$$

$$\mathsf{SensedDivider} \equiv \mathsf{SensedDashedDivider} \sqcup \mathsf{SensedSolidDivider} \quad (20)$$

The confusion tables (Tables 1(a)–1(c)) show joint probabilities of an event and its detection by a sensor. These probabilities can be represented as conditional probabilities using the definition of conditional events in terms of their intersections, such as: $P(\text{Solid}_{\text{GT}} \mid \text{Dashed}_{\text{S}}) = P(\text{Solid}_{\text{GT}} \sqcap \text{Dashed}_{\text{S}})\,/P(\text{Dashed}_{\text{S}}) = 0.05/0.75 = 0.07$, where the subscript GT denotes "ground truth" and S stands for "sensed". The sensor model from Table 1b (for instance) can now be elegantly formulated as a set of axioms as follows:

$$P(\mathsf{DashedDivider}|\mathsf{SensedDashedDivider}) = 0.93 \quad (21)$$

$$P(\mathsf{SolidDivider}|\mathsf{SensedDashedDivider}) = 0.07 \quad (22)$$

$$P(\mathsf{DashedDivider}|\mathsf{SensedSolidDivider}) = 0.20 \quad (23)$$

$$P(\mathsf{SolidDivider}|\mathsf{SensedSolidDivider}) = 0.80 \quad (24)$$

An analogue set of axioms is used for confusion Table 1. For recognition tasks the sensor model can be translated as follows:

$$P(\mathsf{Divider}|\mathsf{SensedDivider}) = 1 - \text{false alarm rate} = 0.77 \quad (25)$$

$$P(\mathsf{SensedDivider}|\mathsf{Divider}) = \text{hit rate} = 0.51 \quad (26)$$

With the previous axioms, the $\mathcal{T}$Box is fully specified.

DL is used in this work as a specification language from which a Bayesian description is derived. In the present context, a DL description is used to encode high-level knowledge, such as the permitted driving directions. The use of a DL-based probabilistic logics gives us guarantees concerning expressivity and complexity that are not available when one resorts to full first-order probabilistic logic.

4 Coding and running the scenario

Most axioms presented in the previous section are within the basic definitions of CR$\mathcal{ALC}$. However, the original role hierarchies are not even within the scope of $\mathcal{ALC}$ (and, consequently, not within CR$\mathcal{ALC}$). In our previous publication [30] we circumvented these axioms to construct a TBox with the confines of CR$\mathcal{ALC}$. In this paper we wish to follow a different strategy.

As any TBox in CR$\mathcal{ALC}$ is turned into a Bayesian network upon inference, we have handled hole hierarchies and disjoint concepts within the transformation TBox→Bayesian network. Concepts are translated into nodes of the network in such a way that concepts that are to the left are translated into parents of concepts that are to the right. To handle disjoint concepts, we create or-exclusive nodes that are always set to true in the network. We can then determine whether a lane is *up* or *down* using Formulae (16), (17) and (18). Even though concepts in these formulae are defined using roles, we compute their probabilities by conditioning on concepts GoingUp, GoingDow, OnTwoWayRoad, OnTwoWayRoad. The same procedure is used to determine the kind of "divider" that is related to each lane; that is, which divider is to the east of a lane. In this work we focus on the cardinal direction calculus to determine in which lane the vehicle is located. We can determine $P(eq(vehicle, lane))$ using data in the $\mathcal{A}$Box; for instance, if the sensor indicates a divider to the east of the vehicle, and we can then infer the kind of divider for each lane. Hence we can compute the probability of each lane being an ego-lane.

Given the formalisation presented in Section 3 (and the consideration above), the system generated the Bayesian network represented in Figure[5] 2 , where the nodes in red are observed variables, i.e. sensors' states. A detail of this network is shown in Figure 3, that represents a Bayesian Network for one individual (out of the 5 interconnected nets shown in Figure 2). Besides the information in the $\mathcal{A}$Box, there is evidence in the nodes that represent disjoint concepts (that is, the nodes that encode or-exclusive relations) and nodes that indicate whether a network represents a lane, a vehicle, or a divider.

It is now possible to answer the queries specified in Section 3, which correspond to the following:

1. $\text{argmax}_{\mathsf{li}} P((v : \mathsf{li} : eq))$, i.e. li is the lane with maximum probability of being the vehicle's (v) ego-lane .
2. $\forall i : P(\mathsf{li} : \mathsf{Up})$, i.e. for each lane li, the probability of being a Up lane.

Consulting the network in Figure 2 for all of the eight possible states of the three sensors, we obtain the answers presented in Tables 2 and 3 for the queries 1 and 2 respectively (we employ the abbreviations $STWR$ for SensedOnTwoWayRoad and SDD for SensedDashedDivider).

Table 2 shows the most probable lane on which the vehicle v is driving ($\text{argmax}_{li} P((v, \mathsf{li} : eq))$), given the evidences, represented on the first three columns. The first line of the table, for instance, represents the state where the GPS obtained GoingDown, the map sensed that the vehicle was on a *one way road* and the vision system sensed a *solid divider*. Given these evidences the node li with the highest probability (on the network of Figure 2) was l1. This case is shown in Figure 4(c). On the second line of Table 2, however, the GPS and the map sensor remained in the state just described, but the vision sensed a dashed divider (instead of a solid one). In this case, there were two hypotheses with equal probabilities[6]: l2 and l3 (as repre-

[5] For colour image, please refer to the electronic version of this paper.
[6] They differ on the third decimal digit.

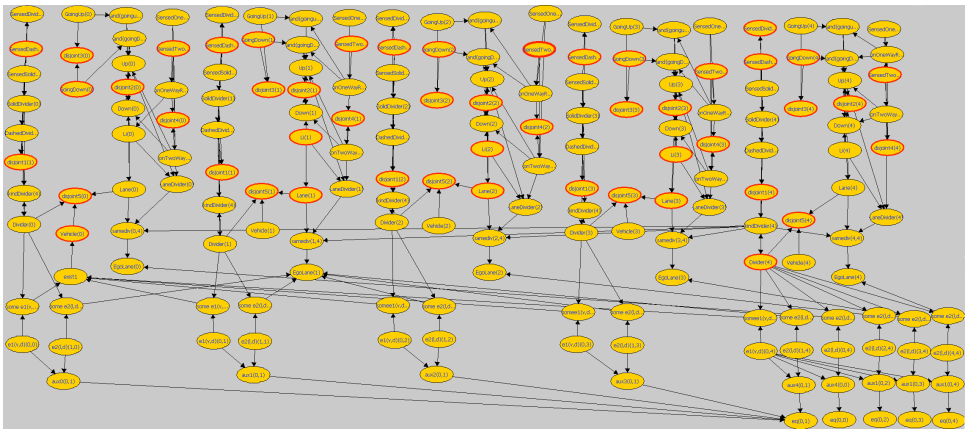


Figure 2. Bayesian Network representing a traffic domain.

sented in Figures 4(a) and 4(b) respectively). The remainder cases on Table 2 are analogous.

Table 3 represents the probabilities for each of the li lanes to be a Down lane, given the evidence on the first three columns. The probability of Up is the complement of the values stated in the table. Take for instance the first line, the highest probability for l1, l2 and l3 is Down, which is consistent with the evidence GoingDown (for the vehicle) and SensedOnOneWayRoad. Similarly for the remainder sensor states represented in the table.

Table 2. Answer to query 1: the probability on the *ego-lane* given the evidence A (expressed on the first three columns).

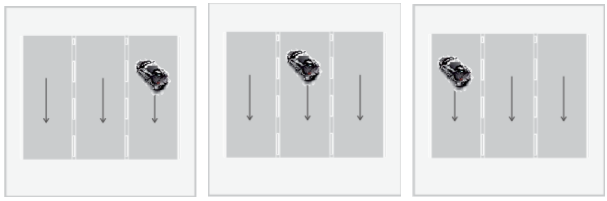
GPS	map	video	$\text{argmax}_{li} P((v, \text{li} : eq \mid A))$
GoingUP	STWR	SDD	
0	0	0	l1
0	0	1	l2 ∨ l3
0	1	0	l2 ∨ l3
0	1	1	l3
1	0	0	l1
1	0	1	l2 ∨ l3
1	1	0	l1
1	1	1	l2

Table 3. Answer to query 2: the probability for the *lane's driving direction* given the evidence A (expressed on the first three columns).

GPS	map	video	l1	l2	l3
GoingUP	STWR	SDD	$P(\text{l1:Down} \mid A)$	$P(\text{l2:Down} \mid A)$	$P(\text{l3:Down} \mid A)$
0	0	0	0.98	0.99	1.00
0	0	1	0.98	0.99	1.00
0	1	0	0.50	0.5	1.00
0	1	1	0.05	0.5	1.00
1	0	0	0.00	0.00	0.00
1	0	1	0.00	0.00	0.00
1	1	0	0.00	0.50	0.99
1	1	1	0.00	0.50	0.99

In this work the queries presented in Table 2 and 3 were run offline. However, for small size scenarios they could be executed in real time.

A network is generated for each individual; in total we have five individuals: 3 lanes, 1 vehicle and 1 divider. The generation of the resulting network is a non-trivial task that is obviously simplified by the use of a probabilistic description language.



(a) (b) (c)

Figure 4. Three traffic scenarios.

5 Concluding remarks

In this paper we have extended our previous efforts on encoding spatial domains with a probabilistic description logic. We still employed CR$\mathcal{ALC}$ as the basic description language but added features that affect the translation of terminologies into Bayesian networks; namely, we added the ability to handle the role hierarchies and the disjoint concepts that appear in spatial domains. The development of more general inference algorithms is the object of our future work.

Overall, the representation of qualitative spatial reasoning with description logics is a recent endeavour [8]. The major difficulty of this task, which we still face in our work, is the representation of transitive relations. Decidability of description logic representations of spatial formalisms were investigated in [20, 8] for a combination of $\mathcal{ALC}$ with a decidable constraint system (called $\mathcal{ALC}(C)$, where C is the constraint system). The investigation of probabilistic extensions of $\mathcal{ALC}(C)$, and whether decidability is maintained, is an interesting issue for future research.

Acknowledgements

This work has been partially supported by Fapesp Project 2008/03995-5 (LogProb). Paulo Santos acknowledges support from CAPES and CNPq, Brazil. Valquiria Fenelon is a graduate student sponsored by CAPES, Brazil. Fabio Cozman acknowledges support from CNPq, Brazil.

REFERENCES

[1] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, *Description Logic Handbook*, Cambridge University Press, 2002.

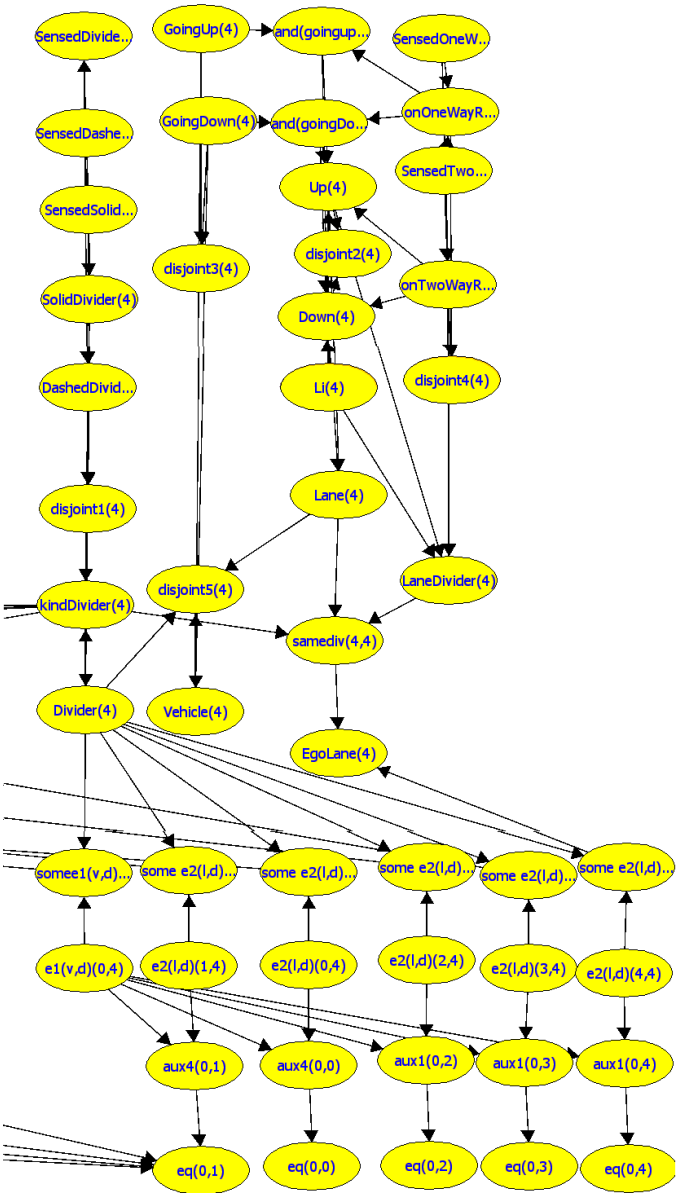


Figure 3. Detail of the Bayesian Network shown in Figure 2.

[2] F. Bacchus, A. Grove, J. Y. Halpern, and D. Koller, 'From statistical knowledge bases to degrees of belief', *Artificial Intelligence*, **87**, 75–143, (1996).

[3] B. Bennett, A. Cohn, and D. Magee, 'Enforcing global spatio-temporal consistency to enhance reliability of moving object tracking and classification', *Künstliche Intelligenz*, **2**, 32–35, (2005).

[4] A. G. Cohn and S. M. Hazarika, 'Qualitative spatial representation and reasoning: An overview', *Fundamenta Informaticae*, **46**(1-2), 1–29, (2001).

[5] A. G. Cohn and J. Renz, 'Qualitative spatial representation and reasoning', in *Handbook of Knowledge Representation*, 551–596, Elsevier, (2008).

[6] F. G. Cozman and R. Polastro, 'Complexity analysis and variational inference for interpretation-based probabilistic description logics', in *Proc. of the Conference on Uncertainty in Artificial Intelligence*, (2009).

[7] F. G. Cozman and R. B. Polastro, 'Loopy propagation in a probabilistic description logic', in *SUM*, pp. 120–133, (2008).

[8] M. Cristani and N. Gabrielli, 'Practical issues of description logics for spatial reasoning', in *Proc. of the AAAI Spring Symposium Benchmarking of Qualitative Spatial and Temporal Reasoning Systems*, pp. 5–10, (2009).

[9] P. C. G. da Costa and K. B. Laskey, 'Of Klingons and starships: Bayesian logic for the 23rd century', in *Conference on Uncertainty in Artificial Intelligence*, (2005).

[10] C. D'Amato, N. Fanizzi, and T. Lukasiewicz, 'Tractable reasoning with Bayesian description logics', in *SUM*, pp. 146–159, Berlin, Heidelberg, (2008). Springer-Verlag.

[11] C. P. de Campos, F. G. Cozman, and J. E. O. Luna, 'Assembling a consistent set of sentences in relational probabilistic logic with stochastic independence', *Journal of Applied Logic: Combining Probability and Logic*, **7**, 137–154, (2009).

[12] M. Dürig and T. Studer, 'Probabilistic ABox reasoning: preliminary results', in *Description Logics*, pp. 104–111, (2005).

[13] J. Fernyhough, A. G. Cohn, and D. C. Hogg, 'Constructing qualitative event models automatically from video input', *Image and Vision Computing*, **18**, 81–103, (2000).

[14] A. U. Frank, 'Qualitative spatial reasoning: Cardinal directions as an example', *International Journal of Geographical Information Science*, **10**(3), 269–290, (1996).

[15] S. M. Hazarika and A. G. Cohn, 'Abducing qualitative spatio-temporal histories from partial observations', in *Proc. of KR*, pp. 14–25, Toulouse, France, (2002).

[16] B. Hummel, *Dynamic and Mobile GIS: Investigating Changes in Space and Time*, chapter Map Matching for Vehicle Guidance, CRC Press, 2006.

[17] B. Hummel, *Description Logic for Intersection Understanding at the Example of Urban Road Intersections*, Phd Thesis, April 2009.

[18] M. Jaeger, 'Probabilistic reasoning in terminological logics', in *Principles of Knowledge Representation (KR)*, pp. 461–472, (1994).

[19] G. Ligozat, 'Reasoning about cardinal directions', *J. Vis. Lang. Comput.*, **9**(1), 23–44, (1998).

[20] C. Lutz and M. Milicic, 'A tableau algorithm for DLs with concrete domains and GCIs', *Journal of Automated Reasoning*, **38**(1–3), 227–259, (2007).

[21] T. Matsuyama and V. S. Hwang, *SIGMA: A Knowledge-Based Image Understanding System*, Plenum Press, New York, U.S., 1990.

[22] J. C. McCall and M. M. Trivedi, 'Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation', *IEEE Transactions on Intelligent Transportation Systems*, **7:1**, 20–37, (2006).

[23] B. Milch and S. Russell, 'First-order probabilistic languages: into the unknown', in *International Conference on Inductive Logic Programming*, (2007).

[24] B. Neumann and R. Möller, 'On scene interpretation with description logics', *Image Vision Comput.*, **26**(1), 82–101, (2008).

[25] A. Nikolov, V. Uren, E. Motta, and A. de Roeck, 'Using the Dempster-Shafer theory of evidence to resolve ABox inconsistencies', in *Workshop on Uncertainty Reasoning for the Semantic Web*, (2007).

[26] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[27] D. Randell, M. Witkowski, and M. Shanahan, 'From images to bodies: Modeling and exploiting spatial occlusion and motion parallax', in *Proc. of IJCAI*, pp. 57–63, Seattle, U.S., (2001).

[28] R. Reiter and A. Mackworth, 'A logical framework for depiction and image interpretation', *Artificial Intelligence*, **41**(2), 125–155, (1989).

[29] P. Santos, H. Dee, and V. Fenelon, 'Qualitative robot localisation using information from cast shadows', in *Proc. of IEEE International Conference on Robotics and Automation*, Kobe, Japan, (2009).

[30] Paulo E. Santos, Britta Hummel, Valquiria Fenelon Pereira, and Fabio G. Cozman, 'Probabilistic logic encoding of spatial domains', in *First International Workshop on Uncertainty in Description Logics (UniDL)*, (2010).

[31] M. Schmidt-Schauss and G. Smolka, 'Attributive concept descriptions with complements', *Artificial Intelligence*, **48**, 1–26, (1991).

Agent Control by Adaptive Neighborhoods

Frank Dylla and **Arne Kreutzmann**[1]

Abstract. Autonomous agents, which are supposed to coexist with humans in the future, must be aware of every-day regulations, e.g. traffic regulations or rules of politeness in crowded places. In this paper we show how qualitative representations can be applied for the formalization and execution of agent behavior. Previous work has shown that conceptual neighborhood structures and its refinement to action-augmented conceptual neighborhood structures are suited for defining agent behavior on an abstract level. Nevertheless, these structures are limited wrt. the actual control of agents. We propose an approach, called *adaptive conceptual neighborhoods*, to overcome these problems. Conceptual neighborhood structures are refined on-the-fly for a situation at hand by means of Monte-Carlo simulation.

1 INTRODUCTION

A considerable part of everyday human activities is guided by regulations, for example, regulations on how to behave in traffic scenarios, recommendations on how to use escalators, rules on how to enter subways and buses, or rules of politeness at bottlenecks. Most of these rules have in common that they are usually formulated in natural language and hence, extensively use *qualitative terms* to describe spatial situations and actions. For example, in traffic laws qualitative concepts are used to describe relevant situations (e.g. *ahead* or *left*) and also the "correct" behavior of agents (e.g. *go straight* or *turn left*) in these situations. Another feature is that most of the rules depend on the agent's *role* in a particular situation. What an agent is allowed to do, may depend on whether he is a pedestrian or on the kind of vehicle she is using.

Representations of rule-compliant behavior, of course, are not limited to navigation. Examples of rule sets guiding the behavior of agents can also be found in sports, games, expert recommendation systems, and so on. Rule sets need to be made explicit and be formalized at different stages when artificial agents or multi-agent systems are specified or implemented. First, rules can be used to specify the desired behavior of an artificial agent (for instance a mobile robot or an autonomous vehicle) such that an implemented system can be tested against these specifications. Rules may also be used to actively control an artificial agent, for example, when we wish to restrict possible trajectories of a mobile system. Formal encodings of rules are also crucial for implementing control systems that observe and judge the behavior of other agents. Finally, rule sets need to be formalized in order to evaluate them according to given criteria, to find gaps, inconsistencies, or deadlocks. For instance, if a rule set describes how *two* agents have to behave in specific situations, one could investigate how this rule set would perform in more complex situations involving more than two agents: Is the rule set still sound in the sense that its intentions (e.g., collision avoidance) are met if all agents act in compliance with the rules? And, is the rule set complete in the sense that it covers all possible situations? In this paper we are concerned with the aspect of how qualitative representations can be used for formalizing agent behavior on a high level of abstraction without explicit need to care about active control of agents, e.g. wether an agent has to turn by 10 or 15 degrees.

Qualitative spatial representation and reasoning (QSR) deals with spatial terms as used in the rules mentioned above. QSR investigates how continuous properties of the world can be represented by a finite set of relations such that predictions about entities can be derived, even if only incomplete knowledge is available [3]. With a qualitative approach one is able to abstract from physical details while formulating and applying regulation specific behavior. Additionally, qualitative approaches are considered to be closer to how humans deal with commonsense knowledge compared to quantitative approaches [21]. Therefore, QSR approaches are well suited for representing regulations in natural language.

A *qualitative calculus* consists of a finite set of relations, e.g. $\{<, =, >\}$, and a set of operations to draw conclusions. Freksa introduced the notion of *conceptual neighborhood* (CN) for qualitative calculi [9]. Two relations are neighbored, if they can be transformed into each other without resulting in a third relation in between, e.g for two points on a line $<$ and $>$ are not neighbored as by continuous transformation relation $=$ must be traversed in between. Thus, wrt. agent control an edge in the CN structures reflects that there exist some actions such that the changeover occurs, but nothing is said about how the actions look like. Therefore, in previous work we present action-augmented CN ($a\mathcal{CN}$) where we extend CN by the notion of actions for the agents involved [5, 8]. Because of its adaptable granularity and its high expressiveness we applied the Oriented Point Relation Algebra ($\mathcal{OPRA}_m$) $a\mathcal{CN}$ structure in the domain of rule-compliant and collision-free vessel navigation [6, 7, 24, 25]. Nevertheless, other orientation calculi could be used as well, for example, due to a common reference frame induced by junctions and the limited space considered, Cardinal Directions are well suited to model right of way rules in road traffic [20]. Although, satisfying behavior was generated in most cases, collisions may still appear as, e.g. spatially extended agents are abstracted to points and $a\mathcal{CN}$ abstracts away from the temporal dimension, i.e. the structure does not reflect *when* a change occurs. Additionally, projections into the future are problematic as the result of an action may be ambiguous wrt. the resulting relation.

To overcome these problems, we introduce *adaptive conceptual neighborhoods* ($ad\mathcal{CN}$). The neighborhood graph is approximated by Monte-Carlo simulations in fixed time steps by means of the situation at hand, including knowledge (and noise) about the concrete situation and assumptions about the kinematics of the agents involved. Due to the discrete nature of this method, the graph may contain changeovers between not neighboring relations. These gaps are filled

[1] University of Bremen, Germany, email: {dylla,kreutzma}@sfbtr8.uni-bremen.de

by means of the $a\mathcal{CN}$ structure. The result is a graph with directed edges between relations, where edges are labeled with actions and transition probabilities. Such a structure is better suited for planning and executing the behavior of an agent than a simple $a\mathcal{CN}$ structure.

The paper is structured as follows. In Section 2 we introduce the field of QSR, including the Oriented Point Relation Algebra ($\mathcal{OPRA}_m$) and the notion of conceptual neighborhood. In Section 3 we present the concept of adaptive neighborhoods and how they are derived by Monte-Carlo approximation. In Section 4 we apply $ad\mathcal{CNG}$s in the SailAway system, a demonstrator for vessel navigation according to official collision regulations. We finalize the paper with concluding remarks in Section 5.

2 Qualitative spatial representation and reasoning

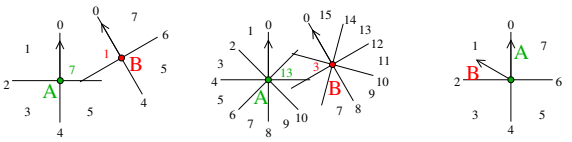
Although the world is infinitely complex and our knowledge of the world is limited, i.e. incomplete, biological systems, especially humans, function quite well within this world without understanding it completely [13]. Humans understand physical mechanisms such as bathtubs, indoor or outdoor navigation, bicycling, microwave ovens, and so on. Qualitative Reasoning (QR) is concerned with capturing such everyday commonsense knowledge of the physical world with a limited set of symbols and allows for dealing with the knowledge without numerical values [3, 21]. The subfield of qualitative reasoning that is concerned with representations of space is called *Qualitative Spatial Reasoning* (QSR).

In the remainder of this section we introduce QSR in more detail. First, the essence of qualitative spatial representations in general is presented, followed by a brief introduction to the Oriented Point Relation Algebra ($\mathcal{OPRA}_m$), the calculus used in the remainder of this paper. After explaining the notion of conceptual neighborhood we finally, introduce the qualitative reasoning toolbox SparQ we apply in our system.

2.1 Qualitative spatial representations

A qualitative spatial description captures distinctions between objects that make an important qualitative difference but ignores others. In general, objects are abstracted to geometric primitives, e.g. points, lines, or regions. The ability to focus on the important distinctions and ignore the unimportant ones is an excellent way to cope with incomplete knowledge [13]. Cohn & Hazarika summarize that the essence of qualitative spatial reasoning is to find ways to represent continuous properties of the world, also called continuities, by discrete systems of symbols, i.e. a finite vocabulary [3]. These symbols describe the relationships between objects in a specific domain. Therefore, they are called *relations*. The domain is given by the set of objects, i.e. geometric primitives, considered. Relations may describe different aspects of space as topology (e.g. 'outside' or 'inside'), orientation (e.g. 'right', 'left', 'ahead' or 'behind'), location (e.g. 'here', 'on the market place', or 'Bremen'), distance (e.g. 'close' or 'distant'), size (e.g. 'small' or 'large'), or shape (e.g. 'cube', 'circle', etc.).

A complete model for a certain domain is called a *qualitative calculus*. It consists of the set of relations between objects from this domain and the operations defined on these relations. Among set theoretic operations, the two most important operations are a shift in perspective (*converse* operation) and the integration of local knowledge of two overlapping sets of objects into survey knowledge (*composition* operation). Simplified, if we know, for example, that object B is right of A a change in perspective from A to B reveals that A is left of B (converse). From knowing that B is left of A and C is left of B we can infer that C is also left of A (composition).



(a) $\vec{A}\ {}_2\angle^1_7\ \vec{B}$ (b) $\vec{A}\ {}_4\angle^3_{13}\ \vec{B}$ (c) $\vec{A}\ {}_2\angle 1\ \vec{B}$

Figure 1. Two oriented points related at different granularities.

Many different qualitative spatial calculi, which can be roughly classified as *topological* and *positional* calculi [11], have been presented in recent years. As collision detection can be reduced to relative orientation knowledge over time[2] we only need to consider the aspect of relative orientation. We choose the Oriented Point Relation Algebra ($\mathcal{OPRA}_m$) [18] as its granularity, i.e. its number of base relations, can be adapted regarding the problem at hand, and other relative orientation calculi can be represented in terms of $\mathcal{OPRA}_m$ relations [5].

2.2 The oriented point relation algebra: $\mathcal{OPRA}_m$

The $\mathcal{OPRA}_m$ calculus [17, 18] relates two oriented points $\vec{A}$ and $\vec{B}$ (points in the plane with an additional direction parameter) and describes their relative orientation towards each other. $\mathcal{OPRA}_m$ is well suited for dealing with objects that have an intrinsic front or move in a particular direction and can be abstracted as points. The granularity factor $m > 0 \in \mathbb{N}$ determines the number of distinguished relations. For each of the points, m lines are used to partition the plane into $2m$ planar and $2m$ linear regions. Figure 1 shows the partitions for the cases $m = 2$ (a) and $m = 4$ (b). The orientation of the two points is depicted by the arrows starting at $\vec{A}$ and $\vec{B}$, respectively. The regions are numbered from 0 to $(4m - 1)$, region 0 always coincides with the orientation of the point. An $\mathcal{OPRA}_m$ relation $rel_{\mathcal{OPRA}_m}$ consist of pairs (reg_i, reg_j) where reg_i is the region of $\vec{A}$ in which $\vec{B}$ falls into, while reg_j is the region of $\vec{B}$ that contains $\vec{A}$. They are usually written as $\vec{A}\ {}_m\angle^j_i\ \vec{B}$ with $i, j \in \mathcal{Z}_{4m}$[3]. Thus, the examples in Figure 1 depict the relations $\vec{A}\ {}_2\angle^1_7\ \vec{B}$ and $\vec{A}\ {}_4\angle^3_{13}\ \vec{B}$. Additional relations describe situations in which both oriented points coincide. In these cases, the relation is determined by the region reg of $\vec{A}$ the orientation arrow of $\vec{B}$ falls into as illustrated in Figure 1(c). These relations are written as $\vec{A}\ {}_2\angle reg\ \vec{B}$ ($\vec{A}\ {}_2\angle 1\ \vec{B}$ in the example).

2.3 Conceptual neighborhood

Conceptual neighborhood extends static qualitative representations by interrelating the discrete set of base relations by the temporal aspect of transformation of the basic entities: *Two spatial relations of a qualitative spatial calculus are* ***conceptual neighbors****, if they can be continuously transformed into each other without resulting in a third relation in between [9].*

The definition of conceptual neighborhood originates from work on time intervals and thus, only continuous transformations on intervals (shortening, lengthening, and shifting) are considered. Later, the definition was also interpreted spatially. For moving objects we

[2] The principle of *constant bearing*: if two straight moving objects are approaching each other and their relative angle does not change over time, a collision will occur.

[3] $\mathcal{Z}_{4m}$ defines a cyclic group with $4m$ elements.

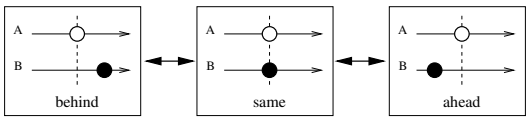


Figure 2. The relations of the Point Calculus arranged as a conceptual neighborhood graph $\mathcal{CNG}$.

can say two relations are conceptual neighbors, if continuous motion of the objects can cause an immediate transition between these two relations. Similar considerations are proposed by Ligozat [15] and Galton [12].

For instance, imagine two boats in a race. The possible relations in our representation are *behind* ($<$), *same* ($=$), and *ahead* ($>$). The vessels are able to move forward with changing speed. Assume A is *behind* B. Observing the scene a few minutes later shows that now A is *ahead* of B. But this is not the whole story. Assuming continuous motion it is not possible for A to overtake B without passing B at some time, i.e. being at the *same* level. Therefore, *ahead* and *behind* are conceptual neighbors of *same*, but not *ahead* and *behind*.

The conceptual neighborhood relation, denoted by $\sim$, between the base relations $\mathcal{BR}$ of a qualitative calculus is often described in form of a *conceptual neighborhood graph* $\mathcal{CNG} = \langle \mathcal{BR}, \sim \rangle$ as illustrated in Fig. 2 for the Point Algebra. A set of base relations which is connected in the $\mathcal{CNG}$ is called a *conceptual neighborhood*. For convenience, we introduce a function $\mathtt{cn} : \mathcal{BR} \rightarrow 2^{\mathcal{BR}}$ which yields all conceptual neighbors for a given base relation $b \in \mathcal{BR} : \mathtt{cn}(b) = \{b' | b \sim b'\}$.

The term *continuous transformation* is a central concept in the definition of conceptual neighborhood. Detailed investigations on different aspects of continuity are, for example, presented in [1, 12, 19]. Conceptual neighborhood on the qualitative level corresponds to continuity on the geometric or physical level: continuous processes map onto identical or neighboring classes of descriptions [10]. However, the term *continuous* with regard to transformations needs a grounding in spatial change over time. We define continuous transformation as continuous motion of a moving agent, e.g. a robot R. This can be described by the function $pos(R) : T \rightarrow P$, where T is a set of times and P is a set of possible positions of R. Now assuming T and P being topological spaces, the motion of R is continuous, if the function $pos(R)$ is continuous [12].

Conceptual neighborhoods and neighborhood-based reasoning are suitable models for how the world could evolve in terms of transitions between qualitative relations. Nevertheless, for tasks like navigation, action planning, as well as behavior monitoring and its interpretation, it is crucial that the $\mathcal{CNG}$s reflect the properties and capabilities of the represented agents so that neighborhood induces direct reachability in the physical world. In its general form a $\mathcal{CNG}$ represents arbitrary dynamics of the objects involved. If two objects are in relation r then the conceptual neighborhood only defines that for any $r' \in \mathtt{cn}(r)$ there exists some action causing a transition from r to r'. Many of these changes are not applicable at all or are most unlikely to occur considering agents or robots in the real world. Additionally, it is not represented which actions have to be executed to achieve a certain transition. Thus, conceptual neighborhood in its original definition is not sufficient from an agent control perspective.

For this reason the notion of conceptual neighborhood was extended to *action-augmented conceptual neighborhood* ($a\mathcal{CN}$) including an explicit representation for the actions causing a change in the relation between two objects [5].

Overall, three main aspects affect the *action-augmented conceptual neighborhood graph* ($a\mathcal{CNG}$) for a given spatial calculus in the context of robot navigation:1) the robot kinematics (motion capabilities), 2) whether the objects may move simultaneously, and 3) whether objects may coincide in position or not (superposition). For example, if we reconsider the boat race example with 'A is *behind* B' and assume that B is definitely faster than A, it will never happen that 'A is on the *same* level as B'. For representing conceptual neighbors of a relation regarding specific actions we introduce a refined neighborhood function $\mathtt{acn}$:

$$\mathtt{acn}(r, a_1, a_2) = \{r' | O_1\ r\ O_2 \wedge r \sim r' \wedge\ r' \text{ is possible, if}$$
$$[\text{ object } O_1 \text{ performs action } a_1 \wedge \text{object } O_2 \text{ performs action } a_2]\},$$

with relation r is the current relation between the two objects O_1 and O_2. The $a\mathcal{CNG}$ is then defined as $a\mathcal{CNG} = \langle \mathcal{BR}, E, g \rangle$ with $e = (v, w) \in E$ if $w \in \mathtt{acn}(v, a_1, a_2)$. Additionally, the labeling function g gives the actions that can cause the transition: $g(e) = (a_1, a_2)$. Considering actions ($\top$) which deliver arbitrary motion behavior for both objects $\mathtt{acn}(r, \top_1, \top_2)$ is equal to $\mathtt{cn}(r)$.

2.4 SparQ

The qualitative spatial reasoning toolbox SparQ [23] provides a platform for making calculi and reasoning techniques developed in the QSR community available. SparQ supports diverse qualitative reasoning tasks for binary and ternary calculi, to name the most common ones: qualification, computing with relations (e.g. composition, converse), and constraint-based reasoning. The focus is to provide an implementation of QSR techniques that is tailored towards the needs of application developers. In its current version, SparQ offers a broad variety of qualitative calculi, among them $\mathcal{OPRA}_m$, other point- and line-based calculi, as well as the region-based calculus RCC-8. The toolbox can be integrated easily into applications through a TCP/IP interface.

3 ADAPTIVE NEIGHBORHOODS

The $a\mathcal{CNG}$ introduced above defines which transitions are possible if actions $a_i\ i \in 1, 2$ are executed, but makes no claim about *when* these changes will take place. Considering specific scenarios reveals that the concrete neighborhood structure depends highly on application-specific parameters and the current situation. Starting with a situation S_0 at time point t_0 and constraining the $a\mathcal{CNG}$ to the transitions that are possible at $t_0 + \Delta_t$ results in an *adaptive conceptual neighborhood graph* ($ad\mathcal{CNG}$). Doing so consecutively for more than one step states in which time interval which relations are possible. Therefore the resulting graph grows along the time axis. Consequently the distance is implicitly given, but rather in the amount of time left (i.e. the number of steps left) than as a numerical value or qualitative relation.

If perfect knowledge about the actions, action's parameters respectively, and the resulting movements are known, one could calculate the exact time point when a transition in the $a\mathcal{CNG}$ would take place. But in most cases these parameters are unknown. Additionally, the time complexity for finding a numerical solution is very high. Therefore, we apply Markov chain Monte-Carlo (MCMC) methods to simulate the situation at hand, resulting in $ad\mathcal{CNG}^{\star}$, which is an approximation of the true $ad\mathcal{CNG}$.

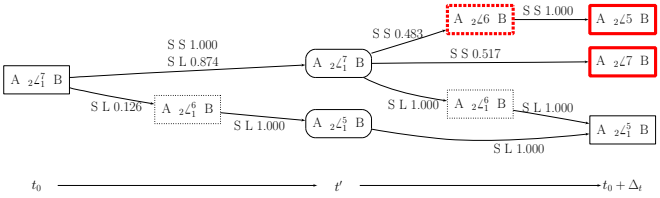


Figure 3. An example of an adaptive conceptual neighborhood graph with two vehicles A and B, where A only goes *straight* (S) and B can also turn left (L), resulting in the two possible action combinations as shown in the above figure. The boxes represent the states at certain time points; the rounded ones are at an intermediate step t'; the dotted represent insertions from the $a\mathcal{CNG}$; the bold red ones indicate collisions.

Monte-Carlo approximation of neighborhood graphs A qualitative action such as *turn right*, is generally modeled as a fixed turn rate, but due to environmental effects like rough ground or limitations in the accuracy of the motion control, the effects may differ. So instead of using a single fixed number for each action, the actions are defined given a closed set of possible parameters, e.g. intervals.

Given a uniform distribution over the parameter set for an action, a number of samples is drawn and simulated for a fixed amount of time Δ_t, the duration of an action. Depending on the speed of change and the duration of the actions a number of intermediate steps are calculated as well, so that application *critical* situation can be identified. At each step all samples with the same qualitative scene description are composed into a single node in the graph, resulting in all possible qualitative descriptions of the scene at that time point. The connections from the nodes at one time point to the ones at the next are given by the transitions of the samples. Each edge therefore describes which actions (including the parameters with an estimated transition probability) result in which next qualitative description.

When using snapshots like above, certain configurations are likely to be missed, e.g. $\mathcal{OPRA}_m$ relations of linear regions, because they only hold for a very small amount of time. In some cases they are even instantaneous. This may result in a graph which includes transitions of relations that are not conceptual neighbors. Using SparQ (see Sec. 2.4) the intermediate states along the *shortest path* in the $a\mathcal{CNG}$ are calculated and inserted into the $ad\mathcal{CNG}^\star$. An example is shown in Figure 3. Starting at relation $_2\angle_1^7$ the MC simulation derives (given the kinematic model) that agents A and B end up in relation $_2\angle_1^5$ after Δ_t, if A executes a *straight* (S) motion and B a *left turn* (L). Nevertheless, with probability 0.874 the objects remain in $_2\angle_1^7$ until $t' = t_0 + \frac{1}{2}\Delta_t$ (i.e. one intermediate step) and changes to $_2\angle_1^5$, with traversing $_2\angle_1^6$ in between, afterwards. With probability 0.126 the change to $_2\angle_1^5$ occurs before t'. In contrast, if both agents perform a *straight* motion, they remain in $_2\angle_1^7$ until t'. Afterwards, they end up in $_2\angle 5$ (with $_2\angle 6$ in between) with probability 0.483 and in $_2\angle 7$ with 0.517. As both paths end up in a 'same' relation, which reflects a collision, both agents must not continue to move straight on.

Considering the example above in a complete context (i.e. A and B can both turn $60°s^{-1}$ left and right and can go straight) employing 1000 samples and doing a 5 step look ahead, each including one intermediate step, are calculated within 590 ms.[4] After the 5 steps *only* 20 different scene descriptions were generated which is far less then the ones resulting from the $a\mathcal{CNG}$, which would be all possible scene descriptions for two objects, i.e. the number of base relations

[4] Intel®Core™2 Duo E8400 @ 3.00GHz

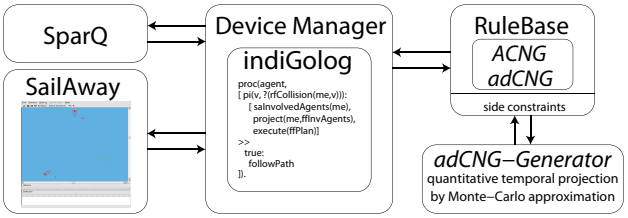


Figure 4. The overall architecture of the SailAway agent.

(72). With increasing number of objects the number of scene descriptions generated with $ad\mathcal{CNG}$ only varies a little, whereas the number of descriptions generated with $a\mathcal{CNG}$ increases significantly.

Collision detection The coincidence of oriented points ('same' relation) clearly resembles a collision. And thus, it is straightforward to model collisions by these relations. As the objects in the domain are extended objects and not single points, a collision detection has to be performed during graph construction. The collision detection is performed on the polygonal approximation of the convex hull of the agents, possibly enlarged to include a minimum safety distance.

If a collision is detected the entities are viewed as colliding and are therefore qualitatively represented as coinciding. This can result in an inconsistent description of a scene, e.g. if A and B coincide, B and C coincide, but A and C do not coincide, which is an inconsistency because if A, B, and C were points then A would also coincide with C. Nevertheless we are only interested in the detection of collisions at this step, the inconsistency does not pose any problems.

4 AGENT CONTROL

In this section we describe how we apply adaptive conceptual neighborhoods in order to control agents in a dynamic environment regarding a set of regulations. Based on previous work (SailAway: [6, 7, 24, 25]) we apply our approach to the domain of collision free vessel navigation. For reasoning and deriving a qualitative model in $\mathcal{OPRA}_m$ from the data of the simulator, we apply the qualitative spatial reasoning toolbox SparQ [23].

Although, our approach can be also integrated in other agent control frameworks, the general architecture is based on the INDIGOLOG framework [4, 22], an incremental variant of the Situation Calculus [16] based programming and planning language GOLOG [14] with a well-defined interface structure to connect to additional modules. GOLOG is a planning and programming language, i.e. the problem domain can not only be specified declaratively in terms of fluents (properties of the world) and actions which affect these fluents, but also imperative programming constructs like conditional branch or while loops are provided to prespecify partial solutions. Notably, Bhatt & Loke present an approach to a general integration of QSR in the Situation Calculus with neglecting the actual execution of spatial actions [2].

In the following, we introduce the system architecture, and go into detail on two of the modules: the RuleBase and the agent code.

4.1 System architecture

The overall architecture (Figure 4) consists of 5 modules: 1) the SailAway simulator, 2) the spatial reasoning toolbox SparQ, 3) the INDIGOLOG framework including the agent specification, 4) the RuleBase, and 5) the $ad\mathcal{CNG}$ generator.

SailAway simulates the environment of an open water plain with different types of vessels, e.g. motor vessels and sailing vessel, and immobile objects, e.g. buoys. Periodically, the module provides a metrical world model including poses and velocities of vessels, the positions of immobile objects, and the direction and speed of the wind. Via a device manager the INDIGOLOG framework hands over the data to SparQ (see Section 2.4) for qualification. The relations derived are directly available for the agent to decide on appropriate actions, while the metrical data is not presented to the agent. The *RuleBase* contains the qualitative specifications of regulations the agents have to follow. A specification consists of a set of start conditions, intermediate configurations, potential restrictions on actions the agents are allowed to execute, and end configurations. If the agent detects a situation where perhaps a rule needs to be be applied, i.e. due to specific fluent values, a request to the RuleBase is triggered. Based on the rule specifications and the quantitative data (with noise added) the $adCNG$ generator simulates the outcome of actions the involved agents are (assumed to be) able to execute (see Section 3). In the end an appropriate action sequence, i.e. no forbidden relations occur during simulation, is selected and handed over to the agent, which thereupon can execute the actions.

In the following, we will go into more detail on the RuleBase module and a simple INDIGOLOG agent.

4.2 RuleBase: collision regulations

Traffic regulations for sea navigation have been defined in the International Regulations for Preventing Collisions at Sea (ColRegs) of the International Maritime Organization (IMO). For each pair of vessels, the rules define which one has to give way (burdened vessel) and which is the privileged one (it is possible that both vessels are burdened). The roles of the vessels and thus, which behavior is expected, are determined by the types of the vessels (e.g. sailing vessels, motor vessels, etc.) and the course they are approaching. Reasonable avoidance behavior of burdened vessels is described in specific patterns in supplemental textbooks.

For example, Rule 14(a) of the ColRegs says: *When two power-driven vessels are meeting head-on or nearly head-on courses so as to involve risk of collision each shall alter her course to starboard so that each shall pass on the port side of the other.*

These regulations are represented in the form of abstract finite state machines (FSM) and build the basis for generating the $adCNG$. Therefore, physical details like speed or distance do not have to be considered in the rule representation. Exemplary, the formal definition of rule 14(a) is given in Figure 5. This specific rule is triggered if both vessels are of type 'MotorBoat' (*About*) and if under the assumption of straight motion a collision in one of the relations $\angle 6$ to $\angle 10$ is detected within a specific interval of time (*Detect*). For more convenient agent programming sets of relations could be labeled with linguistic terms, e.g. *headon_collision* for relations $\angle 6$ to $\angle 10$. Then both agents are only allowed to turn starboard, i.e. right, until they pass on each others port (left) side (*Start*). As long as the agents have not passed the situation has to be considered further (rel $\in \{ {}_4\angle_i^j \mid i,j \in 1 \dots 4\}$) until an end configuration is reached (rel $\in \{ {}_4\angle_i^j \mid i,j \in 5 \dots 7\}$ then Finished).

4.3 The vessel control agent

For the INDIGOLOG agent only the qualitative model is available. Each agent can perform the following actions: light, normal and hard turns to the left and right as well as moving straight. Additionally, the agent can simultaneously accelerate or decelerate. In summary, this allows each agent to perform 21 different actions. The resulting trajectory in SailAway depends on the current speed, the type of the vessel, and possible further influences, e.g. the wind for a sailing boat.

About	:	$A_{\text{type}} \in \{$ MotorBoat $\}$ and $B_{\text{type}} \in \{$ MotorBoat $\}$
Detect	:	rel $\in \{ {}_4\angle 6 \ \dots \ {}_4\angle 10 \ \}$
Start	:	$A_{\text{action}} \in$ turnRightActions and $B_{\text{action}} \in$ turnRightActions then PassLeft else Invalid
PassLeft	:	rel $\in \{ {}_4\angle_i^j \mid i,j \in 1 \dots 4\}$ then PassLeft rel $\in \{ {}_4\angle_i^j \mid i,j \in 5 \dots 7\}$ then Finished else Invalid

Figure 5. Formal representation of Rule 14(a) for two vessels A and B as a FSM, with the first lines as the precondition and the rest are the states and their transitions.

A basic agent can be specified as follows. In general, he follows a path (**followPath**) until some potential collision with another vessel is detected. For doing so, an arbitrary vessel is picked nondeterministically (**pi(v)**) and checked whether a collision is possible (**?(rfCollision(me,v))**. To execute avoiding maneuver preferentially, a prioritized interrupt ($\langle\rangle$) is used. If a potential collision is detected the agent senses for other agents in proximity, which interfere with the execution of a regulation (*saInvolvedAgents(me)*. A request to the RuleBase module is sent to derive appropriate actions (*project(ffInvAgents)*) with the result in fluent (*ffPlan*).

Based on the current world model the RuleBase derives the $adCNG$ by simulation. Given that graph, the most appropriate actions can be chosen. Possible transitions are evaluated wrt. collision avoidance regulations. Those actions are chosen which are closest to the shortest path from start to end configuration, but not those with any chance of a transition into a 'same' relation, i.e. a collision.

If more than two vessels are involved in a setting, potential contradictions of the individual regulations in the global context can be found out not only by bad evaluation values of the actions to execute, but also by constraint-based reasoning as performed in [5, 7] with the SparQ toolbox. The plan derived is handed over to the agent (*ffPlan*) where it is executed. In this simple agent we haven't modeled that the plan execution has to be monitored. If other agents behave different to the model assumed, execution must be terminated and a new plan needs to be derived.

With several agents of this kind, we are able to generate collision-free and rule-compliant (as far as combinations of individual regulations are consistent) behavior.

5 CONCLUSION

In this paper we have shown how vessel behavior can be controlled in a collision-free and rule-compliant manner by means of qualitative spatial representation and reasoning.

We presented the SailAway system and defined a basic INDIGOLOG agent. Regulations are formalized in a purely qualitative manner. Based on these formalizations, assumptions on agent kinematics, and the current world model the $adCNG$, and thus, the next actions, can be derived. This tight integration of neighborhood-based reasoning and Monte-Carlo simulation allows for reasonable agent control. The high-level finite state machine description of the rule abstracts away from details, like *how hard to turn or when to turn at*

all. However all these details are automatically calculated by our approach, thus taking the ambiguity of the natural rule description into account and situation adequate avoidance maneuvers are generated.

In future work, we will compare how much benefit exact physical models give over abstracted qualitative models wrt. diverse optimality criteria. To make the demonstrator more realistic, we will convert the SailAway simulator to deal with maps on the S57 standard (used for professional navigation devices in vessels).

ACKNOWLEDGEMENTS

We would like to thank Diedrich Wolter and the anonymous referees for their comments, which helped to improve this paper considerably. The work was carried out in the DFG Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition. Financial support by the Deutsche Forschungsgemeinschaft (DFG) is gratefully acknowledged.

REFERENCES

[1] Brandon Bennett and Antony P. Galton, 'A unifying semantics for time and events', *Artificial Intelligence*, **153**(1-2), 13–48, (March 2004).

[2] Mehul Bhatt and Seng Loke, 'Modelling dynamic spatial systems in the situation calculus', *Spatial Cognition and Computation*, **8**(1), 86–130, (2008).

[3] Anthony G. Cohn and Shyamanta M. Hazarika, 'Qualitative spatial representation and reasoning: An overview', *Fundamenta Informaticae*, **46**(1-2), 1–29, (2001).

[4] Giuseppe De Giacomo and Hector Levesque, 'An incremental interpreter for high-level programs with sensing', in *Logical Foundation for Cognitive Agents: Contributions in Honor of Ray Reiter*, eds., Hector J. Levesque and Fiora Pirri, 86–102, Springer, Berlin, (1999).

[5] Frank Dylla, *An Agent Control Perspective on Qualitative Spatial Reasoning*, volume 320 of *DISKI*, Akademische Verlagsgesellschaft Aka GmbH (IOS Press); Heidelberg, Germany, 2008.

[6] Frank Dylla, 'Qualitative spatial reasoning for navigating agents', in *Behaviour Monitoring and Interpretation - Ambient Assisted Living*, eds., Björn Gottfried and Hamid Aghajan, chapter 5, 98–128, IOS Press, (2009).

[7] Frank Dylla, Lutz Frommberger, Jan Oliver Wallgrün, Diedrich Wolter, Bernhard Nebel, and Stefan Wölfl, 'SailAway: Formalizing navigation rules', in *Proceedings of AISB Symposium on Spatial Reasoning and Communication*, Edinburgh, UK, (2007).

[8] Frank Dylla and Jan Oliver Wallgrün, 'Qualitative spatial reasoning with conceptual neighborhoods for agent control', *Journal of Intelligent and Robotic Systems*, **48**(1), 55–78, (2007).

[9] Christian Freksa, 'Conceptual neighborhood and its role in temporal and spatial reasoning', in *Proceedings of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning*, eds., Madan G. Singh and Luise Travé-Massuyès, pp. 181–187, North-Holland, Amsterdam, (1991). Elsevier.

[10] Christian Freksa, 'Spatial cognition – An AI perspective', in *Proceedings of 16th European Conference on AI (ECAI 2004)*, (2004).

[11] Christian Freksa and Ralf Röhrig, 'Dimensions of qualitative spatial reasoning', in *Proceedings of the III IMACS International Workshop on Qualitative Reasoning and Decision Technologies – QUARDET'93*, eds., N. Piera Carreté and M. G. Singh, pp. 483–492. CIMNE Barcelona, (1993).

[12] Antony Galton, *Qualitative Spatial Change*, Oxford University Press, 2000.

[13] Benjamin Kuipers, *Qualitative Reasoning: Modeling and simulation with incomplete knowledge*, MIT Press, Cambridge, Massachusetts, USA, 1994.

[14] Hector J. Levesque, Raymond Reiter, Yves Lesperance, Fangzhen Lin, and Richard B. Scherl, 'GOLOG: A logic programming language for dynamic domains', *Journal of Logic Programming*, **31**(1-3), 59–83, (1997).

[15] Gerard Ligozat, 'Qualitative triangulation for spatial reasoning.', in *Spatial Information Theory: A Theoretical Basis for GIS, (COSIT'93), Marciana Marina, Elba Island, Italy*, eds., Andrew U. Frank and Irene Campari, volume 716 of *LNCS*, 54–68, Springer, (1993).

[16] J. McCarthy and P. J. Hayes, 'Some philosophical problems from the standpoint of artificial intelligence', *Meltzer and Michie, Machine Intelligence 4*, 463–502, (1969).

[17] Reinhard Moratz, 'Representing relative direction as a binary relation of oriented points', in *ECAI*, eds., Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, pp. 407–411, Riva del Garda, Italy, (2006). IOS Press.

[18] Reinhard Moratz, Frank Dylla, and Lutz Frommberger, 'A relative orientation algebra with adjustable granularity', in *Proceedings of the Workshop on Agents in Real-Time and Dynamic Environments (IJCAI 05)*, pp. 61–70, Edinburgh, Scotland, (July 2005).

[19] Philippe Muller, 'A qualitative theory of motion based on spatio-temporal primitives', in *KR'98: Principles of Knowledge Representation and Reasoning*, eds., Anthony G. Cohn, Lenhart Schubert, and Stuart C. Shapiro, 131–141, Morgan Kaufmann, San Francisco, California, (1998).

[20] Florian Pommerening, Stefan Wölfl, and Matthias Westphal, 'Right-of-way rules as use case for integrating golog and qualitative reasoning', in *KI*, eds., Bärbel Mertsching, Marcus Hund, and Muhammad Zaheer Aziz, volume 5803 of *Lecture Notes in Computer Science*, pp. 468–475. Springer, (2009).

[21] Jochen Renz, *Qualitative Spatial Reasoning with Topological Information*, number 2293 in Lecture Notes in Computer Science, Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[22] Sebastian Sardina, 'Indigolog: An integrated agent arquitecture: Programmer and user manual', Technical report, University of Toronto, (2004).

[23] Jan Oliver Wallgrün, Lutz Frommberger, Diedrich Wolter, Frank Dylla, and Christian Freksa, 'A toolbox for qualitative spatial representation and reasoning', in *Spatial Cognition V: Reasoning, Action, Interaction: International Conference Spatial Cognition*, pp. 79–90, Bremen, Germany, (2006).

[24] Diedrich Wolter, Frank Dylla, Lutz Frommberger, Jan Oliver Wallgrün, Bernhard Nebel, and Stefan Wölfl, 'Qualitative spatial reasoning for rule compliant agent navigation', in *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference (FLAIRS-2007)*, Key West, FL, USA, (May 2007). Short paper.

[25] Diedrich Wolter, Frank Dylla, Stefan Wölfl, Jan Oliver Wallgrün, Lutz Frommberger, Bernhard Nebel, and Christian Freksa, 'SailAway: Spatial cognition in sea navigation', *Künstliche Intelligenz*, **22**(1), 28–30, (2008).

Modeling Motion Event Using QSR

Rupam Baruah[1] and **Shyamanta M. Hazarika**[2]

Abstract. Representation of high level semantics of dynamic scenes has become an active area of research in cognitive vision. Qualitative spatial representation and reasoning finds application in such domains as video understanding at the highest level is mostly qualitative. In this paper, we propose a formalism based on qualitative spatial representation for describing some events resulting out of motion of spatial entities. We focus on two aspects, namely, orientation and relative direction of motion and describe motion events based on a combination of 25 orientation relations and 24 relative direction relations.

1 INTRODUCTION

Representation of and reasoning with time and space in a qualitative manner has been an active area of research within the artificial intelligence (AI) community. Qualitative Spatial Reasoning (QSR) strives to provide calculi that allow a machine to represent and reason with spatial entities without resort to traditional quantitative techniques. Qualitative spatial representations addressing many different aspects of space including topology, orientation, shape, size and distance have been put forward. An account of work done in qualitative spatial reasoning is surveyed in [2]. Different aspects of space that have been treated in a qualitative manner are topology, cardinal direction, shape, distance etc.

Qualitative spatial reasoning is a knowledge representation technique which is very close to how human beings perceive things in real life. In that sense, we can say that it is a representation of our commonsense knowledge about various domains. Some of the areas where QSR has been applied are Geographical Information System (GIS), robotics, natural language processing, computer vision etc. In a formalism using QSR, typically a set of Jointly Exhaustive and Pairwise Disjoint ($JEPD$) relations are defined between spatial entities. There are different techniques for reasoning. One is the formulation of the problem as a constraint satisfaction problem (CSP).

One interesting application of QSR is in cognitive vision. Cognitive vision covers areas like surveillance, navigational domain etc. We can use QSR to provide high level symbolic representation of events in cognitive vision applications. In such applications, QSR abstracts unnecessary details and removes error and uncertainty in input. For example, in a surveillance application, we may like to describe the situation of an attack in a qualitative way. In a navigational domain, we may like to represent a situation where a vehicle overtakes another on the right.

In this paper, we have proposed a QSR based formalism to represent events in a navigational domain. The spatial entities in this case may be vehicles, human beings, cattle etc. In our work, we focus on two aspects, namely, orientation and relative direction of motion. We have abstracted the spatial entities by Minimum Bounding Rectangles and defined orientation relations based on the model of Goyal and Egenhofer [5]. An intrinsic frame of reference is assumed for orientation. We have defined a notion of intrinsic direction of motion of a spatial object and relative direction of motion of two spatial objects.Below, we explain motivation behind our work and mention some related work in the area.

1.1 Motivation

In many dynamic systems, motion patterns occur. In order to analyse these motion patterns for making predictions or to study behaviours of systems, formal methods are required to deal with them. This has gained importance as a number of technologies have been devised which allow objects to get tracked precisely. In order to identify patterns, the first aspect is representation of events that constitute the pattern. In cognitive vision applications, it is desired that this representation is close to human perception of events. Therefore, qualitative spatial reasoning can be an important tool here to remove unnecessary details and uncertainty from input. Using QSR , we can provide a high level representation of events which may be more useful than quantitative details. Once events are properly defined, we can look for patterns by performing online or offline analysis of input data.

1.2 Related Work

From the vast literature on QSR, we would like to pick some notable work that is relevant to our formalism. Over the years different qualitative constraint calculi have been proposed to handle orientation. Frank [3] suggested different methods for describing cardinal direction of a point; this was with respect to a reference point in geographic space. Ligozat [8] studied the computational properties of cardinal algebra. Another notable work includes Star calculus by Renz and Mitra [13] and double cross calculus by Freksa [4]. Goyal and Egenhofer [5] proposed a direction relation matrix for two dimensional spatial entities. Skaidopolous and Koubarakis [14] developed reasoning algorithms for this calculus. Based on this work, rectangular cardinal directions have been introduced by Navarrete and Sciavicco [11]. Our approach for handling orientation is based on the work of Goyal and Egenhofer [5]. In our approach, spatial entities are represented by Minimum Bounding Rectangles (MBR). The sides of the MBR are parallel to some orthogonal basis in a two-dimensional euclidean space. In the case of direction, work is reported where spatial objects are abstracted as points, line segments or rectangular regions. Mention may be made of qualitative trajectory calculus [15] and dipole relation algebra [9]. In the first, spatial entities are considered as points and in the second, they are considered as directed line

[1] Computer Science & Engg. Department, Jorhat Engineering College, Jorhat, Assam, India. email: rupam.baruah.jec@gmail.com

[2] Computer Science & Engg. Department, Tezpur University, Tezpur, Assam, India. email: smh@tezu.ernet.in

segments. As for motion patterns, Muller [10] defined motion classes in his spatio-temporal theory of motion. Hazarika [6] mentioned behaviour patterns in his work on spatio-temporal continuity.

Table 1. Orientations relations using Allens interval algebra

Abbvr.	Relation Name	Definition (Using IA)
FL	FrontLeft	{m,b} X {mi,bi}
F	Front	{fi,o,d,s,f,e,si,oi}X {mi,bi}
FR	FrontRight	{mi,bi} X {mi,bi}
BL	BackLeft	{m,b} X {m,b}
B	Back	{fi,o,si,oi,d,s,f,e} X {m,b}
BR	BackRight	{mi,bi} X {m,b}
L	Left	{m,b} X {d,s,f,e}
R	Right	{mi,bi} X {d,s,f,e}
L&F	Left&Front	{m,b} X {si,oi}
L&B	Left&Back	{m,b} X {fi,o}
R&F	Right&Front	{mi,bi} X {si,oi}
R&B	Right&Back	{mi,bi} X {fi,o}
EOL	ExtendedOnLeft	{m,b} X {di}
EOR	ExtendedOnRight	{mi,bi} X {di}
EIF	ExtendedInFront	{di} X {mi,bi}
EIB	ExtendedInBack	{di} X {m,b}
I	Inside	{d,s,f,e} X {d,s,f,e}-{(e,e)}
EQ	Equal	{(e,e)}
OAL	OverlapAlong	{fi,o,si,oi,di} X {d,s,f,e}
OAC	OverlapAcross	{d,s,f,e} X {fi,o,si,oi,di}
OOL	OverlapOnLeft	{o,fi} X {fi,o,si,oi,di}
OOR	OverlapOnRight	{si,oi} X {fi,o,si,oi,di}
OIF	OverlapInFront	{di} X {si,oi}
OIB	OverlapInBack	{di} X {fi,o}
FO	FullOverlap	{di} X {di}

In the framework we have proposed, we have considered spatial entities in two dimensional form, using the abstraction of MBR. MBRs have been used for such purpose in GIS applications [11]. Here, our approach moves away from the ones where entities are represented as points or line segments. Representation of spatial objects in two dimensional form allows us to handle aspects like orientation and size at finer details. In this respect, our work is similar to the work of Fernyhough, Cohn and Hogg [7], where spatial objects are represented as rectangles. But we considered the cases where the spatial object may span more than one tile in the direction relation matrix and given formal definition of orientation relations in terms of Allen's interval algebra [1]. Moreover, the cases where the MBRs may overlap are also taken into consideration. As for direction of motion, we have considered relative direction of motion at a finer level by considering inclination of spatial objects to the right and to the left of the axes of projection. For both the aspects, JEPD relations are defined formally and composition tables and conceptual dependency diagrams are presented.

2 THE FRAMEWORK

Below, we describe the framework proposed for representation of motion events. At first, the spatial model is introduced and then we proceed to define JEPD relations for handling orientation and relative direction of motion.

2.1 THE SPATIAL MODEL

In the qualitative framework proposed, each spatial entity is modeled by a minimum bounding rectangle (MBR). We assume that sides of the MBR are parallel to the axes of some orthogonal basis in an two-dimensional euclidean space. Each spatial entity is assumed to have a natural front. The constraint of having the sides of MBRs paralleled to axes of an orthogonal basis in $2D$ could introduce loss of information $vis\text{-}a\text{-}vis$ actual direction of motion. But this is taken care of by the relative direction relations.

2.2 REPRESENTATION OF ORIENTATION

Goyal and Egemhofer [5] proposed a calculus for representing cardinal directions between extended spatial entities. Their calculus consists of a 3 X 3 direction relation matrix which represents the 9 sectors formed by the minimum bounding rectangles of an extended spatial entity. Skaidopolous and Koubarakis [14] developed reasoning algorithms for this calculus. Based on this work, rectangular cardinal directions have been introduced by Navarrete and Sciavicco [11]. In Figure 1, we introduce the framework for orientation. We assume spatial entities to have a natural front. It is assumed that the entity moves in the direction of its front. The direction of the front will define other orientations such as back, left and right. A spatial entity may span more than one tile, but it is important to note that since the entity is represented by an MBR, some combinations are not possible. In Table 1, the set of orientation relations are defined using the semantics of Allens interval algebra [1]. In Figure 2, we illustrate one case of *FrontLeft* and *Right&Front* relations.

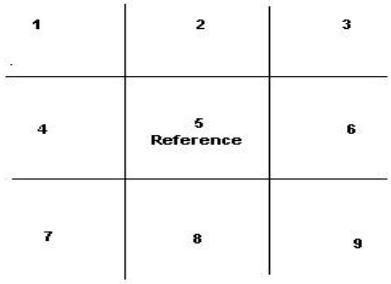


Figure 1. Framework for orientation relations

The set of orientation relations defined are Jointly Exhaustive and Pairwise Disjoint ($JEPD$). In rectangle algebra, Condotta, Balbiani and Farinasdel [12]have shown that there are 169 orientation relations that can hold between two rectangles whose sides are parallel to the axes of projection. The relations were formally defined using Allen's interval algebra [1]. In defining orientation relations in our framework, these 169 relations have been distributed into different orientation relations. So, for the relations to be Jointly Exhaustive, the total number of ordered pairs in all the orientation relations must be 169. This has been verified.

It is Pairwise Disjoint (PD) because the intersection of any two relations is null. The set of orientation relations is closed under composition and converse. *Equal* is the identity relation. These orientation relations tell us how a spatial entity, represented by its MBR, is located in space with respect to a reference entity. This information alone is not sufficient if our focus is on the direction in which the entity is moving. For example, let us consider two spatial entities x and y and let y be the reference. Let x be in tile 1 i.e. x *FrontLeft* y. Now, the front of x may be along any of the four directions indicated by axes of projection. We have assumed that the entity moves in the direction of its front. So, we need to define additional relations

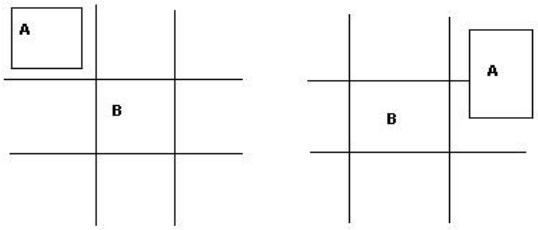


Figure 2. FrontLeft and Right&Front Relations

for handling relative direction of entities in motion. The set of orientation relations is closed under composition and converse. Let X be the primary, Y be the reference and let o be a relative orientation relation and $X\ o\ Y$. For computing the converse, when we consider X as the reference and draw the direction relation matrix. Then Y must be in a single tile or multitile configuration with respect to X. From rectangle algebra [12], we know that this relative orientation between two rectangles can be defined using Allen relation [1] and as the relations are JE, this ordered pair will be included in exactly one orientation relation. It is verified that taking converse twice gives back the original relation for all the orientation relations.

2.3 DIRECTION OF MOTION

Since the sides of an MBR are parallel to the axes of projection, there are four major directions of motion that we can consider. These are along positive and negative directions of the axes of projection. Each of these major directions is made finer by considering the cases where an entity may be inclined to the right or to the left.

2.3.1 Intrinsic Direction of Motion

For representation of direction of motion, we divide an MBR into eight direction segments as shown in the Figure 3 below :

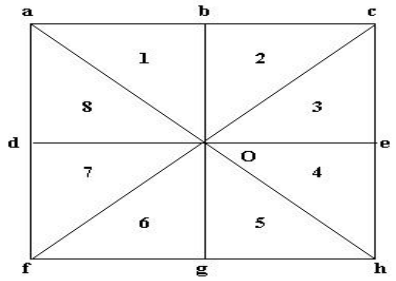


Figure 3. An MBR divided into eight tiles

Let b, e, g and d be the midpoints of the lines ac, ch, fh and af respectively. The point of intersection of the line segments ah, cf, de and bg is designated as o. The MBR is divided into eight tiles and each tile is given an integer i such that $1 \leq i \leq 8$. There are major directions of motion that we consider first. These are along positive and negative directions of X-axis and Y-axis. With the help of tile numbers or line segments, we represent the intrinsic direction of motion of an entity. For example, if the entity moves along the line segment ob, then this movement is parallel to positive Y-axis. Similarly, if the entity moves along the direction specified by the line segment od, then this movement is along the direction of the negative X-axis. Movement parallel to axes of projection can be specified by four line segments ob, og, oe and od. We know that this direction of movement may not always be parallel to one of the axis of projection. For example, if we consider the positive X-axis as the major direction of motion, then two new cases can be introduced by considering the movement direction of the entity to be titled to the left or tilted to the right. So, if the major direction of motion of the entity is along oe (positive X-axis), then we can think about a direction of motion that is left-inclined to this major direction of motion. In the tile representation, this movement will be along a direction included within segment number 3 in Figure 3. As another example, if the major movement direction is og (negative Y-axis), then a movement along a direction included within segment 6 will specify a right-inclined movement along this major direction of motion.

Considering this, if we think about intrinsic direction of movement of a single spatial entity, there are four major directions of movement; Movement along a major direction has three variants: right inclination, left inclination or neutral. For example, movement along the direction of line segment ob is one major movement direction (positive Y-axis) and movement along segments 1 and 2 are two inclinations of it. Therefore, in total we need to consider 12 intrinsic directions of movement of a single spatial entity.

2.3.2 Relative Direction of Motion

In order to define relative directions of motion, we need to consider two spatial entities. For defining qualitative relations for this, at first we consider movement along major directions only. Let P be the primary object and R be its reference and the direction of motion of both primary and reference objects are represented by a set of directed line segments for convenience. In this paper, four relation names, namely, *Same*, *Opposite*, *LR* and *RL* are introduced. *Same* has the intended interpretation that both reference and primary are moving along the same direction and this direction is parallel to one of the axis of projection as illustrated in Figure 4.

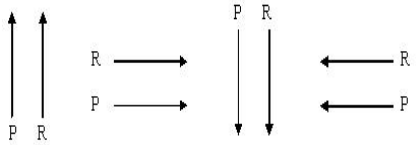


Figure 4. Same

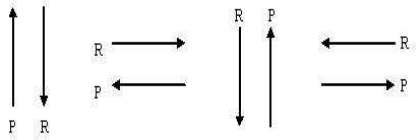


Figure 5. Opposite

Opposite has the intended interpretation that both primary and reference are moving in opposite direction and this direction is parallel to one of the axis of projection which is illustrate this in Figure 5. *LR* has the intended interpretation that the movement of the primary

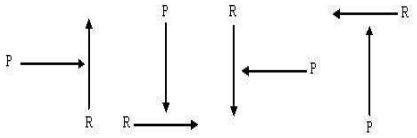


Figure 6. LR

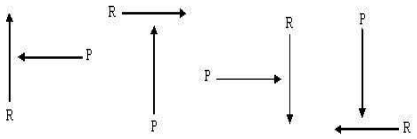


Figure 7. RL

is from left to right with respect to the reference and *RL* has the intended interpretation that the movement of the primary is from the right to left with respect to the reference as shown if Figure 6 and in Figure 7 respectively. In both these cases, the movement is parallel to an axis of projection. Here, we introduce a notation for representing relative direction of motion. We write an ordered pair where the first element specifies the intrinsic direction of the reference and the second element specifies the intrinsic direction of the primary. For example, if we write (oe, oe), it means both reference and primary are moving along the positive X-axis direction. Therefore the *Opposite* relation can be specified as {(oe, eo),(of, fo),(od, do),(ob, bo)}.

Now, we introduce the notion of left and right inclination into relative direction of motion. Two keywords, namely, *Left* and *Right* are introduced for handling this. Let us take an example where both the reference and the primary are moving along positive Y-axis direction but the primary is having a left inclination. We call this relation *LeftSame*. In direction segment representation this can be defined as {(ob, 1),(oe, 3),(og, 5),(od, 7)}. We take another example where the primary is moving from left to right with respect to the reference (LR) and the reference is having a right inclination. We call this relation *LRRight*. If the reference movement is along positive Y-axis, then this can be specified as {(2, do)}. Considering the movement of the reference along other major directions, the *LRRight* relation can be defined as {(2, do),(4, bo),(6, eo),(8, go)}.

It is important to note that these relative direction relations are no way dependent on orientation relations defined earlier. For example, the primary may be in the back of the reference and in that position, both may be directed along any of the directions described.

2.3.3 Relative Direction Relations

Below, we enlist relative direction relations along with their intended interpretation.

1. Same (S): Both primary and reference move in the same direction
2. LeftSame (LS): Same direction, primary inclined to left
3. RightSame (RS): Same direction, primary inclined to right
4. SameLeft (SL): Same direction, reference inclined to left
5. SameRight (SR): Same direction, reference inclined to right
6. SameDiagonal (SD): Same direction, both inclined
7. Opposite (O): Primary and reference are in opposite direction
8. LeftOpposite (LO): Opposite direction, primary inclined to left
9. RightOpposite (RO): Opposite direction, primary inclined to right
10. OppositeLeft (OL): Opposite direction, reference inclined to left
11. OppositeRight (OR): Opposite direction, reference inclined to right
12. OppositeDiagonal (OD): Opposite direction, both inclined
13. LR: Primary in left to right direction, both not inclined
14. LeftLR (L-LR): Left to right direction, primary inclined to left
15. RightLR (R-LR): Left to right direction, primary inclined to right
16. LRLeft (LR-L) : Left to right direction, reference inclined to left
17. LRRight (LR-R): Left to right direction, reference inclined to right
18. LRDiagonal (LRD): Left to right direction, both inclined
19. RL: Primary in right to left direction, both not inclined
20. LeftRL (L-RL) : Right to left direction, primary inclined to left
21. RightRL (R-RL): Right to left direction, primary inclined to right
22. RLLeft (RL-L): Right to left direction, reference inclined to left
23. RLRight (RL-R): Right to left direction, reference inclined to right
24. RLDiagonal (LRD): Right to left direction, both inclined

The set of relative direction relations is Jointly Exhaustive and Pairwise Disjoint ($JEPD$). If we consider intrinsic direction of movement of a single spatial entity, there are 12 directions in which an entity may be headed. For defining relative direction of motion, these 144 combinations of intrinsic motion have been distributed into different relations. We have shown that each relation is defined as a set of ordered pairs. For relations to be JE, it is enough to check that all these 144 combinations are distributed over the relative direction relations. This has been verified. The set of relations is PD because the intersection of any two relations, defined as set of ordered pairs, is null. The set of direction relation is closed under composition and converse. *Same* is the identity relation. For computing converse of any relative direction relation, we need to consider the semantics of the relation. The frame of reference is intrinsic. For example, let us assume that the primary is in LR orientation with respect to the reference i.e. its movement is in left to right direction with respect to the reference. In such a case, the movement of the reference with respect to the primary is from right to left. So, the converse of the LR relation is RL. As another example, let us consider $LeftOpposite$ relation. Both primary and reference are moving in opposite direction and the primary in inclined to the left. For the converse, both are still moving in opposite direction but since the roles are reversed, this time the reference is inclined to the left. So, the converse relation is $OppositeLeft$.It is verified that taking converse twice gives the original relation.

2.4 COMPOSITION AND CONCEPTUAL DEPENDENCY

Composition of binary relations is an important operation in qualitative spatial reasoning. In Table 2, we present the composition table for single tile orientation relations with themselves and in Table 3, the composition of *Same* type direction relations with themselves is presented.

Conceptual neighbours of a relation R are those relations that can hold between the spatial entities because of continuous change arising out of motion. For example, we assume that *Same* relation holds between two entities x and y at any point of time. Because of movement of one or both the entities, this relation may change at subsequent time point. But after *Same*, the relation that holds between x and y can not be *Opposite*. Let x be the primary and let y be the reference. The possibilities are: x becomes inclined to the left or to the right or y becomes inclined to the left or to the right or both

Table 2. Composition of single tile orientation relations

	r	fr	br	l	bl	fl
r	r	fr	br	l	b, bl,eib	f, fl, eif
fr	fr, r&f	fr	r, fr, br, r&f, r&b, eor	f, fr, fl, eif, l, l&f	bl, l, l&b, l&f, eol	f, fr, fl, eif
br	r, br, r&b	fr, r, br, r&f, r&b, eor	br	r, l, b, r&b, l&b, eib	b, bl, eib	l, bl, l&b, fl, l&f, eol
l	r, l	f, fr, fl, eif	b, br, bl, eib	l	bl	fl
bl	b, bl, br, eib, l, l&b	allrel	b, bl, br, eib, r, r&b	l, bl, l&b	bl	br, b, bl, eib
fl	r, fr, r&f	f, fl, fr, eif	r, fr, r&f, br, r&b, eor	f, fr, fl, eif, l, l&f	bl, l, fl, l&f, l&b, eol	fl

Table 3. Composition of Same type direction relations

	s	ls	rs	sl	sr	sd
s	s			sl	sr	
ls	ls			sd	sd	
rs	rs			sd	sd	
sl		s				sl, sr
sr			s			sl, sr
sd		ls, rs	ls, rs			sd

become inclined. Accordingly, the possible relations that may hold after *Same* are *LeftSame*, *RightSame*, *SameLeft*, *SameRight* or *SameDiagonal*. So, these relations are conceptual neighbors of *Same*. The conceptual dependency is expressed in the form of a graph where each node represents a relation and an edge is drawn between two nodes if they are conceptual neighbours. In Figure 8, we show the conceptual neighbourhoods for all the orientation relations defined earlier.

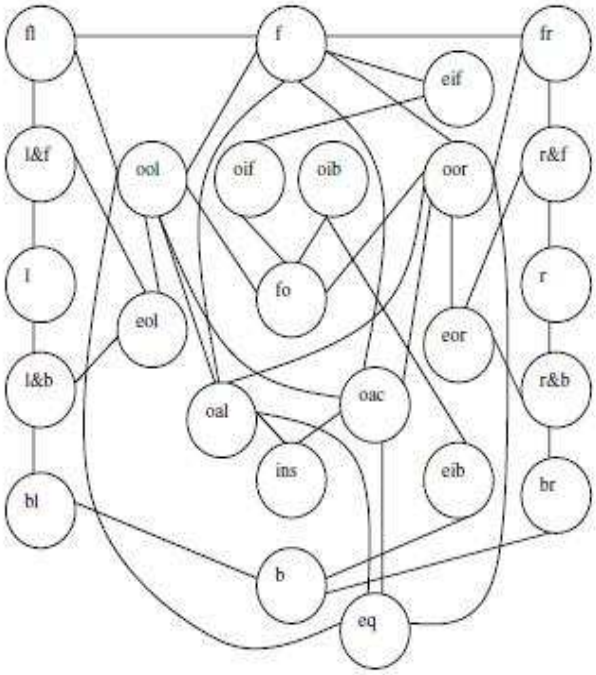


Figure 8. Conceptual Dependency Diagram for Orientation Relation

3 APPLICATION

For illustrating an application where the proposed framework can be used, we would like to consider traffic analysis. We will describe the event of one vehicle crossing another from left to right in the front without any overlap of their MBRs. Let y be the reference entity and let x be the primary entity. Let $t1$, $t2$ and $t3$ be three intervals of time. These intervals of time are related by Allen relation such that $t1$ meets $t2$ and $t2$ meets $t3$. We introduce a notation where O is an orientation relation and d is a direction relation and we write it as x O_d y, which means that x O y & x d y. Note that the notation O_d does not mean that we are using a calculus by combining orientation and direction relations. Since the vehicles are moving, the orientation relations holding between them will change with time. Moreover, at any point of time, a relative direction relation will hold between them. During the first time interval $t1$, the primary will be in the front left of the reference and it will have a left to right direction of motion with respect to the reference. In the next time interval $t2$, the relative orientation of the primary will change and it will come to the front of the reference. Below, we enumerate the relations that should hold at different intervals of time:

- $t1$: x FrontLeft$_{LR}$ y
- $t2$: x Front$_{LR}$ y

- $t3$: x FrontRight$_{LR}$ y

Within this domain, different events such as a vehicle following another, a vehicle overtaking another and many other motion events can be represented.

4 CONCLUSION AND FUTURE WORK

We have defined a set of $JEPD$ relations for orientation and a set of $JEPD$ relations for relative direction of motion of spatial entities. The set of orientation relations is closed under composition and converse and has an identity relation. Same is true about the relative direction relations. Spatial entities considered are not points or directed line segments. We have represented an entity in its two dimensional form. An example event in navigational domain is represented using the framework. Future work includes developing reasoning techniques for the formalism and testing it in a real application.

REFERENCES

[1] J.F. Allen, 'Maitaining knowledge about temporal intervals', *Communication of the ACM*, **26(11)**, 832–843, (1983).

[2] A.G. Cohn and S.M. Hazarika, 'Qualitative spatial representation and reasoning: an overview', *Fundamenta Informaticae*, **46(1-2)**, 1–29, (2001).

[3] A. U. Frank, 'Qualitative spatial reasoning about cardinal directions', *In Proceedings of the 7th Austrian Conference on Artificial Intelligence*, 157–167, (1991).

[4] C. Freksa, 'Using orientation information for qualitative spatial reasoning', *Theories and Methods of Spatio-Temporal reasoning in Geographic Space 639, Lecture Notes in Computer Science*, (1992).

[5] R.K. Goyal and M.J. Egenhofer, 'Similarity of cardinal directions', *Advances in Spatial and Temporal Databases, Lecture Notes in Computer Science*, **2121**, 36–58, (2001).

[6] S.M. Hazarika, 'Pattern mining as abduction from snapshots to spatio-temporal patterns', *In Proceedings of the International Conference on Advanced Computing and Communications, ADCOM 2007*, 289–294, (2007).

[7] A.G.Cohn J. Fernyhough and D.C.Hogg, 'Constructing qualitative event models automatically from video input', *Image and Vision Computing*, **18**, 81–103, (2000).

[8] G. Ligozat, 'Reasoning about cardinal directions', *Journal of Visual Languages and Computing*, **9**, 23–44, (1998).

[9] R. Moratz, J. Renz, and D. Wolter, 'Qulatitative spatial reasoning about line segments', *In Proceedings of ECAI 2000*, 234–238, (2000).

[10] P. Muller, 'Topological spatio-temporal reasoning and representation', *Computational Intelligence*, **18(3)**, 420–450, (2002).

[11] I. Navarrete and G. Sciavicco, 'Spatial reasoning with rectangular cardinal direction relations', *In Proceedings of the ECAI-06 Workshop on Spatial and Temporal Reasoning*, 1–10, (2006).

[12] J. Condotta P. Balbiani and C.L. Farinasdel, 'A new tractable subclass of the rectangle algebra', *In Proceedings of the 16 th International Joint Conference on Artificial Intelligence*, 442–447, (1999).

[13] J. Renz and D. Mitra, 'Qualitative direction calculi with arbitrary granularity', *In PRIAI:2204: Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligence*, 65–74, (2004).

[14] S. Skiadopoulos and M. Koubarakis, 'On the consistencyof cardinal direction constraints', *Artificial Intelligence*, **163(1)**, 91135, (2005).

[15] N. Van de Weghe, B. Kuijpers, P. Bogaert, , and Ph. De Maeyer, 'A qualitative trajectory calculus and the composition of its relations', *In Proceedings of the Inernational. Conerence. on Geospatial Semantics, Lecture Notes in Computer Science*, **3799**, 60–76, (2005).

Proceedings

Spatio-Temporal Dynamics. Editors: Mehul Bhatt, Hans Guesgen, and Shyamanta Hazarika. Workshop Proceedings of European Conference on Artificial Intelligence (ECAI 10), Lisbon, Portugal, 2010.

Also published online as part of Report Series of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition, Universität Bremen / Universität Freiburg. SFB/TR 8 Reports, Bremen, Germany.