Exploiting Qualitative Spatial Neighborhoods in the Situation Calculus

Frank Dylla and Reinhard Moratz

SFB/TR 8 Spatial Cognition
University of Bremen
Bibliothekstr. 1, 28359 Bremen, Germany
{dylla, moratz}@sfbtr8.uni-bremen.de

Abstract. We present first ideas on how results about qualitative spatial reasoning can be exploited in reasoning about action and change. Current work concentrates on a line segment based calculus, the dipole calculus and necessary extensions for representing navigational concepts like *turn right*. We investigate how its conceptual neighborhood structure can be applied in the situation calculus for reasoning qualitatively about relative positions in dynamic environments.

1 Introduction

Most current reliable robot systems are based on a completely determined geometrical world model. The applied metric methods are tending to fail in frequently changing spatial configurations and when accurate distance and orientation information is not obtainable. Qualitative representation of space abstracts from the physical world and enables computers to make predictions about spatial relations, even when precise quantitative information is not available [4]. Important aspects are topological and positional (orientation and distance) information about in most cases physically extended objects. Calculi dealing with such information have been well investigated over recent years and give general and sound reasoning strategies, e.g. about topological relations between regions as in RCC-8 [37,38], about the relative position orientation of three points as in Freksa's Double-Cross Calculus [13] or about orientation of two line segments as in the Dipole Calculus [32,42]. For reasoning with calculi as the above mentioned ones standard constraint-based reasoning techniques can be applied. In [42] Schlieder sketched how a qualitative calculus like the Dipole Calculus might be applied to robot navigation.

In this paper we want to show how to use conceptual neighborhoods [12] for combining Qualitative Spatial Reasoning (QSR) approaches with Reasoning about Action and Change (RAC) approaches for the purpose of robot navigation and path-planning. For the first sketch of our ideas we chose the Dipole Calculus (QSR) and the Situation Calculus [29] respectively Golog [28] (RAC) as agent control language.

Two relations are conceptual neighbors if their spatial configurations can be continuously transformed into each other with only minimal change, e.g. in RCC-8 two disconnected regions (configuration 1) cannot overlap (3) without being externally connected (2) in between. Therefore 1 and 2, as well as 2 and 3 are conceptually neighboring relations but not 1 and 3. Expressing these connections between the relations leads to conceptual neighborhood graphs (CNH-graph). For further motivation for qualitative spatial reasoning we refer to [14] .

Frameworks for reasoning about action and change, e.g. the Situation Calculus [29] based programming language Golog [28], also provide facilities for representing and reasoning about sets of spatial locations. Current variants are able to deal with e.g. concurrency [7], continuous change [22] or decision theory [15]. The Golog framework has been applied successfully in real world domains, e.g. in the RHINO museum tour guide project [3] , or for playing robotic soccer in the RoboCup tournaments [10]. Additionally we can integrate navigational and non navigational actions, e.g. $say(.)$, $pick(.)$, or $look(.)$, without any extra effort.

Unfortunately no formal spatial theory, e.g. for relative position terms like left, right or behind, is defined within for dealing with underspecified, coarse knowledge. Therefore every project modeling dynamic domains needs the naive formalization of such a theory by the developer again and again, although such concepts have been formally investigated.

We present first ideas about qualitative navigation on the basis of oriented line segments, which we consider a valuable starting point. In this context we will show one way how the results about conceptual neighborhood can be applied in the Situation Calculus resp. Golog. In the first stage of this work we will only look at simulated scenarios to omit additional complexity caused by real robot control.

The long term goal is a general representation and usage of qualitative spatial concepts about relative position like e.g. $left$, $right$, or $inFrontOf$ within the Situation Calculus or, the programming language Golog, e.g. providing action facilities like $go(leftOf, exhibit_7)$. We do not only expect such formal qualitative concepts being useful in agent programming but also in human machine interaction [44,30].

In this paper we will present several variants of the Dipole Calculus at different levels of granularity and their corresponding conceptual neighborhoods. We present necessary extensions for expressing robot navigation tasks more adequately. Without doing so we would not be able to formalize navigational behavior with the basic translational and rotational commands intrinsic to every robot system. After a brief introduction to the Situation Calculus and the programming language Golog we will present a first approach combining the Dipole Calculus with the Situation Calculus and Golog. We will clarify our ideas with concrete examples and end with final conclusions.

2 Qualitative Spatial Reasoning

Qualitative Spatial Reasoning (QSR) is an abstraction that summarizes similar quantitative states into one qualitative characterization. From a cognitive perspective the qualitative method *compares* features of the domain rather than *measuring* them in terms of some artificial external scale [13]. The two main directions in QSR are topological reasoning about regions, e.g. the RCC-8, and positional reasoning about point configurations. An overview is given in [5].

Solving navigation tasks involves reasoning about paths as well as reasoning about configurations of objects or landmarks perceived along the way and thus requires the representation of orientation and distance information [41,25]. Many approaches deal with global allocentric metrical data. For many navigational tasks we do not need absolute allocentric information about the position, instead we need relative egocentric representations and a fast process for updating these relations during navigation [45,46].

Several calculi dealing with relative positional information have been presented in recent years. Freksa's double cross calculus [13] deals with triples of points and can also be viewed as a calculus dealing with positional binary relations between a dipole and a point. Schlieder proposed a calculus based on line segments with clockwise or counter clockwise orientation of generating start and end points in [42]. He presented a CNH-graph of 14 basic relations. Isli and Cohn [24] presented a ternary algebra for reasoning about orientation. This algebra contains a tractable subset of base relations.

Moratz et al. [32] extend Schlieder's calculus. In a first variant ten additional relations are regarded, where the two dipoles meet in one point, resulting in a relation algebra in the sense of Tarski [26] with 24 basic relations. Also an extended version is introduced such that spatial configurations can be distinguished in a more fine grained fashion. An application oriented calculus dealing with ternary point configurations (TPCC) is presented in [31]. It is suited for both, human robot communication [30] and spatial reasoning in route graphs [31]. Even more fine grained calculi can be used to do path integration for mobile robots [33]. In [47] a line segment approach for shape matching in a robotic context is presented. In this context the line extraction are derived efficiently in $O(n \log n)$ by using the method of Discrete Curve Evolution [47]. In [1] qualitative spatial calculi are linked to ontological engineering.

2.1 Neighborhood-Based Reasoning

Neighborhood-based reasoning describes whether two spatial configurations of objects can be transformed into each other by small changes [12]. The conceptual neighborhood of a qualitative spatial relation which holds for a spatial arrangement is the set of relations into which a relation can be changed with minimal transformations, e.g. by continuous deformation. Such a transformation can be a movement of one object of the configuration in a short period of time. On the discrete level of concepts, neighborhood corresponds to continuity on the geometric or physical level of description: continuous processes map onto identical or neighboring classes of descriptions [14]. Spatial neighborhoods are very natural perceptual and cognitive entities and other neighborhood structures can be derived from spatial neighborhoods, e.g. temporal neighborhoods. The term continuous in the presence of transformation or deformation needs a grounding in spatial change over time. From our point of view the continuous transformation is the continuous motion of a robot r. This can be described by the function $pos(r) : T \to P$, 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 the function $pos(r)$ is continuous [18]. Detailed work on different aspects of continuity were investigated in [2,6,16,17,23,35]. Based on different definitions of continuity different neighborhood graphs may arouse. This is also true for different robot kinematics, e.g. comparing differential drive robots versus omnidirectional drive robots.

A movement of an agent can then be modeled qualitatively as a sequence of neighboring spatial relations which hold for adjacent time intervals. Using this qualitative representation of trajectories neighborhood-based spatial reasoning can be used as a simple, abstract model of robot navigation and exploration. Neighborhoods can be formed recursively and represented by hierarchical tree or lattice structures.

Schlieder [42] mapped orientation onto ordering. He defined the orientation on triangles and for every set with more than three points recursively for every triangle. He extracted 14 basic relations to reason about ordering of line segments[1]. The conceptual neighborhood graph is shown in Fig. 3. The labels are illustrated in Fig. 4.

From the neighborhood graphs of the individual relations, the neighborhood graph of the overall configuration must be derived. In this global neighborhood graph, spatial transformations from a start state to a goal state can be determined. It has been investigated to use the neighborhood graph of two objects for spatial navigation [42]. It has not been investigated yet how a neighborhood graph for a configuration of more complex or even several objects can be constructed using efficient, qualitative methods based on local knowledge.

A problem for the efficient construction of neighborhood graphs for multiple objects is the combinatorial explosion due to the combined neighborhoods of all objects. A potential solution to this problem is to locally restrict the combination of transitions. If we partition the environment of the moving agent into small parts, then only the neighborhood transition graph for these smaller spatial configurations needs to be considered.

2.2 Dipole Relation Algebra

In [32] a qualitative spatial calculus dealing with two directed line segments, in the following also called *dipole*, as basic entities was presented. These dipoles are used for representing spatial objects with intrinsic orientation. A dipole A is defined by two points, the start point s_A and the end point e_A. The presented calculus deals with the orientation of two dipoles. An example of the relation $lrrr$ is shown in Fig. 1. The four letters denote the relative position (e.g. *left* or *right*) of one of the points to the other dipole:

$$A\ lrrr\ B := A\ l\ s_B \wedge A\ r\ e_B \wedge B\ r\ s_A \wedge B\ r\ s_A$$

Based on a two dimensional continuous space, $\mathbb{R}^2$, the location and orientation of two different dipoles can be distinguished by representing the relative position of start and end points. This means *left* or *right* and the same *start* or *end* point if no more than three points are allowed on a line, and without this restriction *back*, *interior* and *front* additionally (Fig. 2).

The first view leads to 24 *jointly exhaustive and pairwise disjoint* (jepd) basic relations, i.e. between any two dipoles exactly one relation holds at any time. Additionally they build up a relation algebra with 24 basic relations. These relations build up a quite coarse distinction between different orientations, thus we will call this algebra ($\mathcal{DRA}_c$). A visualization is given in Fig. 4. Because of forming a relation algebra standard constraint-based reasoning techniques can be applied. The unrestricted version leads to a relation algebra with 72 basic relations. We will call this fine grained algebra $\mathcal{DRA}_f$. For a detailed description of the calculus' properties we refer to [32].

[1] 16 potential triangle configurations, but two configurations are geometrically impossible.

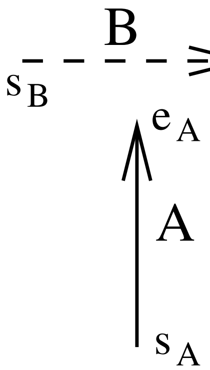

Fig. 1. The lrrr orientation relation between two dipoles

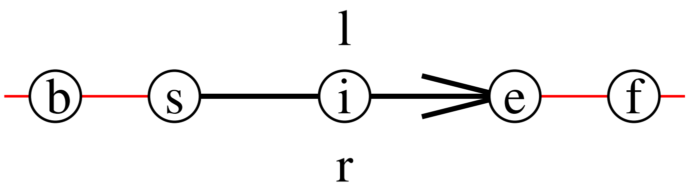

Fig. 2. Extended dipole point relations

2.3 Extended Dipole Relation Algebra

Unfortunately $\mathcal{DRA}_f$ may not be sufficient for robot navigation tasks, because even in this finer grained version many different dipole configurations are pooled in one relation. Thus we extend the representation with additional orientation knowledge and derive a more fine grained relation algebra with additional orientation distinctions. We will call this $\mathcal{DRA}_{fp}$.

Fig. 5 for example visualizes the large configuration space for the *rrrr* relation. This might lead to quite squiggly paths if using these concepts for robot navigation. Other relations being extremely coarse are *llrr*, *rrll* and *llll*. We would expect a more goal directed behavior breaking up the relations by regarding the angle spanned by the two dipoles qualitatively. This gives us an important additional distinguishing feature with four distinctive values. These qualitative distinctions are parallelism (P) or anti-parallelism (A) and mathematically positive and negative angles between A and B, leading to three refining relations for each of the four above mentioned relations (Fig. 6). Thus we call this algebra $\mathcal{DRA}_{fp}$ being an extension of the fine grained relation algebra $\mathcal{DRA}_f$ with additional distinctions based on "parallelism".

For the other relations a '$+$', '$-$', 'P', or 'A' is already determined by the original base relation. We give a complete list of the resulting $\mathcal{DRA}_{fp}$ algebra:

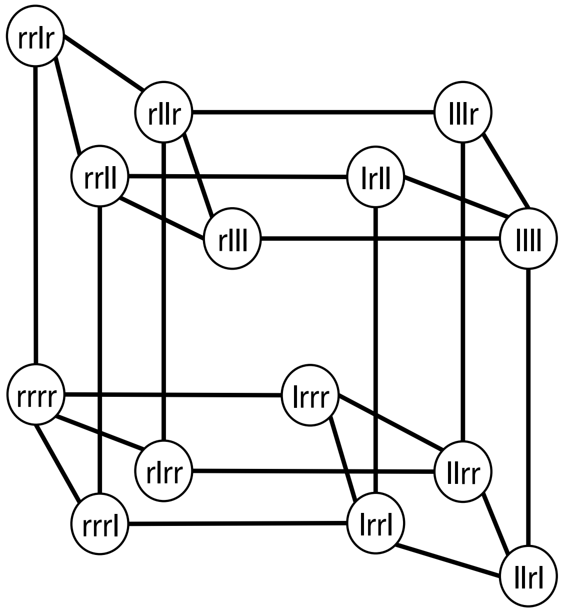

Fig. 3. The conceptual neighborhood graph for the 14 basic relations by Schlieder

1. Original relations from $\mathcal{DRA}_c$:
   (a) Extended relations (12):
   $rrrr+$, $rrrrA$, $rrrr-$, $rrll+$, $rrllP$, $rrll-$, $llrr+$, $llrrP$, $llrr-$, $llll+$, $llllA$, $llll-$
   (b) Unmodified relations (20):
   $rrrl-$, $rrlr+$, $rlrr+$, $rllr+$, $rlll+$, $lrrr-$, $lrrl-$, $lrll-$, $llrl-$, $lllr+$ $ells+$, $errs-$, $lere-$, $rele+$, $slsr+$, $srsl-$, $lsel-$, $rser+$, $seseP$, $esesA$
2. Additional cases on one line[2], $seseP$ and $esesA$ are already defined in 1.(b):
   (a) Basic Allen cases (12):
   $ffbbP$, $efbsP$, $ifbiP$, $bfiiP$, $sfsiP$, $beieP$, $bbffP$, $bsefP$, $biifP$, $iibfP$, $sisfP$, $iebeP$
   (b) Converse cases relative to Allen (12):
   $ffffA$, $fefeA$, $fifiA$, $fbiiA$, $fseiA$, $ebisA$, $iifbA$, $eifsA$, $isebA$, $bbbbA$, $sbsbA$, $ibibA$
3. Other additional cases:
   (a) Without converse (12):
   $lllb+$, $llfl-$, $llbr+$, $llrf-$, $lirl+$, $lfrr-$, $lril-$, $lrri+$, $blrr-$, $irrl-$, $frrr+$, $rbrr+$
   (b) The converse (12):
   $lbll-$, $flll+$, $brll-$, $rfll+$, $rlli-$, $rrlf+$, $illr+$, $rilr-$, $rrbl+$, $rlir+$, $rrfr-$, $rrrb-$

[2] For a relation algebra about this subset of $\mathcal{DRA}_f$ see [40].

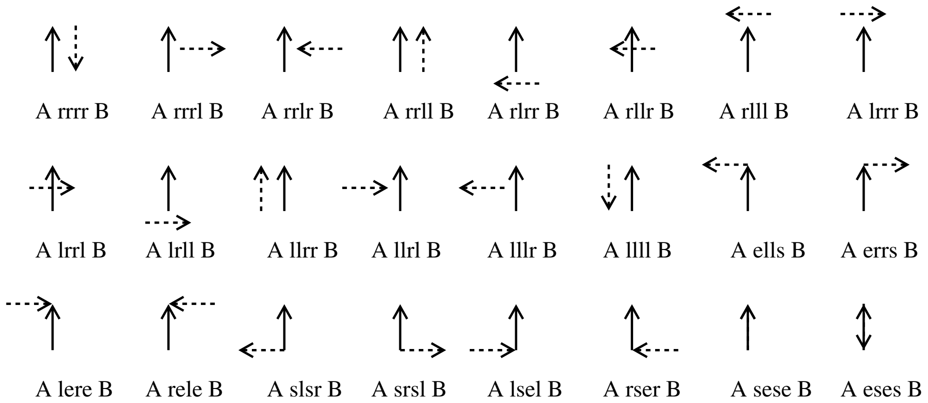

Fig. 4. The 24 atomic relations of the coarse dipole calculus. In the dipole calculus orthogonality is not defined, although the visual presentation might suggest this

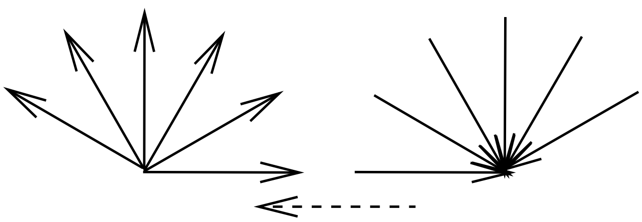

Fig. 5. Pairs of dipoles (A: solid, B: dashed) subsumed by the same relation A(rrrr)B

For lack of space we refer to our web page[3] for the CNH-graphs and CNH-tables for $\mathcal{DRA}_c$, $\mathcal{DRA}_f$ and $\mathcal{DRA}_{fp}$. Restricting to relations suited for robotic navigational tasks where dipoles represent solid objects[4] we end up with only 40 base relations, thus giving us a condensed CNH-graph.

[3] WWW.SFBTR8.UNI-BREMEN.DE/PROJECT/R3/CNH/

[4] Other non solid objects like doorways may also be represented by dipoles.

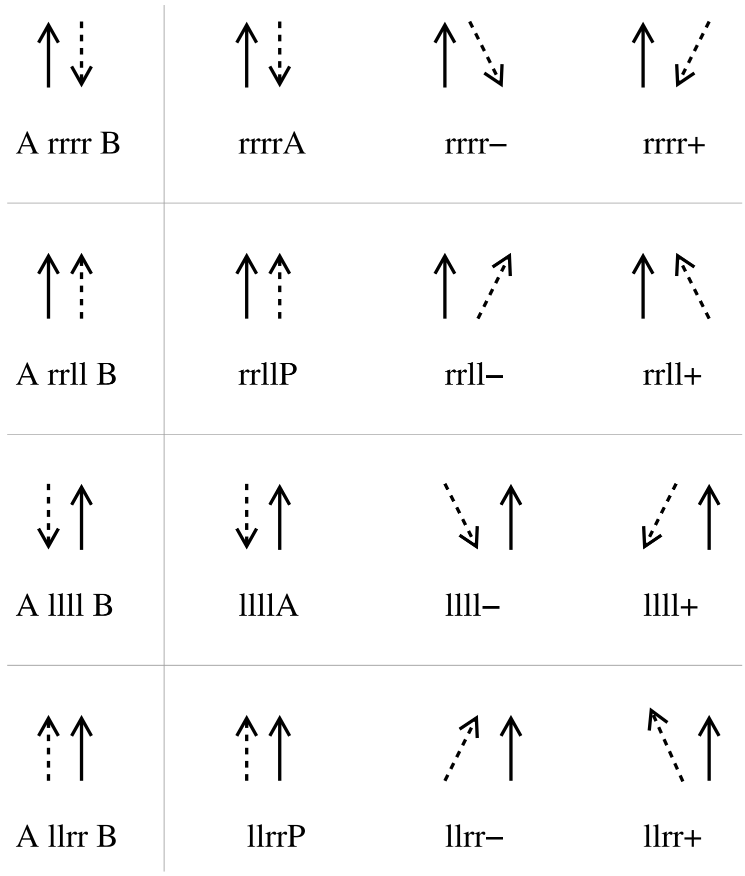

Fig. 6. Refined base relations in $\mathcal{DRA}_{fp}$

3 The Situation Calculus

The situation calculus is a second order language for representing and reasoning about dynamic domains. Although many different variants have been developed from the original framework for dealing with, e.g. concurrency [7], continuous change [20,22] or uncertainty [19], all dialects are based on three sorts: *actions*, *situations* and *fluents*.

All changes in the world are caused by an action a_i in the specific situation s_i resulting in the successor situation s_{i+1}. The special constant S_0 denotes the *initial situation* where no action has been performed before. The binary function $s_{i+1} = do(a_i, s_i)$ starting from S_0 together with a sequence of actions forms a history. Actions are only applicable in the specific situation if preconditions hold which are axiomatized by the predicate $Poss(a, s)$. Fluents are features of the world that might change from situation to situation, e.g. the agents $position$ is changed by a go-action. Two fluent types can be distinguished. Relational fluents describe truth values while functional fluents hold general values and both might change over situations. They are denoted by predicate, or function symbols holding the situation as last argument. The action effects on fluents are axiomatized in so called successor state axioms (SSA) [39]. The general form of an SSA for a relational fluent F with its parameters $\boldsymbol{x}$ is

$$
\begin{aligned}
Poss(a, s) \Rightarrow (F(\boldsymbol{x}, do(a, s)) = true \equiv \\
&\text{the execution of } a \text{ makes } F(\boldsymbol{x}, s) \text{ true} \\
&\vee F(\boldsymbol{x}, s) \text{ already true and } a \text{ makes no change}).
\end{aligned}
$$

With a basic action theory as presented in [27], namely the action precondition axioms, the successor state axioms, the initial situation and an additional unique names axiom a domain model can be formalized.

Golog [28] is a programming language based on the situation calculus for specifying complex tasks like those typically found in robotic scenarios. Golog offers programming constructs well known from imperative programming languages like *sequence*, *if-then-else,*, *while* and *recursive procedures*. Additionally, a *nondeterministic choice* operator is provided to choose from the given alternatives during runtime. Another important difference compared to most other programming languages is the notion of a *test condition*, which in general can be an arbitrary first order sentence.

We give a list of common programming constructs offered by Golog. The e_i denote legal Golog programs:

1. a: primitive actions (actions in the situation calculus)
2. $[e_1, \ldots, e_n]$: sequence of actions
3. $?(c)$: test whether condition c is true, with c denoting an *arbitrary* first order formula
4. $if(c, e_1, e_2)$: conditional execution of e_1 if c evaluates true and e_2 otherwise
5. $while(c, e)$: while c is true e will be executed
6. $e_1|e_2$: nondeterministic action choice, such that either e_1 or e_2 is executed
7. $star(e)$ or $\star(e)$: nondeterministic iteration, i.e. e is repeated an arbitrary number of times
8. $pi(\vartheta, e)$ or $\mathit{\Pi}(\vartheta, e)$: nondeterministic argument choice, i.e. choose an argument term t and proceed with e substituting all occurrences of ϑ with t
9. P_i: procedures, including recursion

Golog programs, also called procedures, can be viewed as macros for complex actions which are mapped onto primitive actions in the situation calculus. With the above given features Golog serves as integrative framework for programming and planning in dynamic domains. Central for the semantics is the ternary relation $Do(\delta, s, s')$ which is a mapping onto a situation calculus formula. Roughly spoken $Do(\delta, s, s')$ means that given a program δ the situation s' is reachable starting in s. Several extensions e.g. dealing with concurrency [7,36], continuous change [22], sensing [8], probabilistic projections [21] or decision theory [11,43] have been presented. Very important for defining our task are sensing and exogenous actions for on-line robot control. Both actions bind the given results to one or more fluents such that they can be used for controlling the agent. With the help of sensing actions the agent is able to obtain environment information actively. If the robot wants to deliver a letter to a specific person he has to check actively whether the person talking to is the right recipient, e.g. by $get\_name\_of\_person(name)$ with the fluent $name$ holding the result afterwards. In contrast exogenous actions are asynchronous events in the environment, e.g. if someone starts talking to the robot an action with the content might be written in the history by $speech\_input("Could\ you...")$. Modeling reacting to speech by a sensing action would lead to a quite introverted agent only willing to react if he "likes to".

4 Examples

We have presented on the one hand the situation calculus as framework for reasoning about action and change, which spatial relations are build on an absolute geometrical coordinate system. On the other hand we presented the line segment based dipole calculus together with its conceptual neighborhood (CNH) graph for reasoning about relative position. The CNH-graph describes possible qualitative transitions between adjacent relative configurations by continuous motion.

Regarding only two dipoles (compare to Fig. 1 with the dashed dipole representing an agent and the solid dipole a static object) in $\mathcal{DRA}_c$ the term $behind$ may be defined by relation *rlrr* and *lrll*, and $front$ by *rlll* and *lrrr*. In the following we will restrict dipoles to representing only solid objects.

4.1 General Assumptions and Definitions

Below we will use our newly developed dipole calculus $\mathcal{DRA}_{fp}$, because we consider $\mathcal{DRA}_f$ not being fine grained enough, especially in the context of turning operations. As stated above the CNH-graph is presented on our web page[4]. We define the symmetric binary relation $cnh(p, q)$ holding if two relations p and q are conceptually neighboring. We denote the set of all defined dipoles in the domain with D.

A simple object is a single dipole. A complex object is a polygon, i.e. a sequence of n dipoles $R_i \in D$ where two consecutive dipoles share a common point. For a closed complex object R_0 and R_n must share a common point as well. How such representations can be extracted efficiently and in a compact manner is shown in [47]. The set of all objects is denoted O.

Modeling a robot domain in the situation calculus at least one fluent $pos(s)$ for holding the recent position of the agent A relative to one object resp. dipole is needed: $pos(s) = \langle r_i, o\rangle$ with $r_i \in \mathcal{DRA}_{fp}$ and $o \in O$, i.e. the relation $A\ (r_i)\ o$ holds. In general there will be more than one dipole, or object present in an environment. Therefore more than one positional fluent relative to different dipoles will be necessary for sound navigation, i.e. $pos_j(s) = \langle r_{i_j}, o_j\rangle$ with $j = 1 \ldots n$ and n representing the number of necessary dipoles. In this paper we will show by example, that not all dipoles are important for doing so. In future we have to investigate which dipoles are essential.

In our examples we consider only the basic navigational action $go(r_i, o)$. The precondition that the agent is not blocked holds at any time. Other actions dealing with relative positional information in a domain are e.g. transporting an object R from current position to destination $\langle r_{dest}, o_{dest}\rangle$: $bring(R, r_{dest}, o_{dest})$ or informational questions about spatial configurations.

Because of restricting dipoles to representing only solid objects we can denote subsets (not necessarily disjoint) of relations suitable for intra-object, agent-object and inter-object relations, regarding a dipole and an object. As defined above the subsequent dipoles of the intra-object description need to share a common point. Therefore only relations containing an e or s are suitable for object descriptions. For the sake of simplicity we omit the case of an internal connection of two dipoles. If we assume the agent not being allowed to touch any other object only relations without sharing a start, end or internal point are applicable. Thus we can define a subset of relations $\mathcal{DRA}_{fp}^{object}$ suitable for intra-object definition.

$$\mathcal{DRA}_{fp} \supset \mathcal{DRA}_{fp}^{object} = \\ \{\text{ells+,errs-,lere-,rele+,slsr+,srsl-,lsel-,rser+}\}$$

For agent-object relations all other relations except relations containing an internal dipole connection are suitable, for inter-object relations all $\mathcal{DRA}_{fp}$ relations except relations with overlapping dipoles may be used.

4.2 Naive Implementation for Two Dipoles

In a first step we show how a CNH structure might be represented in the situation calculus for two dipoles representing an agent A and an arbitrary object R. The successor state axiom for the go-action looks the same as in other domain models without a formal qualitative spatial theory:

$$\begin{aligned} Poss(a, s) \Rightarrow [&pos(do(a, s)) = \langle r_j, o\rangle \equiv \\ &a = go(r_j, o) \vee \\ &[pos(s) = \langle r_j, o\rangle \wedge a \neq go(r_k, o_l)]] \end{aligned}$$

The formula describes the SSA for the go action. If action a is possible in situation s, the fluent $pos(s)$ holds for $\langle r_j, o\rangle$ in the successor situation (i.e. after executing a the agent is r_j relative to object o), iff the go action just defined to go there, or the agent was already in that position in the originating situation and no go action told to go in any other relation to any other object. But the graph structure of the dipole calculus helps us for the definition of the preconditions by exploiting the adjacency of the conceptual neighborhood structure. A movement of the agent to a relative position towards the object is only possible if he is already in a conceptually neighboring configuration. This results in:

$$Poss(go(r_j, o), s) \Leftrightarrow pos(s) = \langle r_i, o\rangle \wedge cnh(r_i, r_j).$$

Assuming an agent A and an object R being in relative position $A(\text{lrrr-})B$ with the goal of being $A(\text{ffffA})B$. The situation calculus and CNH-graph will give the same solution, namely two options to go around R. We sketch the action sequence resp. the transition through neighboring CNH-graph states in Fig. 7.

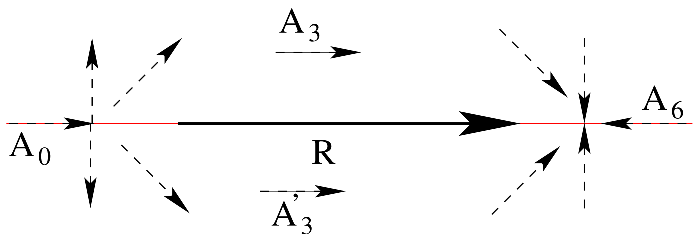

Fig. 7. Simple example with two options for agent A going round object R

4.3 Complex Objects (Going round the Kaaba)

Now we present an example for a complex object. One of the tasks during the hadsch (the great Muslim pilgrimage) is rounding the Kaaba (a cubic building in the main mosque in Mekka) seven times. The knowledge about the Kaaba k (compare Fig. 8) can be represented as follows:

$$R_0(errs-)R_1 \wedge R_1(errs-)R_2 \wedge R_2(errs-)R_3 \wedge R_3(errs-)R_0$$

The agent A may start in position A_0 with $A_0(\mathrm{rrllP})R_0$. At this time the other walls of the Kaaba are of no interest for determining the relative position. Going round the corner of R_0 and R_1 we may get the relations shown in Fig. 9. There are other options traversing the neighborhood graph while turning around, e.g. if the robot starts turning a little earlier. Here we wanted to state the existence of at least one possibility how to turn around the corner.

Looking at all relations for a round trip an analogous situation holds at each corner while the other sides provide no useful knowledge. Thus in this example only two sides are sufficient for describing the relative position of an agent towards the complete object. We expect this being true for more complex, but convex objects, although we have no formal proof so far.

4.4 Going Towards Macro Definitions

After extracting the neighborhoods for one complex action like “turn right” we are now heading for some sort of macro definition such that an agent is able to perform a “turn right” on the basis of line segments and imprecise orientation information. We are now looking at the turn problem from a rather communicative perspective.

Imagine being blind, standing at a wall with the task of turning right at the next corner with arbitrary angle and describing it to an external person. The only sensor is one’s own right hand extended to the right front which can be seen as some sort of coarse “orientation sensor” transferring the task to a robot. One way describing the process of the first right turn in Fig. 8 might be:

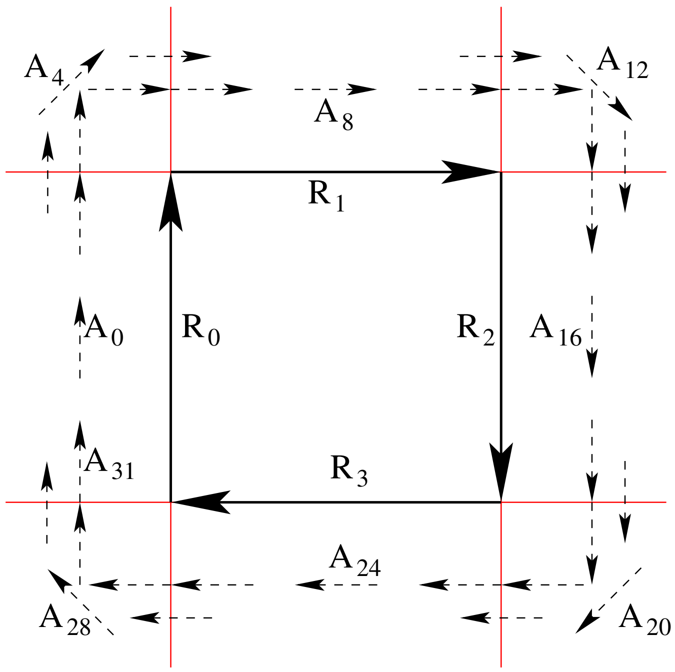

Fig. 8. 32 different qualitative positions an agent A might traverse going round the Kaaba $\{R_0...R_3\}$

1. Position yourself parallel to the wall and follow the wall until you feel an edge (A_1). If one is moving away or coming closer to the wall it has to be adjusted until the robot is again parallel to the wall.
2. Go a little straight ahead so that the edge is to the right of you (A_2), i.e. the next wall comes just into reach on the right side.
3. Turn (in a bow) right around the corner until you are parallel to the next wall (A_3-A_5).
4. Go a little straight ahead until the first wall just gets out of reach (A_6).
5. Go straight ahead until the corner is right behind you (A_7).
6. Follow the wall (A_8).

All the named actions can be modeled as local behaviors and with the help of the base relations presented in $\mathcal{DRA}_{fp}$. If for example loosing parallelism ($A(rrllP)R_0$) to a wall while following it, we have to look whether we have a mathematically positive or negative orientation towards the relating dipole and turn accordingly. We will take such descriptions as a basis for our macro definitions. At first glance the relations of the ($\mathcal{DRA}_{fp}$) might seem to be too fine grained to represent a simple behavior like turning right adequately. But without the additional relations compared to the ($\mathcal{DRA}_c$) we have not found a way for making the transition from one reference dipole to another (from R_0 to R_1) possible, which is necessary to model going round a corner.

R_x / A_y	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
R_0	$rrllP$	$rrllP$	$rrllP$	$rrllP$	$rrll+$	$rrlf+$	$rrlr+$	$rrfr+$	$rrrr+$
R_1	$rrrr-$	$rrrb-$	$rrrl-$	$rrbl-$	$rrll-$	$rrllP$	$rrllP$	$rrllP$	$rrllP$

Fig. 9. The relations $(A_y(r_{y,x})R_x)$ for an idealized turn right

4.5 Macro Definitions

We now have to define an action macro within the framework such that we can model the desired behavior, turning right at a specific corner with the help of neighborhood transitions. Regarding the visualization in Fig. 9 we defined a procedure as shown in Alg. 1. A preliminary to executing this description is being able to percept and distinguish the dipoles.

Turn right at the next opportunity defined in the terms of an initial situation and a goal state might for example lead to the following description:

– $S_0 : pos(R_i, rrllP)$, i.e $A(rrllP)R_i$ with R_i being an aritrary wall the agent stands next to with the heading in the just about correct direction.
– $Goal : A(rrllP)R_j$ if the relation $R_i(errs-)R_j$ holds[5].

This means in the initial situation only the wall to the right (R_i) and the dipole which is the next connected one (R_j) in the direction of movement is relevant. We omit the case where a straight wall might be represented with several dipoles being connected via a relation such as *bsefP*. In the goal situation only the relative position towards R_j is important. Thus we have to look how the transitions from $pos(rrllP, R_i)$ to $pos(rrllP, R_j)$ can be defined. Thus in the beginning R_i is the *main anchor* for the relative position while in the end it is R_j. During the period of turning around the corner described by $R_i(errs+)R_j$ the according relation has to be kept as *auxiliary anchor* in mind. Thus we have to split the $pos(.)$ fluent in pos_{main} and pos_{aux}.

Following this result we have to reformulate the precondition and successor state axiom such that only the main anchor is changed by a movement action and the auxiliary anchor is unaffected:

$$
\begin{aligned}
Poss(go(r_j, o), s) &\Leftrightarrow pos_{main}(s) = \langle r_i, o\rangle \wedge cnh(r_i, r_j) \\
Poss(a, s) &\Rightarrow [pos_{main}(do(a, s)) = \langle r_j, o\rangle \equiv \\
&\qquad a = go(r_j, o) \vee \\
&\qquad [pos_{main}(s) = \langle r_j, o\rangle \wedge a \neq go(r_x, o_x)]]
\end{aligned}
$$

The question now is where R_i and R_j should flip from serving as main anchor to auxiliary anchor, and vice versa. Analyzing Fig. 9 shows the relations from A_0 to A_3 relative to R_0 being stable, whereas A_5 to A_8 is stable for R_1. Therefore the

[5] We omit the symmetric case of $R_i(rele+)R_j$ here.

anchors need to be changed around the relation concerning position A_4. Additionally we have to introduce several new actions which allows us for example to set a specific dipole as auxiliary anchor ($set\_pos_{aux}(R_x)^6$) and to switch between the two anchors ($switch\_pos_{main}\_pos_{aux}$). We also introduce the special turning actions $rotate\_right(o_i, r_{i_j})$ and $rotate\_left(o_i, r_{i_j})$, although they have almost the same formalization as the $go()$ action except not all conceptual neighbors are reachable, namely the ones needing translation to be reached. In Alg. 1 we use $pos_{main}(R_x, r_x)$ as abbreviation for $pos_{main}(s) = \langle R_x, r_x \rangle$. Additionally a coarse environment description as presented for the Kaaba in section 4.3 is given.

In the initial situation we only know the agent being next to R_0, thus defining pos_{main} and no auxiliary dipole is set. Given the task of turning right at the next possibility (compare $turn\_next\_right$ in Alg. 1) we have to bind a fluent R_i to the current value of the main anchor and look for the next right turn R_j in our environment description. As stated above we omit the case of several dipoles representing a straight wall. Now we have the relevant information available for turning right at the next opportunity.

The $turn\_right(R_0, R_1)$ procedure makes the robot turn at the specific corner between R_0 and R_1. In the lines one to three we have to check whether R_0 is the main position anchor as well as R_0 and R_1 really form a right corner. Next we set the auxiliary anchor to R_1. So far we have not checked whether we are in the right orientation relative to R_0. So we have to check and turn accordingly. Now we need R_1 as main anchor (line 10). The rest of the procedure is coded straightforward according to Fig. 9. In the end R_1 is the main anchor for the robot's position and we do not need pos_{aux} anymore. After executing this procedure the agent is in position A_8 according to Fig. 8.

5 Conclusion and Outlook

We presented our approach that showed how the concept of conceptual neighborhood can be exploited for reasoning about relative positional information in the situation calculus in the absence of precise quantitative information. We introduced an extended dipole relation algebra $\mathcal{DRA}_{fp}$ better suited for spatial navigation. We expect that every qualitative calculus can be translated in a straightforward manner naively onto preconditions and successor state axioms using its conceptual neighborhood feature. We have shown by example that not all dipoles of a complex object are necessary to determine the relative position towards the object. We expect the results for connected complex objects being applicable for several not connected dipoles. Additionally we extracted several subsets of the base relations for representing a complex object and dynamic agent behavior.

Future work will deal with the question of how to keep the position representation small for more than one dipole respectively object. A naive implementation would lead to a combinatorial explosion, because the relative position of the agent has to be traced for every single dipole. The presented approach with the two anchors will be problematic in general, a set of nearest dipoles as a basis for the positional fluents will be more adequate. A coarse grid partitioning the eight directions *ahead, ahead-left, adjacent-left, behind-left, behind, behind-right, adjacent-right and ahead-right* and the agent in

[6] We are using the term *nil* to reset the fluent.

Algorithm 1 A first approach defining a *turn right* macro

Initial Situation S_0:

$pos_{main}(R_0, rrllP)$, i.e. $A(rrllP)R_0$ and
$pos_{aux}(nil, nil)$

procedure $turn\_next\_right$

$(\Pi\ R_i)[\ pos_{main}(R_x, r_x), ?(R_x = R_i),$
$\quad(\Pi\ R_j)[\ ?((R_i(errs-)R_j \vee (R_i(rele+)R_j)),$
$\quad\quad turn\_right(R_i, R_j)]]$

procedure $turn\_right(R_0, R_1)$

$?(R_0(errs-)R_1 \vee R_0(rele+)R_1),$
$(\Pi R_i)[pos_{main}(R_x, r_x),$
$?(R_i = R_0)],$
$set\_pos_{aux}(R_1),$
if $(r_x = rrll-)$ **then**
$\quad rotate\_right(R_0, rrllP)$
else if $(r_x = rrll+)$ **then**
$\quad rotate\_left(R_0, rrllP)$
end if
$switch\_pos_{main}\_pos_{aux}$, // $pos_{main} = R_1$
$go(rrrb-, R_1)$, // (A1)
$go(rrrl-, R_1)$, // (A2)
$go(rrbl-, R_1)$, // (A3)
$go(rrll-, R_1)$, // (A4a)
$switch\_pos_{main}\_pos_{aux}$, // $pos_{main} = R_0$
$go(rrll+, R_0)$, // (A4b)
$switch\_pos_{main}\_pos_{aux}$, // $pos_{main} = R_1$
$go(rrllP, R_1)$, // (A5a)
$switch\_pos_{main}\_pos_{aux}$, // $pos_{main} = R_0$
$go(rrlf+, R_0)$, // (A5b)
$go(rrlr+, R_0)$, // (A6)
$go(rrfr-, R_0)$, // (A7)
$go(rrrr+, R_0)$, // (A8)
$switch\_pos_{main}\_pos_{aux}$, // $pos_{main} = R_1$
$set\_pos_{aux}(nil)$.

the middle as presented in [9,34] will serve as a starting point. For each direction the most valuable dipole or object will be held together with the relation between the agent and the dipole. The definition of the term 'most valuable dipole' is one of the major tasks to solve in this context. We will also look on the effects allowing dipoles to represent non-solid entities, e.g. doorways, and potentials to define some sort of general macro definitions for $turnLeft$ and $turnRight$ or $GoAround$ by paths in the conceptual neighborhood graph.

6 Acknowledgment

The authors like to thank Diedrich Wolter, Christian Freksa, Jochen Renz, and Marco Ragni for fruitful discussions and impulses. And we would like to thank Marc-Björn Seidel for computing the neighborhood graph for different calculi, as well as the anonymous reviewers for their valuable comments. Our work was supported by the DFG Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition.

References

1. John Bateman and Scott Farrar. Modelling models of robot navigation using formal spatial ontology. In *Proceedings of Spatial Cognition Conference 2004 (this volume)*, 2004.
2. Brandon Bennett and Antony P. Galton. A unifying semantics for time and events. *Artificial Intelligence*, 153(1-2):13–48, March 2004.
3. Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hahnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3–55, 1999.
4. Anthony G. Cohn. Qualitative spatial representation and reasoning techniques. In Gerhard Brewka, Christopher Habel, and Bernhard Nebel, editors, *KI-97: Advances in Artificial Intelligence, 21st Annual German Conference on Artificial Intelligence, Freiburg, Germany, September 9-12, 1997, Proceedings*, volume 1303 of *Lecture Notes in Computer Science*, pages 1 – 30, Berlin, 1997. Springer.
5. Anthony G. Cohn and Shyamanta M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46(1-2):1–29, 2001.
6. E. Davis. Continuous shape transformation and metrics of shape. In *Fundamenta Informaticae*, volume 46, pages 31–54. May 2001.
7. Giuseppe De Giacomo, Yves Lésperance, and Hector J. Levesque. ConGolog, A concurrent programming language based on situation calculus. *Artificial Intelligence*, 121(1–2):109–169, 2000.
8. Giuseppe De Giacomo and Hector Levesque. An incremental interpreter for high-level programs with sensing. In Hector J. Levesque and Fiora Pirri, editors, *Logical Foundation for Cognitive Agents: Contributions in Honor of Ray Reiter*, pages 86–102. Springer, Berlin, 1999.
9. J. Fernyhough, A.G. Cohn, and D. Hogg. Constructing qualitative event models automatically from video input. *Image and Vision Computing*, 18:81–103, 2000.
10. A. Ferrein, C. Fritz, and G. Lakemeyer. On-line decision-theoretic golog for unpredictable domains. In *Proc. of 4th International Cognitive Robotics Workshop*, 2004.
11. Alexander Ferrein, Christian Fritz, and Gerhard Lakemeyer. Extending DTGolog with Options. In *Proc of the 18th International Joint Conference on Artificial Intelligence*, 2003.

12. Christian Freksa. Conceptual neighborhood and its role in temporal and spatial reasoning. In Madan G. Singh and Luise Travé-Massuyès, editors, *Proceedings of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning*, pages 181 – 187, North-Holland, Amsterdam, 1991. Elsevier.
13. Christian Freksa. Using orientation information for qualitative spatial reasoning. In A. U. Frank, I. Campari, and U. Formentini, editors, *Theories and methods of spatio-temporal reasoning in geographic space*, pages 162 – 178. Springer, Berlin, 1992.
14. Christian Freksa. Spatial Cognition - An AI Prespective. In *Proceedings of 16th European Conference on AI (ECAI 2004)*, 2004.
15. Christian Fritz. Integrating decision-theoretic planning and programming for robot control in highly dynamic domains. Master's thesis, RWTH Aachen (Knowledge-based Systems Group), Aachen, Germany, 2003.
16. Anthony Galton. Space, time, and movement. In Oliviero Stock, editor, *Spatial and Temporal Reasoning*, pages 321–352. Kluwer Academic Publishers, Dordrecht, 1997.
17. Antony Galton. *Qualitative Spatial Change*. Oxford University Press, 2000.
18. A.P. Galton. Continuous motion in discrete space. In A.G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proc. 7th Internat. Conf. on Principles of Knowledge Representation and Reasoning (KR2000)*, pages 26–37. Morgan Kaufmann, San Francisco, CA, 2000.
19. Henrik Grosskreutz. Probabilistic projection and belief update in the pGOLOG framework. In *CogRob-00 at ECAI-00*, 2000.
20. Henrik Grosskreutz and Gerhard Lakemeyer. cc-Golog: Towards more realistic logic-based robot controllers. In *AAAI-00*, 2000.
21. Henrik Grosskreutz and Gerhard Lakemeyer. Turning high-level plans into robot programs in uncertain domains. In *ECAI-00*, 2000.
22. Henrik Grosskreutz and Gerhard Lakemeyer. On-line execution of cc-Golog plans. In *IJCAI-01*, 2001.
23. Shyamanta M. Harzarika and Anthony G. Cohn. Qualitative spatio-temporal continuity. *Lecture Notes in Computer Science*, 2205:92–107, 2001.
24. Amar Isli and Anthony G. Cohn. An algebra for cyclic ordering of 2d orientations. In *AAAI/IAAI*, pages 643–649, 1998.
25. B. J. Kuipers. Representing knowledge of large-scale space. Technical Report 418, 1977.
26. P Ladkin and R Maddux. On binary constraint problems. *Journal of the Association for Computing Machinery*, 41(3):435–469, 1994.
27. H. Levesque, F. Pirri, and R. Reiter. Foundations for the situation calculus. In *Linköping Electronic Articles in Computer and Information Science*, volume 3. 1998.
28. 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.
29. J. McCarthy. Situations, actions and causal laws. Technical report, Stanford University, 1963.
30. R. Moratz, T. Tenbrink, J. Bateman, and K. Fischer. Spatial knowledge representation for human-robot interaction. In Christian Freksa, W. Brauer, C. Habel, and K. F. Wender, editors, *Spatial Cognition III*. Springer, Berlin, Heidelberg, 2002.
31. Reinhard Moratz, Bernhard Nebel, and Christan Freksa. Qualitative spatial reasoning about relative position: The tradeoff between strong formal properties and successful reasoning about route graphs. In Christian Freksa, Wilfried Brauer, Christopher Habel, and Karl Friedrich Wender, editors, *Spatial Cognition III*, volume 2685 of *Lecture Notes in Artificial Intelligence*, pages 385–400. Springer, Berlin, Heidelberg, 2003.
32. Reinhard Moratz, Jochen Renz, and Diedrich Wolter. Qualitative spatial reasoning about line segments. In *Proceedings of ECAI 2000*, pages 234–238, 2000.

33. Reinhard Moratz and Jan Oliver Wallgrün. Propagation of distance and orientation intervals. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3245–3250, 2003.
34. A. Mukerjee and G. Joe. A qualitative model for space. In *Proc. of AAAI-90*, pages 721–727, Boston, MA, 1990.
35. Philippe Muller. A qualitative theory of motion based on spatio-temporal primitives. In Anthony G. Cohn, Lenhart Schubert, and Stuart C. Shapiro, editors, *KR'98: Principles of Knowledge Representation and Reasoning*, pages 131–141. Morgan Kaufmann, San Francisco, California, 1998.
36. Javier Pinto. *Temporal Reasoning in the Situation Calculus*. PhD thesis, Department of Computer Science, University of Toronto, 1994.
37. D. A. Randell and A. G. Cohn. Modelling topological and metrical properties of physical processes. In R. J. Brachman, H. J. Levesque, and R. Reiter, editors, *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, pages 357–368. Morgan Kaufmann, 1989.
38. David A. Randell, Zhan Cui, and Anthony Cohn. A spatial logic based on regions and connection. In Bernhard Nebel, Charles Rich, and William Swartout, editors, *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pages 165–176. Morgan Kaufmann, San Mateo, California, 1992.
39. R. Reiter. *Knowledge in Action*. MIT Press, 2001.
40. Jochen Renz. A spatial odyssey of the interval algebra: 1. directed intervals. In *IJCAI*, pages 51–56, 2001.
41. Thomas Röfer. Route navigation using motion analysis. In Christian Freksa and David M. Mark, editors, *Cosit 1999*, pages 21–36. Springer, Berlin, 1999.
42. C. Schlieder. Reasoning about ordering. In A. U. Frank and W. Kuhn, editors, *Spatial Information Theory - A Theoretical Basis for GIS (COSIT'95)*, pages 341–349. Springer, Berlin, Heidelberg, 1995.
43. Mikhail Soutchanski. An on-line decision-theoretic golog interpreter. In *IJCAI*, pages 19–26, 2001.
44. Thora Tenbrink, Kerstin Fischer, and Reinhard Moratz. Spatial strategies in linguistic human-robot communication. In Christian Freksa, editor, *KI-Journal: Special Issue on Spatial Cognition*. arenDTaP Verlag, 2002.
45. R.F. Wang and E.S. Spelke. Updating egocentric representations in human navigation. *Cognition*, 2000.
46. R.F. Wang and E.S. Spelke. Human spatial representation: Insights from animals. *Trends in Cognitive Sciences*, 6(9):376–382, 2002.
47. Diedrich Wolter and Longin Jan Latecki. Shape matching for robot mapping. In Chengqi Zhang, Hans W. Guesgen, and Wai K. Yeap, editors, *Proceedings of 8th Pacific Rim International Conference on Artificial Intelligence*, Auckland, New Zealand, August 2004.