J Intell Robot Syst (2007) 48:55–78
DOI 10.1007/s10846-006-9099-4

Qualitative Spatial Reasoning with Conceptual Neighborhoods for Agent Control

Frank Dylla · Jan Oliver Wallgrün

Received: 16 March 2006 / Accepted: 03 September 2006 /
Published online: 8 December 2006

Abstract Research on qualitative spatial reasoning has produced a variety of calculi for reasoning about orientation or direction relations. Such qualitative abstractions are very helpful for agent control and communication between robots and humans. Conceptual neighborhood has been introduced as a means of describing possible changes of spatial relations which e.g. allows action planning at a high level of abstraction. We discuss how the concrete neighborhood structure depends on application-specific parameters and derive corresponding neighborhood structures for the $\mathcal{OPRA}_m$ calculus. We demonstrate that conceptual neighborhoods allow resolution of conflicting information by model-based relaxation of spatial constraints. In addition, we address the problem of automatically deriving neighborhood structures and show how this can be achieved if the relations of a calculus can be modeled in another calculus for which the neighborhood structure is known.

Key words conceptual neighborhood · model-based relaxation · qualitative spatial reasoning

1 Introduction

Qualitative spatial reasoning (QSR) is an established field of research investigating qualitative representations of space that abstract from the details of the physical world together with reasoning techniques that allow predictions about spatial relations, even when precise quantitative information is not available [2]. QSR is typically realized in form of calculi over sets of spatial relations (e.g. *left-of* or *north-of*). These

F. Dylla (✉) · J. O. Wallgrün
SFB/TR 8 Spatial Cognition, Universität Bremen, Bibliothekstr. 1,
28359 Bremen, Germany
e-mail: dylla@sfbtr8.uni-bremen.de

J. O. Wallgrün
e-mail: wallgruen@sfbtr8.uni-bremen.de

calculi are well-suited to serve as simple and efficient abstractions of the world, e.g. for robot navigation [5] or communication purposes [16].

A multitude of spatial calculi has been proposed during the last two decades, focusing on different aspects of space (mereotopology, orientation, distance, etc.) and dealing with different kinds of objects (points, line segments, extended objects, etc.). The two main research directions in QSR are topological reasoning about regions [7, 25, 28] and positional reasoning about configurations of point objects [8, 10, 15, 18, 27] or line segments [5, 20, 30]. Calculi dealing with such information have been well investigated over the recent years and provide general and sound reasoning mechanisms. An overview is given in [3].

An important concept in the context of robot navigation and agent control—and the main topic of this text—is the notion of conceptual neighborhood between spatial relations. Conceptual neighborhood has been introduced as a means of describing possible changes of spatial relations which e.g. allows action planning at a high level of abstraction.

As we will show in this text the concrete neighborhood structure, however, depends very much on application-specific parameters like what kind of continuous transformations have to be considered for the objects involved (e.g. locomotion, deformation, etc.), whether objects can be transformed simultaneously, or whether two objects can occupy the same space or not. The exemplary calculus we will investigate in this text is the Oriented Point Relation Algebra ($\mathcal{OPRA}_m$) with adjustable granularity [17, 18] for reasoning about the orientation relations between oriented points. We will analyze its conceptual neighborhood structures as they arise under different conditions in the context of robot navigation.

After this analysis, we will demonstrate one way in which conceptual neighborhoods can be beneficially employed for controlling mobile robots, namely the application of conceptual neighborhoods for a model-based resolution of conflicts in spatial information stemming from different knowledge sources. Standard constraint satisfaction techniques for relational constraints [14] will be used to check for consistency, while the conceptual neighborhoods will be used to incrementally relax the spatial constraints until an optimal consistent relaxation is found with respect to an application-dependent cost or distance function.

Since conceptual neighborhood structures depend on the given scenario and, in addition, qualitative spatial calculi themselves are often developed or adapted for a specific task, methods for automatically deriving or verifying properties like composition tables and neighborhood structures for new calculi are required [6]. We address this problem by showing that neighborhood structures for two other orientation calculi, the FlipFlop calculus [15] and the fine-grained Dipole Relation Algebra [5], can be automatically computed by first modeling the relations of these calculi in $\mathcal{OPRA}_m$, then generating neighboring configurations in $\mathcal{OPRA}_m$, and finally translating the consistent ones back into the original calculus.

The text is structured as follows: We begin by briefly introducing the most relevant concepts with respect to QSR and in particular the $\mathcal{OPRA}_m$ calculus in Section 2. In the next section, we present the notion of conceptual neighborhood structures and investigate the different neighborhood structures of $\mathcal{OPRA}_m$. Section 4 is concerned with the application of conceptual neighborhoods for resolving conflicting information by relaxation. And in Section 5, we address the problem of deriving conceptual neighborhood structures automatically.

2 Qualitative Spatial Reasoning and the $\mathcal{OPRA}_m$ Calculus

In this section, we give a brief overview on spatial calculi and qualitative spatial reasoning and introduce the Oriented Point Relation Algebra ($\mathcal{OPRA}_m$), which will accompany us through the rest of the text. The $\mathcal{OPRA}_m$ calculus has been chosen as it is algebraically well defined and thus well qualified for being integrated into robot control architectures.

2.1 Qualitative Spatial Calculi

A qualitative spatial calculus defines operations on a finite set $\mathcal{R}$ of spatial relations, like *left-of*, *north-of*, *overlap*, etc. The spatial relations are defined over a usually infinite set of spatial objects, the domain D (e.g. points, line segments, regions, etc.). In this text, we will mainly consider *binary calculi* in which $\mathcal{R}$ consists of binary relations $R \subseteq D \times D$.

The set of relations $\mathcal{R}$ of a spatial calculus is typically derived from a jointly exhaustive and pairwise disjoint (JEPD) set of *base relations* $\mathcal{BR}$ so that each pair of objects from D is contained in exactly one relation from $\mathcal{BR}$. Every relation in $\mathcal{R}$ is a union of a subset of the base relation. Since spatial calculi are typically used for constraint reasoning and unions of relations correspond to disjunction of relational constraints, it is common to speak of disjunctions of relations as well and write them as sets $\{B_1, ..., B_n\}$ of base relations. $\mathcal{R}$ is then either taken to be the powerset $2^{\mathcal{BR}}$ of the base relations (all unions of base relations) or a subset of the powerset. In order to be usable for constraint reasoning, $\mathcal{R}$ should contain at least the base relations, the empty relation $\emptyset$, the universal relation $U = D \times D$, and the identity relation $Id = \{(x, x) | x \in D\}$. $\mathcal{R}$ also needs to be closed under the operations defined in the following.

As the relations are subsets of tuples from the same Cartesian product, the set operations union, intersection, and complement can directly be applied:

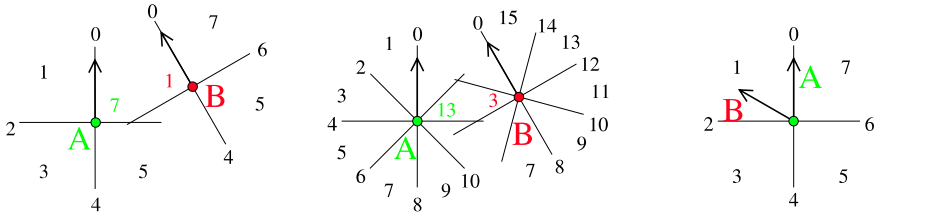
$$\begin{array}{ll} \text{Union:} & R \cup S = \{\ (x, y) \mid (x, y) \in R \vee (x, y) \in S\ \} \\ \text{Intersection:} & R \cap S = \{\ (x, y) \mid (x, y) \in R \wedge (x, y) \in S\ \} \\ \text{Complement:} & \overline{R} = U \setminus R = \{\ (x, y) \mid (x, y) \in U \wedge (x, y) \notin R\ \} \end{array}$$

where R and S are relations from $\mathcal{R}$.

In addition, two more operations are defined which allow derivation of new facts from given information, *conversion* and *composition*:

$$\begin{array}{ll} \text{Converse:} & R^{\smile} = \{\ (y, x) \mid (x, y) \in R\ \} \\ \text{Composition:} & R \circ S = \{\ (x, z) \mid \exists y \in D : ((x, y) \in R \wedge (y, z) \in S)\ \} \end{array}$$

The composition operation is especially important for constraint reasoning since it describes what relations may hold between objects A and C given what is known about the relation between A and B and the relation between B and C. For instance, from knowing that A is *north-of* B and B is *north-of* C it follows that A is *north-of* C as well. The composition operation is often given in form of look-up tables called *composition tables*.



(a) $m = 2$: $\vec{A}\ {}_2\angle^1_7\ \vec{B}$ (b) $m = 4$: $\vec{A}\ {}_4\angle^3_{13}\ \vec{B}$ (c) case where $\vec{A}$ and $\vec{B}$ coincide: $\vec{A}\ {}_2\angle 1\ \vec{B}$

Figure 1 Two oriented points related at different granularities.

2.2 $\mathcal{OPRA}_m$: A Calculus for Reasoning about Oriented Points

The domain of the Oriented Point Relation Algebra ($\mathcal{OPRA}_m$) [17, 18] is the set of oriented points (points in the plane with an additional direction parameter). The calculus relates two oriented points with respect to their relative orientation towards each other. An oriented point $\vec{O}$ can be described by its Cartesian coordinates $x_O, y_O \in \mathbb{R}$ and a direction $\phi_{\vec{O}} \in [0, 2\pi]$ with respect to an absolute reference direction and thus $D = \mathbb{R}^2 \times [0, 2\pi]$.

The $\mathcal{OPRA}_m$ calculus is suited for dealing with objects that have an intrinsic front or move in a particular direction and can be abstracted as points. The exact set of base relations distinguished in $\mathcal{OPRA}_m$ depends on the granularity parameter $m \in \mathbb{N}$. For each of the two related oriented 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 $\mathcal{OPRA}_m$ base relation $rel_{\mathcal{OPRA}_m}$ consists of a pair (i, j) where i is the number of the region of $\vec{A}$ which contains $\vec{B}$, while j is the number of the region of $\vec{B}$ that contains $\vec{A}$. These relations are usually written as $\vec{A}\ {}_m\angle^j_i\ \vec{B}$ with $i, j \in \mathcal{Z}_{4m}$.[1] 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 base relations called *same relations* describe situations in which both oriented points coincide. In these cases, the relation is determined by the number s of the region of $\vec{A}$ into which the orientation arrow of $\vec{B}$ falls (as illustrated in Figure 1c). These relations are written as $\vec{A}\ {}_2\angle s\ \vec{B}$ ($\vec{A}\ {}_2\angle 1\ \vec{B}$ in the example).

The complete set $\mathcal{R}$ of $\mathcal{OPRA}_m$ relations is the powerset of the base relations described above.

2.3 $\mathcal{OPRA}_m$ Notations and Abbreviations

In the following we will on the one hand deal with *normal points*, e.g. P being defined by their position in the plane ($P = (x_P, y_P) \in \mathbb{R}^2$). On the other hand, we will talk about *oriented points* as required for the $\mathcal{OPRA}_m$ calculus, written as $\vec{O}$.

[1] $\mathcal{Z}_{4m}$ defines a cyclic group with $4m$ elements.

The following notations will be used assuming A, B and C are normal points: The direction ϕ^{BC} is defined as the direction from B towards C. We write $\vec{A}^{BC}$ for the oriented point $((x_A, y_A), \phi^{BC})$. It has the same position as the normal point A and the direction ϕ^{BC}. We just write $\vec{A}$ if the direction is unknown or unspecified, e.g. if we want to define an oriented point that coincides with A but can have an arbitrary direction. Note that $\vec{A}^{AB}$, $\vec{A}^{AC}$, and $\vec{A}$ are three different oriented points coinciding in position but possibly differing in orientation. Additionally, we want to emphasize that oriented point names like $\vec{A}^{AC}$ are only identifiers we use for making their role intuitively comprehensible. The knowledge that one oriented point either coincides with or is oriented towards another has to be explicitly represented by respective relations.

As mentioned, we will speak about disjunctions of base relations instead of unions and write them as sets. We will use the abbreviation $\vec{A}\ {}_m\angle^{\{k-l\}}_{\{i-j\}}\ \vec{B}$ with $i, j, k, l \in \mathcal{Z}_{4m}$ for the disjunction

$$\bigvee_{a=i}^{j} \bigvee_{b=k}^{l} \vec{A}\ {}_m\angle^b_a\ \vec{B}.$$

A $*$ abbreviates all members 0 to $(4m-1)$ of $\mathcal{Z}_{4m}$ and $\{i, j\}$ a disjunction of i and j such that for example $\vec{A}\ {}_m\angle^*_{\{i,j\}}\ \vec{B}$ denotes

$$\left(\bigvee_{b=0}^{4m-1} \vec{A}\ {}_m\angle^b_i\ \vec{B}\right) \vee \left(\bigvee_{b=0}^{4m-1} \vec{A}\ {}_m\angle^b_j\ \vec{B}\right).$$

2.4 Constraint Reasoning with Spatial Calculi

The relations $\mathcal{R}$ of a spatial calculus are often used to formulate constraints about the spatial configuration of objects from the domain of the calculus. The resulting spatial constraint satisfaction problem (CSP) then consists of a set of variables $V = \{v_1, ..., v_n\}$ (one for each spatial object considered) and a set of constraints $C_1, ..., C_m \in \mathcal{R}$. Each variable v_i can take values from the domain of the utilized calculus. CSPs are often described as constraint networks which are complete labeled graphs $CN =< V, l >$ where the node set is the set of variables of the CSP and the labeling function $l : V \times V \to \mathcal{R}$ labels each edge with the constraining relation from the calculus. Figure 2 shows a constraint network for $\mathcal{OPRA}_2$.

A CSP is consistent if an assignment for all variables to objects of the domain can be found, that satisfies all the constraints. Spatial CSPs usually have infinite domains and thus backtracking over the domains cannot be used to determine

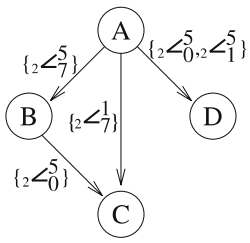


Figure 2 A constraint network over $\mathcal{OPRA}_m$ relations. All edges not shown are labeled with universal relation U and thus unconstrained.

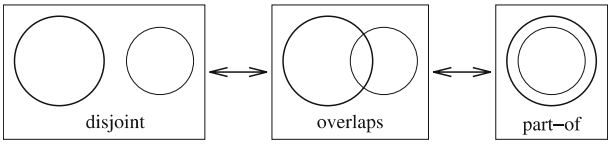


Figure 3 The relations of a simple region based calculus arranged as a neighborhood graph.

consistency. Therefore, special techniques for CSPs with relational constraints have been developed [14].

Besides consistency, weaker forms of consistency called *local consistencies* are of interest in QSR, as they can be utilizable to decide or approximate consistency under specific conditions. One important form of local consistency called *path-consistency* is directly connected to the composition operation as it means that for every triple of variables each consistent evaluation of the first two variables can be extended to the third variable in such a way that all constraints are satisfied. Path-consistency can be enforced in $O(n^3)$ time for binary constraints where n is the number of variables, for instance with the algorithm by van Beek [31].[2]

3 Conceptual Neighborhood and Robot Navigation

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 [29]. In this section, we will introduce the notion of conceptual neighborhood and neighborhood-based reasoning for modeling how the world could evolve in terms of transitions between qualitative relations. We will investigate continuous transformations, a fundamental concept for the definition of conceptual neighborhoods, and discuss the term in the context of robot motion capabilities and other relevant properties of the objects involved. Based on these, we will derive different neighborhood structures for $\mathcal{OPRA}_m$.

3.1 Conceptual Neighborhood

The notion of conceptual neighborhood has been introduced by Freksa [9, 10]. Two spatial relations of a qualitative spatial calculus are conceptually neighbored if they can be continuously transformed into each other without resulting in a third relation in between. For instance, imagine two disks A and B in the plane which can move in arbitrary direction. If we distinguish the three relations *disjoint*, *overlaps*, and *part-of* (see Figure 3), it is possible to directly get from relation A *disjoint* B to A *overlaps* B, e.g. by continuously moving A. In contrast, going from A *disjoint* B to A *part-of* B by continuous motion is not possible without passing through the relation A *overlaps* B. *Disjoint* and *overlaps* are therefore conceptual neighbors, written as *disjoint* $\sim$ *overlaps*, while *disjoint* and *part-of* are not (*disjoint* $\not\sim$ *part-of*).

[2]Since this is a syntactic algorithm purely based on the defined composition and converse operations, it only computes path-consistency if the calculus at hand meets specific conditions (cf. [26] for details).

The conceptual neighborhood relation between the base relations $\mathcal{BR}$ of a qualitative calculus is often described in form of the *conceptual neighborhood graph* $\mathcal{CNG} =< \mathcal{BR}, \sim >$ as illustrated in Figure 3. For convenience, we also introduce a function $\mathtt{cn} : \mathcal{BR} \rightarrow 2^{\mathcal{BR}}$ which yields all conceptual neighbors for a given base relation b:

$$\mathtt{cn}(b) = \{b' | (b, b') \in \sim\}$$

A *conceptual neighborhood* is a set of base relations which is connected in the $\mathcal{CNG}$. Later in the text, we utilize a function that takes a relation given in form of a set S of base relations and yields a coarser relation in which all conceptual neighbors of the base relations have been added. This function called `relax` is defined as follows:

$$\mathtt{relax}(S) = \left(\bigcup_{b \in S} \mathtt{cn}(b) \right) \cup S$$

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 [11]. Spatial neighborhoods are very natural perceptual and cognitive entities. However, the term continuous with regard to transformations needs a grounding in spatial change over time. Different kinds of transformations considered like locomotion, growing or shrinking, or deformation will result in different neighborhood structures. We will look at this in more detail in the next section.

In the context of robot navigation, we are particularly interested in continuous locomotion of the robot. The movement of an agent over time can be modeled qualitatively as a sequence of neighboring spatial relations which hold for adjacent time intervals. The edges in the $\mathcal{CNG}$ can be labeled by qualitative actions of the agent that cause this particular neighborhood transition (e.g. turn left until a change in the perceived spatial relation occurs). This facilitates action planning on a high level of abstraction. Using the neighborhood graph for navigation has for instance been proposed in [30]. A more detailed consideration of neighborhood-based planning is presented in [5].

The idea of conceptually neighbored relations between two objects can be extended to configurations involving more than two objects which leads to neighborhood graphs of complex configurations. In these, spatial transformations from a start configuration to a goal configuration can be determined.

Modeling the relations from a naive point of view, i.e. by keeping track of all relations between all objects, leads to combinatorial explosion, as for example shown in [24]. This can be seen as an allocentric approach. One way to reduce these complexity issues is to shift to an egocentric perspective by only considering neighborhoods of relations to a selected set of reliably recognizable objects [5].

3.2 Continuous Transformation

The term *continuous transformation* is a central concept in the definition of conceptual neighborhood. Detailed investigations on different aspects of continuity have been presented in [1, 4, 12, 13, 21]. The original definition of conceptual neighborhood originates from work on time intervals and therefore only the continuous transformations shortening and lengthening of intervals were considered. When

transferring conceptual neighborhood to spatial relations, only vague discriminations were made between different types of transformations, e.g. between transformations in size or transformations in position, although different types of neighborhoods were already mentioned in [10]. For navigation and action planning it is crucial that the $\mathcal{CNG}$s reflect the capabilities of the agent so that neighborhood induces direct reachability in the physical world.

Overall, three main aspects affect the neighborhood structure for a given spatial calculus in the context of robot navigation:

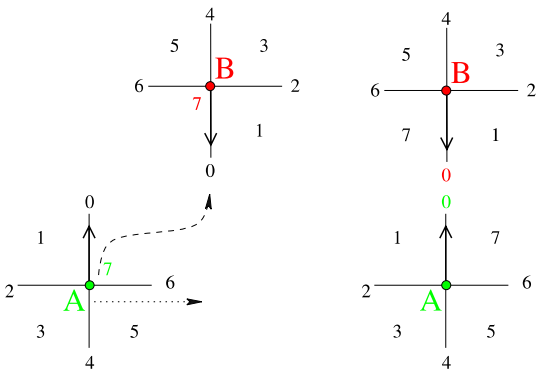
- The robot kinematics (motion capabilities)
- Whether the objects may move simultaneously
- Whether objects may coincide in position or not (superposition)

Restrictions in motion capabilities and number of objects moving will affect which relations are connected in the $\mathcal{CNG}$. To give an example, let us assume that $\mathcal{OPRA}_2$ relation $\vec{A}\ {}_2\angle_7^7\ \vec{B}$ holds between an agent A and some static object B. The orientations correspond to the intrinsic fronts of both objects (cf. Figure 4). If our robot is equipped with an omnidrive allowing it to drive sideways, it can reach configuration $\vec{A}\ {}_2\angle_0^0\ \vec{B}$ directly by moving to the right side. A robot only outfitted with a differential drive has to traverse (notated as $\rightsquigarrow$) other configurations before reaching the desired configuration, e.g. $\vec{A}\ {}_2\angle_7^7\ \vec{B} \rightsquigarrow \vec{A}\ {}_2\angle_0^7\ \vec{B} \rightsquigarrow \vec{A}\ {}_2\angle_1^7\ \vec{B} \rightsquigarrow \vec{A}\ {}_2\angle_1^0\ \vec{B} \rightsquigarrow \vec{A}\ {}_2\angle_0^0\ \vec{B}$.

In contrast, if the related objects cannot take the same position (no superposition), for instance because they are both solid physical objects, then relations which represent such configurations are not feasible and thus are missing in the $\mathcal{CNG}$. To simplify matters, we will talk about solid and non-solid objects in the remainder of the text, though the reasons for not allowing superposition can be different.

In the following, we will systematically derive the neighborhood structures for the $\mathcal{OPRA}_m$ calculus starting out with very simple robot kinematics and ending with the most general case of the neighborhood structure for objects which can move in arbitrary direction. The considered situations are the following:

1. One object moving, rotation only
2. Two objects moving, rotation only
3. Two objects moving, translation only, solid objects



(a) $\vec{A}\ {}_2\angle_7^7\ \vec{B}$ (b) $\vec{A}\ {}_2\angle_0^0\ \vec{B}$

Figure 4 Possible conceptual neighborhood structures regarding different motion capabilities of agent A with respect to a static object B. Direct transition is possible from 4(**a**) to 4(**b**) if agent A is able to move sidewards (*dotted line*), whereas several neighborhood transitions are necessary if not (*dashed line*).

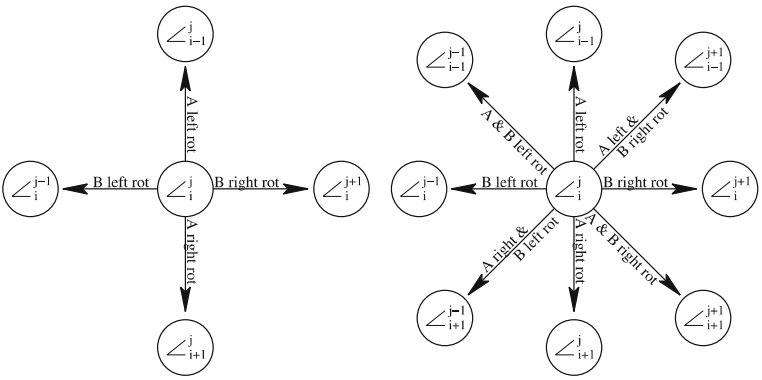
4. Two objects moving, translation only, non-solid objects
5. Two objects moving, *either* translation *or* rotation, non-solid objects
6. Two objects moving, unconstrained motion, non-solid objects

3.3 Neighborhood Structure of $\mathcal{OPRA}_m$

For understanding the neighborhood structures of $\mathcal{OPRA}_m$, we start by considering the simplest case in which only one of the two related solid objects is allowed to rotate while the other does not move at all.

3.3.1 Single Rotating Object

Imagine a robot R represented by the oriented point $\vec{R}$ standing in a room together with a stable object with a fixed intrinsic front, e.g. a locker ($\vec{L}$). Rotating on the spot will lead to a change in relative position of the locker compared to the robot's own intrinsic front, but the robot's relative position to the locker does not change. Therefore, the $\mathcal{OPRA}_m$ relation representing the situation $(\vec{R}\ {}_m\angle_i^j\ \vec{L})$ only changes in i. Turning left results in a decrease of i by one, and a right turn in an increase by one. This means that for all $i, j \in \mathcal{Z}_{4m}$, $\mathtt{cn}({}_m\angle_i^j) = \{{}_m\angle_{i-1}^j, {}_m\angle_{i+1}^j\}$.[3] Reversing the roles of $\vec{L}$ and $\vec{R}$ entails the same changes in j: $\mathtt{cn}({}_m\angle_i^j) = \{{}_m\angle_i^{j-1}, {}_m\angle_i^{j+1}\}$. Figure 5a illustrates this neighborhood structure with the involved actions of R annotated to the edges. If we allow both objects to occupy the same location, we also get a general formula for the neighborhood relation between same relations: $\mathtt{cn}({}_m\angle i) = \{{}_m\angle(i-1), {}_m\angle(i+1)\}$ where the first is either caused by R turning left or L turning right, and vice versa for the latter.



(a) Only one object is able to rotate.

(b) Both objects are able to rotate.

Figure 5 Possible neighborhood transitions of $\mathcal{OPRA}_m$ base relations for rotating solid objects.

[3]Note, that $\mathcal{Z}_{4m}$ is defined as a *cyclic* group so that no modulo operation is required.

3.3.2 Both Objects Rotating

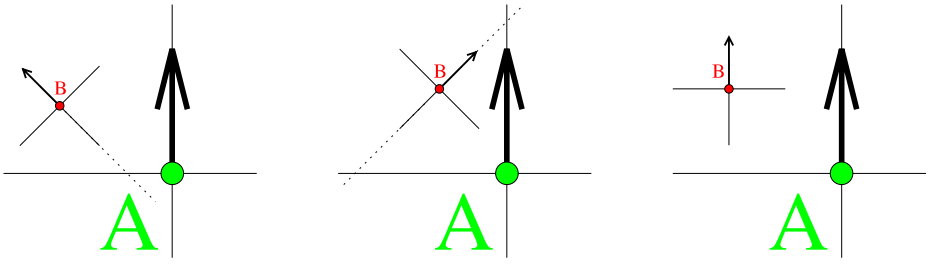
If both objects are allowed to rotate simultaneously, additional neighborhood transitions are possible. For object $\vec{A}$ and $\vec{B}$ both rotating left $\vec{A}\ {}_m\angle_{i-1}^{j-1}\ \vec{B}$ can occur. Both rotating right may result in $\vec{A}\ {}_m\angle_{i+1}^{j+1}\ \vec{B}$. For both rotating opposingly $\vec{A}\ {}_m\angle_{i-1}^{j+1}\ \vec{B}$ and $\vec{A}\ {}_m\angle_{i+1}^{j-1}\ \vec{B}$ are possible. The complete annotated neighborhood structure is illustrated in Figure 5b.

3.3.3 Translating Solid Objects

We will now take a look at possible neighborhood transitions if only translation is possible and both objects cannot occupy the same location. Just allowing translation means that our exemplary robot can only move forward in the direction it is facing or backwards in the opposing direction. For the following observations it does not matter if only one or both objects may move at the same time and we will thus treat both cases here. As translations are much more complex to analyze for $\mathcal{OPRA}_m$ than rotations, we will restrict ourselves to $m = 2$.

We first look at the consequences of moving either $\vec{A}$ or $\vec{B}$ individually. Let us assume the relation $\vec{A}\ {}_2\angle_1^5\ \vec{B}$ as given. In Figure 6, we illustrate different situations subsumed by the given relation. In the first case (cf. Figure 6a), $\vec{A}\ {}_2\angle_2^5\ \vec{B}$ can be reached if $\vec{A}$ moves forward or $\vec{B}$ backwards. With $\vec{A}$ moving backwards $\vec{A}\ {}_2\angle_1^4\ \vec{B}$ is a valid transition. $\vec{B}$ moving forward in the given situation would result in no change. The second case is similar (cf. Figure 6b). $\vec{A}\ {}_2\angle_1^6\ \vec{B}$ follows from $\vec{A}$ moving forward or $\vec{B}$ backwards, and $\vec{A}\ {}_2\angle_0^5\ \vec{B}$ from $\vec{B}$ moving forward. The third case is special, because it will only occur if the orientations of $\vec{A}$ and $\vec{B}$ are identical. Then both relation parts change by one to $\vec{A}\ {}_2\angle_2^6\ \vec{B}$.

Table I summarizes these results and actions are assigned to the particular neighborhood transitions. As long as the objects are not allowed to superpose, simultaneous movement of both objects does not result in additional neighborhood relations. A similar analysis needs to be performed for other relations possible between A and B accordingly. Due to space restrictions, we will leave out the details here and move directly to the case of non-solid objects.



(a) ${}_2\angle_2^5 \vee {}_2\angle_1^4$ (b) ${}_2\angle_1^6 \vee {}_2\angle_0^5$ (c) ${}_2\angle_2^6$

Figure 6 Possible neighborhood transitions for relation $\vec{A}\ {}_2\angle_1^5\ \vec{B}$ and translation only.

Table I The neighborhood transitions corresponding to Figure 6.

$\vec{A}\ {}_2\angle{}_1^5\ \vec{B} \sim$			
A fwd	*A* bwd	*B* fwd	*B* bwd
${}_2\angle{}_2^5 \vee {}_2\angle{}_1^6 \vee {}_2\angle{}_2^6$	${}_2\angle{}_1^4$	${}_2\angle{}_0^5$	${}_2\angle{}_2^5 \vee {}_2\angle{}_1^6 \vee {}_2\angle{}_2^6$

3.3.4 Translating Non-solid Objects

We now take a look at non-solid objects. Allowing objects to superpose means that transitions between same and non-same relations are now possible. With only object B moving, the objects can only take the same positions for the cases with $\vec{A}\ {}_2\angle{}_i^0\ \vec{B}$ and $\vec{A}\ {}_2\angle{}_i^4\ \vec{B}$. These cases are illustrated in Figure 7. The first case (B_1 and B_2) reveals that this situation can only be reached by forward motion of $\vec{B}$. It always results in the relation $\vec{A}\ {}_2\angle(4+i)\ \vec{B}$ (in general $\vec{A}\ {}_m\angle(2m+i)\ \vec{B}$). The second case ($B_3$ and B_4) can only migrate to a same relation by backward motion of $\vec{B}$ and results in $\vec{A}\ {}_2\angle i\ \vec{B}$ (or $\vec{A}\ {}_m\angle i\ \vec{B}$ in general).

Now we allow both objects to move. In the case of *solid* objects we did not get additional neighborhood transitions, as forward motion of one object implied the identical transitions as backward motion of the other object. For *non-solid* objects this is not the case. Consider the different situations described in Figure 8, all represented by relation $\vec{A}\ {}_2\angle{}_1^7\ \vec{B}$. If $\vec{A}$ and $\vec{B}$ have the right velocities, a same relation will occur. The first case will result in $\vec{A}\ {}_2\angle 7\ \vec{B}$, the second in $\vec{A}\ {}_2\angle 6\ \vec{B}$, and the third in $\vec{A}\ {}_2\angle 5\ \vec{B}$.

We have now analyzed rotation and translation individually. The neighborhood transitions for the case of two rotating solid objects are a superset of those for two translating solid objects. For robots that move about by alternatingly rotating on the spot and driving straight (which we will consider in the example in Section 4), the corresponding neighborhood structure is hence the one given in Figure 5b with additional action labels for forward and backward motion as suggested in Table I. Additional neighborhood transitions arise when allowing concurrent translation and rotation (differential drive) but will not be considered here any further. Instead we will now look at the most general case of unconstrained motion as e.g. provided by an omnidrive.

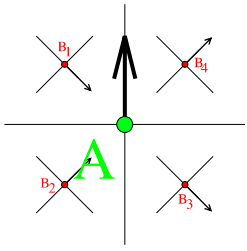
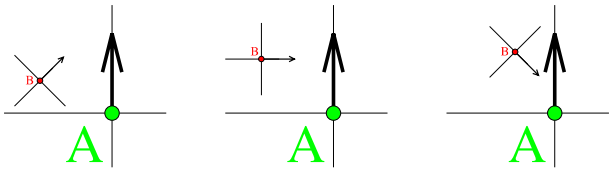


Figure 7 Some situations resulting in same relations by straight forward or backward motion of one objects.



(a) ${}_2\angle 7$ (b) ${}_2\angle 6$ (c) ${}_2\angle 5$

Figure 8 Some situations resulting in same relations by straight forward motion of both objects.

3.3.5 Neighborhood Structure for Unconstrained Motion

For objects A and B being solid objects in the general relation $\vec{A}\ {}_m\angle_i^j\ \vec{B}$ and able to perform any simultaneous combination of rotation and translation in an arbitrary direction as e.g. provided by an omnidrive, eight neighborhood transitions are possible. As already indicated in Figure 4, unconstrained motion now means that it is for instance possible to directly reach relation ${}_2\angle_0^0$ via sideway motion. Overall, all combinations of increasing and decreasing i and j by one are now possible:[4]

$$\begin{aligned}\mathtt{cn}_s({}_m\angle_i^j) = \{\ & {}_m\angle_{i-1}^{j-1}, {}_m\angle_{i-1}^{j}, {}_m\angle_{i-1}^{j+1}, {}_m\angle_{i}^{j-1},\\ & {}_m\angle_{i}^{j+1}, {}_m\angle_{i+1}^{j-1}, {}_m\angle_{i+1}^{j}, {}_m\angle_{i+1}^{j+1}\ \}\end{aligned}$$

The result is the same neighborhood structure as in Figure 5b. However, here all transitions can also be achieved by translation actions and can in principle result from moving only one of the objects.

If we have non-solid objects, same relations can be reached as well and same relations can either change into different same relations or back to non-same relations:

$$\begin{aligned}\mathtt{cn}_{ns}({}_m\angle_i^j) = \{ & {}_m\angle(i-1), {}_m\angle(i+1), {}_m\angle(-j), {}_m\angle(2m-j),\\ & {}_m\angle i, {}_m\angle(2m+i)\} \cup \mathtt{cn}_s({}_m\angle_i^j)\\ \mathtt{cn}_{ns}({}_m\angle s) = \{ & {}_m\angle(s+1), {}_m\angle(s-1), {}_m\angle_*^s, {}_m\angle_*^{2m-s}, {}_m\angle_s^*, {}_m\angle_{2m+s}^*\}\end{aligned}$$

We have now derived the neighborhood structures of the $\mathcal{OPRA}_m$ calculus for different scenarios. As briefly discussed above, these neighborhood structures can for instance be employed for qualitative action planning. In the following, we will address another application of conceptual neighborhoods, namely the model-based resolution of conflicts in spatial information.

4 Dealing with Conflicting Information: Relaxing Constraint Networks

As suggested in [19], conceptual neighborhoods offer a suitable way to deal with conflicting information. In the following, we will demonstrate this by showing how they can serve as a domain-specific heuristic to resolve contradictions. The distance

[4]The index s is used in the following for the case of solid objects, while ns stands for non-solid objects.

between two relations in the neighborhood graph can be used to formulate appropriate distance measures between constraint networks which allows searching for the minimal relaxation of the inconsistent information. We start out by first illustrating the problem with a small example and by defining what we mean by relaxation. We then proceed by showing how conceptual neighborhood relations can be used to formulate application-specific distance functions and finally address the problem of determining the minimal relaxation with respect to a chosen distance function.

4.1 Inconsistent Information and Relaxations

Let us consider a small example involving three robots as illustrated in Figure 9. The robots A, B, C are solving some task as a team and all have a limited view angle (indicated by the dashed lines) which in this case means that A only sees B, B only sees C and C only sees A. They represent information about the current situation in the form of qualitative spatial relations from the $\mathcal{OPRA}_2$ calculus and have the ability to communicate with each other to exchange information on what they currently perceive. This results in a common world model including information not available to the individual robot (e.g. the information about the other robot that is currently outside their view). However, the combined world model does not need to be consistent. In our case, due to slight errors in the perception of robot B, the relations perceived by the robots could be as shown in Table II.

The corresponding constraint network is inconsistent as the configuration described by these three relations is not satisfiable in the domain of oriented points. For instance, the composition of $\vec{A}\ {}_2\angle_7^5\ \vec{B}$ and $\vec{B}\ {}_2\angle_0^5\ \vec{C}$ yields $\vec{A}\ {}_2\angle_{\{5-7\}}^{\{5-7\}}\ \vec{C}$, while on the other hand taking the converse of $\vec{C}\ {}_2\angle_1^7\ \vec{A}$ yields $\vec{A}\ {}_2\angle_7^1\ \vec{C}$. Since the intersection of these two results is empty, the constraint network cannot be consistent, which can even be detected by just using a simple path-consistency algorithm like [31].

In such a situation, it is desirable to slightly relax the given constraints until a consistent constraint network is found instead of completely discarding the inconsistent information. As constraints in our case are disjunctions of base relations from a particular spatial calculus written as sets of base relations, we can use set inclusion to define what we mean by relaxation of an individual constraint: Constraint C_1 is a relaxation of constraint C_2 if and only if $C_2 \subseteq C_1$. We can extend this notion to complete constraint networks over the same set of variables by defining that

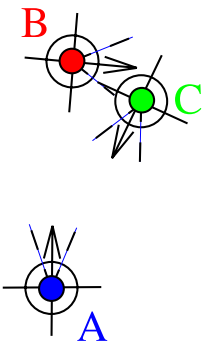


Figure 9 A scene with three robots represented in $\mathcal{OPRA}_2$.

Table II The $\mathcal{OPRA}_2$ relations for the real situation in Figure 9 and the relations perceived by the individual robots.

Perceived situation	Real situation
$\vec{A}\ {}_2\angle^5_7\ \vec{B}$	$\vec{A}\ {}_2\angle^5_7\ \vec{B}$
$\vec{B}\ {}_2\angle^5_0\ \vec{C}$	$\vec{B}\ {}_2\angle^5_7\ \vec{C}$
$\vec{C}\ {}_2\angle^7_1\ \vec{A}$	$\vec{C}\ {}_2\angle^7_1\ \vec{A}$

$CN_1 =< V, l_1 >$ is a relaxation of $CN_2 =< V, l_2 >$ (written $CN_2 \subseteq CN_1$) if and only if for all $v_i, v_j \in V, l_2(v_i, v_j) \subseteq l_1(v_i, v_j)$ holds.

4.2 Distance Functions Based on Conceptual Neighborhoods

We usually are not interested in an arbitrary relaxation of the original network that is consistent, but are looking for *minimal relaxations* with respect to the domain and task at hand. To define minimality, we need a distance function over constraint networks and, as we will show, distances between relations in the conceptual neighborhood graph are well-suited as a basis to formulate such distance functions.

We start by defining the distance d_b between two base relations $r_i, r_j \in \mathcal{BR}$ of a calculus as the length of the shortest path connecting them in the neighborhood graph $\mathcal{CNG}$:

$$d_b(r_i, r_j) = x \iff x \text{ is the length of the shortest path connecting } r_i \text{ and } r_j \text{ in } \mathcal{CNG}$$

We extend this distance measure to constraints which are disjunctions of base relations written as sets and thus also to conceptual neighborhoods. The intuition behind this extension is that the distance should correspond to the number of times we have to apply the `relax` operation from Section 3.1 to one of the sets until the result contains all the relations from the other set. This corresponds to taking the symmetric Hausdorff distance between the two sets based on d_b. We therefore define the distance $d_c(S, R)$ between two sets of base relations $S, R \in 2^{\mathcal{BR}}$ as:

$$d_c(S, R) = \max\{h(S, R), h(R, S)\}$$

where

$$h(X, Y) = \max_{x \in X} \left(\min_{y \in Y} d_b(x, y) \right)$$

How this distance function for individual constraints should be extended to constraint networks over the same set of variables is dependent on the concrete application considered since the resulting distance function describes what we consider similar in the given context. Things that one might want to minimize in a given application could for example be the following:

– the number of corresponding constraints that differ in constraint networks CN_1 and CN_2:

$$\texttt{ConstraintsChanged}(CN_1, CN_2) = |\ \{(v_i, v_j) \mid\ v_i, v_j \in V \wedge l_1(v_i, v_j) \neq l_2(v_i, v_j)\}\ |$$

– the maximal distance between corresponding constraints as given by d_c:

$$\texttt{MaxChangedDistance}(CN_1, CN_2) = \max_{v_i, v_j \in V} d_c\left(l_1(v_i, v_j), l_2(v_i, v_j)\right)$$

– the overall sum of distances between corresponding constraints as given by d_c:

$$\texttt{SumOfDistances}(CN_1, CN_2) = \sum_{v_i, v_j \in V} d_c\left(l_1(v_i, v_j), l_2(v_i, v_j)\right)$$

Such measures can be arbitrarily combined to define an overall distance function d_{CN} over the space of constraint networks over the same set of variables and the same set of relational constraints, for example by taking the weighted sum:

$$\begin{aligned} d_{CN}(CN_1, CN_2) = {} & \alpha\ \texttt{ConstraintsChanged}(CN_1, CN_2) \\ & + \beta\ \texttt{MaxChangedDistance}(CN_1, CN_2) \\ & + \gamma\ \texttt{SumOfDistances}(CN_1, CN_2) \end{aligned}$$

4.3 Minimal Relaxations

Finding the minimal relaxation means that we have to solve a combinatorial optimization problem in which the cost function is given by our distance function d_{CN} with respect to the original constraint network. Relaxed constraint networks are constructed by applying the `relax` function to individual constraints. This means that a constraint is replaced by the coarsest relation with distance one according to the distance function d_c.

Let $\mathcal{RC}_{CN}$ be the set of all constraint networks which are relaxations of a given constraint network CN and consistent:

$$\mathcal{RC}_{CN} = \{\ N \mid CN \subseteq N \wedge\ N \text{ is consistent}\ \}$$

The goal of the relaxation process now is to find a constraint network $CN^* \in \mathcal{RC}_{CN}$ that is closest to CN according to the distance function d_{CN} chosen. We call such a network a *minimal consistent relaxation* of CN:[5]

$$CN^* = \operatorname*{argmin}_{N \in \mathcal{RC}_{CN}} d_{CN}(N, CN)$$

The number of possible relaxations is n^l where n is the number of constraints (which is proportional to $|V|^2$) and l is the diameter of the constraint graph which means a constraint can be relaxed at most l times. If the distance to the original network grows monotonically whenever the `relax` operator is applied to a constraint, a general algorithm similar to Dijkstra's shortest path algorithm can be used to find a minimal relaxation. Constraint networks would then be stored in a priority queue sorted by increasing distance from the original constraint network. In every step the first network is taken from the queue and checked for consistency. If it is not consistent several new networks are generated by applying `relax` to one of

[5]Note, that CN^* is not well-defined, as several minimal consistent relaxations may exist.

the constraints of the current network. These new networks are then sorted into the queue. If the inspected network is consistent, it has to be a minimal consistent relaxation.

However, for certain distance functions it is possible to directly enumerate the relaxations in order of increasing distance without the need of keeping multiple constraint networks stored in a queue. Provided we have such an enumeration function `generateRelaxation`(N) that generates the next relaxation from the currently considered relaxation N with respect to the distance function, and we have a function `consistency`(N) that decides the consistency of network N, we can define a recursive function `mcr`(CN) that computes a minimal consistent relaxation of a given network CN as follows:

$$\texttt{mcr}(CN) = \begin{cases} CN & : \text{ if } \texttt{consistent}(CN) \\ \texttt{mcr}(\texttt{generateRelaxation}(CN)) & : \text{ otherwise} \end{cases}$$

In our example case, we are dealing with noisy sensor information. Therefore, we want our overall distance function d to combine `MaxChangedDistance` and `SumOfDistances` in a way that `MaxChangedDistance` is given precedence over `SumOfDistances`. This means, that given two different relaxations CN_1, CN_2 of CN, $\texttt{MaxChangedDistance}(CN_1, CN) > \texttt{MaxChangedDistance}(CN_2, CN)$ implies $d(CN_1, CN) > d(CN_2, CN)$. Only if `MaxChangedDistance` is the same for CN_1 and CN_2, `SumOfDistances` would be used to decide which one is closer to CN. Employing this distance is very appropriate for our application since it prefers relaxations with multiple small changes over few large changes while, as a second criterion, minimizing the overall sum of changes. In applications in which few large outliers are more likely than many small changes, a combination of `ConstraintsChanged` and `SumOfDistances` would have been the better choice.

For many applications, a reasonable distance function will give strict precedence of one criterion over another. In principal, these can still be formulated as a weighted sum by choosing the weights accordingly. Moreover, in most cases it should be possible to formulate direct enumeration algorithms. The enumeration function we used for our example together with the networks considered can be found on our website.[6] For simplicity we assume that our robots can either rotate on the spot or move straight in the direction they are facing and employ the corresponding neighborhood structure as discussed in Section 3.3.4. The result is one of the following two possible minimal consistent relaxations:

$$\begin{array}{l} \vec{A}\ {}_2\angle_7^5\ \vec{B} \\ \vec{B}\ {}_2\angle_7^5\ \vec{C} \\ \vec{C}\ {}_2\angle_1^7\ \vec{A} \end{array} \quad \text{and} \quad \begin{array}{l} \vec{A}\ {}_2\angle_7^5\ \vec{B} \\ \vec{B}\ {}_2\angle_7^6\ \vec{C} \\ \vec{C}\ {}_2\angle_1^7\ \vec{A} \end{array}$$

The original situation is correctly described by the first of these two solutions but the other would have been just as likely given the inconsistent information. Even if the agents follow a conservative policy and only accept what holds in all minimal consistent relaxations, they still are able to determine the position of the robot not visible to them, just not the exact orientation of C with respect to B.

[6] http://www.sfbtr8.uni-bremen.de/project/r3/relaxation/

Even though we can generate relaxations in order of increasing distance to the original constraint network, the huge number of possible relaxations means that only for problems in which the minimal consistent relaxations are rather close to the original network can be relaxed in appropriate time. In our case, it would be reasonable to assume that no perceived relation will be further from the actual relation by more than 2 with respect to d_b. Therefore, if no corresponding consistent relaxation with `MaxChangedDistance` ≤ 2 would have been found, the relaxation process could have been stopped, as obviously something unexpected must have happened.

To summarize, as we have shown conceptual neighborhoods can serve as a meaningful and well-defined basis for optimizing the relaxation process in constraint networks. Minimal relaxation can be defined based on minimal neighborhood distance. However, as a prerequisite the adequate neighborhood structure for the application at hand needs to be known. Since manually deriving neighborhood structures can be a very tedious process, tools to automate the derivation are required. We will now present a method that exploits the formalization of the relations of a new calculus in another calculus with known neighborhood structure to automatically deduce the neighboring structure of the new calculus.

5 Deriving Conceptual Neighborhoods for Qualitative Calculi in $\mathcal{OPRA}_m$

Since not always adequate calculi are available for specific tasks, new ones have to be developed. Determining the operations and properties of a new calculus is an error-prone and time-consuming process. Modeling one spatial calculus in another one has been shown to be helpful for determining the properties and operations of a new calculus. In [22, 23] the Dependency Calculus is presented and its properties are derived by mapping it onto the Region Connection Calculus [25]. In [6] it is shown how $\mathcal{OPRA}_m$ representations of other orientation calculi can be utilized to automatically compute composition tables.

This section proceeds in a similar way. We introduce two more orientation calculi, then show how to model their base relations in $\mathcal{OPRA}_m$, and finally use these models and our knowledge about $\mathcal{OPRA}_m$ neighborhood structures to derive the neighborhood relations of the two calculi. The calculi considered are the FlipFlop calculus (FFC) and the fine-grained Dipole Relation Algebra ($\mathcal{DRA}_f$).

5.1 The FlipFlop Calculus (FFC)

The FlipFlop calculus proposed in [15] describes the position of a point C (the referent) in the plane with respect to two other points A (the origin) and B (the relatum) as illustrated in Figure 10. FFC relations are only defined for $A \neq B$. The following base relations are distinguished: C can be to the **l**eft or to the **r**ight of the oriented line going through A and B, or C can be placed on the line resulting

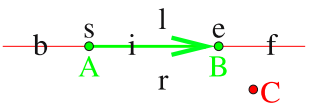


Figure 10 The reference frame for the FlipFlop Calculus.

in one of the five relations **i**nside, **f**ront, **b**ack, **s**tart ($C = A$) or **e**nd ($C = B$) resulting in 7 base relations overall. A FFC relation rel_{FFC} is written as $A, B\ rel_{FFC}\ C$, e.g. A, B **r** C as depicted in Figure 10.

5.2 Encoding the FlipFlop Calculus in $\mathcal{OPRA}_m$

To encode the seven ternary FlipFlop base relations $A, B\ rel_{FFC}\ C$ in $\mathcal{OPRA}_1$ based on the three normal points A, B, and C, we use the two oriented points $\vec{A}^{AC}$ and $\vec{B}^{BC}$. To distinguish the cases in which C coincides with either A or B, we make use of a third oriented point $\vec{C}$ with one distinct but arbitrary orientation. In [6] a different $\mathcal{OPRA}_1$ representation of FFC relations has been introduced, but the formalization we use here is better suited to address neighborhood structures.

We first consider the five FFC relations for which $A \neq B \neq C$ holds: **r**ight, **l**eft, **f**ront, **i**nside, and **b**ack. We need three $\mathcal{OPRA}_1$ relations to unambiguously describe these. $\vec{A}^{AC}\ {}_1\angle_i^j\ \vec{B}^{BC}$ will describe the position of C relative to the reference line $\overline{AB}$. The other two relations ($\vec{A}^{AC}\ {}_1\angle_0^j\ \vec{C}$ and $\vec{B}^{BC}\ {}_1\angle_0^j\ \vec{C}$) are needed for modeling $A \neq C$ and $B \neq C$ for these five cases and will differ for the **s**tart and **e**nd cases.

Assume C is located on the left side of $\overline{AB}$. B is now to the right of $\vec{A}^{AC}$ and A to the left of $\vec{B}^{BC}$ resulting in $\mathcal{OPRA}_1$ relation $\vec{A}^{AC}\ {}_1\angle_3^1\ \vec{B}^{BC}$. As long as C stays on the left side of the line $\overline{AB}$, the two oriented points stay in this relation. If C moves to a position in front of $\overline{AB}$ two things will change because both oriented points rotate simultaneously and their orientations now coincide: B is now in front of $\vec{A}^{AC}$ and A is behind $\vec{B}^{BC}$ yielding the $\mathcal{OPRA}_1$ relation $\vec{A}^{AC}\ {}_1\angle_0^2\ \vec{B}^{BC}$. The **i**nside case in which C lies between A and B results in $\vec{A}^{AC}\ {}_1\angle_0^0\ \vec{B}^{BC}$ and the **b**ack case in $\vec{A}^{AC}\ {}_1\angle_2^0\ \vec{B}^{BC}$. Moving C to the right of the reference line yields $\vec{A}^{AC}\ {}_1\angle_1^3\ \vec{B}^{BC}$.

For **s**tart and **e**nd the position of C coincides with either A or B. Relation **e**nd for example is defined by $A \neq B = C$. Since in this case the orientation of $\vec{B}^{BC}$ is not fixed, the result is a disjunction of four $\mathcal{OPRA}_1$ relations that we write with the *-notation introduced in Section 2.3: $\vec{A}^{AC}\ {}_1\angle_0^*\ \vec{B}^{BC}$. Additionally, we have to define that $B = C$ by $\vec{B}^{BC}\ {}_1\angle*\ \vec{C}$. At first glance, the two cases $\vec{A}^{AC}\ {}_1\angle_0^0\ \vec{B}^{BC}$ and $\vec{A}^{AC}\ {}_1\angle_0^2\ \vec{B}^{BC}$ contained in the disjunction above seem to conflict with the formalizations of **f**ront and **i**nside, but this ambiguity is resolved by the difference in the relation between $\vec{B}^{BC}$ and $\vec{C}$. Similarly, the **s**tart relation results in the $\mathcal{OPRA}_1$ relations $\vec{A}^{AC}\ {}_1\angle_*^0\ \vec{B}^{BC}$ and $\vec{A}^{AC}\ {}_1\angle*\ \vec{C}$. We thus have now a disjoint and exhaustive description of the FFC relations in $\mathcal{OPRA}_1$ which is summarized in Table III.

Table III A mapping of FlipFlop relations to $\mathcal{OPRA}_1$ relations.

$A, B\ r_{FFC}\ C$	$\vec{A}^{AC}\ r_{\mathcal{OPRA}_1}\ \vec{B}^{BC}$	$\vec{A}^{AC}\ r_{\mathcal{OPRA}_1}\ \vec{C}$	$\vec{B}^{BC}\ r_{\mathcal{OPRA}_1}\ \vec{C}$
front	$\vec{A}^{AC}\ {}_1\angle_0^2\ \vec{B}^{BC}$	$\vec{A}^{AC}\ {}_1\angle_0^*\ \vec{C}$	$\vec{B}^{BC}\ {}_1\angle_0^*\ \vec{C}$
end	$\vec{A}^{AC}\ {}_1\angle_0^*\ \vec{B}^{BC}$	$\vec{A}^{AC}\ {}_1\angle_0^*\ \vec{C}$	$\vec{B}^{BC}\ {}_1\angle*\ \vec{C}$
inside	$\vec{A}^{AC}\ {}_1\angle_0^0\ \vec{B}^{BC}$	$\vec{A}^{AC}\ {}_1\angle_0^*\ \vec{C}$	$\vec{B}^{AC}\ {}_1\angle_0^*\ \vec{C}$
start	$\vec{A}^{AC}\ {}_1\angle_*^0\ \vec{B}^{BC}$	$\vec{A}^{AC}\ {}_1\angle*\ \vec{C}$	$\vec{B}^{BC}\ {}_1\angle_0^*\ \vec{C}$
back	$\vec{A}^{AC}\ {}_1\angle_2^0\ \vec{B}^{BC}$	$\vec{A}^{AC}\ {}_1\angle_0^*\ \vec{C}$	$\vec{B}^{BC}\ {}_1\angle_0^*\ \vec{C}$
left	$\vec{A}^{AC}\ {}_1\angle_3^1\ \vec{B}^{BC}$	$\vec{A}^{AC}\ {}_1\angle_0^*\ \vec{C}$	$\vec{B}^{BC}\ {}_1\angle_0^*\ \vec{C}$
right	$\vec{A}^{AC}\ {}_1\angle_1^3\ \vec{B}^{BC}$	$\vec{A}^{AC}\ {}_1\angle_0^*\ \vec{C}$	$\vec{B}^{BC}\ {}_1\angle_0^*\ \vec{C}$

5.3 Deriving the $\mathcal{CNG}$ for FFC

Now we derive the general neighborhood structure of FFC from the general neighborhood structure of $\mathcal{OPRA}_1$ described in Section 3.3.5. Since A and B are not allowed to coincide in the FFC definition, we can restrict ourselves to consider cases in which only C is allowed to move. Nevertheless, the corresponding oriented point $\vec{C}$ is part of all three relations in the $\mathcal{OPRA}_1$ formalization and thus all three relations can change simultaneously.

The general idea of the algorithm for deriving the $\mathcal{CNG}$ is to combine the individual conceptual neighbors of the three $\mathcal{OPRA}_1$ relations and check the resulting constraint networks for consistency to find out whether the generated formalizations describe valid FFC relations. We abbreviate the $\mathcal{OPRA}_1$ model $\vec{A}^{AC}\ S_1\ \vec{B}^{BC} \wedge \vec{A}^{AC}\ S_2\ \vec{C} \wedge \vec{B}^{BC}\ S_3\ \vec{C}$ of a FCC relation (as provided by Table III) as a triple (S_1, S_2, S_3) where the S_i again are sets of base relations. The set of combinations that need to be checked for a given triple (S_1, S_2, S_3) can then be specified with the help of the `relax` function (defined in Section 3.1) as $(\texttt{relax}(S_1) \times \texttt{relax}(S_2) \times \texttt{relax}(S_3)) \setminus (S_1 \times S_2 \times S_3)$.

For the FFC relation **f**ront, we check all combinations from the following three sets except those which are combinations of base relations already contained in S_1, S_2, and S_3:

$$\texttt{relax}(S_1) = \texttt{relax}(\{{}_1\angle_0^2\}) = \{{}_1\angle_0^2, {}_1\angle_3^1, {}_1\angle_3^2, {}_1\angle_3^3, {}_1\angle_0^1, {}_1\angle_0^3, {}_1\angle_1^1, {}_1\angle_1^2, {}_1\angle_1^3\}$$

$$\texttt{relax}(S_2) = \texttt{relax}(S_3) = \texttt{relax}({}_1\angle_0^*) = \{{}_1\angle_0^*{}_1\angle_3^*, {}_1\angle_1^*, {}_1\angle*\}$$

From all these combinations that were checked for consistency with the SparQ[7] toolbox [32], all consistent networks fall into one of three classes from Table III. These are $({}_1\angle_3^1, {}_1\angle_0^*, {}_1\angle_0^*)$ for **l**eft, $({}_1\angle_1^3, {}_1\angle_0^*, {}_1\angle_0^*)$ for **r**ight, and $({}_1\angle_0^*, {}_1\angle_0^*, {}_1\angle*)$ for **e**nd. These three FFC relations are indeed the correct conceptual neighbors of the **f**ront relation. Applying this method to the other base relations yields the $\mathcal{CNG}$ shown in Figure 11.

5.4 The Fine-grained Dipole Relation Algebra ($\mathcal{DRA}_f$)

A dipole is an oriented line segment as e.g. determined by a start and an end point. We will write $\vec{d}_{AB}$ for a dipole defined by start point A and end point B. The idea

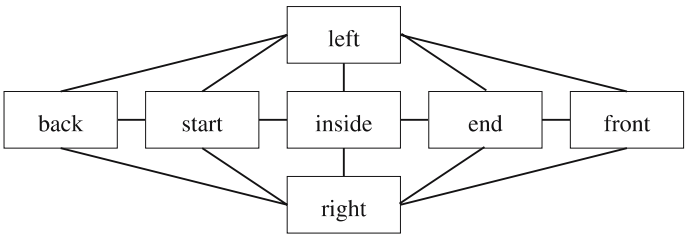


Figure 11 The conceptual neighborhood structure of the FlipFlop Calculus.

[7] http://www.sfbtr8.uni-bremen.de/project/r3/sparq/

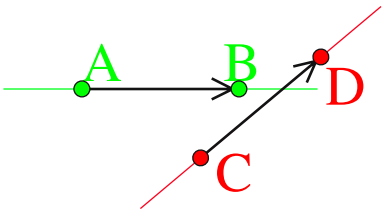


Figure 12 A dipole configuration: $\vec{d}_{AB}$ *rlll* $\vec{d}_{CD}$ in the fine-grained dipole relation algebra ($\mathcal{DRA}_f$).

was first introduced by Schlieder [30] and extended in [20]. The fine-grained dipole calculus ($\mathcal{DRA}_f$) [5] describes the orientation relation between two dipoles $\vec{d}_{AB}$ and $\vec{d}_{CD}$. Each base relation is a 4-tuple (r_1, r_2, r_3, r_4) of FlipFlop relations. r_1 describes the relation of C with respect to the dipole $\vec{d}_{AB}$, r_2 of D with respect to $\vec{d}_{AB}$, r_3 of A with respect to $\vec{d}_{CD}$, and r_4 of b with respect to $\vec{d}_{CD}$. The relations are usually written without the commas and brackets, e.g. *rrll*. The example in Figure 12 shows the relation $\vec{d}_{AB}$ *rlll* $\vec{d}_{CD}$. $\mathcal{DRA}_f$ has 72 base relations.

5.5 Encoding $\mathcal{DRA}_f$ in $\mathcal{OPRA}_m$

We will now give a mapping of the 72 base relations of the fine-grained Dipole Relation Algebra ($\mathcal{DRA}_f$) to $\mathcal{OPRA}_1$ based on the points A, B, C, and D belonging to the $\mathcal{DRA}_f$ relations. Analogous to the FFC formalization, we need the oriented points $\vec{A}^{AC}$, $\vec{B}^{BC}$, and $\vec{C}$ for r_1, $\vec{A}^{AD}$, $\vec{B}^{BD}$, and $\vec{D}$ for r_2, $\vec{C}^{CA}$, $\vec{D}^{DA}$, and $\vec{A}$ for r_3, and $\vec{C}^{CB}$, $\vec{D}^{DB}$, and $\vec{B}$ for r_4.

Since each of the four symbols of a $\mathcal{DRA}_f$ relation is derived in the same way, just for different triples of points, and each is describing a FlipFlop relation, we begin by providing the FlipFlop encodings from Table III, but with the concrete oriented points replaced by variables X, Y, and Z (see Table IV). Table V lists the

Table IV Describing the local $\mathcal{DRA}_f$ point configurations (cf. Table III).

$A, B\ r_{FFC}\ C$	$\vec{X}^{XZ}\ r_{\mathcal{OPRA}_1}\ \vec{Y}^{YZ}$	$\vec{X}^{XZ}\ r_{\mathcal{OPRA}_1}\ \vec{Z}$	$\vec{Y}^{YZ}\ r_{\mathcal{OPRA}_1}\ \vec{Z}$
front	$\vec{X}^{XZ}\ {}_1\angle_0^2\ \vec{Y}^{YZ}$	$\vec{X}^{XZ}\ {}_1\angle_0^*\ \vec{Z}$	$\vec{Y}^{YZ}\ {}_1\angle_0^*\ \vec{Z}$
end	$\vec{X}^{XZ}\ {}_1\angle_0^*\ \vec{Y}^{YZ}$	$\vec{X}^{XZ}\ {}_1\angle_0^*\ \vec{Z}$	$\vec{Y}^{YZ}\ {}_1\angle*\ \vec{Z}$
inside	$\vec{X}^{XZ}\ {}_1\angle_0^0\ \vec{Y}^{YZ}$	$\vec{X}^{XZ}\ {}_1\angle_0^*\ \vec{Z}$	$\vec{Y}^{XZ}\ {}_1\angle_0^*\ \vec{Z}$
start	$\vec{X}^{XZ}\ {}_1\angle_*^0\ \vec{Y}^{YZ}$	$\vec{X}^{XZ}\ {}_1\angle*\ \vec{Z}$	$\vec{Y}^{YZ}\ {}_1\angle_0^*\ \vec{Z}$
back	$\vec{X}^{XZ}\ {}_1\angle_2^0\ \vec{Y}^{YZ}$	$\vec{X}^{XZ}\ {}_1\angle_0^*\ \vec{Z}$	$\vec{Y}^{YZ}\ {}_1\angle_0^*\ \vec{Z}$
left	$\vec{X}^{XZ}\ {}_1\angle_3^1\ \vec{Y}^{YZ}$	$\vec{X}^{XZ}\ {}_1\angle_0^*\ \vec{Z}$	$\vec{Y}^{YZ}\ {}_1\angle_0^*\ \vec{Z}$
right	$\vec{X}^{XZ}\ {}_1\angle_1^3\ \vec{Y}^{YZ}$	$\vec{X}^{XZ}\ {}_1\angle_0^*\ \vec{Z}$	$\vec{Y}^{YZ}\ {}_1\angle_0^*\ \vec{Z}$

Table V Instantiation mapping of X, Y, Z in Table IV according to the position in the $\mathcal{DRA}_f$ relation tuple.

position		X	Y	Z
1st position	(r_1)	A	B	C
2nd position	(r_2)	A	B	D
3rd position	(r_3)	C	D	A
4th position	(r_4)	C	D	B

instantiations that have to be chosen for X, Y, and Z for each of the four symbols. We give an example of a complete translation of a $\mathcal{DRA}_f$ relation:

$$\begin{aligned}\vec{d}_{AB}\, rfll\, \vec{d}_{CD} \equiv\ & \vec{A}^{AC}\ {}_1\angle_1^3\ \vec{B}^{BC} \wedge \vec{A}^{AC}\ {}_1\angle_0^*\ \vec{C} \wedge \vec{B}^{BC}\ {}_1\angle_0^*\ \vec{C} \\ \wedge\ & \vec{A}^{AD}\ {}_1\angle_0^2\ \vec{B}^{BD} \wedge \vec{A}^{AD}\ {}_1\angle_0^*\ \vec{D} \wedge \vec{B}^{BD}\ {}_1\angle_0^*\ \vec{D} \\ \wedge\ & \vec{C}^{CA}\ {}_1\angle_3^1\ \vec{D}^{DA} \wedge \vec{C}^{CA}\ {}_1\angle_0^*\ \vec{A} \wedge \vec{D}^{DA}\ {}_1\angle_0^*\ \vec{A} \\ \wedge\ & \vec{C}^{CB}\ {}_1\angle_3^1\ \vec{D}^{DB} \wedge \vec{C}^{CB}\ {}_1\angle_0^*\ \vec{B} \wedge \vec{D}^{DB}\ {}_1\angle_0^*\ \vec{B}\end{aligned}$$

5.6 Deriving the $\mathcal{CNG}$ for $\mathcal{DRA}_f$

For deriving the $\mathcal{CNG}$ for $\mathcal{DRA}_f$ we use the same algorithmic scheme as for FFC: composing the individual neighbors of the relations occurring in the $\mathcal{OPRA}_1$ translation, checking the results for consistency, and retranslating the consistent ones.

For a single FFC relation about 140 potential neighboring relations exist. Applying this scheme naively to $\mathcal{DRA}_f$ relations, we get about 140^4 potential neighbors for each relation. However, after eliminating the inconsistent ones for FFC, we ended up with an average of four consistent networks per FFC relation. We therefore chose to employ these results to reduce the number of networks that need to be checked for consistency for $\mathcal{DRA}_f$. The constraints being responsible for the inconsistencies in the FFC networks will also be contained in the potential neighbors for $\mathcal{DRA}_f$ relations and thus the overall constraint network would be classified as inconsistent as well. After that, only an average of $5^4 - 1 = 624$ potential neighboring configurations needed to be checked per $\mathcal{DRA}_f$ relation.

Out of the potential neighbors of *rfll* only 14 are consistent. The neighboring relations are: *rrll*, *rlll*, *rele*, *rlli*, *rrlf*, *fll*, *brll*, *rrbl*, *ffbb*, *efbs*, *ifbi*, *bfii*, *sfsi*, and *ffff*. For solid objects, the relations *efbs* and *sfsi* would not have been neighbors of *rfll*.

To give another example, for relation *ebis* 30 neighbors were derived: *fbii*, *ibib*, *ells*, *errs*, *eses*, *lbll*, *rbrr*, *rlir*, *lril*, *llll*, *lllb*, *lllr*, *llrl*, *llrr*, *lril*, *lrll*, *lrlr*, *lrrl*, *lrri*, *lrrr*, *fll*, *frrr*, *illr*, *rlll*, *rlli*, *rllr*, *rlrl*, *rlrr*, *rrll*, and *rlrl*.

The method for deriving neighborhood structures described in this section yields the correct neighborhood graphs for the two calculi considered. However, whether it is applicable for qualitative calculi in general is still an open question. So far, manual verification of the neighborhoods graphs computed for different orientation calculi indicates that this might be the case, but further research is needed on this issue.

6 Conclusion

In this paper, we investigated the idea of conceptual neighborhood in the context of robot control tasks and world modeling. We have shown that conceptual neighborhoods are well-suited to provide information about how the world might develop on a high level of abstraction. Using the example of the $\mathcal{OPRA}_m$ calculus, we studied how neighborhood structures are affected by such task and environment-specific parameters like robot motion capabilities, the dynamics of the objects involved, and whether objects are able to superpose or not. As an exemplary application of conceptual neighborhoods, we showed how these structures can be exploited for resolving conflicting knowledge about the world, e.g. as arising through noisy sensor readings.

As often no adequate calculus is available for a specific task, new ones have to be developed and methods to automate this are needed. We showed that expressing the relations of a new calculus in a different well-specified calculus is one way in which this automation can be achieved.

The $\mathcal{OPRA}_m$ calculus is a very suitable calculus for formalizing other orientation calculi as it is expressive enough to unambiguously describe the relations of most other orientation calculi as configurations of a rather small number of oriented points. We presented the $\mathcal{OPRA}_m$ formalizations for the FlipFlop Calculus and the fine-grained Dipole Relation Algebra, and will continue to do so for other spatial calculi.

Other benefits of the mappings between different calculi still need to be investigated. In addition, the idea of complex neighborhood graphs that represent possible changes of spatial relations for more than two objects needs further investigation. In principle, the approach presented in Section 5 computes neighboring configurations for $\mathcal{OPRA}_m$ configurations of more than two objects. We intend to continue work in this direction in order to improve the applicability of neighborhood-based planning in complex dynamic situations.

Acknowledgements The authors would like to thank Lutz Frommberger, Diedrich Wolter, Reinhard Moratz, and Christian Freksa for fruitful discussions and impulses. Our work was supported by the DFG Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition.

References

1. Bennett, B., Galton, A.P.: A unifying semantics for time and events. Artif. Intell. **153**(1,2), 13–48 (2004), March
2. Cohn, A.G.: Qualitative spatial representation and reasoning techniques. In: Brewka, G., Habel, C., Nebel, B. (eds.) KI-97: Advances in Artificial Intelligence, 21st Annual German Conference on Artificial Intelligence, Freiburg, Germany, September 9-12, 1997. Proceedings, vol. 1303 of Lecture Notes in Computer Science, pp. 1–30. Springer, Berlin Heidelberg New York (1997)
3. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: an overview. Fundam. Inform. **46**(1,2), 1–29 (2001)
4. Davis, E.: Continuous shape transformation and metrics of shape. Fundam. Inform. **46**, 31–54 (2001), May
5. Dylla, F., Moratz, R.: Exploiting qualitative spatial neighborhoods in the situation calculus. In: Freksa, C., Knauff, M., Krieg-Brückner, B., Nebel, B., Barkowsky, T. (eds.) Spatial Cognition IV. Reasoning, Action, Interaction: International Conference Spatial Cognition 2004, vol. 3343

of Lecture Notes in Artificial Intelligence, pp. 304–322. Springer, Berlin, Heidelberg, New York (2005)
6. Dylla, F., Wallgrün, J.O.: On generalizing orientation information in $\mathcal{OPRA}_m$. In: Proceedings of the 29th German Conference on Artificial Intelligence (KI, June 2006), Bremen, Germany (2006)
7. Egenhofer, M.J.: A formal definition of binary topological relationships. In: 3rd International Conference on Foundations of Data Organization and Algorithms, New York, NY, USA, pp. 457–472. Springer, Berlin Heidelberg New York (1989)
8. Frank, A.: Qualitative spatial reasoning about cardinal directions. In: Proceedings of the American Congress on Surveying and Mapping (ACSM-ASPRS), Baltimore, MD, USA, pp. 148–167 (1991)
9. Freksa, C.: Conceptual neighborhood and its role in temporal and spatial reasoning. In: Singh, M.G., Travé-Massuyès, L. (eds.) Proceedings of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning, pp. 181–187. North-Holland, Amsterdam (1991)
10. Freksa, C.: Using orientation information for qualitative spatial reasoning. In: Frank, A.U., Campari, I., Formentini, U. (eds.) Theories and Methods of Spatio-temporal Reasoning in Geographic Space, pp. 162–178. Springer, Berlin Heidelberg New York (1992)
11. Freksa, C.: Spatial cognition – an AI prespective. In: Proceedings of 16th European Conference on AI (ECAI 2004) (2004)
12. Galton, A.: Continuous motion in discrete space. In: Cohn, A., Giunchiglia, F., Selman, B. (eds.) Proceedings 7th Internat. Conf. on Principles of Knowledge Representation and Reasoning (KR2000), pp. 26–37. Morgan Kaufmann, San Francisco, CA (2000)
13. Galton, A.: Qualitative Spatial Change. Oxford University Press, London, UK (2000)
14. Ladkin, P., Reinefeld, A.: Effective solution of qualitative constraint problems. Artif. Intell. **57**, 105–124 (1992)
15. Ligozat, G.: Qualitative triangulation for spatial reasoning. In: Frank, A.U., Campari, I. (eds.) Spatial Information Theory: A Theoretical Basis for GIS, (COSIT'93), Marciana Marina, Elba Island, Italy, vol. 716 of Lecture Notes in Computer Science, pp. 54–68. Springer, Berlin Heidelberg New York (1993)
16. Moratz, R.: Intuitive linguistic joint object reference in human-robot interaction. In: Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI), Boston, MA. AAAI, Menlo Park, CA (2006), July
17. Moratz, R.: Representing relative direction as binary relation of oriented points. In: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI, August, 2006), Riva del Garda, Italy (2006)
18. Moratz, R., Dylla, F., Frommberger, L.: A relative orientation algebra with adjustable granularity. In: Proceedings of the Workshop on Agents in Real-time and Dynamic Environments (IJCAI, July 05), Edinburgh, Scotland (2005)
19. Moratz, R., Freksa, C.: Spatial reasoning with uncertain data using stochastic relaxation. In: Brauer, W. (ed.) Fuzzy-Neuro Systems 98, pp. 106–112. Infix; Sankt Augustin (1998)
20. Moratz, R., Renz, J., Wolter, D.: Qualitative spatial reasoning about line segments. In: Horn, W. (ed.) Proceedings of the 14th European Conference on Artificial Intelligence (ECAI), Berlin, Germany. IOS Press, Amsterdam, The Netherlands (2000)
21. Muller, P.: A qualitative theory of motion based on spatio-temporal primitives. In: Cohn, A.G., Schubert, L., Shapiro, S.C. (eds.) KR'98: Principles of Knowledge Representation and Reasoning, pp. 131–141. Morgan Kaufmann, San Francisco, CA (1998)
22. Ragni, M., Scivos, A.: Dependency calculus: Reasoning in a general point relation algebra. In: Proceedings of the 28th German Conference on Artificial Intelligence (KI 2005), pp. 49–63 (2005)
23. Ragni, M., Scivos, A.: Dependency calculus reasoning in a general point relation algebra. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, 30 July–5 August, 2005, pp. 1577–1578. Professional Book Center, Denver, CO (2005)
24. Ragni, M., Wölfl, S.: Temporalizing spatial calculi – on generalized neighborhood graphs. In: Proceedings of the 28th German Conference on Artificial Intelligence (KI 2005) (2005)
25. Randell, D.A., Cui, Z., Cohn, A.: A spatial logic based on regions and connection. In: Nebel, B., Rich, C., Swartout, W. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR'92), pp. 165–176. Morgan Kaufmann, San Mateo, CA (1992)

26. Renz, J., Ligozat, G.: Weak composition for qualitative spatial and temporal reasoning. In: Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming (CP 2005), pp. 534–548, Sitges (Barcelona), Spain (2005), October
27. Renz, J., Mitra, D.: Qualitative direction calculi with arbitrary granularity. In: Zhang, C., Guesgen, H.W., Yeap, W.-K. (eds.) PRICAI 2004: Trends in Artificial Intelligence, 8th Pacific RimInternational Conference on Artificial Intelligence, Auckland, New Zealand, Proceedings, vol. 3157 of Lecture Notes in Computer Science, pp. 65–74. Springer, Berlin Heidelberg New York (2004)
28. Renz, J., Nebel, B.: On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. Artif. Intell. **108**(1,2), 69–123 (1999)
29. Röfer, T.: Route navigation using motion analysis. In: Freksa, C., Mark, D.M. (eds.) Spatial Information Theory: Foundations of Geographic Information Science. Conference on Spatial Information Theory (COSIT), pp. 21–36. Springer, Berlin Heidelberg New York (1999)
30. Schlieder, C.: Reasoning about ordering. In: Spatial Information Theory: A Theoretical Basis for GIS (COSIT'95), vol. 988 of Lecture Notes in Computer Science, pp. 341–349. Springer, Berlin Heidelberg New York (1995)
31. van Beek, P.: Reasoning about qualitative temporal information. Artif. Intell. **58**(1-3), 297–321 (1992)
32. Wallgrün, J.O., Frommberger, L., Wolter, D., Dylla, F., Freksa, C.: A toolbox for qualitative spatial representation and reasoning. In: Spatial Cognition 2006, Bremen, Germany (2006)