Streets to the $\mathcal{OPRA}$— Finding your destination with imprecise knowledge

Dominik Lücke
University of Bremen
SFB/TR8
Bremen, Germany

Till Mossakowski
DFKI GmbH Bremen
Safe and Secure Systems
Bremen, Germany

Reinhard Moratz
National Center for
Geographic Information and Analysis
Department of Spatial Information
Science and Engineering
Orono, Maine, USA

Abstract

Qualitative spatial calculi offer a method to describe spatial configurations in a framework based on a finite set of relations that abstracts from the underlying mathematical theory. But an open issue is whether they can be employed in applications. Further their cognitive adequacy is questionable or not investigated at all. In this paper we investigate the applicability of $\mathcal{OPRA}$ to navigation in street networks that are described via local observations. Further we scrutinize whether a description of directions that is deemed cognitively adequate and can be described in $\mathcal{OPRA}$ can perform that task. We are using an environment that we developed ourselves for these experiments, the used algorithms and the program itself are explained in detail.

1 Introduction

Since the emergence of Allen's interval algebra [Allen, 1983] qualitative spatial and temporal reasoning has become an interesting field in artificial intelligence research. A lot of the tools used and later refined for reasoning tasks has already been introduced by Allen, i.e. composition based reasoning. A multitude of qualitative spatial and temporal calculi have been defined dealing with different aspects of space and time. In the field of spatial calculi, we can spot two big classes of calculi, this is the ones dealing with topological aspects of space like $\mathcal{RCC}$ [Randell *et al.*, 1992] and others dealing with directions either with a local or global reference frame. $\mathcal{OPRA}$ is a calculus dealing with directions having a local reference frame. It is based on oriented points, i.e. points in the plane that have a position and an orientation. A feature of the $\mathcal{OPRA}$ calculus is its adjustable granularity, in fact for each $m \in \mathbb{N}$ with $m \geq 1$ a version of the $\mathcal{OPRA}$ calculus exists. The reference frame for $\mathcal{OPRA}_2$ is shown in Figure 1. The position of the basic entity of the $\mathcal{OPRA}$ calculus, the *oriented point*, is shown as the black dot in the middle and its direction as the arrow. $\mathcal{OPRA}_2$ means that the plane is divided into sectors by two intersecting lines with all angles between adjacent lines being the same. The lines and their intersection point divide the plane into one sector that is a point (the intersection point itself) four sectors on the lines and four

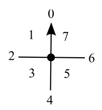

Figure 1: $\mathcal{OPRA}_2$ reference frame

planar sectors. If we call the point in Figure 1 A and another point B, we can determine in which sector B with respect to A lies. With rising granularity the relations of the $\mathcal{OPRA}$ calculus grow finer and finer and their number rises making reasoning very time consuming.

Although there are many qualitative spatial calculi and even more publication about them, only initial steps have been made towards applicability of qualitative spatial calculi to problems that arise in the real world. Moreover, for many calculi it is known that *algebraic closure* only approximates consistency, but it is not know if this approximation is "good enough" for tasks at hand.

We investigate the applicability of the $\mathcal{OPRA}$ calculus (with reasonable granularity) to navigation problems in a street network. For this task, we only rely on knowledge that a person can observe at the *decision points*, i.e. the crossings, of a street network in a qualitative way. In Figure 2 such a crossing is shown. The person driving in the car knows where

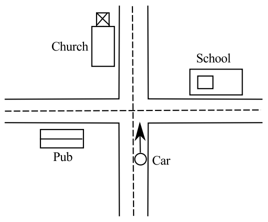

Figure 2: A crossing

she comes from and can observe that the street with the pub is to the left, the one with the church is straight ahead and the one with the school is to the right. But she cannot observe where the airport at the other end of the city is with respect to this. Further knowledge can be deduced from the observed one, but that knowledge is only as good as is the reasoning for the calculus at hand. The have to ask the question, if this knowledge is good enough. What helps us in this case is the fact that we are navigating in a grid that is pre-defined by the given street network. But there are still open questions, is the "straight ahead" or "left" defined by $\mathcal{OPRA}$ the "straight ahead" or "left" as perceived by humans.

As an overall scenario consider that a swarm of robots is exploring an unknown street network (or interior of a building). The robots can make observations at any crossing with respect to a qualitative calculus (in our case $\mathcal{OPRA}$) and they know what a street looks like, i.e. the connections between crossings. The robots can exchange and integrate the data they obtained, but they cannot triangulate their positions. When their work is done, a network of local observations is obtained, but nothing is known so far about non-local constraints. In that this means that all these non-local constraints are only restricted by the universal relations so far. So the issue is the non-existence of non-local knowledge in our network. It is desirable to refine those universal relations in a way that all relations that cannot hold with respect to the algebraic properties of the calculus at hand are thrown out. The standard approach in qualitative spatial reasoning is applying algebraic closure on the network. This approach is basically just an approximation, but this approximation might be good enough.

Research on "wayfinding choremes" by A. Klippel et al. [Klippel and Montello, 2007; Klippel *et al.*, 2005] claims a *cognitively adequate* representation of directions on decision points, i.e. crossings in our street networks. Basically there are 7 choremes that describe turning situations at crossings as depicted in Figure 3. These choremes are ignorant of the

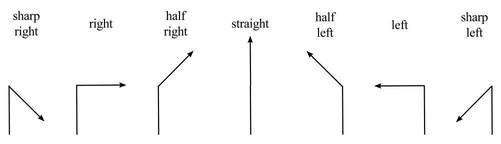

Figure 3: The seven wayfinding choremes

situation of "going back", which is formalized in $\mathcal{OPRA}$. Furthermore, for our navigation task the situation of running into a dead end can always appear and we need the possibility of turning around and leaving that dead end. The derivation of these choremes in based upon a sectorization of a circle as shown in Figure 4. With these sectors we would have

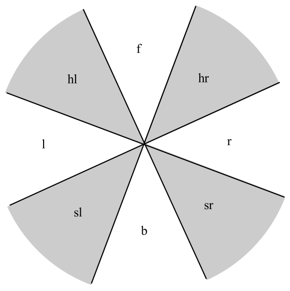

Figure 4: Sectors of a circle for wayfinding choremes

the choice of directions from l, r, f, b in Figure 2, sharp or half turns do not occur there. This sectorization clearly has a "back" sector and is quite close to the definition of the $\mathcal{OPRA}$ relations. The main difference is the lack of relations on a line. The size of the sectors in Figure 4 is only approximately described by Klippel. We are going to simulate these sectorization by $\mathcal{OPRA}$ relations of adequate granularity. Where the choice of granularity is a tradeoff between the minimum size of sectors and reasoning efficiency. We will use these Klippel's sectors encoded in $\mathcal{OPRA}$ to navigate our street network and examine its impact on the reasoning qualities.

We apply our techniques for techniques for deriving observations in $\mathcal{OPRA}$ and in the representation of Klippel's sectors in $\mathcal{OPRA}$ to test data to gain knowledge their fitness for navigation tasks in street networks. Since we believe that the best test data for street networks are the real ones, we use descriptions of street networks compiled out of maps from OpenStreetMap[1].

2 The $\mathcal{OPRA}$ calculus

The basic entity of the $\mathcal{OPRA}$ calculus are *oriented points*, these are points that have a position given by coordinates and an orientation. This orientation can be given as an angle with respect to an axis. A configuration of oriented points is shown in Figure 5.

Definition 1 (Oriented Point)**.** An *oriented point* is a tuple $\langle p, \varphi \rangle$, where p is a coordinate in $\mathbb{R}^2$ and φ an angle to an axis.

We also can describe an oriented point as a tuple of points $\langle p_0, p_i \rangle$ being located at p_0 and pointing to p_1. hence the direction is given by the vector from p_0 to p_1. From this description, we can compute the angle φ to the axis easily. By disregarding the lengths of the vectors, we arrive at Definition 1. The $\mathcal{OPRA}$ calculus defines relations between such

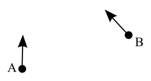

Figure 5: Oriented points

pairs of oriented points. These relations are of adjustable granularity, where this granularity is denoted by the index m of $\mathcal{OPRA}_m$. For the introduction of relations the plane around each oriented point is sectioned by m lines with one of them having the same orientation φ as the oriented point. The angles between all lines have to be equal. The sectors are numbered from 0 to $4m - 1$ counterclockwise. The label 0 is assigned to the direction with the same orientation as the oriented point itself. Such a sectioning is shown in Figure 6 this is in fact Figure 5 with the sectioning introduced. In fact, we introduce a set of angles

$$\bigcup_{0 \le i < 2m} \left\{ \left[i\frac{\pi}{m} \right], \left[i\frac{\pi}{m}, (i+1)\frac{\pi}{m} \right] \right\}$$

[1]http://www.openstreetmap.org/

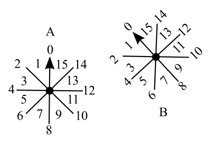

Figure 6: Oriented points with sectors

to partition the plane into the described sections. To introduce $\mathcal{OPRA}$ relations between two oriented points o and q, we need to distinguish between the two cases, if $\mathrm{pr}_1(o) = \mathrm{pr}_1(q)$ or not, where pr_1 is the projection to the first component of a tuple. I.e. we need to distinguish if both points have the same position in the plane.

A good auxiliary construction to introduce $\mathcal{OPRA}_m$ relations are *half relations*.

Definition 2. For two oriented points o and q we call $o \rhd q$ the *half relation* from o to q.

If we want to annotate a sector i or granularity m to a half relation, we shall write $o_m \rhd_i q$. A half relation determines the number i of the sector around o where q lies in if $\mathrm{pr}_1(o) \neq \mathrm{pr}_1(q)$ and the sector around o into q points into, if $\mathrm{pr}_1(o) = \mathrm{pr}_1(q)$. E.g. in Figure 6 the oriented points B lies in sector 13 of A and we obtain the half relation $A_4 \rhd_{13} B$. And for A with respect to B we get $B_4 \rhd_3 A$.

First we consider the case of $\mathrm{pr}_1(o) \neq \mathrm{pr}_1(q)$. We then get the $\mathcal{OPRA}_m$ relation from o to q as the product of $o_m \rhd_i q$ and $o_m \rhd_j q$, we will write this as $o_m\angle_i^j$. And for $\mathrm{pr}_1(o) = \mathrm{pr}_1(q)$, we get the $\mathcal{OPRA}_m$ relations as the product of $o\ s\ q$ and $o_m \rhd_j q$ written as $o_m\angle_s^j q$, where s is a special symbol describing the coincidence of the position of points.

The composition and converse tables for $\mathcal{OPRA}$ need to be calculated for any granularity of this calculus, fortunately there is a quite efficient algorithm for this task [Mossakowski and Moratz, to appear].

3 Factorizing the $\mathcal{OPRA}$ to cognitive adequacy

Investigations of Alexander Klippel et al. [Klippel *et al.*, 2005] investigated sector models as shown in Figure 7 for

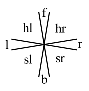

Figure 7: Klippel's relations

navigation tasks and claim their cognitive adequacy. They are using eight sectors

f	front
hl	half left
l	left
sl	sharp left
b	back
sr	sharp right
r	right
hr	half right

for their model. Nothing is said about the treatment of the borders of the sectors, i.e. about which sector the separating line belongs to, if it belongs to any. This question needs to be solved for simulating such a sectioning by a qualitative spatial calculus.

We are encoding Klippel's approach into $\mathcal{OPRA}_8$ (see Figure 8) and $\mathcal{OPRA}_{16}$ to be able to define f, b, l and r

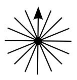

Figure 8: $\mathcal{OPRA}_8$

sectors that are suitably small and to get constraint network sizes that still can be handled by algebraic reasoners. The reasoner `GQR` [Gantner *et al.*, 2008] already needs 14GB of memory to start up with the $\mathcal{OPRA}_{16}$ composition table, without precaching the composition table for all general relations. For having suitably small sectors, we unite the $\mathcal{OPRA}_m$ ($m \in 2^n$ and $n > 2$) sectors via a mapping d as following.

$$\begin{aligned} f &\mapsto \{0, 1, 2, 4m-1, 4m-2\} \\ l &\mapsto \{m-2, m-1, m, m+1, m+2\} \\ b &\mapsto \{2m-2, 2m-1, 2m, 2m+1, 2m+2\} \\ r &\mapsto \{3m-2, 3m-1, 3m, 3m+1, 3m+2\} \end{aligned}$$

The Klippel sectors hl, sl, sr and hr are formed by the remaining $\mathcal{OPRA}$ sectors. For $n = 2$ the sectors would overlap with this approach. We decided to add the border lines of f, b l and r to the respective relations, since this still yields sectors for these relations for $m \mapsto \infty$ for $\mathcal{OPRA}_m$. With this we would recover $\mathcal{OPRA}_2$ from Klippel's approach for $m \mapsto \infty$. To apply this sectioning to $\mathcal{OPRA}_m$, for all sets $d_1, d_2 \in d(K)$ apply $d_1 \times d_2$ where K are Klippel's sectors, and add the sets $\{s\} \times d_1$ we call these sets D. From these sets of sectors we can easily define predicates $p_1 \ldots p_8$ that are $true$ if and only if a certain $\mathcal{OPRA}_m$ relation belongs to such a set lifted to $\mathcal{OPRA}_m$.

Example 3. We want to encode Klippel's sectioning into the sectioning of $\mathcal{OPRA}_8$, which has the half relations $0 \ldots 31$. With the above definitions we obtain the mapping

$$\begin{aligned} f &\mapsto \{30, 31, 0, 1, 2\} \\ hl &\mapsto \{3, 4, 5\} \\ l &\mapsto \{6, 7, 8, 9, 10\} \\ sl &\mapsto \{11, 12, 13\} \\ b &\mapsto \{14, 15, 16, 17, 18\} \\ sr &\mapsto \{19, 20, 21\} \\ r &\mapsto \{22, 23, 24, 25, 26\} \\ hr &\mapsto \{27, 28, 29\} \end{aligned}$$

This mapping of Klippel's sectors to the sectors of $\mathcal{OPRA}_8$ is shown in Figure 9. Please note that the $\mathcal{OPRA}_8$ emulations of f, l, b and r are still quite big sectors with 22.5°. Another drawback is that all sectors are the same size.

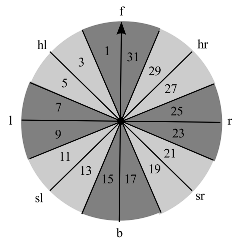

Figure 9: Mapping Klippel to $\mathcal{OPRA}$

Example 4. We can get smaller sectors by encoding Klippel's sectors into the sectors for $\mathcal{OPRA}_{16}$ as

$$\begin{array}{rcl}
f & \mapsto & \{62, 63, 0, 1, 2\} \\
hl & \mapsto & \{3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\} \\
l & \mapsto & \{14, 15, 16, 17, 18\} \\
sl & \mapsto & \{19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29\} \\
b & \mapsto & \{30, 31, 32, 33, 34\} \\
sr & \mapsto & \{35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45\} \\
r & \mapsto & \{46, 47, 48, 49, 50\} \\
hr & \mapsto & \{51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61\}
\end{array}$$

The sectors for f, l,b and r now have a size of $10.25°$ and the remaining sectors are bigger than them, what is closer to Klippel's intention. The issue with working with $\mathcal{OPRA}_{16}$ is already the sheer size of the composition table with 4160^2 entries and the long descriptions of constraint networks in 4160 base relations.

In the end we have a trade-off between staying close to Klippel's intentions, which can be done by a high arity $\mathcal{OPRA}$ calculus and the possibility to perform reasoning over constraint networks. But for our task of navigation the reasoning results to not have to be perfect, they just need to be good enough. Hence, we hope that on constraint networks of reasonable size $\mathcal{OPRA}_8$ and $\mathcal{OPRA}_{16}$ do the job. It would also be nice to have high arity $\mathcal{OPRA}$ calculi for having the possibility of being able to compare the impact of the size of f, l b and r in more detail.

4 From observations to a constraint network

As stated it is our aim to investigate navigation based on local observations using the $\mathcal{OPRA}$ calculus. A good source for realistic data about street networks is the world itself. We are using street networks that have been retrieved from OpenStreetMap[2], make local observations on them and formalize these observations in $\mathcal{OPRA}$. We simplify the OpenStreetMap data in the sense that we abstract from bends in streets. Our streets are just straight lines. With algebraic reasoning global knowledge can be deduced from local observations. For algebraic reasoning we use the tools `GQR` [Gantner *et al.*, 2008] and `SparQ` [Wallgrün *et al.*, 2006, 2009]. Using this overall knowledge, we navigate through the described street network.

[2]`http://www.openstreetmap.org/`

In the rest of this section we are using the street network in Figure 10 as the source for our examples. In our street

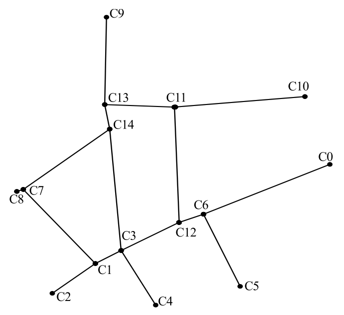

Figure 10: A street network

networks, we label crossings by C_i with $i \in \mathbb{N}$. Please note that our definition of crossings at this point includes dead-ends. In our example these are the dots. The lines depict streets between crossings. We call crossings C_i and C_j with $i \neq j$ that are connected by a street *adjacent*.

4.1 Local Observations

It is our aim to navigate with knowledge that people can make at crossings. When walking to a crossing, you know where you came from and hence your orientation. Further you can see which orientation the other streets at the crossing have with respect to your orientation. And of course you know that streets are streets with a crossing at both ends. You do not know what the situation at any other crossing looks like. This is an abstraction from very short streets.

In the first step of the formalization of our local observations we need to derive oriented points from a given street network. For any point C_i in the network determine the set A of adjacent oriented points. For any $C \in A$ introduce the oriented point $\langle C_i, C \rangle$. For the sake of brevity, we will also write C_iC for such a tuple. As described in Section 2 this representation of an oriented point still contains unnecessary information about the length of the vector from C_i to C, but this does no harm.

Example 5. Consider the network given in Figure 10 and the point C_6. The set of adjacent points to C_6 is $\{C_0, C_5, C_{12}\}$, we hence introduce the set of oriented points $\{\langle C_6, C_0\rangle, \langle C_6, C_5\rangle, \langle C_6, C_{12}\rangle\}$ or written in the short form $\{C_6C_0, C_6C_5, C_6C_{12}\}$.

In the second step, we define the streets. For each oriented point C_iC_j, we define the street via the $\mathcal{OPRA}_m$ relation $C_iC_j \;{}_m\angle_0^0\; C_jC_i$. The oriented point C_jC_i exists, since the streets in our network are not directed and hence if C_j is adjacent to C_i then C_i is adjacent to C_j.

Example 6. For a street in shown in Figure 11. This is the

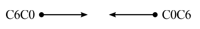

Figure 11: A street

street between the points C_6 and C_0, hence we have introduced the oriented points C_6C_0 and C_0C_6 in the previous step (see Example 5) at the respective locations to point to each other. So we introduce the relation $C_6C_0 \; {}_m\angle_0^0 \; C_0C_6$.

In the third step, we add the local observations. For each oriented point C_iC_j form the set P of oriented points with for each $p \in P$ the properties $pr_1(p) = C_i$ and $pr_2(p) \neq C_j$ hold. Where pr_1 is the projection to the first component of a tuple and pr_2 to the second one. For each $p \in P$, we form the $\mathcal{OPRA}_m$ relation $C_iC_j \; {}_m\angle_s^{C_iC_j \triangleright p} \; p$. Since $C_i = pr_1(p)$, the first half-relation is clearly s, the computation of the second one will be explained in section Section 4.2.

Example 7. We again refer to Figure 10 and the oriented points introduced in Example 5. Consider the oriented point C_6C_0. For this point we get $P = \{C_6C_5, C_6C_{12}\}$ and the relations

$$C_6C_0 \; {}_m\angle_s^{C_6C_0 \triangleright C_6C_5} \; C_6C_5$$
$$C_6C_0 \; {}_m\angle_s^{C_6C_0 \triangleright C_6C_{12}} \; C_6C_{12}$$

In Algorithm 1 we show a slightly optimized version of the described algorithm where steps two and three are amalgamated.

Algorithm 1 Deriving Observations

```
 1: $\mathcal{C}$ is the set of nodes of a street network
 2: $\mathcal{S}$ the set of streets as tuples of start and end points
 3: $O$ is the set of oriented points
 4: $R$ is the set of relations
 5: $m$ is the granularity of the $\mathcal{OPRA}$ calculus
Require: $O = \emptyset$ and $R = \emptyset$ and $m > 0$
 6: Require a correct description of a street network
Require: $\forall C \in \mathcal{C}.\exists s \in \mathcal{S}.C = \mathrm{pr}_1(s) \vee C = \mathrm{pr}_2(s)$
Require: $\forall s \in \mathcal{S}.\exists C_1 \in \mathcal{C}.\exists C_2 \in \mathcal{C}.s = \langle C_1, C_2 \rangle \wedge C_1 \neq C_2$
 7: Introduction of oriented points
 8: for all $C \in \mathcal{C}$ do
 9:     for all $s \in \mathcal{S}$ do
10:         if $\mathrm{pr}_1(s) = C$ then
11:             $O := O \cup \{\langle C, \mathrm{pr}_2(s) \rangle\}$
12:         end if
13:     end for
14: end for
15: Definition of streets and local observations
16: for all $o \in O$ do
17:     $R := R \cup \{o_m\angle_0^0 \langle \mathrm{pr}_2(o), \mathrm{pr}_1(o) \rangle\}$
18:     for all $p \in O$ do
19:         if $\mathrm{pr}_1(o) = \mathrm{pr}_1(p)$ and $\mathrm{pr}_2(o) \neq \mathrm{pr}_2(p)$ then
20:             $R := R \cup \{o_m\angle_s^{o \triangleright p} p\}$
21:         end if
22:     end for
23: end for
24: return $R$
```

If we are working with an approach as suggested by Klippel, we add another step that replaces the $\mathcal{OPRA}$-relations by sets of relations as described in Section 4.3.

4.2 Deriving $\mathcal{OPRA}$-relations

For our observations taken in Section 4.1, we need a way to derive $\mathcal{OPRA}$-relations from tuples of points (or line segments). In particular we need is computation in Algorithm 1 Line 20, where $o \triangleright p$ was not determined so far.

By scrutinizing the definitions of the $\mathcal{OPRA}_m$ relations, we see that for any $C_kC_l \; {}_m\angle_i^j \; C_tC_v$, there is little dependence between the i and j. In fact, the only dependence is on i being s or not. We can distinguish these cases easily by determining if $C_k = C_t$ or not. If $C_k = C_t$, we know that $i = s$ and can determine j as a half relation. If $C_k \neq C_t$, there is no dependence between i and j and we can determine both via half relations. We can apply Algorithm 2 to determine $\mathcal{OPRA}$-relations between two oriented points C_kC_l and C_tC_v. The main issue that is still open is the derivation

Algorithm 2 Computing $\mathcal{OPRA}$-relations

```
 1: $C_kC_l$ oriented point
 2: $C_tC_v$ oriented point
 3: $m$ granularity of $\mathcal{OPRA}$
Require: $m > 0$
 4: if $C_k = C_t$ then
 5:     return ${}_m\angle_s^{C_kC_l \triangleright C_tC_v}$
 6: else
 7:     return ${}_m\angle_{C_kC_l \triangleright C_tC_v}^{C_tC_v \triangleright C_kC_l}$
 8: end if
```

of the half relations. In fact the needed calculation for the $\mathcal{OPRA}$ relations in Algorithm 1 can be reduced to this step (refer to Algorithm 1 Line 20). All other information in the involved $\mathcal{OPRA}$ relations can already be derived directly in that algorithm.

To determine the $\mathcal{OPRA}_m$ half relations between oriented points C_kC_l and C_tC_v, we determine sectors of the unit circle[3] in the Euclidean plane that correspond to those relations. Then, we compute the angle from C_kC_l to C_tC_v and determine into which sector this angle belongs. This directly yields the half relation. In Figure 12 these sectors are shown for $\mathcal{OPRA}_1$ to $\mathcal{OPRA}_4$. By inspecting the definition of

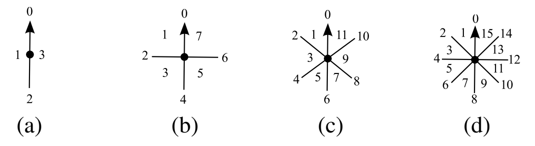

Figure 12: Sectors of the circle for $\mathcal{OPRA}_1$ (a), $\mathcal{OPRA}_2$ (b), $\mathcal{OPRA}_3$ (c), and $\mathcal{OPRA}_4$ (d)

$\mathcal{OPRA}$ relations, we also see that half relations with an even identifier are relations on a line, while the ones with an odd identifier are relations in a plane. For an example inspect Figure 12.

The sectioning for $\mathcal{OPRA}_m$ is done by identifying an angle interval with every element of the cyclic group $\mathbb{Z}_{4m}$ as

$$[i]_m = \begin{cases} \left]2\pi\frac{i-1}{4m}, 2\pi\frac{i+1}{4m}\right[& \text{if } i \text{ is odd} \\ \left\{2\pi\frac{i}{4m}\right\} & \text{if } i \text{ is even} \end{cases}$$

[3]In fact the radius of the circle does not matter, since we are disregarding lengths.

Please note that these intervals are normalized to the representation of angles in the interval $[0, 2\pi[$. For an implementation one can create a look-up-table with the borders of the respective intervals and the respective values for i.

To compute the needed angle from C_kC_l and C_tC_v, we form the vectors

$$\vec{a} = \begin{pmatrix} (C_k)_x - (C_l)_x \\ (C_k)_y - (C_l)_y \end{pmatrix} \qquad \vec{b} = \begin{pmatrix} (C_k)_x - (C_t)_x \\ (C_k)_y - (C_t)_y \end{pmatrix}$$

if $C_k \neq C_t$ and

$$\vec{a} = \begin{pmatrix} (C_k)_x - (C_l)_x \\ (C_k)_y - (C_l)_y \end{pmatrix} \qquad \vec{b} = \begin{pmatrix} (C_t)_x - (C_v)_x \\ (C_t)_y - (C_v)_y \end{pmatrix}$$

if $C_k = C_t$. The operations $(\_)_x$ and $(\_)_y$ denote the projections to the x and y coordinate of a point. The case distinction takes credit to the fact that in the case of positional equality of oriented points the angle between the orientations is used as an $\mathcal{OPRA}$ half relation. We determine the angle ϕ' from $\vec{a}$ to $\vec{b}$, we use the atan2 function which yields values in the interval $]-\pi, \pi]$ as:

$$\phi' = \operatorname{atan2}(\vec{a}_x\vec{b}_y - \vec{a}_y\vec{b}_x, \vec{a}_x\vec{b}_x + \vec{a}_y\vec{b}_y)$$

we normalize our angles to the interval $[0, 2\pi[$ by

$$\phi = \begin{cases} \phi' + 2\pi & \text{if } \phi' < 0 \\ \phi' & \text{if } \phi' \geq 0 \end{cases}$$

to get an angle that ϕ that is compatible to the intervals in our definition of $[i]_m$. To determine the half relation for ϕ, we just need look up the appropriate interval that has been pre-calculated.

4.3 Factorizing the $\mathcal{OPRA}$-relations to cognitive adequacy

Additionally to investigating navigation with $\mathcal{OPRA}$ relations, we also want to emulate relations as proposed by Klippel [Klippel *et al.*, 2005] in $\mathcal{OPRA}_m$. For this reason, we use n unary predicates p_i with $1 \leq i \leq n$ that partition the set of the $\mathcal{OPRA}_m$ base relations. If an $\mathcal{OPRA}_m$ relation $o_m\angle_s^t q$ has been determined between o and q with Algorithm 2, we form the new relation relations

$$o \left\{ r \mid p_i(r) = p_i(_m\angle_s^t) \text{ for } 1 \leq i \leq n \right\} q$$

where r is an $\mathcal{OPRA}_m$ relation. We do this for all pairs of oriented points that haven been introduced in Section 4.2. All other pairs are in the universal relation anyways. For this factorization adjacent sectors will be united to a single relation, but the operation involved works for all kinds of predicates, even tough the usefulness might be questionable in many cases.

Algorithm 3 Factorization (to Klippel's description)

```
 1: $R$ set of determined $\mathcal{OPRA}_m$ relations
 2: $p_i$ with $1 \leq i \leq n$ set of predicates
 3: $R'$ set of output relations
Require: $R' = \emptyset$
 4: for all $o_m\angle_s^t q \in R$ do
 5:     $R_{\text{tmp}} = \emptyset$
 6:     for all $_m\angle_x^y \in \mathcal{OPRA}_m$ do
 7:         prop $= true$
 8:         for $1 \leq i \leq n$ do
 9:             prop := prop $\wedge\, p_i(_m\angle_s^t) = p_i(_m\angle_x^y)$
10:         end for
11:         if prop then
12:             $R_{\text{tmp}} := R_{\text{tmp}} \cup \{_m\angle_x^y\}$
13:         end if
14:     end for
15:     $R' := R' \cup \{oR_{\text{tmp}}q\}$
16: end for
17: return $R'$
```

5 Navigation

Having obtained a description of a street network as an $\mathcal{OPRA}$ constraint network, we are able to apply algebraic closure on them to obtain refined constraint networks. Since we are starting from consistent descriptions, we do not have to fear that algebraic closure detects inconsistencies. In fact, in the descriptions from Section 4.2 and Section 4.3 many universal relations are contained, since we only made local observations. E.g. the relation between $C_{13}C_9$ and C_6C_5 is universal, since these oriented points cannot be observed together locally at a crossing. Algebraic closure only approximates consistency for $\mathcal{OPRA}$, hence our refined constraint networks might be too big, but this is no issue for our navigation task, it might just lead to detours.

Starting from a refined constraint network of a street network, we want to navigate through it (hopefully without taking too many detours). We are going to apply a least angle strategy for navigation with imprecise and maybe faulty data. We can base the navigation on half relations. Just remember the definition of $\mathcal{OPRA}$ relations. If $C_kC_l\ {_m\angle_i^j}\ C_tC_v$, then C_tC_v is in sector i of C_kC_l with granularity m. The way backwards is of no interest for forward navigation. Based on this we introduce weights on half $\mathcal{OPRA}$ relations. Going forward and taking slight bends is normally good for such a navigation, taking sharp bends and going back is bad. We can assign the weights $w(i)$ to $\mathcal{OPRA}_m$ half relations i as

$$w(i) = \begin{cases} i & \text{if } 0 \leq i \leq 2m \\ 4m - i & \text{if } 2m < i < 4m \end{cases}$$

this yields a weight distribution that assigns the lowest weights to going forward and making slight bends.

Example 8. Consider again the sectors for $\mathcal{OPRA}_4$ in Figure 12d. Applying weights with respect to our formula yields the distribution

$$\begin{array}{rr} w(0) = 0 & w(5) = w(11) = 5 \\ w(1) = w(15) = 1 & w(6) = w(10) = 6 \\ w(2) = w(14) = 2 & w(7) = w(9) = 7 \\ w(3) = w(13) = 3 & w(8) = 8 \\ w(4) = w(12) = 4 & \end{array}$$

which is depicted in Figure 13. We can observe that going forward or taking slight bends has small weights whereas going backwards and taking sharp bends leads to high weights.

In the navigation task, we start at a point *from* and want to reach a point to. The current point is $start$ initialized

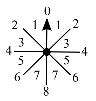

Figure 13: $\mathcal{OPRA}_4$ weight distribution

by *from*. These are point that represent crossings in the street scenario, not oriented points. We determine the set of all $\mathcal{OPRA}$ relations $o_m\angle_i^j p$ with $\mathrm{pr}_1(o) = start$ and $\mathrm{pr}_1(p) = to$. We then form the half relations $o \rhd to$ as

$$o \rhd to = \sum_{\mathrm{pr}_1(p)=to} \left\{ o_m \rhd_i to \mid o_m\angle_i^j p \right\}$$

We then normalize the weights as

$$w = \frac{\sum\limits_{i=o\rhd to} w(i)}{|o \rhd to|} \cdot penalty$$

where *penalty* is a property of $\mathrm{pr}_2(o)$ that is initialized with 1 and incremented by 1 each time $\mathrm{pr}_2(o)$ is visited on a path. This is introduced to make loops bad ways to go and to get out of dead ends. We now take all o with the minimum w, if there is more than one, we choose by fortune. $\mathrm{pr}_2(o)$ becomes our new point *start* and its penalty is increased since it is visited. We repeat this, until *to* is reached. The algorithm for navigation is shown in Line 4.

Algorithm 4 Navigation

```
 1: $from$ start point
 2: $to$ end point
 3: $start = from$
 4: $ROUTE := start$
 5: **while** $start \neq to$ **do**
 6:     $R := \emptyset$
 7:     $W := \emptyset$
 8:     **for all** $p$ with $\mathrm{pr}_1(p) = to$ **do**
 9:         **for all** $o$ with $\mathrm{pr}_1(o) = start$ **do**
10:             **if** $R$ contains a relation $o \rhd p$ **then**
11:                 $R := (R \setminus o \rhd p) \uplus o(\rhd \uplus o_m \rhd_i)p$ if $o_m\angle_i^j p$
12:             **else**
13:                 $R := R \uplus o_m \rhd_i p$ if $o_m\angle_i^j p$
14:             **end if**
15:         **end for**
16:     **end for**
17:     **for all** $r \in R$ **do**
18:         $W := W \cup (r, \mathrm{weight}(r))$
19:     **end for**
20:     $cand := r \in R$ with $w(r) = min$
21:     $next :=$ random element from cand
22:     increase $\mathrm{pr}_2(next).penalty$
23:     $start := \mathrm{pr}_2(next)$
24:     $ROUTE := ROUTE \circ start$
25: **end while**
26: **return** $ROUTE$
```

The assignment of weights is shown in Algorithm 5. Please note that we have used disjoint unions of the half relation symbols in Line 4, since those lead to better navigation results in our first experiments, even for low granularities.

Algorithm 5 Weight assignment: weight

```
 1: $o \rhd p$ is given
 2: $W := 0$
 3: **for all** $r \in \rhd$ **do**
 4:     **if** $0 \leq r \leq 2m$ **then**
 5:         $W := W + r$
 6:     **else**
 7:         $W := W + 4m - r$
 8:     **end if**
 9: **end for**
10: $W := \frac{W}{|\rhd|} \cdot \mathrm{pr}_2(o).penalty$
11: **return** $W$
```

6 Experiments

Finding good data for experiments with navigation based on local observations is a hard task. A big issue is that the maximal size of a street network that we can use for navigation is limited by the number of nodes and by the granularity of the underlying $\mathcal{OPRA}$ calculus. The time needed for applying the algebraic closure algorithm rises steeply with any of these two parameters growing. As a rule of thumb we can say that we can e.g. handle street networks with around 120 to 170 points with $\mathcal{OPRA}_8$ in a reasonable time (2 to 4 hours) when computing algebraic closure with GQR. (However, note that this has to be computed only once, and can then be used for as many navigation tasks as wanted.) On the other hand a network in 170 points in our representation (the reduction of data is described in Figure 4) does not cover big areas in most cases. For example the network shown in Figure 14 that derived from the data on OpenStreetMap (latitude 51.8241200 longitude 9.3117500) for a village with about 1400 inhabitants already has 117 points. Large cities like Paris of course have many more points in our representation and cannot be handled efficiently with the algebraic reasoners. But on the

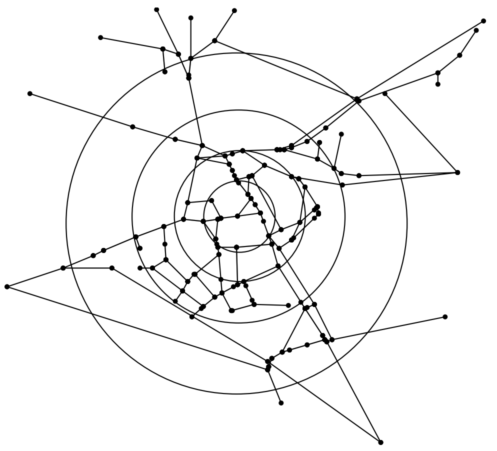

Figure 14: A street network of a village

other hand, we want to observe navigations along paths of very differing lengths, including very long paths to be able to judge the navigation properties of our networks based on local observations under very differing circumstances. For long paths networks of a sufficiently big size are needed. Unfortunately this problem grows even bigger by the fact that the closer we get to boundary of our street network the worse our local observations and refinements will be. For a point in the middle of a street network (as in the inner circle in Figure 14 there are many points around it in all directions with observations being made, putting this point into its place in a qualitative sense. In the middle circle the observations around a certain point already get sparser and information about the points gets less certain, this gets worse in the outer circle. Outside of the outer circle information about the points is very bad. In fact, it turned out, that navigating into the dead ends at the boundary of the map is very alluring, since their position with respect to other points is not very restricted. For meaningful experiments about the navigation performance, the need street networks that are big enough to provide an area in the center for which enough information can be derived.

Our test data has hence to consist of street networks that are small enough to be manageable with qualitative reasoners and that are big enough to yield enough information. For the first requirement networks in no more than 20 points would be nice, for the second one the whole world, since then there would be no boundary problem.

The results of our experiments are available at `http://www.informatik.uni-bremen.de/~till/fuerstenau_K8.html` (for Klippel$_8$) and `http://www.informatik.uni-bremen.de/~till/fuerstenau_O8.html` (for $\mathcal{OPRA}_8$). We have made 66 navigation experiments. The average path length was 16.0 (using $\mathcal{OPRA}_8$ factorized due to Klippel's sectorization of the circle) and 16.2 (using $\mathcal{OPRA}_8$) with our algorithm based on local observations, while that of a shortest path (using the complete map) was 13.2, and that of a uniform shortest path (counting all way lengths as 1) was 11.0. The average length of a random walk was 718.0. As expected, the standard deviation of our algorithm is significantly higher than that of shortest paths:

	Klippel$_8$	$\mathcal{OPRA}_8$	shortest path	uniform shortest path	random walk
mean	16.0	16.2	13.2	11.0	718.0
standard deviation	9.3	9.3	5.1	3.5	519.9

However, our algorithm still performs quite well when compared with shortest paths.

Conclusion

Our experiments show that navigation based on local observations of an agent performs fairly well when compared with shortest paths computed using global map knowledge, and orders of magnitude better than randowm walk.

When making the experiments, we quickly reached the limits of the standard qualitative spatial reasoning tools. The constraint networks generated by our algorithms thus could been seen as a challenge for (further) improving performace of these tools.

Further experiments should be done with different test data. Particularly interesting would be street networks of diverse style. It is e.g. interesting to use layouts of planned and grown cities and villages. Further gyratory traffics (e.g. at Place-Charles-de-Gaulle) of increased interest. With a larger set of experiments, the approach could be used to systematically evaluate street networks with respect to their local navigation quality, and study which features of street networks influence this quality.

References

J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, pages 832–843, 1983.

Z. Gantner, M. Westphal, and S. Wölfl. GQR - A Fast Reasoner for Binary Qualitative Constraint Calculi. In *Proc. of the AAAI-08 Workshop on Spatial and Temporal Reasoning*, 2008.

Alexander Klippel and Daniel R. Montello. Linguistic and nonlinguistic turn direction concepts. In Stephan Winter, Matt Duckham, Lars Kulik, and Benjamin Kuipers, editors, *Spatial Information Theory, 8th International Conference, COSIT 2007, Melbourne, Australia, September 19-23, 2007, Proceedings*, volume 4736 of *Lecture Notes in Computer Science*, pages 354–372. Springer, 2007.

A. Klippel, H. Tappe, L. Kulik, and P. U. Lee. Wayfinding choremes–a language for modeling conceptual route knowledge. *Journal of Visual Languages and Computing*, 16(4):311 – 329, 2005. Perception and ontologies in visual, virtual and geographic space.

T. Mossakowski and R. Moratz. Qualitative reasoning about relative direction on adjustable levels of granularity. *Journal of Artificial Intelligence*, to appear.

D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In Bernhard Nebel, Charles Rich, and William Swartout, editors, *Proc. of KR-92*, pages 165–176. Morgan Kaufmann, 1992.

J. O. Wallgrün, L. Frommberger, D. Wolter, F. Dylla, and C. Freksa. Qualitative Spatial Representation and Reasoning in the SparQ-Toolbox. In T. Barkowsky, M. Knauff, G. Ligozat, and D. R. Montello, editors, *Spatial Cognition*, volume 4387 of *Lecture Notes in Comput. Sci.*, pages 39–58. Springer, 2006.

J. O. Wallgrün, L. Frommberger, F. Dylla, and D. Wolter. SparQ User Manual V0.7. User manual, University of Bremen, January 2009.