Relative Representation of Spatial Knowledge: The 2-D Case

Daniel Hernández*
Institut für Informatik
Technische Universität München
Arcisstr. 21, D-8000 München 2

e-mail: danher@informatik.tu-muenchen.de

Revised Version January 29, 1991

Abstract—*There have been some straightforward efforts to extend Allen's interval-based temporal logic to spatial dimensions by using Cartesian tuples of relations [6]. We take a different approach based on a study of the kind of information that best relates two entities in 2-dimensional space qualitatively. The relevant spatial categories turn out to be "projection" and "orientation". We define a small set of spatial relations and stress the importance of making their* reference frames *explicit. Furthermore, we introduce "abstract maps", an analogical representation that inherently reflects the structure of the represented domain, and demonstrate their use in spatial reasoning. This scheme also facilitates "coarse" reasoning and the hierarchical organization of knowledge. These representational issues form the basis for an experimental system to develop "cognitive maps" from 2-D scanned layout plans of buildings.*

1 Introduction

Cognitive spatial concepts are *qualitative* in nature. Despite the widespread belief that geometric models are "better", in the sense of being more precise, this preciseness is not present (and not needed!) in cognitive models of space. Even more, due to the *limited acuity of perception* [2], geometric models might actually falsify the representation by forcing discrete decisions. We model the qualitativeness of "cognitive space" by using a *relative* representation of spatial knowledge based on *comparative relations* (over selected spatial dimensions) among objects, and between objects and distinguished reference structures (e.g. landmarks and boundaries). Such a relative representation avoids the falsifying effects of exact geometric representations by "not committing" to all aspects of the situation being represented. By doing so, however, it is also "under-determined" in the sense that it might correspond to many "real" situations. The reason why it still can be effectively used to solve spatial problems is that those problems are always embedded in a particular context. We claim that directly modelling this qualitativeness can lead to more intuitive user interfaces for applications such as Geographic Information Systems (GIS), as well as to more efficient ways to handle partial and uncertain spatial information (Freksa argues this convincingly in [3]).

*Research supported by Deutsche Forschungsgemeinschaft under grant Br 609/4-2 and Siemens AG

2 Relative Representations

In a much quoted paper [1] Allen introduced an interval-based temporal logic, in which knowledge about time is maintained qualitatively by storing comparative relations between intervals. There have been some straightforward efforts to extend this temporal approach to spatial dimensions by using tuples of relations [6]. However, while the "cognitive plausibility" of Allen's approach carries over to the one-dimensional spatial case, it gets lost as soon as Cartesian tuples of relations are introduced to handle higher dimensions. People don't go around decomposing the world into three axes and then determining a qualitative relationship for an interval on each of them! We take a different approach based on a study of the kind of information that best relates two entities in 2-dimensional space *qualitatively*.

2.1 2-Dimensional Scenes

The type of spatial scenes that we want to model are *2-D projections* of 3-D scenes, i.e. we allow overlapping objects[1] (see for example the layout plan of an office in Fig. 2a). Furthermore, we ignore shape by considering only either "convex" objects or canonically oriented "delineative rectangles" enclosing the objects to be represented. (Size also turns out to play only an implicit role in the kind of relations we consider.)

Determining relations in an n-dimensional space requires an "external" observer in the $(n+1)$st-dimension. This is also true for $n = 3$, since the passage of time, which constitutes the fourth dimension, is necessary to establish relations in 3-D space. For the 2-D case at hand, we assume an external observer that establishes the relations by looking "from above" (3rd dimension) at the 2-D situation. It is important to distinguish the external observer from a possible "point of view" embedded in the scene. Such points of view are used as reference frames in some types of relations (see deictic type of use below).

While in the temporal domain the "beginning" of an interval comes always before its "end" (due to the irreversibility of time), in the spatial domain (even in the one-dimensional case) beginnings and ends depend on intrinsic properties of the objects described and/or on the point of view of the observer. The issue of the reference frame of a spatial relation becomes crucial in two- and higher-dimensional spaces, as we will see shortly.

2.2 Projection and Orientation

Our goal is to establish qualitative spatial relations between objects in a cognitively plausible way. When comparing the position of two objects in 2D-space there are two relevant dimensions: projection and orientation. By "projection" we mean how the boundaries of the objects relate. It comprises the relations `inclusion`, `overlap`, `tangency` and `disjointness` (see Fig. 1). In a quantitative system one would use distance instead. Distance, however, cannot be determined qualitatively, i.e. by comparing just the two objects involved. It requires an external frame with respect to which the positions of the objects being compared must be established first. Thus, in a sense, projection is used in lieu of "qualitative distance". By "orientation" we mean where the objects are placed relative to one another and comprises the relations `front`, `back`, `left`, `right`, `left-back`, `right-back`, `left-front`,

[1] Note, however, that in the projection it is not possible to distinguish which object is above the other one.

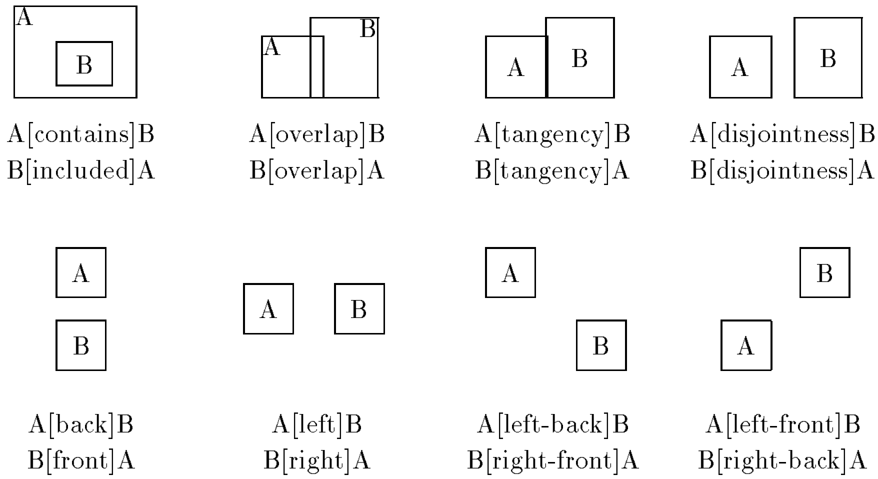


Figure 1: Projection and orientation relations

`right-front`[2]. Here `front`, `back`, `left` and `right` are taken to be abbreviations for "`front` but not `left-front` nor `right-front`" and so on. That is, `left-back`, for example, is not a finer distinction than say `back` or, with other words, all eight orientations have the same resolution. We do allow for various levels of granularity, both on projections and orientations, but in hierarchically organized homogenous layers (see section 3.2). "Factoring out" orientation reduces the number of projection relations from the 13 used by Allen to five (`inclusion` splits in two — see below) and makes them (except for `inclusion`) *symmetric*, for example in 2-D: A [`overlap`] B = B [`overlap`] A.

`Inclusion` is special in several ways. First, it is asymmetrical, i.e. it does make a difference (in 2-D) if A is `included` in B or if A `contains` B, except, of course, in the case of equality (or rather "perfect overlap") of objects, which is treated here as "mutual inclusion". Second, it precludes orientation, i.e. in our framework if two objects are in an `inclusion` relation it is impossible to establish their relative orientation. Third and most important, it is the interface to the hierarchical organization of abstract maps, since all objects `included` in another one can be manipulated at a higher level by manipulating the parent object. For example, all the objects in a room inherit the orientation of that room with respect to others in the building.

2.3 Reference Frames

We have to make explicit not only the *primary object* (the one to be located), the *reference object* (the one in relation to which the primary object is located), and the projection/orientation pair relating them, but also the *reference frame*, i.e. "the orientation that determines the direction in which the primary object is located in relation to the reference object" [9, p. 95]. Given for example a sentence like *The ball is in front of the car*, studies of the use of projective spatial prepositions in natural language reveal three types of reference frames:

[2] The names of the relations were chosen purposely not to coincide with the spatial prepositions to which they are related, because there is more to the use of prepositions in natural language than what a straightforward mapping would suggest.

intrinsic:	when the orientation is given by some inherent property of the reference object (e.g., w.r.t. car front)
extrinsic:	when external factors impose an orientation on the reference object (e.g., w.r.t. actual direction of the motion of the car)
deictic:	when the orientation is imposed by the point of view from which the reference object is seen (from *within* the scene, e.g. by an internal observer or from the speaker's point of view)

Criteria for determining the intrinsic orientation of objects and places are among others: the characteristic direction of motion or use, the side containing perceptual apparatus, the side characteristically oriented towards the observer, the symmetry of objects.

2.4 Expressing Spatial Knowledge

Now, how are these relations used to express spatial knowledge? We maintain knowledge about the relative position of two objects that could be declaratively stated as follows:

```
<primary object, [projection,orientation], ref. object, ref. frame>
```

where the reference frame in respect to which the orientation is determined can be:

1. *implicit*, i.e. the intrinsic orientation of the parent object (the one that includes the reference object) is used as reference. For example, the orientation of the objects in an office is given implicitly by the intrinsic orientation of the office as a whole (which in turn could be determined by such factors as typical entrance or placement of windows).

2. *explicit*, i.e. of the form:

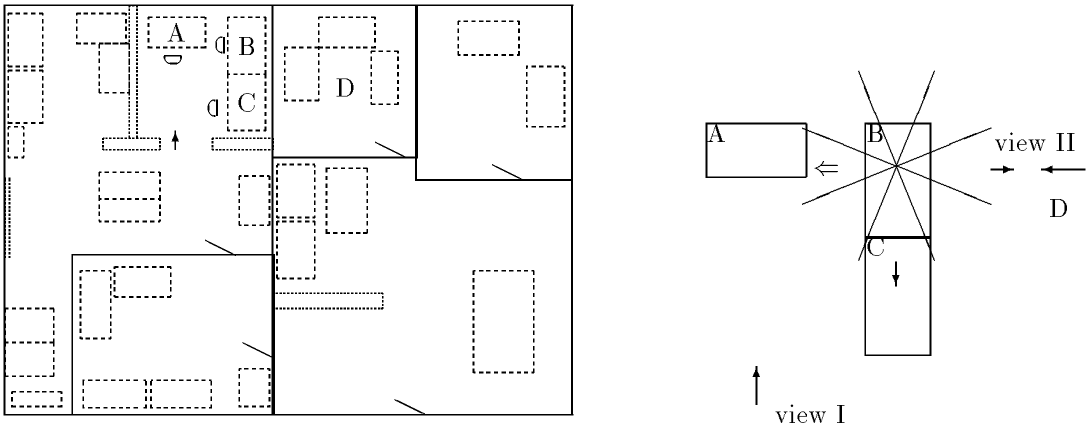
   {`type-of-use` ⟦, `dex-relation` ⟦,`trans`⟧⟧}

   An explicit reference frame must specify its `type-of-use`, which can be one of intrinsic, extrinsic or deictic as explained above, and optionally (indicated by double square brackets), if the `type-of-use` is not `intrinsic`, the `dex-relation`, which relates the external object or observer to the reference object in an intrinsic manner. A further `trans`-formation might be required, to determine the back/front sides of the reference object with respect to the observer ("mirror" transformation: fronts point to each other, backs point in opposite directions; "tandem" transformation: both fronts and both backs point in respective common directions).

The implicit orientation is used as the canonical reference frame. That is, for reasoning purposes, all relations between objects are converted first into the implicit form. This is the most natural representation for most applications (see example below). It corresponds, for example, to the act of an external observer of viewing the layout plan of an office from above such that the labels (assumed to convey the orientation of the office as a whole) are "right side up". On "larger scale" spaces the "preferred" orientation might be given by global directions (north, south, east, west).

2.5 Example

As an example consider the following natural language descriptions of spatial relations (see Fig. 2a/b):



(a) Layout plan of an office (b) Two views on relevant relations

Figure 2: Example of 2-D projection of a scene

(1) *Petra's desk (A) is to the left of Daniel's desk (B).* (Fig. 2b, view I)

(2) *Petra's desk (A) is behind Daniel's desk (B) as seen from Christian's (D) office.* (Fig. 2b, view II)

The relative spatial information contained in these sentences can be expressed declaratively as:

```
(1) REL1 = <A, [d,l], B>

(2) REL1 = <A, [d,b], B, {deictic, REL2, mirror}>
    REL2 = <D, [t,b], B, {intrinsic}>
```

In (1) no point of view is mentioned, so we assume the scene's implicit orientation which is given by the entrances and the position of the windows (view I in Fig. 2b). To see the difference between implicit (see above) and intrinsic orientations, note that A is to the intrinsic *front* of B, since B's intrinsic front as a desk is the side where the drawers open to or where one typically sits (see double arrow in Fig. 2b). However, if we were to store all relations with an intrinsic reference frame, we would have to transform back and forth between the intrinsic and the implicit orientations for most queries. In (2) a point of view is explicitly mentioned. Here an explicit reference frame is necessary to establish the orientation of the point of view with respect to the reference object in an intrinsic manner.

3 Abstract Maps

We could at this point use general constraint satisfaction mechanisms (see e.g. [8] for an overview) to operate on this representation. Note, however, that the relations form a "structural domain": orientations form a circular structure, whereas projections form linear neighborhoods. We would like to use an internal representation that inherently reflects as much as possible the properties of the represented domain (analogical principle). We introduce "abstract maps" containing for each object in a scene a data structure called "rpon" (for relative

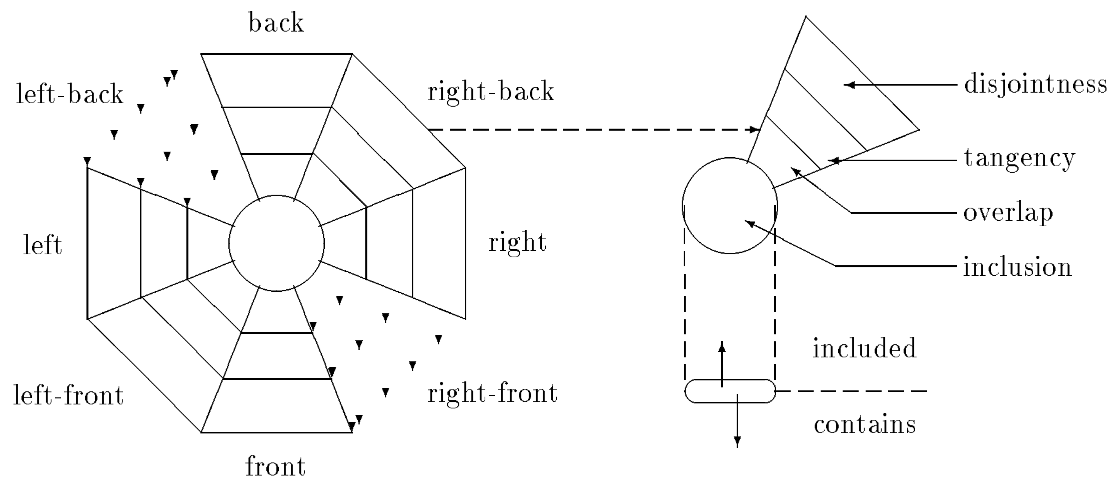


Figure 3: **R**elative **P**rojection and **O**rientation **N**ode

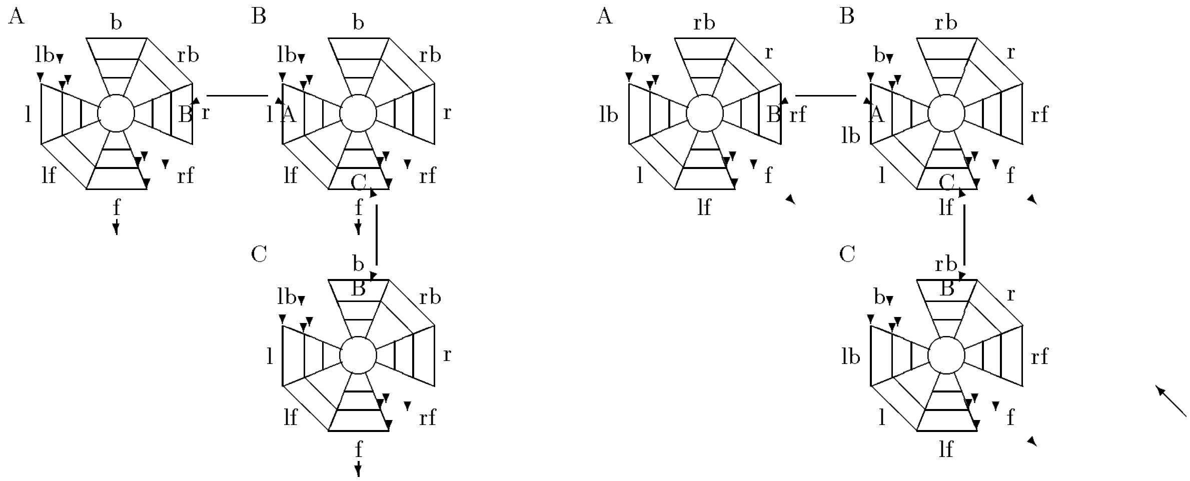
projection and orientation node), which has the same neighborhood structure of the projection and orientation relations. Such a data structure can be *visualized* as an octagon-shaped figure (see Fig. 3). Except for the center, which conveys information on the `inclusion` relation, each of the eight areas corresponding to the eight orientations is subdivided in three slots corresponding to the symmetrical projection relations `overlap`, `tangency` and `disjointness`. A spatial relation between two objects is stored by creating a bidirectional pointer from the projection slot of the given (implicit) orientation of the reference object to the corresponding slot of the "inverse relation" in the primary object. It is important to remember, that rpons represent the neighborhood structure of the relations being used to represent space and *not* the metric of space itself. In particular, even though we tend to *depict* rpons in configurations that resemble the arrangement of the scene being represented (see Figs. 4, 5), every topological deformation thereof would be an equally valid visualization. That is, "internally" only the connections between slots carry information.

3.1 Reasoning with abstract maps

The obvious advantage of these data structures is that, since they have the same structure as the relational domain they represent, operations such as a change in point of view (section 3.1.1) or the composition of relations (section 3.1.2) can be performed efficiently. The simultaneous effect of multiple constraints (section 3.1.3) is an important factor for the effective use of abstract maps in spatial reasoning.

3.1.1 Changing the point of view

A change in point of view, which affects relations with an explicit deictic type of reference frame, can be easily accomplished by "rotating" the labels of the orientation with respect to the intrinsic one. In the example shown in Fig. 4, A is to the left of B (A [d,l] B) as seen from the original point of view (4a). If the point of view changes to be the one depicted in 4b, A will rather be considered to be left-back of B as indicated by the rotated labels. Note that all relations of a given object to others in the scene are updated simultaneously while keeping



(a) Original point of view

(b) Rotation

Figure 4: Change in point of view as simultaneous rotation of labels

projections consistent. So, while B is to the back of C in 4a, their relationship is correctly relabeled in 4b to be "B is right-back of C". Furthermore, this can be done "in parallel" for all objects in a scene.

3.1.2 Computing the composition of spatial relations

An essential operation in every relational representation is the composition of relations: Given the relation between A and B, on the one hand, and between B and C, on the other, we want to know the relation between A and C. For simplicity we will consider projections and orientations separately. We will first look at simple composition tables and discuss then how the regularities found there lead to efficient ways to compute the composition of relations using abstract maps.

Table 1 shows the projection compositions for an assumed constant orientation. For example, given the projection relations A [o,] B and B [o,] C (the orientation, say `back`, is assumed constant for both given relations *and* for the result!), which can correspond to any of the three (qualitatively) different square configurations shown in the right half of Fig. 5b, the resulting relation between A and C is A [{o,t,d},] C, as can be confirmed by looking at the figures (the notation {o,t,d} means that any of `overlap`, `tangency` or `disjointness` might be the case). Except for the relation pairs involving *only* {i,c} (upper left quarter of the table), there are several interesting regularities:

1. The table is "almost symmetric", i.e. given any pair of relations A [x,] B and B [y,] C such that not both x and y are elements of {i,c}, their composition A [t,] C is the same as the composition of A [y,] B and B [x,] C, except that if an i occurs in any of x, y, t it must be substituted by c and conversely c by i. Example: A [o,] B, B [c,] C $\longrightarrow$ A [{c,o,t,d},] C, so A [i,] B, B [o,] C $\longrightarrow$ A [{i,o,t,d},] C. (For this reason and to improve the readability of the table, the vertical order of {i,c} is flipped with respect to the horizontal order.)

2. The relations in the composition are always "conceptual neighbors" (see [4]).

A[]B	B[]C i	c	o	t	d
c	{i,c,o}	{c}	{c,o}	{c,o,t}	{c,o,t,d}
i	{i}	{i,c,o,t,d}	{i,o,t,d}	{t,d}	{d}
o	{i,o}	{c,o,t,d}	{o,t,d}	{t,d}	{d}
t	{i,o,t}	{t,d}	{t,d}	{d}	{d}
d	{i,o,t,d}	{d}	{d}	{d}	{d}

Table 1: Projection compositions for constant orientation

3. The number of relations in the composition tends to decrease from left to right on all rows except the first one.

Using these regularities, the computation of composition can be implemented as propagation of tokens in a rpon-graph. We will omit the details of the propagation algorithm here. The basic idea is to propagate tokens from both rpons for which the composition is being computed, along the pointers to common known relations, and according to well-defined heuristics, as suggested in Fig. 5a.

Table 2 contains the orientation compositions for disjoint objects (i.e. for constant projection = `disjointness`, which is also assumed for the resulting relation between A and C). Figure 5b shows an example: If A is to the `back` of B and B is to the `left` of C, A can be to the `back`, `left-back` or `left` of C as the three square configurations show. The regularities here are straightforward. Given A [,x] B and B [,y] C, the composition A [,t] C contains all orientations that are "in-between" and including x and y (see Fig. 3) on the "shortest path". The following examples illustrate this rule:

1. If $x = y$ then $t = x = y$, since there are no orientations in-between (distance between orientation = 0).

2. For $x =$ `lb` and $y =$ `r` we obtain $t =${`lb,b,rb,r`} (and *not* {`lb,l,lf,f,rf,r`} which are the orientations between x and y on the longer path "the other way around"). Here, $0 <$ distance < 4.

3. If x and y are opposites (e.g. `back/front`, `left/right`, `left-back/right-front`) then $t = \{?\}$, that is *any* of the orientations apply, since both paths between opposites have the same length (distance = 4).

Again, the relations in the composition are always "conceptual neighbors". Furthermore, the composition is symmetric, i.e. t(x, y)=t(y, x).

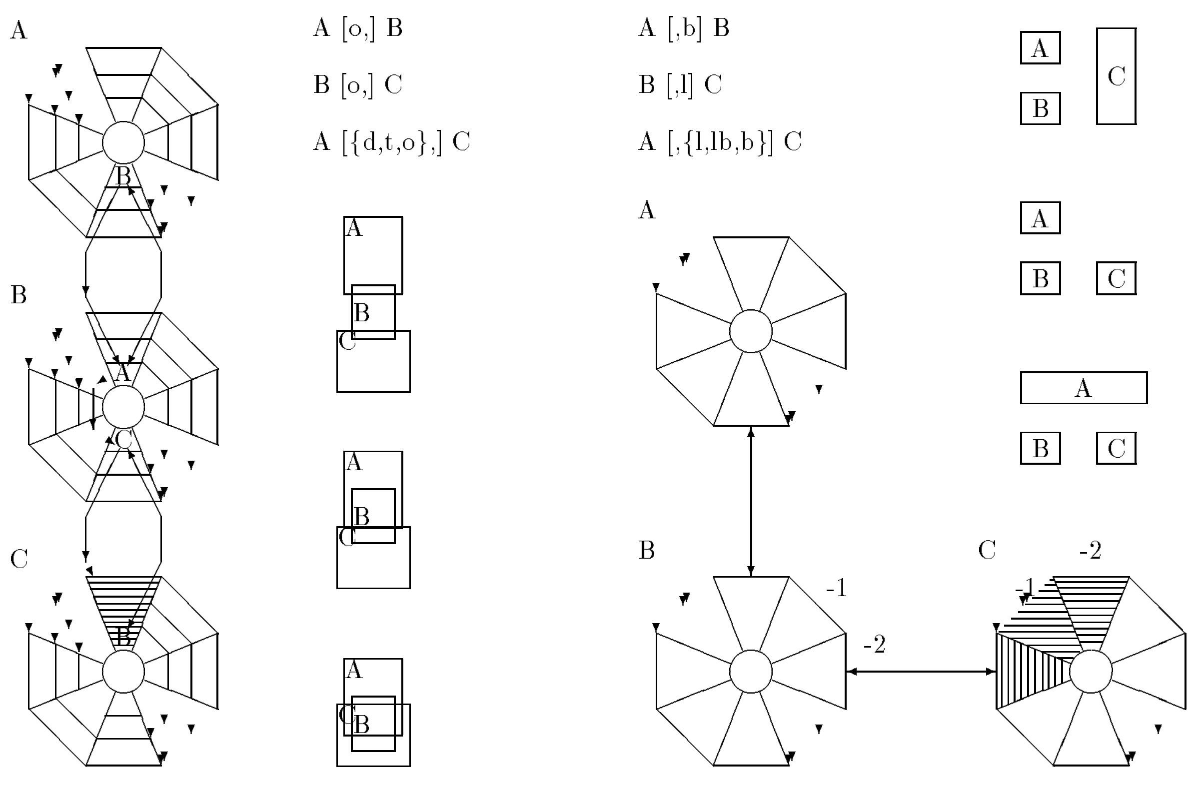
Here, again omitting details, the idea is to find the orientations on the common rpon (B in Fig. 5b) according to the rules given above and then, if desired, propagate the result to the rpon representing the reference object for storage.

3.1.3 The effect of multiple constraints

Given the qualitativeness of the relations used, computing their composition seems to lead to a serious "information degradation" in the sense that the resulting relations are increasingly unspecific (see examples above). Note, however, that compositions are not computed in

A[]B	B[]C							
	b	lb	l	lf	f	rf	r	rb
b	{b}	{b,lb}	{b,lb,l}	{b,lb,l,lf}	{?}	{b,rb,r,rf}	{b,rb,r}	{b,rb}
lb	{lb,b}	{lb}	{lb,l}	{lb,l,lf}	{lb,l,lf,f}	{?}	{lb,b,rb,r}	{lb,b,rb}
l	{l,lb,b}	{l,lb}	{l}	{l,lf}	{l,lf,f}	{l,lf,f,rf}	{?}	{l,lb,b,rb}
lf	{lf,l,lb,b}	{lf,l,lb}	{lf,l}	{lf}	{lf,f}	{lf,f,rf}	{lf,f,rf,r}	{?}
f	{?}	{f,lf,l,lb}	{f,lf,l}	{f,lf}	{f}	{f,rf}	{f,rf,r}	{f,rf,r,rb}
rf	{rf,r,rb,b}	{?}	{rf,f,lf,l}	{rf,f,lf}	{rf,f}	{rf}	{rf,r}	{rf,r,rb}
r	{r,rb,b}	{r,rb,b,lb}	{?}	{r,rf,f,lf}	{r,rf,f}	{r,rf}	{r}	{r,rb}
rb	{rb,b}	{rb,b,lb}	{rb,b,lb,l}	{?}	{rb,r,rf,f}	{rb,r,rf}	{rb,r}	{rb}

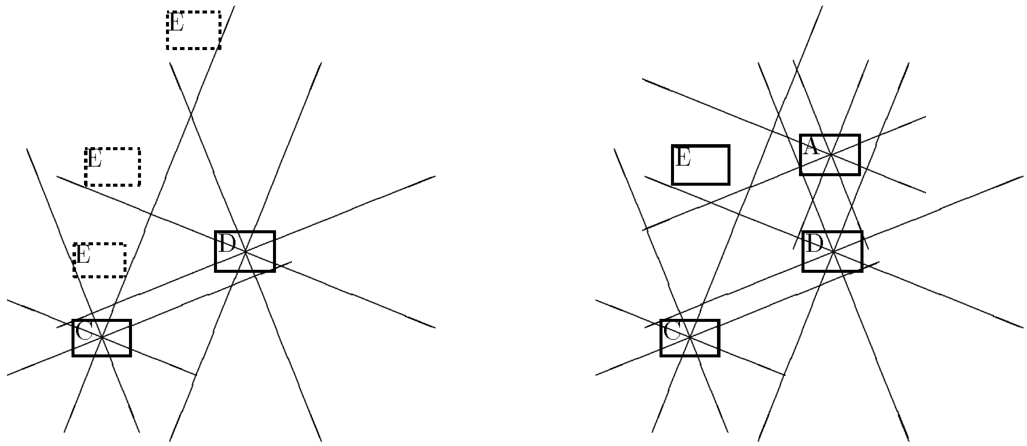
Table 2: Orientation compositions for constant projection



(a) Projections

(b) Orientations

Figure 5: Propagation of tokens in a rpon-graph



(a) E[]D ambiguous given C[]D, C[]E   (b) Adding E[]A further constrains E[]D

Figure 6: The effect of multiple constraints

isolation from other relations possibly constraining the one under consideration. Consider for example the situation depicted in Figure 6a. Given the relation between C and D as C [d,lf] D, the fact that E is to the back of C (E [d,b] C) constrains the relation between E and D only to be "somewhere to the left or to the back" (E [d,{l,lb,b}] D). But if we additionally know that E is to the left of A (E [d,l] A), which in turn is to the back of D (A [d,b] D) —see Fig. 6b— then E is further constrained to be "left-back" with respect to D (E [d,lb] D).

In general the effects of multiple constraints can lead to an arbitrarily precise description of the relative positions of objects. Of course, the general constraint satisfaction problem (CSP) is known to be computationally intractable as the amount of computation needed to establish consistency in a network of relations grows exponentially with the number of relations. The point of introducing abstract maps with their analogical data structures is to reduce this computational overhead through a pre-structuring of the domain, that limits the number of irrelevant cases considered.

3.2 Coarse reasoning and hierarchical organization

Abstract maps also facilitate "coarse" reasoning, i.e. using relations that make less distinctions either to cope with missing information or to simplify reasoning. We could for example only know that the primary object is on the `left` and has `contact` to the reference object. In that case it suffices to distinguish between `left` and `right` in the orientation dimension, and between `contact` and `no-contact` in the projection dimension. In our analogical representation this corresponds to using levels of rpons with less subdivisions (in the visualization of rpons, discs with less subdivisions are superimposed on the original octagon).

As was mentioned in section 2.2, these "levels of granularity" are organized hierarchically in homogenous layers. The projection levels are organized as follows:

$$\left.\begin{array}{l}\left.\begin{array}{l}\text{disjointness}\end{array}\right\}\text{no-contact}\\ \left.\begin{array}{l}\text{tangency}\\ \text{overlap}\\ \text{inclusion}\end{array}\right\}\text{contact}\end{array}\right\}\text{no projection info}$$

Other organizations are possible, for example {o,t,d} = non-containement vs. inclusion or {i,c,o} = overlap vs. no-overlap (solids).

The orientation levels, in order of increasingly fine distinctions, are:

0. no orientation information
1. (a) back, front
   (b) left, right
2. back, front, left, right
3. back, front, left, right, left-back, right-back, left-front, right-front

Note that the meaning of `back`, for example, varies depending on the level of granularity it is used: If the only distinction being made at that level is `back/front`, then it means all locations "behind" a given object; if four orientations are available, it means "all locations behind a given object, which are not considered to be `left` nor `right` of that object"; finally, if eight orientations are available, it means "all locations behind a given object, which are not considered to be `left-back` nor `right-back` of that object". For our purposes it is not necessary to make finer orientation distinctions, even though some existing qualitative navigational systems have 16 or even 32 different orientations. Multiple constraints (see section 3.1.3) allow an arbitrarily precise localization of objects without increasing the number of basic relations.

Spatial reasoning, both changing the point of view and computing the composition of relations, can be done at coarser or finer levels depending on the kind of information available. In particular, if only coarse information is available, the reasoning process is less involved than if more details are known. This is a very intuitive aspect of our approach, which distinguishes it from other frameworks (e.g. using value ranges or confidence intervals) in which less information means more computation.

4 Experimental System

These representational issues form the basis for an experimental system that we envision as follows: The system gets a 2-D scanned layout plan of a building (reduced to the "relevant" surfaces) and an initial position as input[3]. Note that the layout plan is used as a cheap replacement for an actual perceptual system and is not available as such for internal processing (otherwise we would already have a spatial representation!). Based on the layout plan, the first stage of processing generates the partial views — as seen by a hypothetical agent — thus

[3] Since the images consist mostly of straight lines and right angles, we assume the existence of simple recognition mechanisms (classic pattern recognition algorithms or diagrammatic methods such as Funt's retina [5]).

simulating a robot's "perception" while wandering through the building. Those partial views are used next to create a relative, egocentric representation of the perceived room, a process that includes the recognition of spatial boundaries and exit points and the position of other surfaces relative to those boundaries or exits. This *egocentric* representation, in which all information is relative to an agent, is just the first stage in the process of building a "cognitive map" [7]. In order to perform useful spatial reasoning, the agent must be able to transform this representation into non-egocentric forms: an *allocentric* one, in which spatial information is expressed relative to distinguished reference structures, and a *geocentric* one containing abstract topologic and metric relations in a coordinated system of reference frames. A common characteristic of all stages is the hierarchical clustering of relative information. In our setting, the most important kind of information going beyond single rooms is connectivity, which emphasizes that the way (including temporal order) in which we experience our environment strongly influences the representation we build thereof.

Acknowledgements

I would like to thank Wilfried Brauer and Christian Freksa for their comments and support. I also received helpful hints from Scott Freundschuh and Hansgeorg Schlichtmann.

References

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

[2] E. Davis. Solutions to a paradox of perceptual acuity. In *Proceedings of the First Conference on Theoretical Aspects of Knowledge Representation*, Toronto, Oct. 1989.

[3] C. Freksa. Qualitative spatial reasoning. In D. Mark and A. Frank, editors, *Cognitive and Linguistic Aspects of Geographic Space*, NATO Advanced Studies Institute. Kluwer, Dordrecht, 1990.

[4] C. Freksa. Temporal reasoning based on semi-intervals. Technical Report TR-90-016, ICSI, Berkeley, CA, Apr. 1990.

[5] B. V. Funt. Problem solving with diagrammatic representations. *Artificial Intelligence*, 13(3):201–230, 1980.

[6] H.-W. Guesgen. Spatial reasoning based on Allen's temporal logic. Technical report, ICSI, Berkeley, CA, Aug. 1989.

[7] R. Hart and G. Moore. The development of spatial cognition: A review. In R. Downs and D. Stea, editors, *Image and Environment: Cognitive Mapping and Spatial Behaviour.* Aldine, Chicago, 1973.

[8] P. Meseguer. Constraint satisfaction problems: An overview. *AI Communications*, 2(1):3–17, Mar. 1989.

[9] G. Retz-Schmidt. Various views on spatial prepositions. *AI Magazine*, 9(2):95–105, Summer 1988.