

Relations between spacial objects

Simon Fromme

edited on October 30, 2014

Contents

1	Motivation	3
2	Types of geometric relation	3
2.1	Topological relations	3
2.1.1	Prerequisites	3
2.1.2	The 4-Intersection Model	3
2.2	Directional Relations	4
2.2.1	formal definition	4
2.2.2	previous work	5
3	Implementation	11
3.1	Directional relations	11
3.1.1	prerequisites	11

1 Motivation

2 Types of geometric relation

2.1 Topological relations

2.1.1 Prerequisites

To understand the following sections a number of definitions are given that can also be found in most elementary text books on topology and multidimensional analysis.

Let A be a subset of \mathbb{R}^n ($n = 1, 2, 3 \dots$).

Interior

The interior of a set A is denoted by A^o . It is the union of all open subsets of A and hence is a subset of A itself. It is formally defined as

$$A^o := \{x \in \mathbb{R}^n : \exists \epsilon > 0 : U_\epsilon(x) \subseteq A\},$$

where U_ϵ is That means that for every point x in A^o there exists some arbitrary small sphere around x that is still entirely inside of A .

Boundary

The boundary of a set A is denoted by ∂A and is formally defined as

$$\partial A := \{x \in \mathbb{R}^n : \forall \epsilon > 0 : U_\epsilon(x) \cap A \neq \emptyset \wedge U_\epsilon(x) \cap (\mathbb{R}^n \setminus A) \neq \emptyset\}.$$

c The idea of this definition is that an arbitrary small sphere around a point withing ∂A intersects with both the set A and its exterior ($\mathbb{R}^n \setminus A$).

When an object is represented in BRep the boundary ∂A is thus equal to the given $n - 1$ dimensional shell.

Exterior

The exterior of a set A is equivalent to the complement of A denoted by A^c . Thus it holds

$$A^c := \{x \in \mathbb{R}^n : x \notin A\}$$

As entities representing real world objects will naturally be closed.

To formalize topological relations between spacial objects different approaches were taken, one being the 4-Intersection Model which was later extended to a 9-Intersection Model. Since the latter will be used here it will be discussed in the following section.

2.1.2 The 4-Intersection Model

The 4-Intersection Model (4-IM) was described by M. J. Egenhofer and R. Franzosa in 1991 [4] and makes use of point set theory.

Topological relations are handled as topological predicates and thus propositions formed with these predicates can only take the values *true* or *false* (e.g. the proposition “object A *touches* object B” formed with the predicate *touches*). To overcome the limitations of human language the 4-IM defines a set of predicates as signatures of a 2x2 matrix defined as

$$I := \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B \\ \partial A \cap B^\circ & \partial A \cap \partial B \end{pmatrix}.$$

As each matrix element can either be empty (\emptyset) or non-empty ($\neg\emptyset$), $2^4 = 16$ different configuration of the matrix are possible. Eight of those configurations are identified as predicates that occur in real situations

2.2 Directional Relations

Another kind of relation to distinguish are relations that take into considerations the relative directions between objects. Intuitive examples are predicates like *left*, *right*, *above*, *below*, *eastOf*, ...

In the context of city models one might be interested in all buildings west of an exhaust emitting factory, since that is the preferred wind direction in central europe. A geometry based query for bridges might be based on the *above* relationship relative to a specified street and one might want to verify the statement that riverbanks on the northern hemisphere erode more on the right handside relative to the direction of flow by querying for objects on this side of the river.

To decide whether a specific directional relation holds or to find all objects that are related to a given object in terms of a given relation with computational means, it is necessary to clearly define

1. a *set of relations* to be supported by the program and
2. *underlying models* for these relations that the computer then uses to decide whether a relation does hold or not in a given spacial context.

For both questions a number of approaches exist, some of them presented in this work in a first step.

2.2.1 formal definition

When speaking about directional relations it is important to specify a reference frame relative to which these relations are being considered. Following [8] Borrmann and Rank [2] distinguish three different kinds of reference frames, namely

- an *intrinsic* reference frame that is the defined by the inner orientation of the object such as the front side of a house or the right side of a river relative to the flow,

- a *deitic* a reference frame that is dependent on the position of an external observer and
- an *extrinsic* reference frame that is defined by some external reference points.

Taking into account the different meaning of the predicates *left* and *right* in a reference frame relative to an observer or relative to the frontside of a building it becomes clear that a reference frame must be specified when using either of those predicates.

We thus define a directional relation as a tuple (D, RF) where

$$D \subseteq A \times B$$

with A and B being sets of objects and RF a reference frame.

Since directional relationships can be defined unambiguously only for point sized objects it is necessary to work out models to define those relations for spatially extended objects in higher dimensions.

Borrmann and Rank [2] list a number of previous approaches that were developed to address this problem for objects with at least one having dimension 1 or above.

2.2.2 previous work

The most simple yet common approach is to use an objects center of gravity (or some other distinguished point) and to define directional relations between spatially extended objects as 0D-0D-relationships between those points. This is easy and unambiguous, however, it sometimes yields counter intuitive results as seen in Figure 2.1 where the center of gravity of B is *above* the center of gravity of A although a person would probably classify the relationship the other way around. Thus using this simple approach might be easy to implement but sometimes lacks significance in complex scenarios.

In a two-dimensional space Borrmann and Rank [2] distinguish two different approaches to define directional relations between point size objects: the *cone-based* and the *direction-based* model.

In the cone-based model the space around a point gets dissected into a number of equally sized partitions (usually 4 or 8) and the direction of a target point from

In the direction-based model the space is divided into a northern and southern and an eastern and western halfspace relative to some reference point respectively. By projecting a target point on each pair of halfspaces one gets the four possible directions: *northEastOf*, *southEastOf*, *southWestOf* and *northWestOf*.

A different approach to define directional relations between 2D objects is to use their minimum bounding rectangle (MBR) representation.

Definition 1. A minimum bounding rectangle (MBR) is a rectangle, oriented to the x - and y -axes, that bounds a geographic feature or a geographic dataset. It is specified by

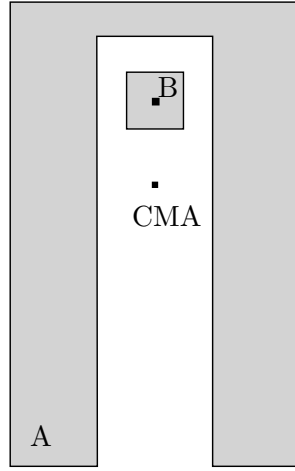


Figure 2.1: counter intuitive result when defining directional relations via the centers of gravity. The relation “*B* is *above A*” would hold in this case although that certainly violates common opinion.

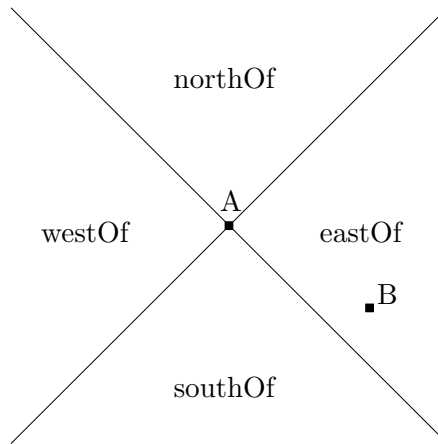


Figure 2.2: According to the cone-based model in 2D-space the point B is located *eastOf* the point A.

two coordinate pairs: x_{\min}, y_{\min} and x_{\max}, y_{\max} .¹

Given the MBR of two objects: $(x_{\min}^{(1)}, y_{\min}^{(1)}, x_{\max}^{(1)}, y_{\max}^{(1)})$ and $(x_{\min}^{(2)}, y_{\min}^{(2)}, x_{\max}^{(2)}, y_{\max}^{(2)})$ one examines the projections onto the x- and y-axis of each bounding box and uses the characterization of 1D interval relations between $[x_{\min}^{(1)}, x_{\max}^{(1)}]$, $[x_{\min}^{(2)}, x_{\max}^{(2)}]$ and $[y_{\min}^{(1)}, y_{\max}^{(1)}]$, $[y_{\min}^{(2)}, y_{\max}^{(2)}]$ respectively.

Having defined n relations between 1D intervals one obtains n^d interval relations in a d -dimensional space that can then be used to define directional relations between the two objects.

Allen [1] classifies 13 relations between (originally temporal) intervals in a 1D space, such as *Interval I_1 is before Interval I_2* or *I_1 overlaps I_2* that are used by other authors for classifications of 1D relationships. Out of those relations Guesgen [6] uses a set of eight (*left*, *attached*, *overlapping*, *inside* and the reversed versions) to describe to and formally reason about spacial relationships in more-dimensional models. This is done by means of d -dimensional relationship tuples, since each dimension is considered to be independent from the others.

An example of this method is pictured in Figure 2.3. If one sticks to the interval relations proposed by Guesgen one would describe the relations of A to B as the tuple (*above*, *overlapping*) where the first tuple entry refers to the x- and the second one to the y-direction. However, the *overlapping* predicate is considered to be rather a topological relation in this thesis.

Papadias et al. [7] use the whole set of Allen’s 13 relations thus obtaining 169 relations in 2D but a distinction between topological and directional relations is not being made.

Borrmann also mentions a model by Goyal [5] that parts the space around a reference object on the basis of its MBR and introduces a matrix that records in which segments a target object falls. This is easily extendable to three dimensions, however, Goyal does not assign names to the different matrix configurations making it unsuitable for a spacial query language.

Coming to the conclusion that little work has been put into developing models on directions in 3D space Borrmann [2] proposes two new models: the *projection based* and the *half space model*.

the projection-based directional model Both models are suited for objects of any dimensions. Spacial relations are examined separately for each axis, thus no combined relations such as *northEastOf* is being computed. In three dimensions six distinct relations are part of the model:

The idea is to extrude the reference object in the direction of one axis (e.g. the x-axis for the *northOf* relation) and to check for intersections with the extrusion. Borrmann distinguishes a relaxed and a strict version of these relations. For the “relaxed” version

¹definition according to http://en.mimi.hu/gis/minimum_bounding_rectangle.html

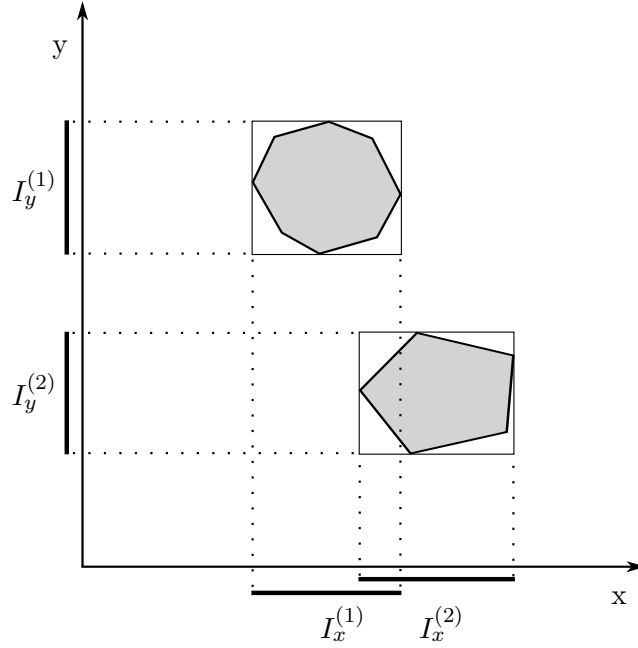


Figure 2.3: Using minimum bounding rectangles to define directional relations in 2D. The MBRs are projected on both the x- and y-axis. Subsequently one-dimensional relations between the projection intervals $I_x^{(1)}$, $I_x^{(2)}$ and $I_y^{(1)}$, $I_y^{(2)}$ are evaluated independently and are then combined in a tuple.

axis relations

x	<i>northOf, southOf</i>
y	<i>eastOf, westOf</i>
z	<i>above, below</i>

to hold it is sufficient for a target object to intersect with the extrusion of the reference object in the respective direction. For the “strict” version it is necessary for the target object to lay fully inside of the extrusion on the respective side of the axis.

This can be formalized in the following way.

relaxed

$$\begin{aligned}
\text{eastOf_proj_relaxed}(A, B) &:\Leftrightarrow \exists a, b : a_y = b_y \wedge a_z = b_z \wedge a_x < b_x, \\
\text{westOf_proj_relaxed}(A, B) &:\Leftrightarrow \exists a, b : a_y = b_y \wedge a_z = b_z \wedge a_x > b_x, \\
\text{northOf_proj_relaxed}(A, B) &:\Leftrightarrow \exists a, b : a_x = b_x \wedge a_z = b_z \wedge a_y < b_y, \\
\text{southOf_proj_relaxed}(A, B) &:\Leftrightarrow \exists a, b : a_x = b_x \wedge a_z = b_z \wedge a_y > b_y, \\
\text{above_proj_relaxed}(A, B) &:\Leftrightarrow \exists a, b : a_x = b_x \wedge a_y = b_y \wedge a_z < b_z, \\
\text{below_proj_relaxed}(A, B) &:\Leftrightarrow \exists a, b : a_x = b_x \wedge a_y = b_y \wedge a_z > b_z.
\end{aligned}$$

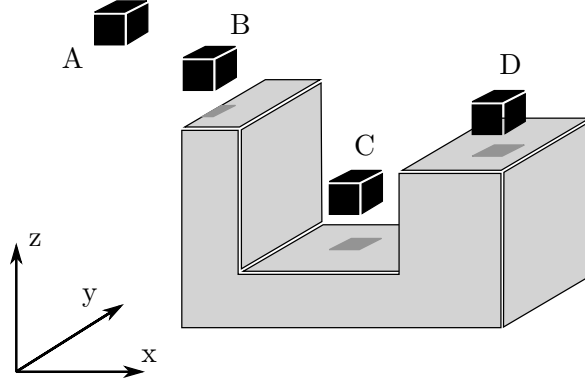


Figure 2.4: direction-based model: According to the relaxed version of the *above* relation the cubes *B*, *C* and *D* would be considered to be *above* the gray object. According to the strict version only the cubes *B* and *C* would be.

strict²

$$\begin{aligned}
\text{eastOf_proj_strict}(A, B) &:\Leftrightarrow \forall b : (\exists a : a_y = b_y \wedge a_z = b_z \wedge a_x < b_x) \wedge \\
&\quad (\nexists a : a_y = b_y \wedge a_z = b_z \wedge a_x \geq b_x), \\
\text{westOf_proj_strict}(A, B) &:\Leftrightarrow \forall b : (\exists a : a_y = b_y \wedge a_z = b_z \wedge a_x > b_x) \wedge \\
&\quad (\nexists a : a_y = b_y \wedge a_z = b_z \wedge a_x \leq b_x), \\
\text{northOf_proj_strict}(A, B) &:\Leftrightarrow \forall b : (\exists a : a_x = b_x \wedge a_z = b_z \wedge a_y < b_y) \wedge \\
&\quad (\nexists a : a_x = b_x \wedge a_z = b_z \wedge a_y \geq b_y), \\
\text{southOf_proj_strict}(A, B) &:\Leftrightarrow \forall b : (\exists a : a_x = b_x \wedge a_z = b_z \wedge a_y > b_y) \wedge \\
&\quad (\nexists a : a_x = b_x \wedge a_z = b_z \wedge a_y \leq b_y), \\
\text{above_proj_strict}(A, B) &:\Leftrightarrow \forall b : (\exists a : a_x = b_x \wedge a_y = b_y \wedge a_z < b_z) \wedge \\
&\quad (\nexists a : a_x = b_x \wedge a_y = b_y \wedge a_z \geq b_z), \\
\text{below_proj_strict}(A, B) &:\Leftrightarrow \forall b : (\exists a : a_x = b_x \wedge a_y = b_y \wedge a_z > b_z) \wedge \\
&\quad (\nexists a : a_x = b_x \wedge a_y = b_y \wedge a_z \leq b_z).
\end{aligned}$$

the half-space based model In the half-space based model one does not consider the extrusion of an object but the entire half space in one direction when checking for intersection of the target with the reference object. If the target object lays completely inside the half-space (e.g. above an object) one considers the “strict” version of the relation to hold.

In the “relaxed” version one does also allow for intersections of the target with the

²It has to be noted that Borrmann made a mistake in his original publication [2] where at some points he mistakenly exchanged *a* for *b* in the definitions of the “strict” relations for the projection-based directional model. These rather apparent mistakes were corrected here.

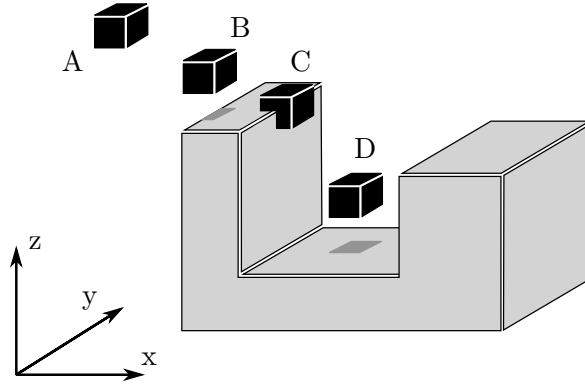


Figure 2.5: half-space based model: According to the relaxed version of the *above* relation the cubes *A*, *B* and *C* would be considered to be *above* the gray object. According to the strict version only the cubes *A* and *B* would be. In contradiction to the direction based model *C* would not be considered to be *above* (relaxed/strict) the gray object

reference object requiring only part of the object to lay further in the direction of the specific relation.

Example: For the relation “*B above_hs_relaxed A*” to hold it is necessary that for every point of *A* there is at least one point of *B* with a greater *z*-component.

Borrmann consecutively arrives at the following formal definitions:

relaxed-version

$$\begin{aligned}
 \text{eastOf_hs_relaxed}(A, B) &: \Leftrightarrow \exists b : \forall a : a_x < b_x, \\
 \text{westOf_hs_relaxed}(A, B) &: \Leftrightarrow \exists b : \forall a : a_x > b_x, \\
 \text{northOf_hs_relaxed}(A, B) &: \Leftrightarrow \exists b : \forall a : a_y < b_y, \\
 \text{southOf_hs_relaxed}(A, B) &: \Leftrightarrow \exists b : \forall a : a_y > b_y, \\
 \text{above_hs_relaxed}(A, B) &: \Leftrightarrow \exists b : \forall a : a_z < b_z, \\
 \text{below_hs_relaxed}(A, B) &: \Leftrightarrow \exists b : \forall a : a_z > b_z.
 \end{aligned}$$

and strict-version

$$\begin{aligned}
 \text{eastOf_hs_strict}(A, B) &: \Leftrightarrow \forall a, b : a_x < b_x, \\
 \text{westOf_hs_strict}(A, B) &: \Leftrightarrow \forall a, b : a_x > b_x, \\
 \text{northOf_hs_strict}(A, B) &: \Leftrightarrow \forall a, b : a_y < b_y, \\
 \text{southOf_hs_strict}(A, B) &: \Leftrightarrow \forall a, b : a_y > b_y, \\
 \text{above_hs_strict}(A, B) &: \Leftrightarrow \forall a, b : a_z < b_z, \\
 \text{below_hs_strict}(A, B) &: \Leftrightarrow \forall a, b : a_z > b_z.
 \end{aligned}$$

3 Implementation

3.1 Directional relations

3.1.1 prerequisites

To determine a set of directional relations to query the specific city model in this work and to decide about which underlying models to use for the implementation of each relation some of the specific characteristics of the city model Karlsruhe and its potential use, that do influence that choice, are being recapped.

1. The city model used in this work is available in a fairly low level of detail (LoD2). This resolution does not allow for the modeling of separate floors or basements which makes it unnecessary to implement directional relations for the z-axis. However, if floors and basement are modeled as separate entities (e.g. in city models of higher resolution) such a query function would be desirable.

Another consequence of this low level of detail is that considering the extrusion of an object when defining a particular directional relation is not very meaningful. This would have applications when querying for columns and walls *above* a certain ceiling or for all building parts *above* the floor plate.

Since a house is modeled in one or two parts only in the given city model the only reasonable queries the author can think of that would use this kind of relations would be queries for the roof of a building (*above_proj_strict/relaxed* house_base_body) or the neighboring buildings in a specific direction (e.g. *eastOf_proj_relaxed* building). These queries seem not important enough to implement them in the API.

2. As a city model will mostly be used in a way that is independent of the way one observes it (in contradiction to e.g. a virtual reality world for the purpose of direct user interaction), it seems appropriate not to include relations that require a deitic reference frame as defined in section 2.2. Thus relations of the kind *leftOf* and *rightOf* that depend upon the position and the view angle of the observer will be omitted.

Hence in the actual implementation the set of the following relations

- *eastOf_hs_relaxed*,
- *westOf_hs_relaxed*,
- *northOf_hs_relaxed*,
- *southOf_hs_relaxed*,

as previously described in section 2.2.2 are being used. In a more detailed model it would be very useful to extend this set but due to the above reasons this does not seem adequate for the special case.

Given the input data in a BRep representation (Borrmann [2] discusses the case for spatial objects in Octree representation) the implementation is fairly easy. If V_{O_1} and V_{O_2} denote the set the set of vertices in 3D space of the spatial objects O_1 and O_2 the following holds.

$$\begin{aligned} \text{eastOf_hs_relaxed}(O_1, O_2) &= \begin{cases} \text{true}, & \max\{v_x : v \in V_{O_2}\} > \max\{v_x : v \in V_{O_1}\} \\ \text{false}, & \text{otherwise} \end{cases} \\ \text{westOf_hs_relaxed}(O_1, O_2) &= \begin{cases} \text{true}, & \min\{v_x : v \in V_{O_2}\} < \min\{v_x : v \in V_{O_1}\} \\ \text{false}, & \text{otherwise} \end{cases} \\ \text{northOf_hs_relaxed}(O_1, O_2) &= \begin{cases} \text{true}, & \max\{v_y : v \in V_{O_2}\} > \max\{v_y : v \in V_{O_1}\} \\ \text{false}, & \text{otherwise} \end{cases} \\ \text{southOf_hs_relaxed}(O_1, O_2) &= \begin{cases} \text{true}, & \min\{v_y : v \in V_{O_2}\} < \min\{v_y : v \in V_{O_1}\} \\ \text{false}, & \text{otherwise} \end{cases} \end{aligned}$$

For optimization purposes the minimum and maximum of the x-, y- and z-component of the vertices of a geometry are saved after parsing to avoid repeated sorting for multiple directional queries of a kind.

References

- [1] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, November 1983.
- [2] A. Borrmann and E. Rank. Specification and implementation of directional operators in a 3d spatial query language for building information models. *Advanced Engineering Informatics*, 23(1):32–44, 2009.
- [3] Simon Daum and André Borrmann. Processing of topological {BIM} queries using boundary representation based methods. *Advanced Engineering Informatics*, (0):–, 2014.
- [4] M. J. Egenhofer and R. Franzosa. Point-set topological spatial relations. *International Journal of Geographic Information Systems*, (5(2)):161–174, 1991.
- [5] Roop Goyal, Roop K Goyal, and K Goyal. Similarity assessment for cardinal directions between extended spatial objects. 2000.
- [6] Hans Werner Guesgen. *Spatial reasoning based on Allen’s temporal logic*. International Computer Science Institute Berkeley, CA, 1989.
- [7] Dimitris Papadias, Timos Sellis, Yannis Theodoridis, and Max J Egenhofer. *Topo-*

logical relations in the world of minimum bounding rectangles: a study with R-trees, volume 24. ACM, 1995.

- [8] Gudula Retz-Schmidt. Various views on spatial prepositions. *AI Mag.*, 9(2):95–105, July 1988.