Human Reasoning About Common Spatial Relations in Daily Conversation

Lucas Freitas[1], **Claudio E. C. Campelo**[1]

[1]Federal University of Campina Grande (UFCG)
Departamento de Sistemas e Computação
Campina Grande – PB – Brazil

Abstract. *When people talk about place, there are a few key parts of the produced sentences that are particularly useful for GIS tasks, spatial relations being the primary concern in this paper. Used to describe the position of an object in relation to another, spatial relations play a crucial role in the proper understanding of spatial communication. In this work, a set of algorithms for generating polygons that match spatial relations used in daily communication are proposed. An experiment is then conducted, not only evaluating the proposed algorithms but also shedding some light on the way people understand spatial relations.*

1. Introduction

Advances in the field of automatic interpretation of natural language, have the potential of enabling the development of many applications. Spatial language in particular, contain information that is useful for solving many real world problems such as aiding in the operation of driverless vehicles.

One piece of crucial information often used by people when describing the location of an object in an environment are spatial relations. They describe how an object is located in a given scenario, in relation to another reference object. In the sentence *The place is near the school, right next to that big old church*, **near** and **next to** are the parts that play the role of spatial relations, as they describe the location of the place in relation to the school and the big old church respectively.

Having systems that understand the spatial relations that people really make use is of utmost importance for the development of geographic aware applications. To address this issue, this paper proposes a set of algorithms that project in a 2D map, some of the most frequently used spatial relations. They take as input the name of a reference object and produce the geometry that best describes the area described by the relation and the object, in an urban environment.

The algorithms are then evaluated, in a data analysis that compares their output with drawings representing spatial relations made by humans in an experiment that also tries to answer some questions on people's mental representations of spatial relations.

The dataset of geometries drawn by participants is made publicly available, hoping that it can contribute to further research in the area.

2. Related Work

Spatial Relations have been studied for a very long time and have been classified into topological, projective and metric [Bucher et al. 2012].

While metric relations are important, humans seem to have a qualitative reasoning of space [Cohn and Renz 2008]. Topological relations describe the positioning of objects in terms of the intersections of their interiors, boundaries and exteriors. They have been extensively studied [Egenhofer and Franzosa 1991, Mark and Egenhofer 1994, Clementini et al. 1994], and in fact are even supported by spatial query languages. Most of the relations explored in this work fall in the directional category.

Directional relations are a common subcategory of projective relations that include daily expressions used in natural language such as “right of”, “in front of” and “between”. Directional relations are ambiguous and need additional contextual information such as Frames of Reference [Clementini 2013]. In his work, Clementini defines a taxonomy of frames of reference, mapping relations to the 5-Intersection model [Clementini and Billen 2006] of projective relations, this gives geometric definitions needed to compute relations. [Clementini and Bellizzi 2019] builds on top of this mapping and present a Java application framework that implements the directional relations for some of the frames of reference.

Despite the fact that most of the relations explored in this work are directional, this is not intended to be a comprehensive list of all relations in this class. In fact, the main focus of the study is to explore a subset of relations, that seem to be among the ones that are most often used when people describe places. This subset includes common expressions that (to the best of our knowledge) have not been explored before such as "Next to", “Near” and “At Street”. The algorithms proposed in this work, try to produce acceptance regions for the relations solely by computing intersections between polygon buffers and streets. Finally, one of our aims is trying to get a grasp of the representation of spatial relations in the minds of humans.

3. Materials and Methods

In order to better understand the way people imagine spatial relations, an experiment was conducted. Through the usage of a web app, participants were told to picture the following scenario:

“Imagine that a friend will give you a ride and tell you over the phone where the car stopped and is waiting. Based on the description he gave you, we ask you to draw on the map the area where you think the car might be.”

The web app then shows up a map with a highlighted landmark and a sentence that describes the location of the car. Figure 1 shows the screen that the participants see when they are supposed to start drawing. The sentence in (2) means *Your ride awaits you at: AT Café Poético’s STREET, NEXT TO Bar do Cuscuz.* The blue capitalized words represent spatial relations while the black ones represent spatial landmarks. Participants drew the regions by clicking on the map and creating points and lines. Each person had to draw five relations (Table 1) for each of the four landmarks.

A street might extend itself for kilometers, this was a concern when designing the experiment, for participants could get tired of drawing really large areas. For this reason, relations “At Street” and “Next to” were combined, this way, participants were supposed to draw a polygon in only a smaller portion of the street.

The drawings were then stored in the GeoJSON format and a CSV of the data is

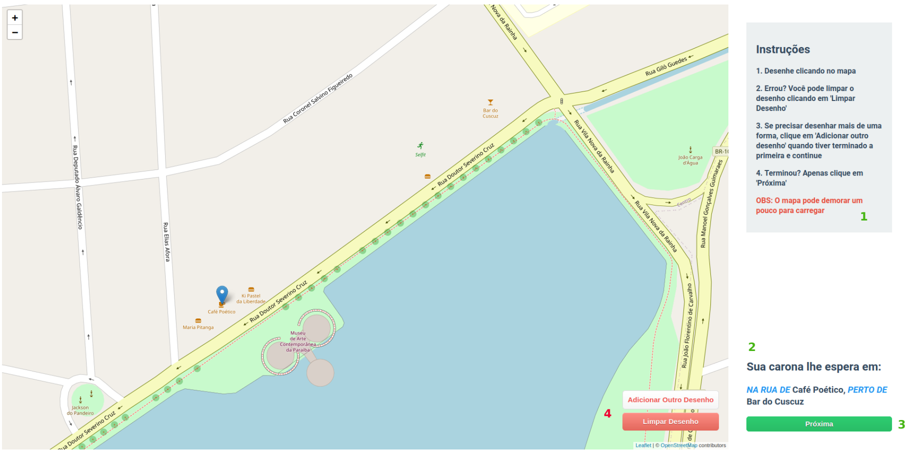

Figure 1. 1 - Drawing Instructions. 2 - Sentence Describing Location. 3 - Next Button. 4 - Drawing Controls

Table 1. Spatial Relations Names

Brazilian Portuguese Relation Name	English Translation
NA FRENTE DE	In front of
NA RUA - PERTO DE	At Street - Next To
ENTRE	Between
AO LADO DE	Next to
À DIREITA DE	Right of

available at GitHub [1].

4. Directional Relations Algorithms

With access to geographic data in a city level, the chosen spatial relations could be implemented with simple algorithms. Some level of uncertainty is taken into account when we don't have enough information about the spatial features (e.g. for a building located at a street corner, if the object's address is not available, it is impossible to know, to which street the facade is turned to, therefore, the disambiguation of the relation "In front of" becomes impractical).

Data is in the heart of geographically aware systems. However, it is not always available in the best format. In a dataset such as OpenStreetMaps many objects are available as polygons representing real world landmarks, but some are only available as latitude/longitude points. As all of the procedures receive as input the geometry corresponding to the reference of the relation, they provide ways of generating the relations for polygons in point and polygon formats.

All functions return a new polygon, representing the **acceptance region**, that describes the relation with respect to the input landmark.

4.1. In Front of

"The car is in front of the university". In this sentence, the "In Front of" relation is used. In most cases if we actually know the street address of the university this relation is

[1] https://github.com/jslucassf/geoinfo-spatial-relations

pretty straightforward, however, for big buildings that face multiple streets this definition becomes faulty, as people can refer to a different street as if being "in front of" as long as some part of the building still faces it.

Algorithm 1 In Front of

```
1:  function FRONTOF(landmark geometry)
2:      if landmark is of type point then
3:          Compute a buffer around landmark
4:          The intersection between the buffer and all streets that intersect it
5:          If this intersection contains more than one street, make a line that goes from landmark to each street
6:          return all pieces of streets which have not been crossed by a line
7:      else
8:          for each side in the landmark polygon do
9:              Compute a one-sided buffer in the line representing the side of the polygon
10:             streetFront = the union of all streets that intersect the one-sided buffer
11:             Compute a buffer between the landmark and streetFront
12:             if There are no other objects inside this buffer then
13:                 streetFront = Union of streetFront and finalFront
14:             end if
15:         end for
16:         return finalFront
17:     end if
18: end function
```

For points, the algorithm returns the pieces of streets that are closer to the landmark. For landmarks that actually contain polygons, we compute a buffer for each of the polygon sides (line 7) check if there are streets that intersect this buffer and make sure that the intersections are not in front of other objects (lines 9 to 10).

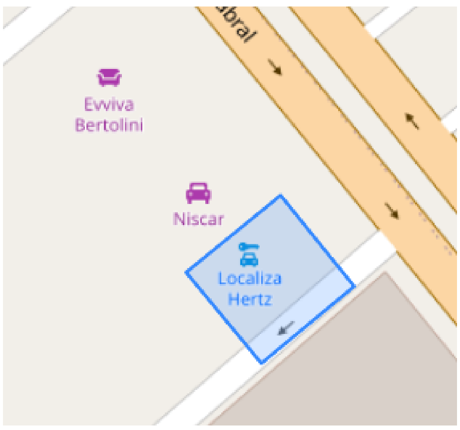

Figure 2. The buffer on one of *Niscar's* sides includes a street, however, the presence of *Localiza Hertz* indicates that the street is in front of the latter landmark not the former.

Algorithm 2 At Street

```
1: function ATSTREET(landmark geometry)
2:     Compute the front of the landmark
3:     for Each street that intersects the landmark's front do
4:         if Area of the intersection between the street and the front is bigger then some threshold then
5:             finalStreet = Union of finalStreet and intersection
6:         end if
7:     end for
8:     return finalStreet
9: end function
```

4.2. At Street

"The car is at the university's street". When the university is a well known landmark in the area, the street in which it is located becomes a common landmark. This relation produces an acceptance region that includes the whole extension of the street.

For this relation, it is important to filter out the parts of streets whose areas are small enough (line 4). As sometimes the crossing between streets is included in the front area (mostly for point landmarks) but only one of the streets is really in front of the landmark.

Once again, If the data includes addresses of the objects, the projection of the acceptance region could be thought as straightforward, however we also consider that people might think that streets that are not the official address of some building but that are adjacent to one of its sides could also be seen as "the building's street".

4.3. Near

The Near relation is quite simple and is the same for points and polygons. A buffer around the landmark represents the region that is *near* it.

Algorithm 3 Near

```
function NEAR(landmark geometry, distance float)
    return a buffer with a distance-sized radius around landmark
end function
```

The distance parameter should be tuned, and probably varies depending on the context (e.g. people who live in smaller cities might consider as near, a distance that is different from people that live in bigger cities).

Future experiments could try to quantify this value, by averaging the distances that people consider as being "near" some landmark.

4.4. Between

Between is the only ternary relation in this list. It defines the position of one object, with respect to two others as in *"The car is between the university and the bookstore"*. For this reason, the function takes in two geometry parameters.

Algorithm 4 Between

1: **function** BETWEEN(landmark1 geometry, landmark2 geometry)
2: Draw a line between the a point in the surface of each of the two geometries
3: Extract two points from the line, that are equidistant from the line center and at a distance *d*rom the line ends
4: Draw a new line between the two points
5: **return** a *d*-sized buffer around the new line
6: **end function**

If one returns a buffer around the line defined by the points in each geometry, the resulting region will include area that is actually outside the desired relation. To fix this issue, the procedure finds points along the line that are located at the same distance from the closest end (line 3, Figure 3), in PostGIS, this could be done using ST_LineInterpolatePoint.

Figure 3. Points that are equidistant.

4.5. Next To

In the "Next to" relation, regions that are immediately next to the landmark but not in front of it are included.

This relation is basically a smaller "Near" minus the front. For this reason a smaller buffer is computed (line 4) and intersected with the street of the landmark. For each street in this intersection a line is drawn starting from the landmark (Figure 4), if this line crosses the difference between intersections and the street (Figure 5), this means that the intersection includes a street that is closer to the landmark, so the one that is farther away is removed (lines 6 to 10). After this, the difference between the resulting area and the front is returned.

4.6. Right of (Left of)

When someone say "The car is to the right of the university", the message can be interpreted in different ways according to one's spatial mental reasoning. If directions are defined based on the observer's point of view, the region defined by the relation may assume a completely opposite position than if directions were defined by the position of the

Algorithm 5 Next To

```
function NEXTTO(landmark geometry)
    Compute landmark front
    Compute landmark street
    Compute a buffer around landmark
    nextInt = the intersection between the buffer and landmark street
    for Each partOfStreet that intersects nextInt do
        Draw a line from landmark to partOfStreet
        if Line do not cross the difference between nextInt and partOfStreet then nextFinal = union between nextFinal and partOfStreet
        end if
    end for
    if Landmark is of type point then
        for Each partOfStreet that intersects nextFinal do
            Get the point in partOfStreet that is closest to landmark
            nextFinal = difference between nextFinal and a small buffer around the closest point
        end for
        return nextFinal
    end if
    return Difference between nextFinal and the landmark front
end function
```

Figure 4. A line starting in the landmark and ending at a street included in the "Near" relation.

Figure 5. The line crosses the difference between the "Near" relation and the street, hence the street is excluded from the result.

reference object itself. Contextual information such as a clear frame of reference is then needed to disambiguate the sentence.

According to [Retz-Schmidt 1988] frames of reference can be classified as **intrinsic** where an intrinsic property of the reference object (such as its front) defines orientation, **extrinsic** where orientation is defined by another external landmark and **deictic** that defines orientation based on an observer's point of view.

The "Right of" function receives a string of text as second argument, representing the type of frame of reference that should be used to define the relation. It can assume the values of two of the three aforementioned types, intrinsic (defines right, based on the landmark front) and deictic (defines right based on the point of view of an observer positioned in front of landmark and looking towards it).

Algorithm 6 Right of

```
1: function RIGHTOF(landmark1 geometry, for text)
2:     if for == "intrinsic" then bufferSide = "left"
3:     else if for == "deictic" then bufferSide = "right"
4:     end if
5:     Compute landmark front
6:     Compute landmark Next To relation
7:     for Each partOfStreet that intersects landmark front do
8:         Draw a line from landmark to the centroid of partOfStreet
9:         Create a one-sided buffer that grows in the direction of the bufferSide variable
10:        for Each polygon in landmark Next To relation do
11:            if polygon intersects buffer then
12:                finalRight = union between finalRight and polygon
13:            end if
14:        end for
15:    end for
16:    return finalRight
17: end function
```

The relation "Next to" includes regions that are positioned immediately to the left and to the right of the landmark. This function builds a buffer (lines 8 and 9) used to test which side of these regions satisfy the relation (lines 11 to 13).

The algorithm for the "Left of" relation is almost the same as this one, the only difference is in lines 2 and 3, where the "left" and "right" values are swapped. For brevity reasons, it is not included here.

5. Frames of Reference and People's Interpretation

As already discussed, correctly interpreting the relation "right of" can be challenging. One of the main interests during the experiments, was to try to understand the spatial reasoning behind people's decisions when interpreting this relation.

The question of interest here is: Which type of frame of reference best describes the reasoning behind people's decisions, when faced with the task of locating a region said to be at the right side of some reference object.

The sentenced used to describe the relation was: "Sua carona está à direita de <landmark>" which translates to "Your ride is to the right of <landmark>". It is important to mention that the sentence choice might have an impact on the results of the analysis.

The terminology in the analysis assumes a **deictic** FoR, therefore, here, when a drawing is said to be to the right of the landmark, this means that it is located to the right side in the perspective of an observer that looks towards the landmark (Figure 6).

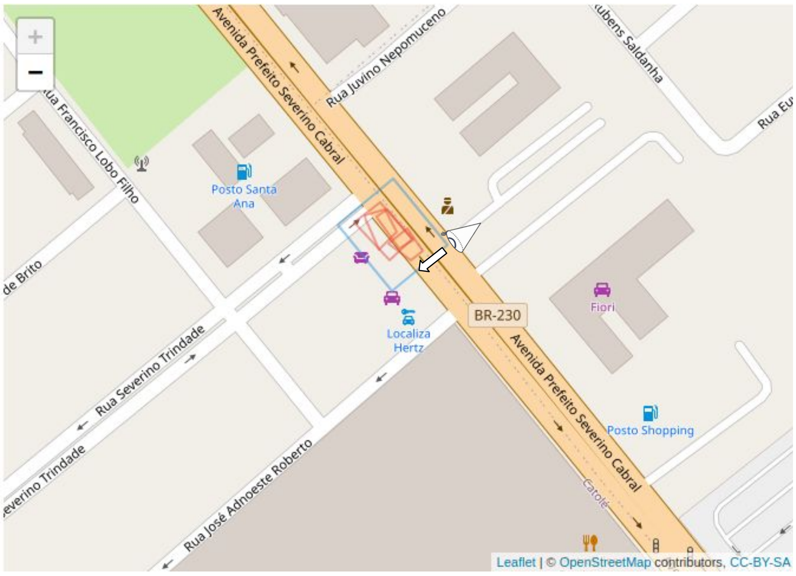

Figure 6. Drawings to the right of the observer's point of view

The drawings were classified using buffers for each of the sides (the blue polygon on Figure 6). Results are shown in Figure 7. In most of the cases, drawings intersected only with the left buffer, this seems to indicate that participants consider the intrinsic properties of the landmark when interpreting the relation, i.e. the reasoning behind the **intrinsic** frame of reference. However, many participants positioned their regions in the right side, in fact for one of the landmarks, this was the case more often than the left side. This raises a few questions. Given the high number of drawings on the right side for landmarks *Café Poético* and *Maria Pitanga*, might this result be affected by the geographical direction towards which the landmarks are facing (these landmarks are neighbors and face the same direction)? What happens if we factor in people's problems in telling left from right or even map reading difficulties? A future work, would be to repeat this experiment, but showing participant's the actual images of landmarks facades.

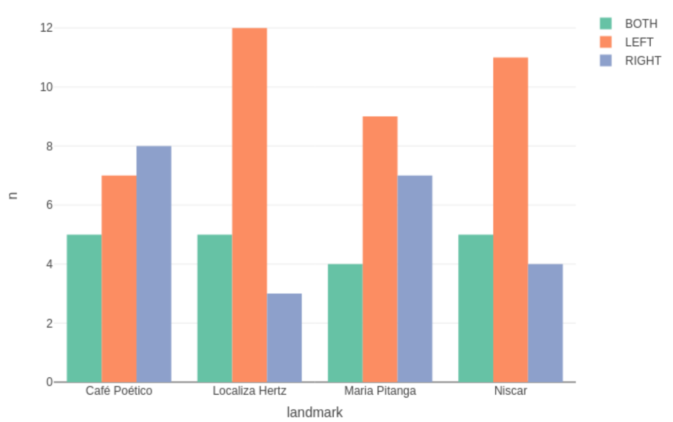

Figure 7. Location of drawings in the "Right of" relation

Another interesting finding is that many participants chose to draw on both sides of the landmark. In another future work, the reason behind this choice could be investigated by identifying and questioning the individuals.

6. Evaluating the Performance of the Proposed Algorithms

In order to evaluate the performance of the algorithms, they were implemented using PostGIS and executed for each of the four landmarks used in the experiments. The regions produced by them was then compared against the collected drawings. One issue with the drawings was that although the experiment defined that participants should imagine the location of a car, some drawings do not intersect streets at all. This might be due to not so clear instructions and a future experiment can try to address this issue. However as the algorithms function in the scope of streets (the regions produced by them are mostly located on the streets), the drawings that do not intersect streets at all were not considered.

6.1. Intersection of Areas

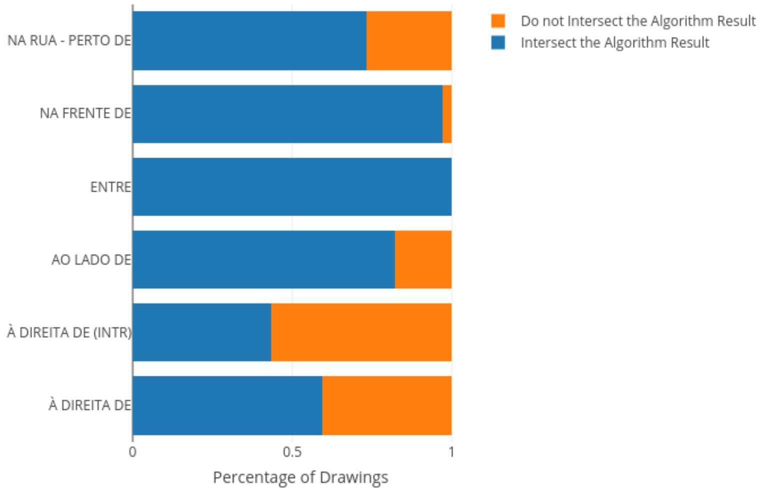

Figure 8. Percentage of drawings that intersect the region produced by the proposed algorithms

The chart presented in Figure 8 shows that for almost all relations, the algorithms produce regions that intersect the majority of drawings made by participants of the experiment. The relation “Right of” got the lowest results however this could be explained by the uncertainty in this relation, explored in Section 5.

6.2. Jaccard’s Similarity Coefficient

A common metric used to access the similarity between sets is Jaccard’s Similarity Coefficient. It expresses how similar two sets are in a scale of 0 to 1 and is computed by the Equation 1.

$$Jaccard(A, B) = A \cap B / A \cup B \tag{1}$$

This metric was used to evaluate how similar are the geometries produced by the algorithms and the drawings made by the participants.

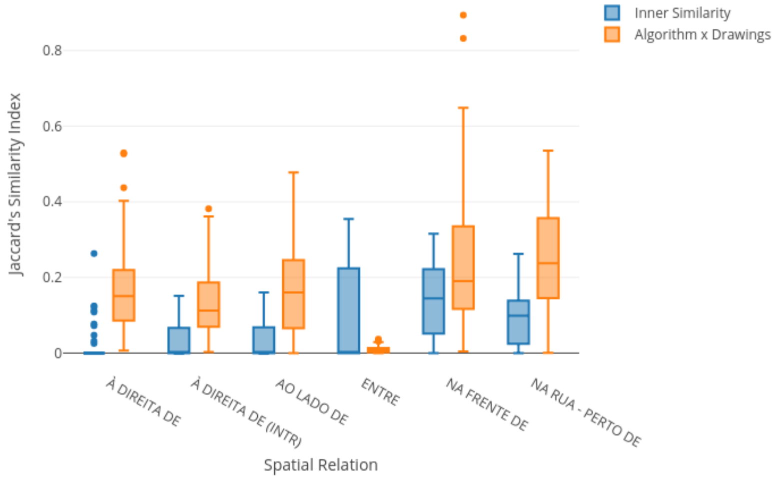

Figure 9. Jaccard's Similarity Coefficient between geometries

In order to assess the complexity of the task, a value to show how similar the drawings made by the participant's are with each other was also computed, here it was called the **inner jaccard**. For each drawing, the Jaccard's Similarity Index with all other drawings in the same category (landmark and spatial relation) is computed, the median result is the inner jaccard and it represents how similar is this drawing to all the others. Figure 9 displays the results of the analysis.

As can be seen in the low inner jaccard values, the drawings themselves are not very similar. This might indicate that people have different understandings of spatial relations. Considering this, the proposed algorithms had modest results comparing to such a diverse set of region polygons, the exception being the "Between" relation, this can be explained by the fact that the region produced by the algorithm is large, for this reason it also intersects all the drawings in the same relation. These results, when coupled with the high intersection percentages shown in Section 6.1, suggest that these algorithms are a good starting point for the implementation of some of the spatial relations that are most used in people's daily language.

7. Conclusion

This paper proposes algorithms to implement some of the spatial relations that are most used by people in natural conversations. It includes an experiment that briefly studies the spatial reasoning behind people's interpretation of these relations. An analysis on the collected data shows that these algorithms hold promissory results in representing human understanding of spatial relations. Another contribution to the field is making available a dataset of more than 400 drawings of spatial relations, allowing further studies on people's interpretation of spatial relations.

References

Bucher, B., Falquet, G., Clementini, E., and Sester, M. (2012). Towards a typology of spatial relations and properties for urban applications. *Usage, Usability, and Utility of 3D City Models–European COST Action TU0801*, page 02010.

Clementini, E. (2013). Directional relations and frames of reference. *GeoInformatica*, 17(2):235–255.

Clementini, E. and Bellizzi, G. (2019). A geospatial application framework for directional relations. *ISPRS International Journal of Geo-Information*, 8(1):33.

Clementini, E. and Billen, R. (2006). Modeling and computing ternary projective relations between regions. *IEEE Transactions on Knowledge and Data Engineering*, 18(6):799–814.

Clementini, E., Sharma, J., and Egenhofer, M. J. (1994). Modelling topological spatial relations: Strategies for query processing. *Computers and Graphics*, 18(6):815–822.

Cohn, A. G. and Renz, J. (2008). Qualitative spatial representation and reasoning. *Foundations of Artificial Intelligence*, 3:551–596.

Egenhofer, M. J. and Franzosa, R. D. (1991). Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2):161–174.

Mark, D. M. and Egenhofer, M. J. (1994). Modeling spatial relations between lines and regions: combining formal mathematical models and human subjects testing. *Cartography and geographic information systems*, 21(4):195–212.

Retz-Schmidt, G. (1988). Various views on spatial prepositions. *AI magazine*, 9(2):95–95.