Human Spatial Reasoning in Everyday Language: Inferring Regions that Describe Spatial Relations

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

[1]Federal University of Campina Grande (UFCG)
Systems and Computing Department
Campina Grande – PB – Brazil

joselucas@copin.ufcg.edu.br, campelo@dsc.ufcg.edu.br

Abstract. *The study of natural spatial language used by humans can foster the development of geographically aware systems that are able to assist us in our daily lives. In this specific type of language, one of the most important features is the usage of spatial relations. 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. The computation of these relations in the application level is important to locate objects in the space or even interpret a location description provided by a human. There are many types of spatial relations such as metric and topological, and in fact implementations of these relations are even available in common spatial database query languages. However, these relations are not easily translated into the ones that people use in daily communication, for which there is still a lack of approaches to estimating regions described by them. In this paper, a set of algorithms that derive polygons that match spatial relations used in daily communication are proposed. The algorithms were evaluated by comparing their outputs with drawings produced by humans. The results indicate that although this is a very difficult task, the proposed procedures produce satisfactory spatial region extents that almost always intersect the ones drawn by the subjects.*

1. Introduction

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.

The emergence of models that are capable of interpreting the spatial relations is of utmost importance for the development of geographic aware applications. Even though methods for generating spatial relations do exist, they tend to be focused on a different set of relations than the ones that are present in the language used by people more often. To address this issue, the main goal of this paper is to propose a set of algorithms that project in a 2D map, some of the spatial relations that are most often used by humans in daily communication. They take as input the geometry of the object that serves as reference in the spatial relation and return the polygon that corresponds to the real-world region described by it.

An experiment was carried out to evaluate the precision of these algorithms, where volunteers were asked to read phrases containing references to landmarks and spatial relationships and then to draw on a map the polygon(s) they thought that best describes the region referred to in the text. Apart from validating the algorithms, the collected data was analyzed with the aim of answering some research questions on 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.

The emergence of algorithms that project the relations that people use the most in daily dialogue, as well as the better understanding of human spatial reasoning, has the potential of enabling the development of many applications, such as geographically aware chatbots, different search interfaces to be used in map services or even improving the operation of driverless vehicles.

This paper is structured in the following manner. The next section explores previous works that can be related to this line of research. After that, Section 3 describes the methods of developing the research as well as the materials used. Section 4 presents the proposed algorithms. Using the data collected in the experiment, section 5 tries to answer a relevant question about the human interpretation of spatial relations. The experiment data is used to evaluate the precision of the algorithms in section 6. Lastly, section 7 presents some final thoughts on the research.

2. Related Work

Spatial Relations have been studied for decades 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 of projective relations [Clementini and Billen 2006], this gives the additional geometric definitions needed to compute relations. [Clementini and Bellizzi 2019] build on top of this mapping and present a Java application framework that implements the directional relations given the assumption that the relations are being interpreted in a few of the frames of reference.

In the present work, a different approach is presented in computing directional relations. A set of algorithms that generate regions that correspond to the relations by computing intersections between buffers around landmarks and nearby streets is proposed. The idea is that these procedures could be used in an application after a stage of entity extraction from natural language, where landmarks and spatial relations are collected, to generate possible projections of spatial relations. Despite most of the relations explored being directional, the set of relations covered by the proposed algorithms 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 although are widely used by people in conversation, to the best of our knowledge have neither been categorized as directional nor explored before such as Next-To, Near and At-Street.

3. Materials and Methods

In a preliminary study, a group of 57 participants were presented with a point in a map and asked to describe its location. The descriptions were then studied and a list of the most frequently mentioned spatial relations was compiled.

These relations were implemented in a database spatial query language and the algorithms designed. In order to evaluate them and also better understand the way people reason about spatial relations, another experiment was carried out. Through the usage of a web app, another group of 20 participants, none of whom participated in the preliminary study, 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 participants of this experiment form a diverse group of people from different backgrounds. However, most of them are students (undergrad and grad) aged between 20 and 35.

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

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. It is also possible to draw multiple disconnected geometries, to support scenarios where a participant wishes to draw on more than one place. Each person had to draw five relations (Table 1) for each of the four landmarks.

Table 1. Spatial Relations Names

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

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

The drawings were then stored in the GeoJSON format and a CSV of the data is available at GitHub [1].

4. Spatial Relations Algorithms

This section presents the algorithms designed to infer spatial extents of regions, vaguely described in terms of different spatial relations to certain landmarks. The algorithms are presented as functions named as spatial relations.

These algorithms must deal with some level of uncertainty when there is insufficient information about the spatial features referred to in the descriptions. For example, for a building located at a street corner, defining its facade may be considerably challenging or even impractical using traditional mapping data, posing even more challenges for modeling some relations, such as In-Front-Of, as the buildings' facade may be extended around the corner.

The lack of geographic data in the appropriate format may also impact the efficacy of this kind of algorithm. For example, in traditional mapping datasets, many spatial extents of landmarks are available, however, many others are represented as single geographic coordinates (points). The algorithms proposed here are able to better infer the spatial extents of regions for polygon inputs. In the absence of this format of data they are also capable of working with point inputs. However, we believe the availability of this kind of data as polygons tends to increase considerably in the next years, contributing directly to the accuracy of systems that will incorporate those algorithms.

All proposed functions take as input a geometry, representing the spatial extent of a landmark, and return a generated polygon, representing the spatial extent of a region that best describes the relation with respect to that landmark. These regions are called here **acceptance regions**. An important observation is that the algorithms work in the scope of streets. This way, when generating the acceptance region to the Right-Of relation for instance, one should expect that the algorithm will produce a region that encompasses the portions of street that are to the right of the landmark.

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

4.1. In Front of

Algorithm 1 implements the relation In-Front-Of. Line 2 tests whether the input geometry is of type point or polygon. For point input geometries a buffer around the input is computed (Line 3), the intersection between this buffer and the nearby streets (Line 4) represents the candidates to be included in the acceptance region. For each candidate, the algorithm tests if there is another object between the input landmark and the candidate and includes the street in the final result, if it does not meet these conditions (Lines 5 to 10).

Algorithm 1 In Front of

```
 1: function INFRONTOF(landmark geometry)
 2:     if landmark is of type point then
 3:         Compute a buffer around landmark
 4:         intStreets = the intersection between the buffer and all streets that intersect it
 5:         for each street in intStreets do
 6:             testLine = a line from landmark to street
 7:             if testLine do not crosses another landmark or street in intStreets then
 8:                 finalFront = Union of street and finalFront
 9:             end if
10:         end for
11:     else
12:         for each side in the landmark polygon do
13:             Compute a one-sided buffer in the line representing the side of the polygon
14:             streetFront = the union of all streets that intersect the one-sided buffer
15:             Compute a buffer between the landmark and streetFront
16:             if There are no other objects inside this buffer then
17:                 finalFront = Union of streetFront and finalFront
18:             end if
19:         end for
20:     end if
21:     return finalFront
22: end function
```

For polygon input landmarks, the procedure is almost the same, with the exception that the buffer used to select the candidate streets as well as the tests that check if a candidate street is really in front of the landmark (Figure 2), can be computed for each of the lines representing sides of the polygon (Lines 12 to 19), this allows the generation of an acceptance region that is much more accurate, for it really represents the full extension of region that is in front of each particular side of landmark, as opposed to an estimate of such region, which is the case for point input landmarks. Line 12 returns the acceptance region produced by the union of street candidates for the appropriate input format (Line 8 for points and Line 17 for polygons).

4.2. At Street

An example sentence that uses this relation is: *"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.

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 intersection between the street and the front is big then
5:             finalStreet = Union of intersection and finalStreet
6:         end if
7:     end for
8:     return finalStreet
9: end function
```

Algorithm 2 computes the front region of the landmark by making use of Algorithm 1 (Line 2). It includes in the acceptance region all streets that intersect the front (Lines 3 to 7). 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 points) but only one of the streets is really in front of the landmark.

If the data includes the 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 implemented in Algorithm 3. It is quite simple and is the same for points and polygons. A buffer around the landmark represents the region that is *near* it (Line 2).

Algorithm 3 Near

```
1: function NEAR(landmark geometry, distance float)
2:     return a buffer with a distance-sized radius around landmark
3: 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, Algorithm 4 takes as input two geometry parameters.

The function draws a line between the two input geometries (Line 2). If a buffer around the line between both input landmarks is returned, the result will include regions

Algorithm 4 Between

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

Figure 2. The street is in front of *Localiza Hertz*, not the input landmark (*Niscar*).

Figure 3. Points that are equidistant.

that are actually outside the desired relation. To fix this issue, the procedure finds two points along the line that are located at the same distance d to each of the lines ends (Line 3, Figure 3), in PostGIS, this could be done using ST_LineInterpolatePoint. A new line between these two points is drawn (Line 4) and a buffer of radius d around it is returned (Line 5). Since d is the exact distance between each input geometry and the new line, a buffer with radius d will not cover any region that is not between the two landmarks.

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, Algorithm 5 starts by computing the landmark front (Line 2) using Algorithm 1. The street relation is also used, so Line 3 uses Algorithm 2 to compute the street relation. A buffer around the landmark is generated (line 4) and intersected with the region that correspond to the street of the landmark (Line 5). For each street in this intersection (Lines 6 to 18) a line is drawn starting from the landmark (Line 7, 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 8 to 10). A special case is when the input geometry is actually in point format, for the front relation for points can include large regions. In this scenario, Lines 13 to 17 find for each street in the resulting area, the point that is closest to the input landmark. Buffers around these points serve as the front relation for the input landmark. After this, the difference between the resulting area and the front is returned (Line 19),

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 = nextFinal minus 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 from landmark to a street in the Near relation.

Figure 5. It crosses the difference between the Near relation and the street.

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 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 Algorithm 6 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, as in Figure 6).

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
```

This function computes the front relation using Algorithm 1. For each of the streets that intersect the front acceptance region, (Lines 7 to 15) it computes a one-sided buffer on a line that goes from the input landmark to the street (Lines 8 and 9). To determine in which side of the line the buffer is generated, the string representing the frame of reference is used (Lines 2 to 4). The relation Next-To includes regions that are positioned immediately to the left and to the right of the landmark. For this reason, it is computed (Line 6). If any of its containing polygons intersects the one-sided buffer, it is included in the final result. Line 16 return the acceptance region, formed by the polygons of the Next-To relation that intersect the one-sided buffer.

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. Evaluating Frames of Reference in the Spatial Reasoning of People

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 decisions when interpreting this relation. The question of interest here is: Which type of frame of reference (FoR) best describes the reasoning behind the decisions of people, when faced with the task of locating a region said to be at the right side of some reference object?

To answer this question, the drawings in the experiment were analyzed. Particularly the ones made when participants were prompted with the sentence that included the Right-Of relation. The translated sentence was: “Your ride is to the right of <landmark>”.

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, which 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.

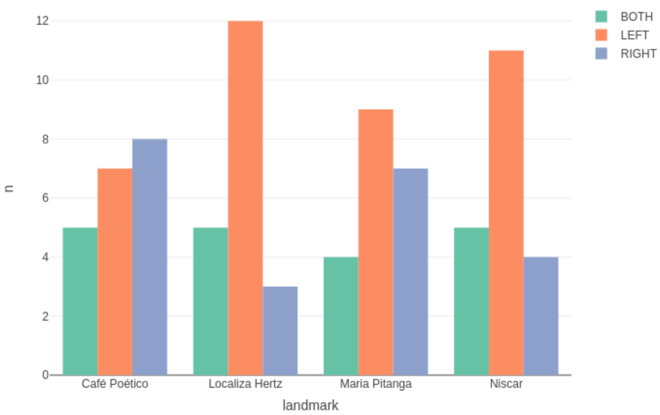

Figure 7. Location of drawings in the Right-Of relation

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 Precision of the Proposed Algorithms

In order to evaluate the precision 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 were 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

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

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

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 conversation. It includes an experiment that evaluates how well

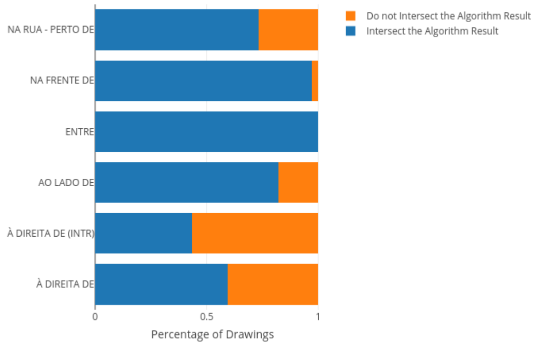

Figure 8. Percentage of drawings that intersect the region produced by the proposed algorithms

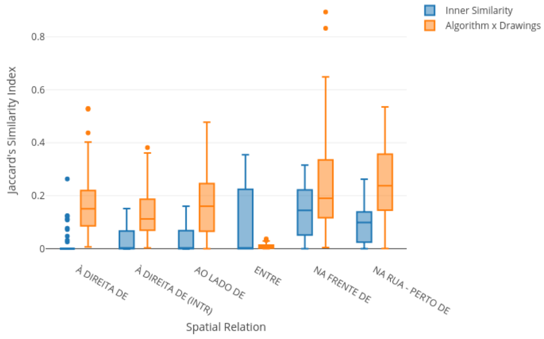

Figure 9. Jaccard's Similarity Coefficient between geometries

the output of the algorithms, match the mental representation of spatial relations in the minds of the participants. The analysis of the collected data shows that this is a difficult problem, however the proposed algorithms hold promissory results, intersecting most of the regions made by the participants and presenting some similarities to them. Another contribution to the field is making available a dataset of more than 400 drawings of spatial relations, allowing further studies on the interpretation of spatial relations by humans.

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.