A splitting line model for directional relations

Kevin Buchin
Dep. of Mathematics and
Computer Science
TU Eindhoven
The Netherlands
k.a.buchin@tue.nl

Vincent Kusters
Institute of Theoretical
Computer Science
ETH Zürich
Switzerland
vincent.kusters@inf.ethz.ch

Bettina Speckmann
Dep. of Mathematics and
Computer Science
TU Eindhoven
The Netherlands
speckman@win.tue.nl

Frank Staals
Dep. of Information and
Computing Sciences
Utrecht University
The Netherlands
f.staals@uu.nl

Bogdan Vasilescu
Dep. of Mathematics and
Computer Science
TU Eindhoven
The Netherlands
b.n.vasilescu@tue.nl

ABSTRACT

Directional relations are fundamental to spatial data queries, analysis and reasoning. Consequently there has been a significant amount of effort to determine directional relations between two regions. However, many existing methods do not perform well when the regions are neighboring or intertwined. In this paper we introduce a new model for directional relations which is based on a splitting line separating the two regions in question. We identify essential quality criteria for directional relation models and translate them into measurable properties of a given splitting line. We present an efficient algorithm that computes an optimal splitting line for two regions and perform extensive experiments. Our results show that the splitting line model captures directional relations very well and that it clearly outperforms existing approaches on pairs of neighboring or intertwined regions.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*

General Terms

Algorithms, Theory

Keywords

Directional relations, splitting line, geometric algorithms

1. INTRODUCTION

Is Sweden south of Norway? And what are the western neighbors of Brazil? Directional relations are frequently

used to select data in spatial databases and are fundamental to spatial data queries, analysis and reasoning [1, 8]. Directional relations are not only used widely in geographic information systems, but also in areas like artificial intelligence [6], computer vision [10], and multimedia [19]. Consequently there has been a significant amount of effort to determine directional relations automatically.

Our motivation to study directional relations also stems from schematized maps like subway maps [17] and (rectangular) cartograms [16]. In these maps the exact geometry is not very important. But for human perception it is essential that spatial relations like topology, relative distance, and orientations are preserved. This is particularly crucial for objects that are close in geographic space. For neighboring regions many of the existing models for directional relations do not give a useful answer. In particular, many existing models do not work for intertwined regions, that is, regions where the centroid of one lies inside the bounding box of the other.

We illustrate these issues with several examples. We distinguish a reference region A, drawn in dark gray, and a target region B, drawn in light green (light gray in grayscale). The region centroids are indicated by circles. We want to determine the direction of the target region with respect to the reference region. For example, in Fig. 1 we ask: "where does Italy lie with respect to

Figure 1: Austria and Italy.

Austria?" We are usually interested in the compass direction that describes the directional relation best. Here we use the term compass directions to refer to the union of the 4 cardinal directions (north, south, west, east) and the 4 ordinal directions (northwest, northeast, southwest, southeast).

A reasonable and commonly used indicator for the directional relation between two regions is the direction between their centroids [9, 12], possibly snapped to one of the 8 compass directions. The centroids tell us that Italy lies to the south of Austria, which is arguably the best answer. Since the direction between the centroids is symmetric, this also

implies that Austria lies north of Italy. However, northeast seems to be the better answer in this case: visually speaking, a significant part of Austria's area lies to the northeast of Italy.

The popular direction-relation matrix model [7] allows for such asymmetric answers. It subdivides the space around the bounding box of the reference region into nine direction tiles and classifies other regions according to the cell of the subdivision they lie in. Generally this model gives a good indication of the compass direction. However, for neighboring regions –in particular, if one country is considerably larger than the other– a large part of the target region might lie inside the bounding box of the reference region, in which case no direction can be determined by the model.

For instance, Paraguay lies completely inside the bounding box of Brazil (Fig. 2), and therefore the direction-relation matrix does not give a useful answer. The compass direction resulting from the centroids in this case is south, while southwest is arguably the better answer. Here both the general direction of the joined border and the relative position of the majority of the area of the respective countries seem to play a major role in perception of directional relations.

Figure 2: Paraguay and Brazil.

Another interesting example are the Gambia and Senegal which are intertwined. For humans it appears that directional relations between intertwined countries are also related to separation. Here the question is: in which direction do we need to move the Gambia to separate it from Senegal? Consequently, people tend to place Senegal to the east or northeast of the Gambia and the Gambia to the west or southwest of Senegal.
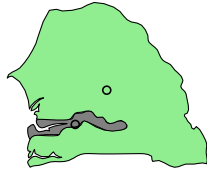
Figure 3: The Gambia and Senegal.

Our final example illustrates how important the direction of a shared border is for directional relations between neighboring countries. Belarus and Ukraine are separated by a nearly horizontal border. Consequently they are often perceived as north-south neighbors, although neither centroids nor the location of the majority of the respective areas point that way.
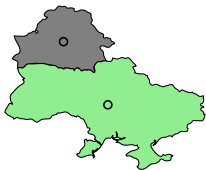
Figure 4: Belarus and Ukraine.

Results and Organization. We present a new model for directional relations between regions which is based on a splitting line separating the two regions in question. Our model performs very well, specifically for neighboring or intertwined regions.

In Section 2 we first survey related work. In Section 3 we identify essential quality criteria for direction relations and translate them into measurable properties of a given splitting line. In Section 4 we present an efficient algorithm that computes an optimal splitting line for two regions. We close with an extensive experimental evaluation in Section 5 where we compare the results of our new method to the compass directions obtained from the region centroids and from the direction-relation matrix. We used a user study to determine appropriate parameters for our method; the details of this study can be found in Appendix A.

2. RELATED WORK

Here we discuss several approaches that determine the directional relation between two regions, represented by arbitrarily shaped polygons in the plane. We denote the reference polygon by A and the target polygon by B.

Haar [9] introduced a triangular model which approximates both the reference and the target polygon by their centroids. It considers four angular regions corresponding to the four cardinal directions extending outwards from the centroid of the reference polygon. The directional relation is determined by testing in which of these regions the centroid of the target polygon lies (Fig. 5 a). This model ignores both shape and size of the polygons and thus can produce inaccurate results. For example, polygon C in Fig. 5 a would be classified as being north of polygon A.
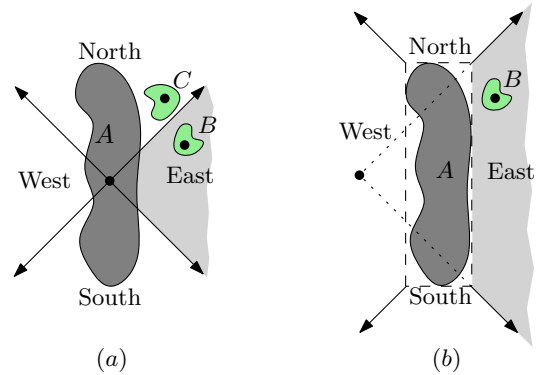
Figure 5: *(a)* **The triangular model,** *(b)* **the cone-based model.**

Peuquet and Ci-Xiang [12] extended the triangular model to take size, shape, and orientation of the polygons into account. Their cone-based model uses the bounding box of the reference polygon when determining the triangular regions, which now extend from the corners of the bounding box (Fig. 5 b). When dealing with intertwined polygons the directional relation is determined using rays shot from the centroid of the target polygon. The use of a bounding box
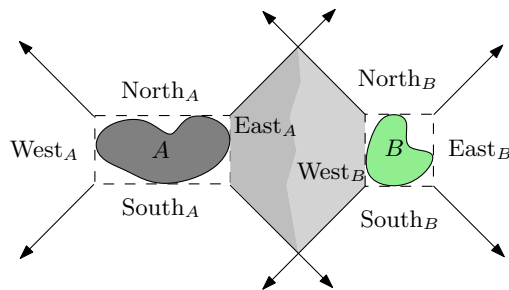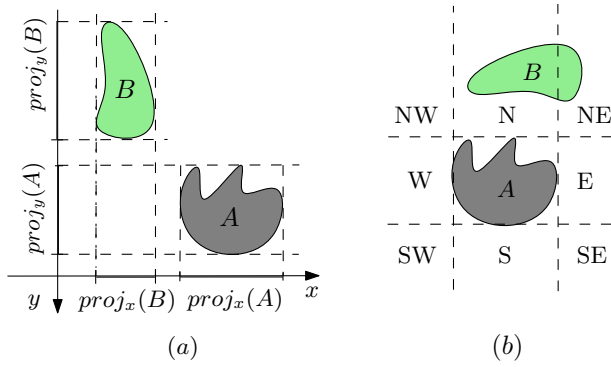
Figure 6: The intersection-based model.

Figure 7: *(a)* **The projection-based model,** *(b)* **the direction-relation mode, the center tile is called** *sameLocation.*

and a centroid to approximate regions can lead to inaccurate results for neighboring regions.

Abdelmonty and Williams [2] presented an intersection-based method which further refines the cone-based model. They consider the intersections between the lines determining the directional regions of the reference and target polygons, passing through the corners of their respective bounding boxes (Fig. 6). This method suffers from the same problems as the previous approach and is symmetric.

Papadias and Theodoridis [11] propose a projection-based model in which the reference and target polygons are approximated by their bounding boxes. The model distinguishes between 169 different directional relation configurations by computing the relations between the projections of the two bounding rectangles onto the x- and y-axes (Fig. 7 a).

Goyal [7] introduced a similar model in which the reference object is approximated by its bounding box and the space surrounding it is partitioned into nine direction tiles (Fig. 7 b). The compass directions between a target and a reference polygon are stored as a 3×3 boolean direction-relation matrix. An entry is true if the intersection between the target polygon and the corresponding direction tile is non-empty. The model can be adapted to return a unique direction by storing in the matrix the proportion of the total area of the target polygon that lies in the corresponding direction tile and selecting the direction in which most of the area of the target polygon lies. The direction-relation matrix
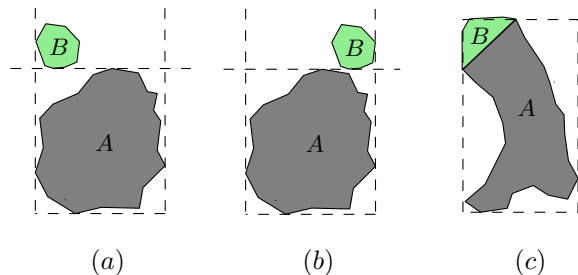

Figure 8: *(a) and (b)* **Projection-based model and direction-relation matrix model:** B **lies** *northwest* **or** *northeast* **of** A **but is classified** *north*, *(c)* **direction-relation matrix model:** B **lies** *northwest* **of** A **but is classified** *sameLocation.*
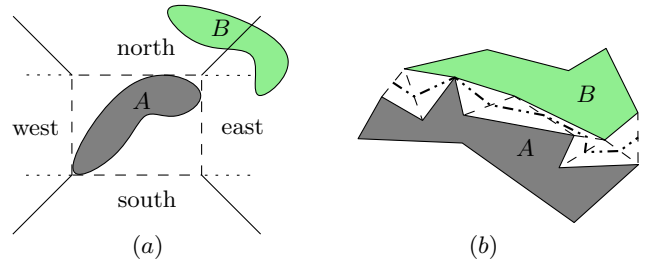

Figure 9: *(a)* **Cone-based directional relations model,** *(b)* **the DVD approach: triangulation (dashed) and DVD (dash-dotted).**

model is not symmetric. Both the projection-based model and the direction-relation matrix model can lead to inaccurate results when the polygons are intertwined or when the target polygon lies partially within the bounding box of the reference polygon (Fig. 8).

Several extensions to the direction-relation matrix model exist. Cicerone and Di Felice [4] study compass directions between regions with uncertain boundaries. They consider such objects to have two boundaries (an inner and an outer boundary) and propose a model that creates a double partition of the plane around the inner and the outer boundary of the region. The resulting 25 distinct direction tiles are stored in a 4-value, 5x5 matrix. Skiadopoulos et al. [14, 13] present an efficient implementation of the direction-relation matrix model that can handle disconnected regions and regions with holes. Chen et al. [3] proposed an extension to the direction-relation matrix model which uses the overlay of two grids of directional tiles. This approach is symmetrical, does not give a unique answer, and is ill-suited to deal with intertwined polygons.

Skiadopoulos et al. [15] presented a cone-based directional relations model for disconnected regions based on the cone-based model [12]. Here only the reference polygon is approximated by its bounding box, as in the direction-relation matrix model. The space around the reference polygon is partitioned into 5 regions using four rays originating from the four vertices of the bounding box (Fig. 9 a).

Deng and Li [5] proposed a statistical approach. They rasterize polygons A and B into small cells, compute the angle between the centroids of each pair of cells, and return the median angle. This method mostly ignores the shape of the polygons and can so lead to inaccurate results.

Yan et al. [18] proposed a method using directional Voronoi diagrams (DVDs). They first triangulate the space between the two polygons A and B. For every triangle that has two edges connecting A to B the DVD contains a line segment connecting the midpoints of these two edges (Fig. 9 b). They then compute a quality value for each direction based on the lengths and directions of the edges of the DVD and output all values. This approach is symmetric and does not return a unique answer. It ignores the area distribution of the polygons and is based on local features of the boundaries which do not necessarily represent the general directions well.

3. THE SPLITTING LINE MODEL

We propose the following quality criteria for a directional relation model. Each criterion is accompanied by a figure to illustrate the concept.

Area. The direction should indicate where the majority of the area of the target polygon B lies with respect to the reference polygon A. This is the most obvious and important factor in deciding the directional relation, and is also utilized by the centroid and matrix methods. This criterion plays an important role in Fig. 1.
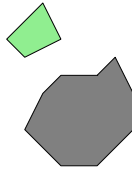
Alignment. If the polygons are close together, then the direction should be perpendicular to the shared border. In the figure, considering only the angle between the centroids might suggest that northwest is the best direction, whereas due to the orientation of the shared border, north is perceived as the better answer here. This criterion is important in Fig. 4, where the shared border between the countries is approximately horizontal.

Removal direction. The direction should indicate how to move A and B apart without incurring intersections. This factor becomes important when the polygons are intertwined, as illustrated in the figure. In this example, the centroid of the reference polygon actually lies left of the centroid of the target polygon. Hence, the centroids method will answer east, whereas west is perceived as the right answer here. A similar situation is illustrated in Fig. 3.
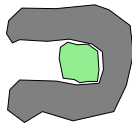
Robustness. Extensions of A or B can affect the direction only if they contain a significant part of the area of A or B. It is important for the mathematical soundsness of the model that the answer is not affected by minor changes (with respect to area) to one of the polygons.
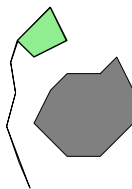
Affine transformation. The directional relation should be consistent under affine transformations. Since the figure is a 90°-rotated version of the figure that accompanies the first paragraph discussing the area criterion, the answer is southwest.
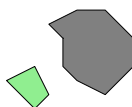
We do not consider symmetry to be an important criterion for a good directional relation model. As illustrated by Fig. 1, it can be argued that the directional relation is not always symmetrical.

We model our criteria as measurable properties of a splitting line ℓ which separates A and B. We associate each side of ℓ with one of the polygons: the A-side and the B-side of ℓ. Parts of both A and B may lie on the "wrong" side of ℓ. We obtain the directional relation from a splitting line ℓ by taking the normal vector on the B-side of ℓ (Fig. 10). The quality of a splitting line is evaluated by a composition of the measures derived from the quality criteria. To determine the directional relation we find the optimal splitting line.

To evaluate the quality of a particular line ℓ we conceptually divide the scene into thin slabs that are perpendicular to ℓ. We then compute a quality value for each slab and finally combine the values for all slabs. As the number of slabs used approaches infinity, and hence the width of a slab approaches 0, the measure for a line can be represented using an integral. For our explanation we will assume the slabs have a width of Δx which makes reasoning about them more intuitive.
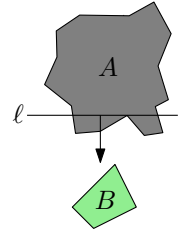

Figure 10: A splitting line ℓ for A and B.

Notation. To formalize our splitting line model we first introduce some notation. Let A be the reference and B be the target polygon. We consider the horizontal splitting line $\ell : y = y_0$ with the A-side above ℓ. Since we take the limit with infinitely thin slabs, we can assume that the intersections of A, B, or the white space (everything but A and B) with a slab $[x : x + \delta x] \times [-\infty : \infty]$ are rectangles which we call *segments* (see Fig. 11). We refer to the
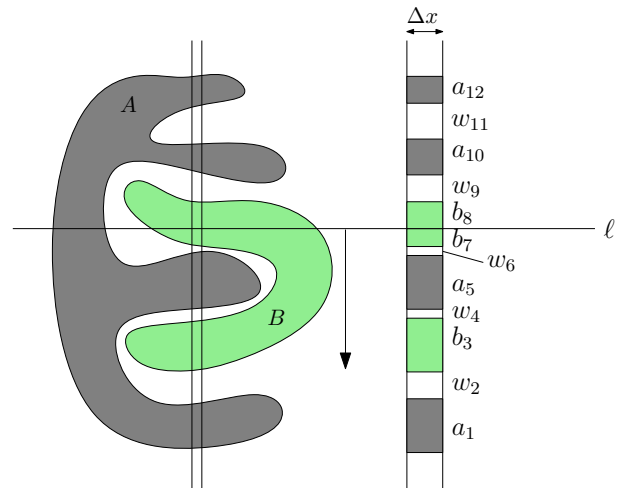

Figure 11: A slab for reference polygon A and target polygon B. The segments are numbered in order, and we have: $TopA = \{a_{10}, a_{12}\}$, $TopB = \{b_8\}$, $TopW = \{w_9, w_{11}\}$, $BottomA = \{a_1, a_5\}$, $BottomB = \{b_3, b_7\}$, **and** $BottomW = \{w_2, w_4, w_6\}$

height of a segment s as $height(s)$. The set of A-segments on the A-side of ℓ is called $TopA$ and the set of A-segments on the B-side of ℓ is called $BottomA$. $TopB$ and $BottomB$ are defined similarly. Let m_1 be the minimum y-coordinate of an A- or B-segment in a slab, and let m_2 be the maximum. We define $TopW$ to be the set of white space segments inside the sub-slab $[x : x+\delta x] \times [y_0 : m_2]$ on the A-side of ℓ; $BottomW$ is defined similarly. The set of all segments between m_1 and m_2 is called $Strip$. For segments s and t we say $s < t$ if all points in s have a lower y-coordinate than the points in t.

The Measure. Our quality measure consists of several components that jointly capture our set of quality criteria. As measure we take the sum of these components weighted by positive real weights ρ_i.

To satisfy the area criterion we ask that the majority of A lies on the A-side of ℓ and the majority of B lies on the B-side of ℓ. Thus we take these quantities, $GoodA \cdot \Delta x$ and $GoodB \cdot \Delta x$ (see Eq. 1 and 2), as components of our measure. They constitute the first two terms in our measure given in Eq 5. The Δx factor is not included in the measure since we later take the integral over the combined measure. The modeling of the area criterion can be refined to not only take into account the side of the line A and B lie in, but also whether A and B are actually facing each other, that is if a slab contains a large fraction of A on the correct side then it should also contain a large fraction of B on the correct side and vice versa (Fig. 12). Furthermore, if there is a large fraction of B on the correct side, only a small fraction of A should be on the wrong side. This is captured by the third, forth and fifth term in Eq 5. To normalize these terms the height of the minimal axis-aligned bounding box of A (resp. B) is used, which we denote by $height(A)$ (resp. $height(B)$).

To satisfy the alignment criterion, ℓ should align well with the border of A (but not necessarily with the border of B). We hence penalize parts of A that are on the correct side but separated from the splitting line by B or white space. Specifically, we negatively weigh such parts within the slab by the amount of B and white space separating it from the line, that is by $AlignmentA \cdot \Delta x$ (see Eq. 3). We weigh the contribution of the slab to this term by the fraction of B in the slab on the correct side. This is the sixth term in Eq 5.

$$GoodA = \sum_{a \in TopA} \frac{height(a)}{area(A)} \qquad (1)$$

$$GoodB = \sum_{b \in BottomB} \frac{height(b)}{area(B)} \qquad (2)$$
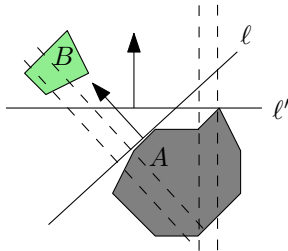
Figure 12: The splitting line ℓ is better than ℓ' as none of the slabs of ℓ' contains both A and B.

$$AlignmentA = \sum_{w \in TopW \cup TopB} \left(\frac{height(w)}{\sum_{s \in Strip} height(s)} \cdot \frac{\sum_{a \in \{s | s \in TopA \wedge w < s\}} height(a)}{area(A)} \right) \qquad (3)$$

$$ObstructB = \sum_{b \in TopB} \left(\frac{height(b)}{area(B)} \cdot \frac{\sum_{a \in \{s | s \in TopA \wedge b > s\}} height(a)}{\sum_{a \in TopA} height(a)} \right) \qquad (4)$$

To satisfy the removal direction criterion we ask that a slab does not contain alternations of A and B. More specifically, we penalize parts of B (resp. A) on the wrong side of the line that have parts of A (resp. B) between it and the splitting line. This results in the terms $ObstructB \cdot \Delta x$ (see Eq. 4, $ObstructA$ is defined analogously) and $ObstructA \cdot \Delta x$. These constitute the final two terms in Eq 5. The resulting splitting line model is consistent under affine translations by definition. Most of the terms used in the measure are not significantly affected by small quantities of A, B, or white space, and therefore satisfy the robustness criterion. The only term that can be affected by such small quantities is the height of the bounding box. However, the overall measure remains robust: if the height of the bounding box is extremely large relative to the area of the region then the corresponding components will simply be very small, thus giving more weight to the more robust components.

$$M = \rho_1 \cdot GoodA + \rho_1 \cdot GoodB \qquad (5)$$
$$\rho_2 \cdot \frac{\sum_{a \in TopA} height(a)}{height(A)} \cdot GoodB +$$
$$\rho_2 \cdot \frac{\sum_{b \in BottomB} height(b)}{height(B)} \cdot GoodA -$$
$$\rho_3 \cdot \frac{\sum_{a \in BottomA} height(a)}{height(A)} \cdot GoodB -$$
$$\rho_4 \cdot \frac{\sum_{b \in BottomB} height(b)}{height(B)} \cdot AlignmentA -$$
$$\rho_5 \cdot ObstructA - \rho_5 \cdot ObstructB$$

The value of M depends on the slab and the splitting line chosen and therefore is a function of x and y. We denote this function by $M(x, y)$. The resulting overall measure is now

$$M_{line}(y) = \int_{-\infty}^{\infty} M(x, y) \, dx. \qquad (6)$$

4. ALGORITHM

In this section we first present an efficient algorithm to compute a description of $M_{line}(y)$. We then describe a discretized version of our algorithm, which we use in our experimental validation.

Computing the function $M_{line}(y)$. Let A and B be the input polygons and assume, without loss of generality, that we want to find the optimal horizontal splitting line with

the A-side above the splitting line. A horizontal splitting line can be specified by its y-coordinate. In this section we show that the derivative of the measure $M_{line}(y)$ consists of pieces that are sums of rational function. The optimal splitting line corresponds to a break point or a zero of the derivative. We show how to efficiently compute a description of $M_{line}(y)$ (and of its derivative) using a sweep line algorithm that sweeps a horizontal line ℓ downwards. However, given the description of $M_{line}(y)$ it is still not feasible to solve the resulting equation. Therefore we also present a discretized version of the sweep line algorithm, which we use in our implementation.

We first illustrate the algorithm to compute a description of $M_{line}(y)$ by the example of the contribution of $GoodA$. As M, $GoodA$ is a function in x and y. The component $\int_{-\infty}^{\infty} GoodA(x, y)\, dx$ is piecewise quadratic in y. To see this, consider a splitting line at $y = y_0$ and $y = y_1 = y_0 - \Delta y$ and assume that there are no vertices of A with y-coordinate between y_0 and y_1. Let P be the set of pieces of the polygon A intersected by the horizontal slab $[y_1 : y_0]$ (see Fig. 13). It follows that the component of $GoodA$ increases by $\delta = \sum_{p \in P} area(p)/area(A)$. The pieces in P are all trapezoids. The area of such a trapezoid is quadratic in Δy. Since δ is a sum of quadratic functions, it is also quadratic in Δy. Overall the component is a piecewise quadratic function in y. While sweeping downwards we need to update this component at a discrete set of events, namely the y-coordinates of the vertices of A. At each event the status at the sweep line, that is, the configuration of trapezoids, changes locally. We can update the quadratic function by replacing the contributions of ending trapezoids by the contribution of new trapezoids. In the same way we can compute the contribution of $GoodB$. Using standard data structures like a sorted list as event queue and a balanced binary search tree for the status we can compute these components as functions in y in $O(n \log n)$ time, where n is the number of vertices of A and B.

y_0
y_1

Figure 13: The intersection of polygon A with the horizontal slab $[y_1 : y_0]$ consists of a set of trapezoids.

The other components are more difficult to compute since they depend on A and on B (and on the white space) and since for some components the denominator is a function in x and/or y. Since they all can be computed in the same way, we describe how to compute them by the example of $ObstructB$, which together with $ObstructA$ is the most complicated component. We subdivide the plane into vertical strips through the vertices of A and B. By this we obtain a linear number of strips which do not contain any vertices of A or B in the interior. We first consider the contribution of a single strip.

Assume we sweep a splitting line from $y = y_0$ to $y = y_1 = y_0 - \Delta y$ with a lower edge of A entering the strip at (x_0, y_0) from the left and leaving it at (x_1, y_1) to the right (see Fig. 14) and with white space below that edge. We denote the x-coordinate of the intersection of the lower edge and a splitting line by $x_L(y)$, which is a linear function in y. We further denote the y-coordinate of this intersection by $y_L(x)$, which is a linear function in x. Let $h_A(x)$ and

$h_A(x)$

y_0
y
y_1

$x_L(y)$
$x_0 \quad x_1$
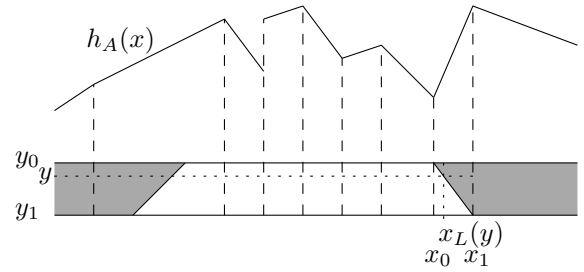
Figure 14: The decomposition of $[y_1 : y_0]$. Parts of polygon A are drawn in grey. The whitespace and B parts are white.

$h_B(x)$ (one such function, $h_A(x)$, is shown in Fig. 14 above the sweep line) denote the height of A and or B in the strip above the sweep line at y_0. Both of these functions are (within the strip) linear functions. Let

$$g(x) = \sum_{b \in TopB} height(b) \cdot \sum_{a \in \{s | s \in TopA \wedge b > s\}} height(a).$$

for the sweep line at y_0. The function $g(x)$ is quadratic.

Now, the contribution of this part up to the sweep line corresponding to y is

$$\int_{x_0}^{x_L(y)} \frac{g(x) + h_B(x)(h_A(x) + y_L(x) - y_0)}{area(B)(h_A(x) + y_L(x) - y_0)}\, dx +$$
$$\int_{x_L(y)}^{x_1} \frac{g(x) + h_B(x)(h_A(x) + y(x) - y_0)}{area(B)(h_A(x + y(x) - y_0))}\, dx.$$

These are integrals over a rational function with a linear function in y on the integral boundary. Therefore the derivative is a rational function in y. The sum of the derivatives over all strips is therefore a sum of rational functions. As description of the contribution we can store $h_A(x)$, $h_B(x)$, $g(x)$ for each strip. We have to update these function each time an edge of A or B intersects the boundary a vertical strip. We therefore have a linear number of events per strip. This yields a quadratic number of events overall. As event queue we use a priority queue. Initially it contains only events corresponding to vertices of B. From such a vertex we examine the incident edges of B and add for such an edge the next intersection point with a strip boundary to the event queue. In this way we store at most a linear number of events at any time. As status structure we store an array with entries corresponding to the strips. The other terms can be handled in the same way. The running time of this algorithm is dominated by the cost of maintaining the event queue, $O(n^2 \log n)$.

Discretized version. If the exact position of the optimal splitting is not essential but a good approximation suffices, then we can compute the measure for discrete steps in y. This has the main advantage of avoiding solving equations involving sums of rational functions and all the associated numerical problems.

We used the discretized version of the algorithm in our experiments. That is, instead of evaluating the measure at dynamically computed event points we decompose the bounding box of A into equal height horizontal strips and evaluate M_{line} at each strip boundary. We also discretize the

computation of M_{line} itself: we divide the scene into equal width vertical slabs, and approximate the intersections of A and B with a slab using a set of rectangular segments. This means that when an edge of either of the polygons enters the slab at height y_0 and leaves it at y_1 we treat it as a horizontal edge at height $(y_0 + y_1)/2$. For the figures in this paper we divide the bounding box of A into 100 horizontal strips, and we use 100 vertical slabs for each computation of M_{line}. For initial experiments we even used only 25 strips and slabs. While 100 strips and slabs should result in better (more exact) results we could in fact not visually distinguish between the two sets. This implies that our approach is "stable" in a certain sense and shows that even a quite simplified version already gives very good results.

5. EXPERIMENTAL VALIDATION

We implemented the discretized version of our algorithm and we performed an experimental validation. For non-neighboring countries our algorithm is nearly equivalent to using the centroid of regions. Hence we focus our evaluation on neighboring regions, specifically, we tested our algorithm on a set of neighboring countries in Europe. Table 1 displays the resulting splitting lines for 24 pairs of input regions using 360 directions. Overall, the splitting line model performs very well, computing the line that best separates the two polygons and aligns with the reference polygon on the shared border. The only exception is SI → IT, where the alignment with the reference polygon (SI) outweighs the fact that the splitting causes a small part of the target polygon (IT) to be cut off.

To obtain the directional relations between two countries, we could simply snap the output of our algorithm run in 360 directions to the compass directions. Instead we opted to limit the number of sweeping directions to the 8 compass directions. To evaluate the performance of this version of our algorithm, we implemented the direction-relation matrix and the centroids algorithms, and compared their performance on a large set of neighboring countries in Europe.

The values of the parameters ρ_i (Table 2), used by both the 8- and 360-directions versions of our algorithm, were selected such that the results of our algorithm match the results of the user study (see Appendix A). In particular, the parameters were not tuned to the validation set. The user study contains 25 questions which use examples from related work, as well as examples designed to assess the influence of the criteria discussed in Section 3, and was completed by 85 people.

The output of our splitting line algorithm for a large set of neighboring countries in Europe is shown in Table 3. Each figure in Table 3 is accompanied by the result of the splitting line, the centroids model, and directional-relation matrix. We write "BE → LU: SE" to indicate that the reference polygon is Belgium, the target polygon is Luxembourg and the splitting line answers southeast. The table is sorted alphabetically by country codes. The directional-relation matrix model and the centroids model show various shortcomings on this test set. The directional-relation matrix fails if a large part of the target polygon lies inside the bounding box of the reference polygon (e.g., BA → ME, BE → LU, ES → PT, or NO → SE). In various cases, the orientation of the shared border is important, e.g., BE → NL, BG → GR, and RS → HU. Since the centroids model does not take the orientation of the border into account, it gives an inaccurate direction in these cases.

Overall, the splitting line performs very well, with very few exceptions. In the cases NO → FI and NO → SE, the reference country has a border which clearly favors a diagonal splitting line: any vertical line either has too much of the reference polygon on the wrong side or too much target polygon between it and the reference polygon. Here the direction given by a good splitting line does not correspond to the direction perceived. Similar situations can be observed in the cases IT → FR, and LV → BY.

ρ_1	1
ρ_2	0.5
ρ_3	3
ρ_4	4
ρ_5	4

Table 2: The parameter values

Table 1: The optimal splitting line for adjacent countries using 360 directions.

AL → ME   AL → MK   AT → CZ   BA → ME   BG → MK   CZ → PL   DE → PL   ES → FR

FR → BE   LV → LT   MD → RO   MK → AL   MK → RS   NL → BE   NL → DE   PL → BY

PL → SK   RO → BG   RO → RS   RS → BA   SI → AT   SI → IT   SK → CZ   SK → HU
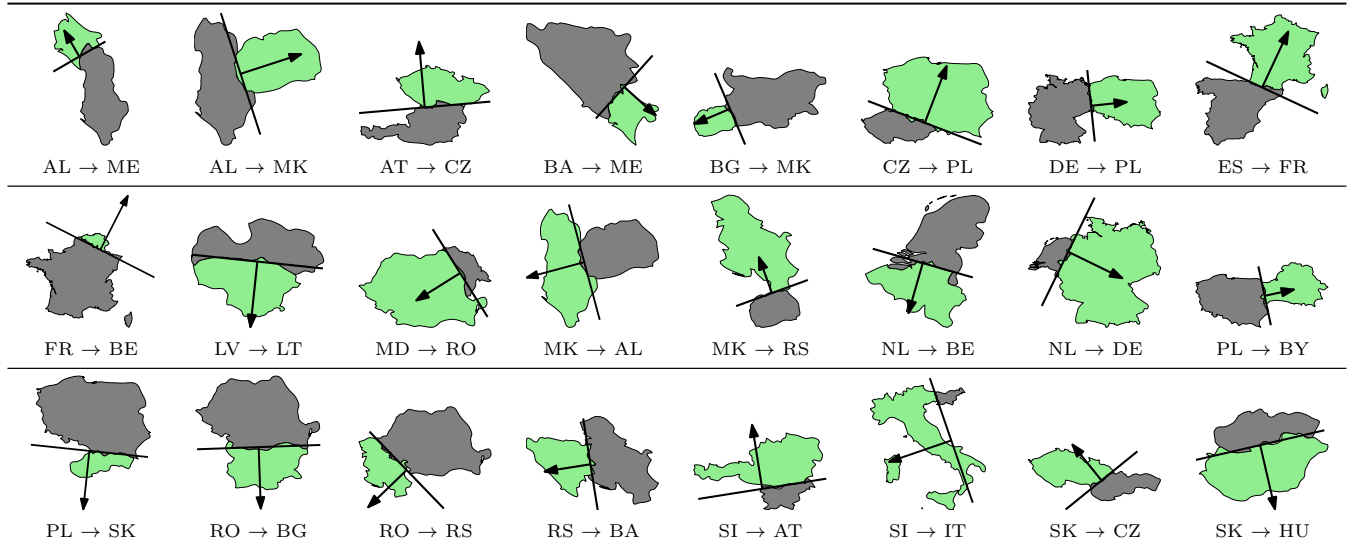
Table 3: The optimal splitting line for adjacent countries.
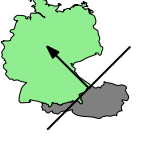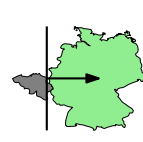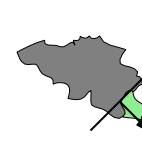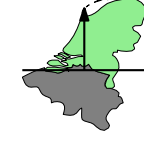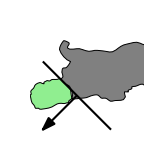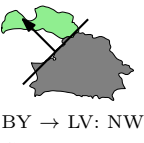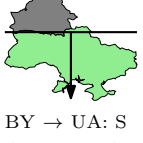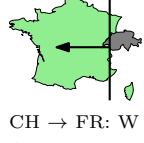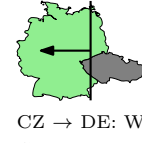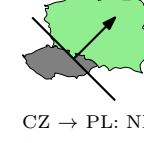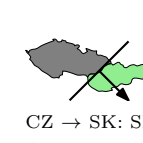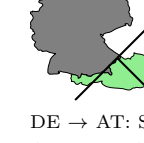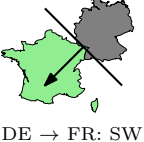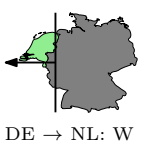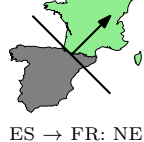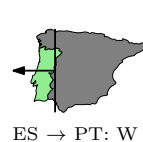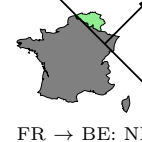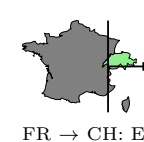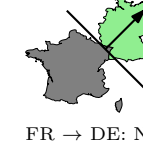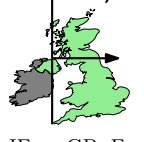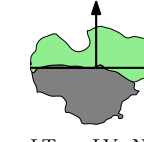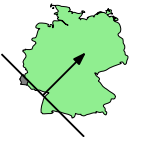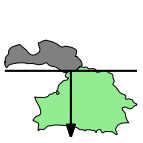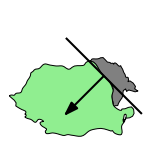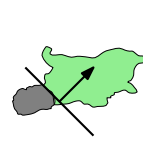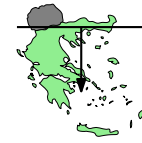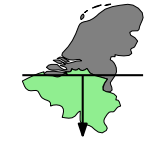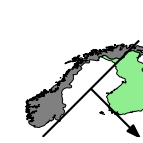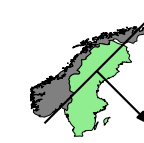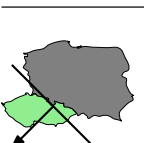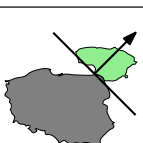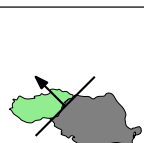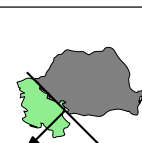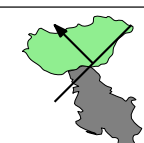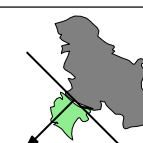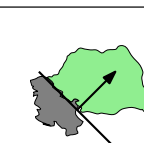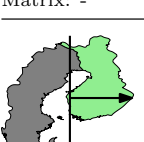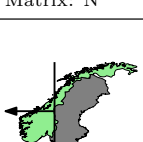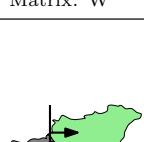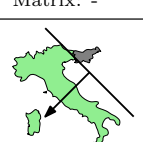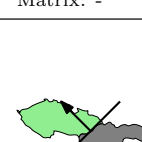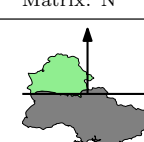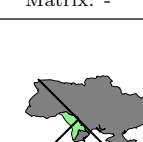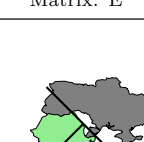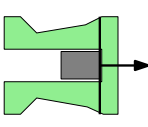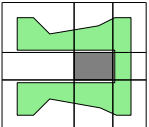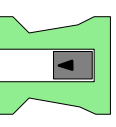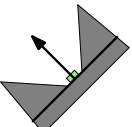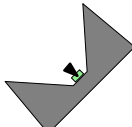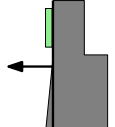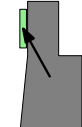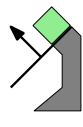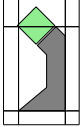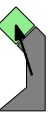
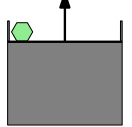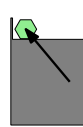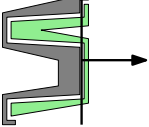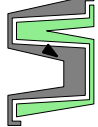|							
AT → DE: NW Centroids: NW Matrix: N	AT → IT: S Centroids: S Matrix: S	BA → ME: SE Centroids: SE Matrix: -	BE → DE: E Centroids: E Matrix: NE	BE → LU: SE Centroids: SE Matrix: -	BE → NL: N Centroids: NE Matrix: N	BG → GR: S Centroids: SW Matrix: S	BG → MK: SW Centroids: W Matrix: W
BY → LV: NW Centroids: NW Matrix: N	BY → UA: S Centroids: SE Matrix: SE	CH → FR: W Centroids: W Matrix: SW	CH → IT: S Centroids: SE Matrix: SE	CZ → DE: W Centroids: W Matrix: NW	CZ → PL: NE Centroids: NE Matrix: N	CZ → SK: SE Centroids: E Matrix: E	DE → AT: SE Centroids: SE Matrix: -
DE → FR: SW Centroids: SW Matrix: SW	DE → NL: W Centroids: W Matrix: W	ES → FR: NE Centroids: NE Matrix: N	ES → PT: W Centroids: W Matrix: -	FR → BE: NE Centroids: NE Matrix: -	FR → CH: E Centroids: E Matrix: -	FR → DE: NE Centroids: NE Matrix: NE	GR → AL: NW Centroids: NW Matrix: -
GR → BG: N Centroids: NE Matrix: N	GR → MK: N Centroids: NW Matrix: -	IE → GB: E Centroids: E Matrix: E	IT → AT: NE Centroids: N Matrix: N	IT → CH: NW Centroids: NW Matrix: -	IT → FR: W Centroids: NW Matrix: W	IT → SI: NE Centroids: NE Matrix: -	LT → LV: N Centroids: NE Matrix: N
LU → DE: NE Centroids: E Matrix: NE	LV → BY: S Centroids: SE Matrix: SE	MD → RO: SW Centroids: SW Matrix: W	MK → BG: NE Centroids: E Matrix: NE	MK → GR: S Centroids: SE Matrix: S	NL → BE: S Centroids: SW Matrix: S	NO → FI: SE Centroids: E Matrix: -	NO → SE: SE Centroids: SE Matrix: -
PL → CZ: SW Centroids: SW Matrix: -	PL → LT: NE Centroids: NE Matrix: N	RO → HU: NW Centroids: W Matrix: W	RO → MD: NE Centroids: NE Matrix: -	RO → RS: SW Centroids: SW Matrix: -	RS → HU: NW Centroids: NW Matrix: N	RS → ME: SW Centroids: SW Matrix: -	RS → RO: NE Centroids: NE Matrix: E
SE → FI: E Centroids: E Matrix: E	SE → NO: W Centroids: NW Matrix: W	SI → HU: E Centroids: E Matrix: NE	SI → IT: SW Centroids: SW Matrix: SW	SK → CZ: NW Centroids: W Matrix: NW	UA → BY: N Centroids: NW Matrix: N	UA → MD: SW Centroids: SW Matrix: -	UA → RO: SW Centroids: SW Matrix: -

Table 4: Examples showing the splitting line, matrix decomposition and the centroids.

SplittingLine	Matrix	Centroids	SplittingLine	Matrix	Centroids	SplittingLine	Matrix	Centroids
E	SW	W	NW	no result	SE	W	no result	NW
NW	N	N	N	no result	NW	E	no result	SE

The splitting line model is, however, particularly suitable for situations in which the two polygons are intertwined or have very irregular shapes, where traditional models such as directional-relation matrix model and the centroids model offer incorrect answers. We illustrate such situations in Table 4. The directional-relation matrix does not give any answer in the situations in which the target polygon is contained inside the bounding box of the reference polygon (examples 2, 3, 4, and 6), and gives answers which do not correspond to the perceived direction in the situations in which one of the eight cells used for determining the direction contains a higher proportion of the area of the target polygon than the other cells (examples 1 and 5). In example 1, the answer is determined by the direction in which B can move away from A without incurring intersections, i.e., east; in example 5, the answer is determined by the orientation of the shared border between the target and the reference polygons, i.e., northwest.

Similarly, the centroids model does not give the desired answer in the situations in which the shared border (examples 2, 3, 4, and 5) or the direction in which the two polygons can be separated without incurring intersections (examples 1, 2, and 6) play the decisive role.

6. CONCLUSION

Many models for directional relations between regions are solely based on where the majority of area of the target region lies relative to the reference region. We believe that this is not sufficient to model directional relations. In particular, for neighboring regions the direction of the common border seems crucial. Additionally, for intertwined regions it seems important how the regions can be separated by translation. Our new model is based on splitting lines. The placement of these splitting lines is computed by optimizing a measure that incorporates all of the above mentioned factors. While we do not believe that the directional relation between regions can be fully captured by a single measure, our experimental evaluation clearly shows the advantages of the splitting line model.

Several challenges remain. If the general direction of the common border between two regions is not straight, there might be no good splitting line, but there might still be a good non-linear separator. For intertwined regions a model that is robust to extensions of insignificant area might not be suitable: these extensions do influence whether the regions can be separated by translation.

In the case that one of eight compass directions has to be chosen, there are several possibilities to apply our model. We choose a direction by comparing the quality of the splitting lines in these directions. But an alternative would be to test more directions and snap to one of the compass directions. It remains open which of these possibilities is most suitable.

In this paper we argued that a directional relation model should allow for asymmetric answers. While this is also indicated by our user study, further cognitive studies would be necessary for a better understanding of this asymmetry.

7. REFERENCES

[1] A. I. Abdelmoty and C. B. Jones. Towards maintaining consistency of spatial databases. In *Proc. 6th International Conference on Information and Knowledge Management (CIKM'97)*, pages 293–300. ACM, 1997.

[2] A. I. Abdelmoty and M. H. Williams. Approaches to the representation of qualitative spatial relationships for geographic databases: A critical survey and possible extensions. In *Proc. Advanced Geographic Data Modeling: Spatial data modeling and query language for 2D and 3D applications (AGDM'94)*, pages 216–221. Springer, 1994.

[3] T. Chen, M. Schneider, G. Viswanathan, and W. Yuan. The objects interaction matrix for modeling cardinal directions in spatial databases. *Lecture Notes in Computer Science*, 5981:218–232, 2010.

[4] S. Cicerone and P. Di Felice. Cardinal relations between regions with a broad boundary. In *Proc. 8th ACM International Symposium on Advances in Geographic Information Systems (GIS'00)*, pages 15–20. ACM, 2000.

[5] M. Deng and Z. Li. A statistical model for directional relations between spatial objects. *Geoinformatica*, 12(2):193–217, 2008.

[6] C. Dewey, M. Knauff, K. Richter, D. Montello, C. Freksa, and E. Loeliger. Direction concepts in wayfinding assistance systems. In *Workshop on*

Artificial Intelligence in Mobile Systems (AIMS'04), pages 1–8, 2004.

[7] R. K. Goyal. *Similarity assessment for cardinal directions between extended spatial objects*. PhD thesis, 2000.

[8] R. K. Goyal and M. J. Egenhofer. Similarity of cardinal directions. In *Proc. 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD'01)*, pages 36–58. Springer, 2001.

[9] R. Haar. Computational models of spatial relations. TR-478 MSC-72-03610, Department of Computer Science, University of Maryland, 1976.

[10] P. Matsakis and L. Wendling. A new way to represent the relative position between areal objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(7):634–643, 1999.

[11] D. Papadias and Y. Theodoridis. Spatial relations, minimum bounding rectangles, and spatial data structures. *International Journal of Geographical Information Science*, 11(2):111–138, 1997.

[12] D. Peuquet and Z. Ci-Xiang. An algorithm to determine the directional relationship between arbitrarily-shaped polygons in the plane. *Pattern Recognition*, 20(1):65–74, 1987.

[13] S. Skiadopoulos, C. Giannoukos, N. Sarkas, P. Vassiliadis, T. Sellis, and M. Koubarakis. Computing and managing cardinal direction relations. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1610–1623, 2005.

[14] S. Skiadopoulos, C. Giannoukos, P. Vassiliadis, T. Sellis, and M. Koubarakis. Computing and handling cardinal direction information. In *Advances in Database Technology (EDBT 2004)*, volume 2992 of *Lecture Notes in Computer Science*, pages 649–650. Springer, 2004.

[15] S. Skiadopoulos, N. Sarkas, T. Sellis, and M. Koubarakis. A family of directional relation models for extended objects. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):1116–1130, 2007.

[16] M. van Kreveld and B. Speckmann. On rectangular cartograms. *Computational Geometry*, 37(3):175–187, 2007.

[17] A. Wolff. Drawing subway maps: A survey. *Informatik - Forschung und Entwicklung*, 22:23–44, 2007.

[18] H. Yan, Y. Chu, Z. Li, and R. Guo. A quantitative description model for direction relations based on direction groups. *Geoinformatica*, 10(2):177–196, 2006.

[19] S. Yau and Q. Zhang. On completeness of reasoning about planar spatial relationships in pictorial retrieval systems. *Communications in Information & Systems*, 4(3):211–234, 2004.

APPENDIX

A. USER STUDY

We performed a user study to determine direction relations. The survey contains 25 questions which use examples from related work as well as examples designed to test for the influence of the shape and orientation of common borders and intertwined parts. The survey was completed by 85 people. We gave the following instructions:

In each question, you are asked to give the position of the green shape relative to the gray shape. In other words, you should complete the statement "The green shape is to the (...) of the gray shape". You are asked to give both a general answer (one of the four cardinal directions) and a more specific answer (one of the four cardinal directions, or one of the ordinal directions). If you think none of the directions match well, then tick the "Unclear" checkbox. Even if you think the direction is unclear, please select general and specific options which match best.

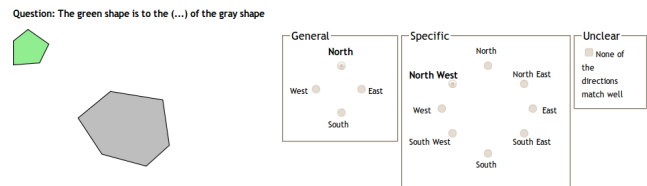These instructions were followed by the example in Fig. 15 together with the following explanation:

Figure 15: Screenshot of the user study.

In the following example, we see that the green shape is to the North West of the gray shape. Hence, we answered "North West" as the specific direction. It is debatable whether the general direction is North or West, but we picked "North" here. The specific direction is pretty clear, so we did not tick the Unclear box.

Table 5: The results of our user study and the three algorithms considered (only 3 of 25 questions shown because of space constraints).