Good NEWS:
Partitioning a Simple Polygon by Compass Directions*

Marc van Kreveld
marc@cs.uu.nl

Iris Reinbacher
iris@cs.uu.nl

Institute of Information and Computing Sciences
Utrecht University, The Netherlands

1 Introduction

Nearly all Internet search engines use term matching to create a list of hits and rank the query results. Normally this yields satisfying results. But when a user seeks for “ruins in Eastern Greece” or “castles near Paris”, standard search engines are less appropriate. The problem is that concepts like “Eastern Greece” and “near Paris” use a spatial relationship with respect to a geographical object, and the terms “Eastern” and “near” themselves are not relevant for term matching. Such problems have lead to research on geographic information retrieval [7, 8]. Addressing spatial searches on the Internet and building a spatially-aware search engine is the focus of the SPIRIT project [6]. By using geometric footprints associated to Web pages, geographic ontologies, methods to determine spatial relationships, and query expansion, spatial searches can most likely be performed much better.

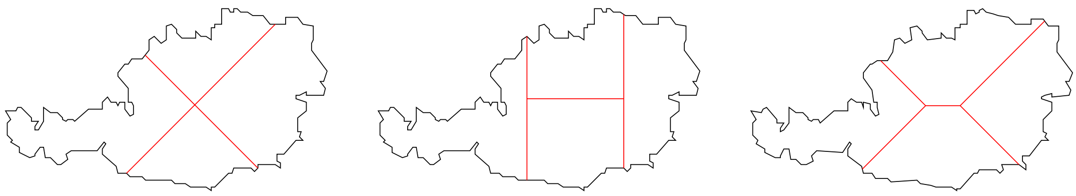

Figure 1: Three partitions of Austria by compass directions.

In this paper we address the problem of dividing a country into four subregions by compass directions (NEWS: North, East, West, South). This is one of the aspects of the SPIRIT project, namely to define and compute spatial relationships that are needed in spatial information retrieval. The partitioning is also useful in geographic user interfaces, where the user can select a region of interest by clicking in the partitioning of the region. The regions of interest, like “the South of Austria”, do not have a well-defined boundary, but are used in a more loose sense. This fuzziness of geographic objects due to language issues is well-known [3, 11]. Spatial relationships between two geographic objects that describe proximity or relative position are discussed in [1, 9].

In order to compute a good NEWS partition, we describe an algorithm to determine — for a simple polygon P and a positive real A — all wedges with a given angle and orientation that contain exactly area A of P. We solve the standard version of the wedge problem in optimal $O(n^2)$ time, and the version where the wedge is restricted to its simply-connected part inside P in $O(n \log n)$ time. However, the corresponding NEWS partitions are computed in $O(n \log n)$ time in both cases.

Related in computational geometry is research on area partitionings and continuous ham-sandwich cuts of polygons [2, 5]. Shermer [10] shows how to partition a simple polygon by a vertical line in two equal-area halves in linear time. Díaz and O’Rourke show in [4] how to partition a convex polygon into equal-size parts using an orthogonal four-partition.

*This research is supported by the EU-IST Project No. IST-2001-35047 (SPIRIT).

2 Criteria for a good NEWS partition

There are many criteria one can use to partition a country into four regions by compass directions. Some criteria are especially relevant for the application in query answering of geographic information retrieval whereas others apply more to geographic user interfaces. We list the most important criteria next:

- the relative orientation of any two points should be conserved (no point in North should be farther to the South than any point in South)
- regions should be non-overlapping but adjacent
- all regions should have the same proportion of the area

The first criterion is essential because is contains the essence of the compass directions. The other criteria seem especially important for the user interface application. We would like to develop a simple, efficient algorithm which works well for most countries. With the criteria in mind, certain algorithmic problems can be formulated to find a NEWS partition of a country. They are stated in the following three suggestions. The partitionings for these suggestions are shown in Figure 1.

Suggestion 1 Compute the center of gravity of the polygon and draw two lines with slope $+1$ and -1 through this point.

Suggestion 2 Use horizontal and vertical lines to iteratively cut the polygon into four regions which each cover 25% of the polygon's area.

Suggestion 3 Divide the polygon into four equal-size regions such that the sum

$$\mathfrak{S} = dist_y(C_N, C_P) + dist_y(C_S, C_P) + dist_x(C_E, C_P) + dist_x(C_W, C_P)$$

is maximized. Here C_P denotes the center of gravity of polygon P, and C_N, C_E, C_W, C_S denote the centers of gravity of the North, East, West, and South region. $dist_x$ and $dist_y$ denote the distance by x-coordinate and by y-coordinate.

The center of gravity in Suggestion 1 can be computed in linear time. For Suggestion 2, we let the x- and y-extent determine whether we first split by horizontal or vertical lines. We can apply the algorithm of Shermer [10] three times, which gives a linear time solution.

In this paper, we will focus on finding a NEWS partitioning by an algorithm following the last suggestion. For now we will consider only the more general setting of partitioning the polygon into not necessarily simply connected NEWS regions. We will prove that a NEWS partitioning by Suggestion 3 can only lead to a partitioning with a shape described in Lemma 1.

We will denote with $\curlyvee\!\!\curlywedge$ a construction made of a vertical line in the middle and two lines with slope $+1$ and -1 at the top and at the bottom. Similarly, we denote with $\succ\!\!\prec$ the rotated shape with a horizontal line as middle part. We call the nodes where the three segments meet the **focal points**. In a $\succ\!\!\prec$ partitioning, we have a West and an East focal point. In a $\curlyvee\!\!\curlywedge$ partitioning, we have a North and a South focal point.

Lemma 1 *If an arbitrary simple polygon P is divided into four (not necessarily connected) parts, such that each part covers exactly 25% of the polygon's area and the sum $\mathfrak{S} = dist_y(C_N, C_P) + dist_y(C_S, C_P) + dist_x(C_E, C_P) + dist_x(C_W, C_P)$ is maximized, then it has inner boundaries shaped $\curlyvee\!\!\curlywedge$ or $\succ\!\!\prec$. Here C_P denotes the center of gravity of P, and C_N, C_E, C_W, C_S denote the centers of gravity of the NEWS regions. Furthermore, the partitioning is unique.*

3 NEWS partitions with arbitrary regions

In this section we give an algorithm to compute all wedges with fixed angle and orientation that contain a given area A. We also show how to compute a NEWS partitioning of a simple polygon following Suggestion 3.

Definition 1 *For a simple polygon P, we call a wedge of the form $(y \geq x + a) \cap (y \geq -x + b)$ that contains 25% of the area of P a* **North-wedge** *of P. East-, South-, and West-wedges are defined similarly.*

Definition 2 *The* **North-trace** *is the locus of all points that are apex of a North-wedge. East-, South-, and West-traces are defined similarly.*

The outline of an algorithm to compute a NEWS partitioning according to Suggestion 3 is as follows:

1. Compute the East-trace T_E and the West-trace T_W of polygon P.
2. Scan T_E and T_W simultaneously from top to bottom, to determine if there is a pair of points $p_E \in T_E$ and $p_W \in T_W$ with the same y-coordinate and p_E to the right of p_W (or coinciding) and which gives a North area of 25% of P.
3. If such a pair exists, return the $\succ\!\prec$ partitioning.
4. Otherwise, compute the $\overset{\curlyvee}{\curlywedge}$ partitioning similarly.

The algorithm can be implemented to run in $O(n^2)$ time, which will be shown next. For convenience, we rotate the polygon by 45 degrees, so that a West-wedge is now a quadrant of the form $(x \leq a) \cap (y \leq b)$.

Lemma 2 *After rotation, the West-trace is an infinite, continuous, piecewise quadratic curve consisting of* $\Theta(n^2)$ *pieces in the worst case, and any line with positive slope intersects it only once.*

We now describe the sweep algorithm that computes the West-trace for the rotated polygon. Since the algorithm is the same for any fixed area of P inside the wedge, we set $A = \text{Area}(P)/4$ and give an algorithm to compute all West-wedge positions that have area A inside the intersection of P and the wedge.

We initialize by computing a vertical line that has area A left of it. Then we determine the highest point of P left of or on this vertical line. The highest point is the first break point p_0 of the trace. The initial part of the trace is a vertical half-line down to p_0. We initialize two lists L_x and L_y with all vertices of P sorted on increasing x- and decreasing y-coordinate, starting at p_0. During the sweep we maintain: a balanced binary search tree $\mathcal{T}$ on the edges of P that intersect the sweep line, the leaf in $\mathcal{T}$ that stores the edge of P vertically below the current position of the trace, and the equation of the currently valid curve. The leaves of $\mathcal{T}$ are linked into a list. One of three event-types can occur:

1. The vertical line through the current position of the trace (the sweep line) reaches a vertex of P.
2. The horizontal line through the current position of the trace reaches a vertex of P.
3. The trace reaches the boundary of P.

The next event can be determined in $O(1)$ time from the equation of the curve, the first element in each list, and the edges of P above and below the current position. An event of type 1 requires an update of $\mathcal{T}$ and possibly the pointer to the leaf with the edge below the trace. If the vertex of P is below the position of the trace, we output the next break point and update the equation of the curve. For the second and third type of event similar actions are taken. Note that the sweep also continues if the position of the trace goes outside P. Later it may enter P again. We conclude:

Theorem 1 *Given a simple polygon P with n vertices and a value A, the set of all positions of apexes of wedges with a fixed shape and orientation that contain a portion of area A of P inside can be computed in optimal* $O(n^2)$ *time.*

Completing the NEWS partitioning by step 2 takes $O(n^2)$ time, so we can compute a NEWS partitioning for Suggestion 3 in $O(n^2)$ time. Actually the running time for computing a trace is $O(n \log n + k)$, where k is the complexity of the trace. In practice k will be much less than quadratic, so the algorithm is efficient on real world data.

We can improve this result to a worst case $O(n \log n)$ time algorithm. We do not compute the whole traces. Instead we determine by binary search and an adaptation of Shermer's algorithm [10] the two vertices of P that have consecutive $(x-y)$-values of their coordinates, and the $\succ\!\prec$ partitioning has its focal points at some $(x-y)$-value in between. Here we can compute the West- and East-trace explicitly by the sweep described above. Now we have at most $O(n)$ events of type 3.

Theorem 2 *Given a simple polygon P with n vertices, we can determine a partitioning by $\succ\!\prec$ or $\overset{\curlyvee}{\curlywedge}$ where each region contains exactly 25% of the area in $O(n \log n)$ time. The partitioning maximizes the sum of distances of centers of gravity* $\mathfrak{S} = dist_y(C_N, C_P) + dist_y(C_S, C_P) + dist_x(C_E, C_P) + dist_x(C_W, C_P)$.

4 Extensions

The three suggestions for partitioning algorithms have been implemented (see Figure 1). The results for ten different country outlines can be found in [12].

The idea of partitioning a simple polygon into four equal-size not necessarily simply-connected parts, while maximizing the sum-of-center-of-gravity distances immediately generalizes to polygons that are not simply connected. The following extensions have been studied more thoroughly, see [12]:

Fair partitioning: Find a simply-connected partition into four regions by $\succ\!\prec$ or $\curlyvee\!\curlywedge$ such that each region has 25% of the area, or decide that no such partition exists.

Maxmin: Find a simply-connected partition by $\succ\!\prec$ or $\curlyvee\!\curlywedge$ that maximizes the area of the smallest region.

Minmax: Find a simply-connected partition by $\succ\!\prec$ or $\curlyvee\!\curlywedge$ that minimizes the area of the largest region.

The fair partitioning uses a modified version of the sweep algorithm. Because of the simply-connectedness of the partition itself we can show the following:

Theorem 3 *For a simple polygon P with n vertices, we can determine in $O(n \log n)$ time if there exists a simply-connected $\succ\!\prec$ or $\curlyvee\!\curlywedge$ partitioning into four simply-connected regions that all have 25% of P's area.*

For the Maxmin and Minmax partitioning the following holds:

Theorem 4 *For a simple polygon P with n vertices, we can compute a simply-connected $\succ\!\prec$ and $\curlyvee\!\curlywedge$ partitioning that minimizes the maximum area subregion or maximizes the minimum area subregion in $O(n^3)$ time.*

Another extension is to define a center region of a country as well. By restating Suggestion 3 as partitioning a polygon into five regions, each with 20% of the area and applying the same proof ideas, we can show that the center is an axis-parallel rectangle and the other boundaries are lines with slope $+1$ or -1. It can also be useful to define a degree of North, East, West and South. This could be done simply by coordinates or by growing the center region of the polygon.

References

[1] A.I. Abdelmoty and B.A. El-Geresy. An intersection-based formalism for representing orientation relations in a geographic database. In *2nd ACM Workshop on Advances in GIS*, pages 44–51, 1994.

[2] P. Bose, J. Czyzowicz, E. Kranakis, D. Krizanc, and A. Maheswari. A note on cutting circles and squares in equal area pieces. In *Proc. FUN with Algorithms '98*, 1998.

[3] P.A. Burrough and A.U. Frank, editors. *Geographic Objects with Indeterminate Boundaries*, volume II of *GISDATA*. Taylor & Francis, London, 1996.

[4] M. Díaz and J. O'Rourke. Ham-sandwich sectioning of polygons. pages 282–286, 1990.

[5] Susan Hert. Connected area partitioning. In *Abstracts 17th European Workshop Comput. Geom.*, pages 35–38. Freie Universität Berlin, 2001.

[6] C.B. Jones, R. Purves, A. Ruas, M. Sanderson, M. Sester, M.J. van Kreveld, and R. Weibel. Spatial information retrieval and geographical ontologies – an overview of the spirit project. In *Proc. 25th Annu. Int. Conf. on Research and Development in Information Retrieval (SIGIR 2002)*, pages 387–388, 2002.

[7] R. Larson. Geographic information retrieval and spatial browsing. In L.C. Smith and M. Gluck, editors, *Geographic Information Systems and Libraries: Patrons, Maps, and Spatial Information*, pages 81–124. 1996.

[8] M. Li, S. Zhou, and C.B. Jones. Multi-agent systems for web-based map information retrieval. In *Proc. 2nd GIScience*, number 2478 in Lect. Notes in Computer Science, pages 161–180, 2002.

[9] D.J. 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.

[10] T. C. Shermer. A linear algorithm for bisecting a polygon. *Information Processing Letters*, 41:135–140, 1992.

[11] L. Talmy. How spoken language and signed language structure space differently. In *Proc. COSIT 2001*, number 2205 in Lect. Notes in Computer Science, pages 274–262, 2001.

[12] M. van Kreveld and I. Reinbacher. Good NEWS: Partitioning a simple polygon by compass directions. Manuscript, 2002.