Circles in the Water: Towards Island Group Labeling

Citation for published version (APA):
van Goethem, A. I., van Kreveld, M. J., & Speckmann, B. (2016). Circles in the Water: Towards Island Group Labeling. In J. A. Miller, D. O'Sullivan, & D. Wiegand (Eds.), *Proc. of the 9th International Conference on Geographic Information Science (GIScience)* (pp. 293-307). (Lecture Notes in Computer Science; Vol. 9927). Springer. https://doi.org/10.1007/978-3-319-45738-3_19

Document license:
Unspecified

DOI:
10.1007/978-3-319-45738-3_19

Document status and date:
Published: 01/01/2016

Document Version:
Accepted manuscript including changes made at the peer-review stage

Circles in the Water: Towards Island Group Labeling*

Arthur van Goethem[1], Marc van Kreveld[2], and Bettina Speckmann[1]

[1] TU Eindhoven, the Netherlands.
[a.i.v.goethem|b.speckmann]@tue.nl
[2] Utrecht University, the Netherlands. m.j.vankreveld@uu.nl

Abstract. Many algorithmic results are known for automated label placement on maps. However, algorithms to compute labels for groups of features, such as island groups, are largely missing. In this paper we address this issue by presenting new, efficient algorithms for island label placement in various settings. We consider straight-line and circular-arc labels that may or may not overlap a given set of islands. We concentrate on computing the line or circle that minimizes the maximum distance to the islands, measured by the closest distance. We experimentally test whether the generated labels are reasonable for various real-world island groups, and compare different options. The results are positive and validate our geometric formalizations.

1 Introduction

Map labeling is a fundamental problem in automated cartography which has received a significant amount of attention both in the GIScience and in the algorithms communities [13]. There are a variety of geometric objects to be labeled, ranging from points (representing locations such as cities), over polylines (representing linear cartographic features such as rivers) and polygons (representing areal features such as lakes), to groups of polygons (representing groups of features such as islands). The basis of all algorithmic work in this area is formed by an extensive set of cartographic guidelines which detail the properties of a high quality labeling (see [5, 8, 14]). These guidelines often lead to optimization problems which can be approached with algorithmic methods.

To generate such guidelines for groups of features, Reimer et al. [12] propose a framework of possible geometric quality measures for the labeling of feature groups. The framework includes the shape of the label, whether it may intersect the features or not, and how the distance between a label and the features is measured. Assuming that the placement of a label is optimal according to some yet unknown measure generates a number of algorithmic optimization problems. Only few of them, namely the simpler ones, can be solved directly with known

* A. van Goethem and B. Speckmann are supported by the Netherlands Organisation for Scientific Research (NWO) under project nos. 612.001.102 and 639.023.208, respectively.

methods. Reimer et al. [12] conclude that new algorithms are needed to advance the state-of-the-art in automated label placement for feature groups.

In this paper we expand on the research in Reimer et al. [12] by solving several of the algorithmic problems in label placement for groups of features. In particular, we focus on min-max distance measures that minimize the maximum distance to the closest point of each feature. This is arguably more natural than the versions that could be solved with known methods. In this setting, we consider straight-line and circular-arc labels, which may or may not intersect the features. We design new algorithms for these optimization problems, analyze their efficiency, and test implementations on island label groups to know how much the automatically generated labels resemble ones we can find in atlases, originally placed by a cartographer.

Contribution. Our input is a set S of k simple polygons $P_1, \ldots, P_k$, with n vertices in total (the *islands*). We refer to labels that are allowed to overlap islands as *general labels* and to labels that are not allowed any overlap as *water labels*. We focus on placing a single label for an island group, in isolation of other features and labels on the map.

We assume that labels are long enough that we can consider complete lines and circles instead of line segments and circular arcs (see Fig. 1). In practice this assumption may not always be true; we will see some examples in the experiments section. Finding optimal labels that are shorter is a considerably more complex placement problem, resulting in higher running times of solutions. Therefore, in this algorithmic study, we limit ourselves to the simpler case where labels span the island group. We will see that the algorithms we obtain are sufficiently complex already.

We give $O(n \log k)$ and $O(nk + k^2 \log k)$ time algorithms for general and water straight-line labels, respectively, in Section 2. For circular-arc labels we give $O(n^2)$ and $O(n^3 \operatorname{polylog} n)$ time algorithms for general and water labels, respectively in Section 3. Our solutions are inspired by free placements in motion planning, facility location, minimum-width annulus computation, and dimensional metrology. In all cases we need to carefully capture the geometry of an optimal placement to arrive at an efficient solution.

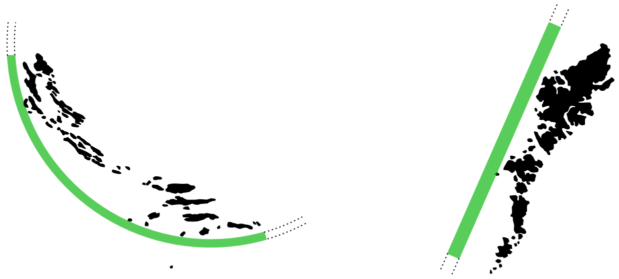

Fig. 1. We assume that labels are long enough to warrant consideration of complete lines and circles. A subsection of the line or circle functions as the final label.

In Section 4 we compare the circular-arc labels generated by our algorithms to manually placed labels for a representative set of island groups. The results are positive and often appear already as good as the manually placed labels. We are hence confident that geometric formalization can capture the implicit connection necessary to associate an island label well with its group.

In Section 5 we discuss further possible extensions of our algorithms, using the min-sum distance measure and allowing labels with a non-zero height.

Related work. Many algorithms have been developed for labeling maps, far too many to list here (for an overview see [13]). The labeling of island groups has received little algorithmic treatment so far. An exception is the work by van Kreveld and Schlechter [9] who give an algorithm that places an intersection-free label in the position minimizing the maximum distance from the label to each island of the group using horizontal straight-line labels. Since only straight labels are handled, the solution is not so general. The other exception is the work of Reimer et al. [12], who review what existing geometric algorithms can be adapted directly to island label placement.

2 Straight-line labels

In this section we show how to compute a straight-line label – either general or water – that minimizes the maximum distance to the islands. For each island, we consider the distance to its closest point. As we assume labels are long enough to be considered as complete lines, we need not compare different length labels reducing the complexity of the problem. This also implies that the distance to a label can be measured perpendicularly to the label. While this assumption restricts the type of labels we can generate, the resulting reduction of complexity of the problem allows us to formulate good polynomial time algorithms. We discuss how to find the actual label placement on this line in Section 4.1.

2.1 General labels

As we assume that labels are complete lines, the minimum distance between an island and a label can always be measured to a point on the convex hull of the

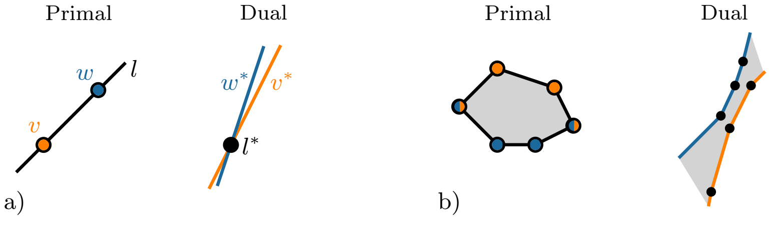

Fig. 2. (a) Two points on a line form two lines intersecting in a point in dual space. (b) The bottom and top of a convex polygon in primal space correspond to the top and bottom of a funnel in dual space.

island. Hence, to find the optimal general label, we first replace every island by its convex hull, taking $O(n)$ time in total [10]. Then we study the problem in dual space, where lines are represented as points. Each line $l = (y = mx + b)$ becomes a point $l^* = (m, -b)$ in dual space. Similarly, each point $p = (p_x, p_y)$ becomes a line $p^* = (y = p_x x - p_y)$ [4]. The k convex polygons in primal space form k funnels in dual space (see Fig. 2). The *top and bottom boundaries* of each funnel are x-monotone polylines. Funnel boundaries belonging to two different islands only intersect when a line in primal space is tangent to both islands. Between each pair of islands there exist at most four tangents, and at most two of these are tangent to any combination of "upper" and "lower" boundaries of the two islands. Thus, the funnel boundaries form a set of $2k$ x-monotone, pairwise 2-intersecting polylines with $O(n)$ vertices total.

For any fixed rotation of the label to be placed, the furthest closest vertex of an island that is below the label in primal space, is on the upper envelope of the lower boundaries in dual space. As all the funnel-boundaries are 2-intersecting, this upper envelope has complexity $O(n)$ and we can compute it in $O(n \log k)$ time by pairwise merging in $\log k$ phases. A similar argument holds for the furthest closest vertex above and the lower envelope.

If we look at a fixed rotation of the label to be placed, this corresponds to finding the optimal position in dual space on a vertical line. As dualization is distance-preserving on any vertical line, the optimal placement for any fixed rotation is exactly centered between the upper and lower envelope. Thus, the optimal solution for any rotation is located on a *centerline*, which also has $O(n)$ complexity.

For each segment s of the centerline we can compute the optimal position in $O(1)$ time. Let x_{start} be the x-coordinate of the start of s. Let d_{start} be the vertical distance to the upper- (or lower-) envelope at x_{start}. While we move along s the distance to the upper envelope changes by a linear factor c_1 in x. For a shift of δ along the x-axis, the vertical distance in dual space is $f_s(\delta) = d_{start} + c_1 \cdot \delta$. The distance to the closest point in primal space is

$$g_s(\delta) = \frac{(d_{start} + c_1 \cdot \delta)^2}{(x_{start} + \delta)^2 + 1}$$

The optimal position is at the minimum over the domain given by segment s. This minimum can be at an endpoint of s or in the middle.

We can compute the upper and lower envelope in dual space in $O(n \log k)$ time and compute the centerline and the optimum on the centerline in $O(n)$ time. Hence, we can find the optimal label in $O(n \log k)$ time.

Theorem 1. *Given a set of k islands having n vertices together, we can compute the straight-line general label optimizing the min-max distance in $O(n \log k)$ time.*

2.2 Water labels

An optimal water label need not have an equal distance to the furthest closest point on either side. Positions that have equal distance to both points may result in labels that intersect one or more islands. As a consequence, the dual of the label need not lie on the centerline in between the upper and lower envelope. In this section we describe an algorithm to compute the optimal solution for straight-line water labels in $O(nk + k^2 \log k)$ time.

Compute the arrangement. We first compute the complete arrangement of all $2k$ funnel boundaries in $O((n + k^2) \log k)$ or $O(nk)$ time.

For the first bound, make an x-sorted list V of funnel boundary vertices in $O(n \log k)$ time. Then do a standard sweep from left to right. The sweep-line intersects $O(k)$ edges simultaneously, leading to the bound.

For the second bound, take the sorted list V and split it into n/k parts such that each part has $O(k)$ vertices. These parts give rise to n/k vertical lines, such that between two vertical lines we need the arrangement of $O(k)$ line segments. We compute the full arrangement of the supporting lines in $O(k^2)$ time and then remove all parts that are on lines but not on the line segments. Then we couple consecutive arrangement parts at the vertical lines into a single arrangement. We spend $n/k \cdot O(k^2) = O(nk)$ time.

Insert the centerline. The centerline has complexity $O(n)$. As all boundaries and the centerline are x-monotone, and all boundaries are either convex or concave, each edge of the centerline can intersect at most $O(k)$ edges of the arrangement. Consequently, the centerline can cross the arrangement $O(nk)$ times. This bound is also realizable in theory.

To insert the centerline, we sort the boundaries and the centerline by increasing slope of the first segment in $O(k \log k)$ time. This uniquely defines the left-most face the centerline is in and we can access it in $O(1)$ time. For each face the centerline crosses do the following. Starting at the left-most vertex of the face-boundary, or the last traversed edges of this face, traverse the upper- and lower-boundary of the face, as well as the centerline simultaneously. To get this working, we maintain the last traversed upper and lower edge of every face, and store that with the face in the arrangement. As soon as we enter a face again, we can proceed where we left off. We traverse each edge at most twice and we make at most an additional $O(nk)$ edges by the intersections between the centerline and the arrangement. Hence, the total time complexity is $O(nk)$.

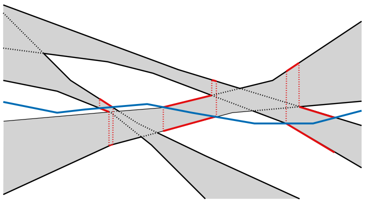

Fig. 3. Three funnels, the illegal cells (grey), the centerline (blue), and the induced closest legal edges (red).

Illegal placements. A line that intersects an island in primal space dualizes to a point inside the funnel of the island in dual space. Hence, all faces of the arrangement covered by a funnel result in labels that intersect one or more islands. Faces covered by a funnel are defined to be *illegal*. An edge of the arrangement separating two illegal faces is also *illegal*, all other edges are *legal*. When we consider the problem for a fixed rotation of the final label, the legal point vertically closest to the centerline in dual space is optimal. The closest legal points form intervals on the edges of the arrangement (see Fig. 3). These *subedges* can either be above, below, or on the centerline.

Find the closest legal subedges. We first remove all illegal edges in $O(n+k^2)$ time. Let a *chain* be a maximal sequence of edges connected by vertices of degree two. The resulting arrangement, excluding the centerline, consists of $O(k^2)$ non-intersecting chains with $O(n+k^2)$ vertices in total. We sweep this arrangement with a vertical line, maintaining the intersected edges and the centerline in a balanced binary search tree to determine the closest legal subedges. There are $O(nk)$ possible intersections with the centerline which we can resolve in $O(1)$ time and $O(k^2)$ vertices at which chains start and end requiring $O(\log k)$ time each to update. Hence, we can find all closest legal subedges above and below the centerline in $O(nk + k^2 \log k)$ time.

The closest point to the centerline may be on the centerline or on the closest legal subedge above or below it. In total we get $O(nk)$ legal subedges and legal edges of the centerline, which we can evaluate in $O(1)$ time each.

Theorem 2. *Given a set of k islands having n vertices together, we can compute the straight-line water label optimizing the min-max distance in $O(nk + k^2 \log k)$ time.*

3 Circular-arc labels

In this section we consider the problem of computing a circle that minimizes the maximum distance to the islands, measured by the distance to the closest point of each island. We approach the problem by computing an annulus of minimum width that touches all islands. The circle in the middle of this annulus is the required solution. The optimal annulus problem has four degrees of freedom as we can express a solution by the coordinates of the center and two radii.

The minimum-width annulus problem has been studied before for point sets [2, 4], and can be solved in $O(n^{3/2+\epsilon})$ time (where $\epsilon > 0$ is a constant that can be chosen). Instead of a point set, we have a set of simple polygons, and the annulus must intersect at least one point of each. We define the *inner circle* as the boundary of the annulus with the smaller radius and the *outer circle* as the boundary with the larger radius. A *contact* is a vertex of the input touching either of the circles, or an edge of the input that is tangent to the outer circle, see Fig. 4. An edge of the input tangent to the inner circle is not a contact, because it cannot contribute to defining the optimal annulus.

3.1 General labels

We begin with the case of general, unrestricted circular-arc labels, which is the minimum-width annulus problem where the annulus must touch every island. Similar to the minimum-width annulus problem for points, any solution having less than four contacts cannot be optimal (proof omitted).

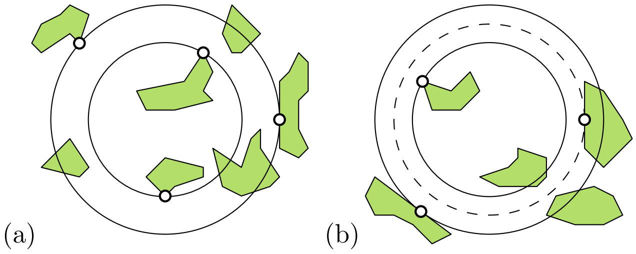

Fig. 4. (a) Annulus touching all islands determined by four contacts (marked), two on the inner circle and two on the outer circle. (b) Annulus partially determined by three contacts with a middle circle that does not intersect any island.

Lemma 3. *The annulus corresponding to the optimal min-max, circular-arc label is determined by at least four contact points.*

The solution to the minimum-width annulus problem for points described in [4] uses the Voronoi Diagram and the Farthest-point Voronoi Diagram to determine the annuli with four contacts that may be the optimal one. The center of each such annulus must lie on a vertex of one of these diagrams, or on an edge-edge intersection of the diagrams. It yields an $O(n^2)$ time algorithm.

We use a similar approach, but need different diagrams. As the annulus is required to overlap or touch each island, the outer radius of the optimal annulus is defined by the closest point of the furthest polygon. Similarly, the inner radius is defined by the polygon with the closest furthest point. We make use of two matching Voronoi Diagrams.

The Farthest-Polygon Voronoi Diagram (FPVD) [6] subdivides the plane in $O(n)$ regions with total complexity $O(n)$. In each region, the same polygon is furthest *and* the same feature (vertex or edge) of that polygon is closest (see Fig. 5(a)). Cheong et al. [6] show that it can be computed in $O(n \log^3 n)$ time.

The Hausdorff Voronoi Diagram (HVD) [7, 11] subdivides the plane in a set of regions, with total complexity $O(n)$. In each region, the same polygon is closest *and* the same feature of that polygon is furthest (see Fig. 5(b)). Here the

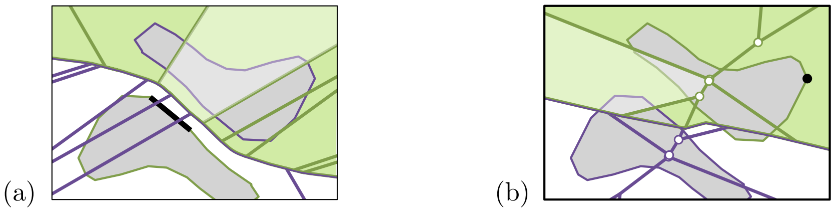

Fig. 5. (a) In each cell of the FPVD the same polygon is furthest and the same feature closest. (b) In each cell of the HVD the same polygon is closest (measured to the furthest point) and the same feature is furthest.

distance between a point x and a polygon P is defined as $\max_{p \in P} d(x, p)$. Hence, the closest polygon is the polygon with the closest furthest point. This diagram can be computed in $O(n \log^4 n)$ time [7].

The optimal annulus must have its center on a vertex of the FPVD, on a vertex of the HVD, or on an intersection of two edges, one of each diagram. For annuli with their center on a vertex we observe that the diagrams have only $O(n)$ vertices, and for each we can determine the width of the corresponding annulus in $O(n)$ time, leading to $O(n^2)$ time for this case. For annuli with their center on an edge-edge intersection we observe that all edges are either straight or parabolic and each diagram has $O(n)$ edges. Thus, there can be at most $O(n^2)$ intersections. For each intersection we know the distance to the inner and outer contacts, and we can determine the width of each annulus in $O(1)$ time.

Theorem 4. *Given a set of islands with n vertices together, we can find a circle minimizing the maximum distance to the closest point of any island in $O(n^2)$ time.*

3.2 Water labels

We next consider the problem of computing a circle that misses all islands and minimizes the maximum distance to them, measured by the distance to the closest point of each island. Phrased in terms of computing an annulus, we want to compute an annulus of minimum width touching every island. For this annulus the additional requirement holds that the *middle* circle, located halfway between the inner- and outer-circle, should not intersect any island.

As before we begin by characterizing properties of an optimal solution by contacts of the three co-centric circles involved and the islands. We can no longer show that there are always four contacts, or two contacts with the inner or outer circle, and hence we cannot use the edges of the FPVD and HVD any longer. Instead we will cover all cases where the annulus is restricted by at least three contacts.

Lemma 5. *An optimal annulus A is of one (or more) of the following types:*

(i) *A has the contacts as if it were an unrestricted annulus (four contacts on outer and inner circle together);*
(ii) *A has at least two contacts on the outer circle and one on the middle circle;*
(iii) *A has at least two contacts on the inner circle and one on the middle circle;*
(iv) *A has at least one contact on the outer circle, one on the inner circle and one on the middle circle.*

With three contacts, the annulus can move its center while retaining these contacts, because there is still one degree of freedom remaining. We can optimize over this degree of freedom and find all locally optimal annuli. As we check all local optimal annuli we will also check the global optimum solution.

This characterization of optimal annuli gives rise to an algorithmic solution. As a tool we will use a data structure that allows us to test whether a query circle intersects any island, which we describe first. We treat the islands as a set of n line segments and observe that a circle does not intersect any island if and only if it does not intersect any line segment (ignoring the case where the circle lies fully inside an island, which can easily be handled).

Lemma 6 (from [3]). *A line segment s intersects a circle C if and only if:*

(i) *exactly one of the endpoints of s lies inside C, or*
(ii) *both endpoints of s lie outside C, the center of C lies in the perpendicular strip of s, and the supporting line of s intersects C.*

The perpendicular strip of s is the strip bounded by the two lines perpendicular to s and each through one endpoint of s (see Fig. 6). Using the lemma we can design a data structure that stores n line segments in a multi-level tree so that for any query circle, we can efficiently decide if it intersects a line segment. Combining the theory of [1] with Lemma 6, and using cutting trees instead of partition trees, we get the following:

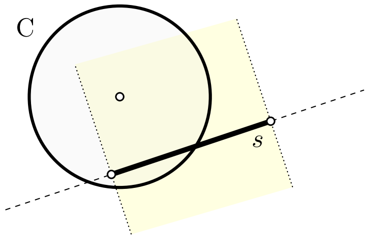

Fig. 6. A line segment s, its supporting line (dashed), its perpendicular strip (yellow), and an intersecting circle C.

Lemma 7. *A set of n line segments can be stored in a data structure of size and preprocessing time $O(n^3 \operatorname{polylog} n)$, such that for any query circle, we can decide in $O(\operatorname{polylog} n)$ time whether it intersects any line segment of the set, where $\operatorname{polylog} n$ stands for $\log^k(n)$ for some constant k.*

We compute an optimal annulus as follows. First, build the data structure T on the n line segments bounding the islands. Second, use the results of the general case to compute $O(n^2)$ annuli. For each such annulus, determine the middle circle and query with it in T to decide if it intersects any island. If not, it is a candidate solution. Third, take any triple of features of the islands and assume them to be contacts. Choose each of the cases (ii), (iii) and (iv) from Lemma 5 and each assignment of contacts for that case.

To treat any of the $O(n^3)$ choices, compute the loci of centers of the corresponding annulus, and the corresponding function describing the widths of these annuli. This is a one-parameter function because we have fixed three out of four degrees of freedom, and we compute it in $O(1)$ time because only three contacts (features of the islands) are involved. We optimize the width function, finding all $O(1)$ local optima. Each gives rise to an annulus, which we test with our data structure T to see if the middle circle intersects any island.

Since we will be testing $O(n^3)$ circles by querying T, and the construction of T takes $O(n^3 \operatorname{polylog} n)$ time, we obtain:

Theorem 8. *Given a set of islands with n vertices together, we can find a circle that does not intersect any island and minimizes the maximum distance to the closest point of any island in $O(n^3 \operatorname{polylog} n)$ time.*

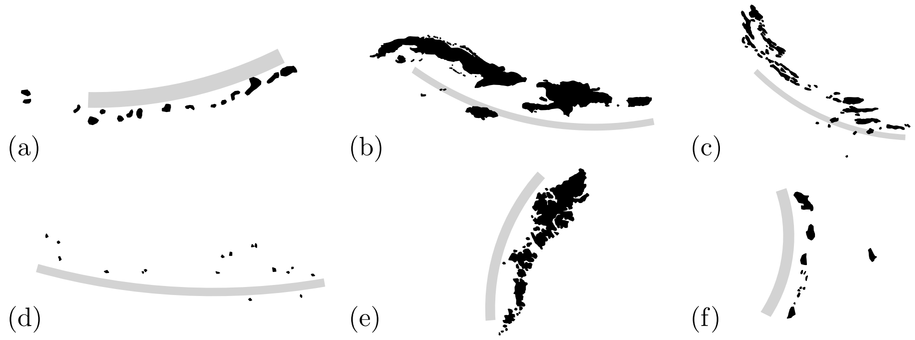

Fig. 7. Digitized island groups and their manual label (grey). (a) Aleutian Islands - Clear arc. (b) Antilles - Different island sizes. (c) Dalmatian Islands - Arc with spread. (d) Caroline Islands - Wide spread. (e) Outer Hebrides - Closely interwoven islands. (f) Windward Islands - Group with outlier.

4 Experiments

We evaluate the quality of the labels generated by our algorithms by performing both a visual analysis and a comparison in numbers with manually placed labels.[1]

4.1 Setup

To determine the quality of the computed labels we compare them to manually generated labels. We use digitized island groups from several atlases together with their respective labels. We selected six candidates to represent a wide range of possible island configurations (see Fig. 7).

In these experiments we focus on circular arc labels as they are more commonly present in maps. Nevertheless, the straight-line labels generated by our algorithms give good results (see Fig. 8) that may directly be used as label positions. The rarity of straight-line labels for island groups makes them less suitable for a direct evaluation though.

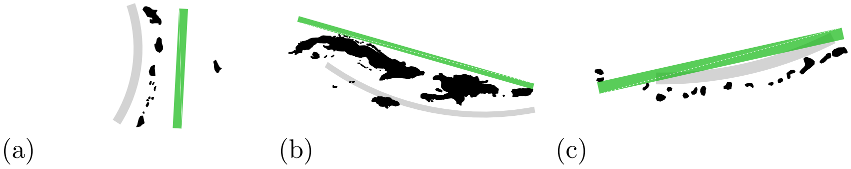

Fig. 8. Island groups with their original, manual label (grey) and the computed label (green). Straight-line water labels under different points of measurement. (a) All points. (b) Centroid. (c) Closest point.

[1] Further results are available online: `http://www.win.tue.nl/~agoethem/labeling`, May, 2016

Besides the straight-line and circular-arc labels discussed in this paper, we also tested other possible optimization measures. Specifically, we tested optimal straight-line and circular-arc labels, using the min-max distance measure, when distance is measured to the centroid or to the complete area of each island. The latter is in fact equivalent to measuring distance to the furthest points. The optimal label placements for these settings can be computed using variations on the techniques described in this paper. To ensure we obtained realistic solutions we computed label positions having the same height as the original manual label. The discussed algorithms can easily be extended to take this into account (see Section 5).

Finally, our algorithms compute only complete lines and circles for label placement. These should still automatically be converted to line segments and circular arcs. We project all vertices of the island onto the computed line (/circle). The final label is the minimal line segment (/circular arc) that contains all projected vertices. As a consequence the computed label position best spans the width of the island group. We note that the actual label may have a different length, but any length label can trivially be computed from this.

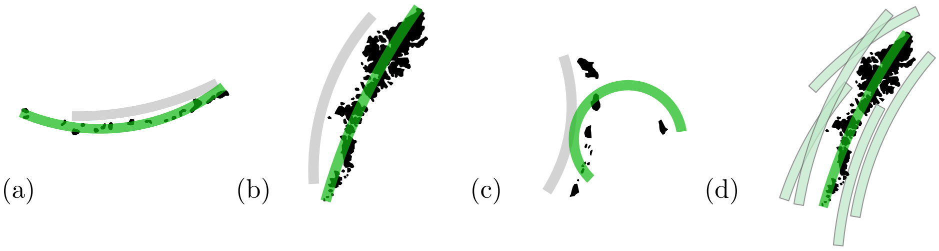

Fig. 9. Island groups with their original, manual label (grey) and the computed label (green). General labels under different points of measurement. (a) Closest point. (b) All points. (c) Closest point. (d) Candidate generation may use general labels to compute several possible high-quality label positions (manual example).

4.2 Visual Inspection

General labels. We observe that the computed labels capture the shape of the island group well when the island group forms a coherent and clear shape (see Fig. 9(a-b)). The label overlaps many islands and is less suitable for label placement, but is a good basis for label candidate generation. By slightly changing the radius and center of the arc we can generate many labels capturing the shape of the group (see Fig. 9(d)).

The min-max distance measure is outlier-sensitive. This is no problem in most groups, but outliers may cause unexpected results (see Fig. 9(c)).

In general, we notice that the effect of the point of measure is small for the min-max measure. When the group contains large islands, however, the effect becomes more prominent. In Fig. 10 the same island group is shown (Antilles) for the three different points of measure. For large islands the closest point, farthest point, and centroid may give significantly different results.

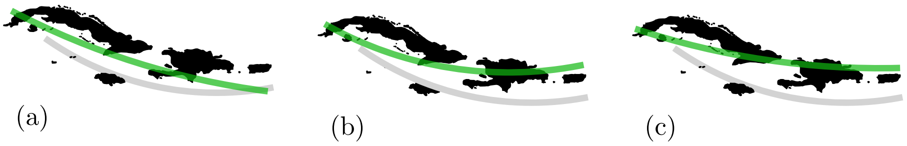

Fig. 10. Antilles islands with the manually placed label (grey) and the computed label (green) for different points of measure: (a) All points. (b) Centroid. (c) Closest point.

Water labels. Generally, water labels give good results (see Fig. 11(a-c,e)). Often the label is located outside the island group following the general shape. When the label is placed inside the group, the same holds for the manual label (see Fig. 11(c-d)).

We observe that, as expected, preventing any overlap between the label shape and the islands may be overly restrictive. In Fig. 11(d) two tiny islands prevent the label from following a more natural shape. Future work may investigate whether it is possible to extend the algorithms to allow a small amount of overlap. A relatively small amount of overlap may be acceptable if this causes a large increase in the quality of the label.

The min-max measure is less outlier sensitive for water labels. The width of the group compensates for the outliers (see Fig. 11(e-f)). As the label cannot intersect any island its shape is more restricted and it is harder for outliers to affect it.

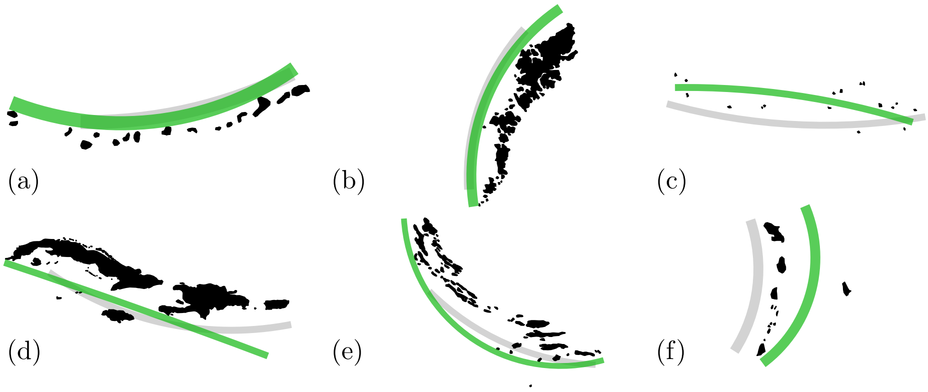

Fig. 11. Island groups with their original, manual label (grey) and the computed label (green). Water labels under different points of measurement. (a) All points. (b) Closest point. (c) Closest point. (d) Centroid. (e) Centroid. (f) Closest point.

4.3 Comparison

To support our visual inspection we also perform a comparison in the line of the research outlined by Reimer et al. [12]. They suggest to analyze what criteria cartographers may subconsciously use when placing island labels, by comparing manually placed island labels from atlases with automatically generated labels according to geometric criteria. For example, suppose we consider the setting with water labels and min-max distance for circular labels, measured to the closest point of each island. If an analysis reveals that cartographers place island labels within a small percentage of what is possible in this setting, then it is likely that cartographers apply this criterion either explicitly or subconsciously. Note that the manually placed label and the automatically generated label may be very different even if cartographers realize nearly the same score on the measure.

For each measure and each setting we compute the label placement that minimizes the distance to the island group and the corresponding distance. We also compute the distance, according to this measure, from the manual label to the island group. For fair comparison we measure distance to the circle concentric to the manual label. The more correlated the distance of the manual label and optimal label position are, the more likely it is that the manual label was placed (subconsciously) according to the given measure. In Table 1 an overview of all distance measures (normalized to the minimal distance) is given for circular arc labels using the min-max distance. We make some observations.

First, the requirement that labels are strictly non-overlapping reduces our ability to optimize the given metrics. Consequently, the manual label for the Antilles is generally ‘better’ than the optimal water label. Second, the outlier sensitivity of the min-max distance measure is reflected in the ‘quality’ of the manual label for the Windward Islands, which appears to ignore a further removed island. Consequently, the manual label is a poor fit in the min-max distance measure. Finally, surprisingly, this preliminary test suggests that the

Circular-arc Min-Max Label

	General Label			Water Label		
	Centroid	All	Closest	Centroid	All	Closest
Aleut. Isl.	3.39	2.61	9.53	1.04	0.97	1.18
Antilles	1.42	1.51	1.57	0.93	0.84	1.07
Carol. Isl.	1.71	1.73	1.78	1.68	1.66	1.72
Dalm. Isl.	1.34	1.40	1.65	1.01	1.02	1.02
Hebrides	2.54	2.44	3.03	1.09	1.11	1.10
Wind. Isl.	3.80	3.81	6.35	2.57	2.35	2.87

Table 1. Distance of the manual label to the island group for the different settings (normalized in comparison to the optimal label placement for that setting). For each combination of island group and label type (general or water), the distance of the label that minimizes its respective measure best is underlined. Note that the manual label may overlap the island group causing labels that are better than “optimal”.

closest point of each island may not be as important as the furthest point or centroid. We should take some care in interpreting these measures though, as the closest point measure is also most easily influenced by a small change in the label shape.

5 Discussion and Future Work

The time bounds we obtain are worst-case time bounds for optimal label placement. They show that optimal placement can be done at least this efficiently. The quadratic and cubic runtime bounds imply that we cannot expect interactive label placement for high-detail island groups. This need not be a problem as the geometry of the problem is likely only marginally affected by the exact details of the island shapes, and line simplification can be applied in preprocessing to reduce the input size. Alternatively, the algorithms could be adapted so that the cubic running time does not show up for realistic inputs.

We sketch two possible extensions of our algorithms and investigate the possibilities for future work.

Min-sum distance. In Section 2 we presented two algorithms for straight-line general and water labels using the *min-max* distance. Both can be extended to the *min-sum* distance as follows. An optimal straight-line label must have equally many islands placed to either side. Hence, for a fixed rotation, the optimal (general) solution can be found in $O(n \log k)$ time. The optimal water label is placed at the legal position closest to the above solution.

In both cases there always exists an optimal solution tangent to an island. Thus, in dual space it is located on an edge of the arrangement. We compute the arrangement in $O((n + k^2) \log k)$ time and detect the faces having an equal number of islands above and below it. We traverse the arrangement and update the required information in constant time per face, resulting in $O((n + k^2) \log k)$ time to find the optimal general and water label.

Non-zero-height labels. For general labels, using a non-zero-height does not change the solution. To place a water label of height h, we can offset all islands by $h/2$ and compute a zero-height label. The vertices of the islands in primal space, however, become circular arcs. Consequently, in dual space we have an arrangement of curves. We can still compute the arrangement, the centerline, and the closest legal segments in $O(nk + k^2 \log k)$ time. A similar approach works for the min-sum distance measure.

Future work. We would like to lift the restriction that the full line or circle must be free for water labels; the part used by the text would be enough. However, the degrees of freedom in the problem increase, and it is unclear whether sufficiently efficient algorithms exist.

Alternatively, it would be interesting to see if we can relax the requirement that a water label cannot have any overlap with the islands. Allowing the label to overlap an outermost strip of fixed width for each island could easily be integrated in our approach. To achieve this, checks of intersection with the middle annulus circle could simply be done with a negatively offset version of the islands. When we instead would require that the label has a maximum amount of overlap in total over all islands, finding the optimal water label is non-trivial.

We note that labeling bodies of water and channels is similar to labeling island groups. In contrast to island group labeling though, we try to place the label in the middle of the body of water or channel. We may achieve this by maximizing the distance to the nearest points outside the body of water, while avoiding any islands. Hence, a max-min distance measure (maximizing the minimum distance) may be more applicable.

Acknowledgments

Thanks to Sarah Lohr, Susanne Heuser, and Andreas Reimer for providing us with the digitized maps.

References

1. P. Agarwal and J. Erickson. Geometric range searching and its relatives. *Contemporary Mathematics*, 223:1–56, 1999.
2. P. Agarwal and M. Sharir. Efficient randomized algorithms for some geometric. optimization problems. *Discrete & Computational Geometry*, 16(4):317–337, 1996.
3. P. Agarwal, M. van Kreveld, and M. Overmars. Intersection queries in curved objects. *Journal of Algorithms*, 15(2):229–266, 1993.
4. M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry.* Springer, 3rd edition, 2008.
5. C. Brewer. *Designing Better Maps: A Guide for Gis Users.* ESRI Press, 2005.
6. O. Cheong, H. Everett, M. Glisse, J. Gudmundsson, S. Hornus, S. Lazard, M. Lee, and H.-S. Na. Farthest-polygon Voronoi diagrams. *Computational Geometry: Theory & Applications*, 44(4):234–247, 2011.
7. F. Dehne, A. Maheshwari, and R. Taylor. A coarse grained parallel algorithm for Hausdorff Voronoi diagrams. In *Proc. of the International Conference on Parallel Processing*, pages 497–504, 2006.
8. E. Imhof. Positioning names on maps. *American Cartographer*, 2(2):128–144, 1975.
9. M. van Kreveld and T. Schlechter. Automated label placement for groups of islands. *Proc. of the 22th International Cartographic Conference*, 2005.
10. A. Melkman. On-line construction of the convex hull of a simple polyline. *Information Processing Letters*, 25(1):11–12, 1987.
11. E. Papadopoulou and D. Lee. The Hausdorff Voronoi diagram of polygonal objects: a divide and conquer approach. *International Journal of Computational Geometry & Applications*, 14(6):421–452, 2004.
12. A. Reimer, A. van Goethem, M. Rylov, M. van Kreveld, and B. Speckmann. A formal approach to the automated labelling of groups of features. *Cartography and Geographic Information Science*, 2015. `http://dx.doi.org/10.1080/15230406.2015.1053986`.
13. A. Wolff and T. Strijk. A map labeling bibliography, 2009. `http://i11www.iti.uni-karlsruhe.de/~awolff/map-labeling/bibliography/`.
14. C. Wood. Descriptive and illustrated guide for type placement on small scale maps. *Cartographic Journal*, 37(1):5–18, 2000.