UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

Time- and Space-Optimal Contour Computation for a Set of Rectangles

Data Structuring Group
Peter Widmayer
Derick Wood

CS-86-35

September, 1986

Time- and Space-Optimal Contour Computation for a Set of Rectangles *

Peter Widmayer †and Derick Wood ‡

September 5, 1986

Abstract

We present the first time- and space-optimal algorithm for the problem of computing the contours of the disjoint polygons defined by the union of n rectangles in the plane. It requires $O(nlogn + e)$ time and $O(n)$ space, where e is the total number of edges in the contour cycles. The space optimality of the solution is demonstrated by way of a combinatorial argument.

1 Introduction

Given a set of n rectangles in the plane with edges parallel to the coordinate axes (isothetic rectangles), we reconsider the problem of computing and reporting the contour of the union of these rectangles. Much work has already been done on this topic, see [2,3,4,6,8,10]. The best results available until now are:

- $O(nlogn + e)$ time and $O(n)$ space algorithms for computing all edges of the contour, where e is the number of edges in the contour, see [2,10]. These can be modified to give an $O(n + e)$ space algorithm for computing the contour cycles. Note that we do not count the space used for the final output, but only space for the intermediate storage of edges that are to be used again by the algorithm. In [8], it has been

*This work was supported by a Natural Sciences and Engineering Research Council of Canada Grant A-5692 and by a grant from the Deutsche Forschungsgemeinschaft Wi 810/1-1

†Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Universität Karlsruhe, Postfach 6980, 7500 Karlsruhe, West Germany

‡Data Structuring Group, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

noticed that only $O(n)$ space need be main memory; $O(e)$ space can be secondary storage.

- an $O(n^2)$ time and $O(n)$ space algorithm, see [6].

We bridge the gap between these two solutions, by providing a solution which is time optimal, that is, requires $O(nlogn + e)$ time and is also space optimal, that is, it requires $O(n)$ space. This not only resolves an open problem, but the proof technique also confirms an empirical conjecture due to Prusinkiewicz and Raghavan which arises from their work in [6], that $O(n)$ space suffices.

2 Time and Space Optimality

We show that two of the above time optimal algorithms, namely those in [2] and [10], can be modified such that they actually report the contour cycles in optimal time and space. To this end, it is sufficient to keep track of all contour cycles that have been started but not yet ended, in a manner similar to the one described in [5].

Let us refer to the plane sweep algorithm of [10] in order to demonstrate briefly how this can be done. This algorithm reports the horizontal edges of the contour; the vertical edges can be deduced from them. We modify the algorithm so that in addition to the plane sweep for detecting contour edges, (tentative) contour cycles are maintained. We distinguish between *outer* or *boundary contour cycles* and *inner contour cycles*, that is the contours of holes in the following. Each tentative, incomplete contour cycle is represented by two points on the sweep line. We assume the edges associated with these points are in the usual cyclic order; clockwise for outer contours and counterclockwise for inner contours. Left vertical edges of rectangles inaugurate outer contours, while right vertical edges inaugurate inner contours. When the sweep line meets a left vertical edge, it may close or partially close contour cycles, see Figure 1. When it meets a right vertical edge, it may combine two contour cycles or widen contour cycles, see Figure 2. When a vertical edge closes a contour cycle, the cycle is finished and can be output (reported) in its entirety. A simple implementation of tentative contour cycles as doubly linked lists of edges ensures that the optimal time bound of $O(nlogn + e)$ is maintained. For the details of keeping track of the (tentative) contour cycles for a similar problem, consult [5].

The storage used by this procedure is clearly bounded by $O(n+c)$, where c is the total number of edges in all contour cycles that have been started, but have not yet been finished, at any stage of the plane sweep. The above algorithm is space optimal if and only if $c = O(n)$. Our contribution in this note is to show that $c = O(n)$ in fact holds. We first prove that the union of

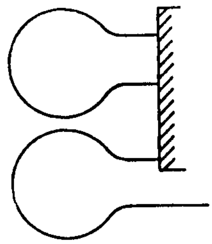

Figure 1: Meeting a left vertical edge

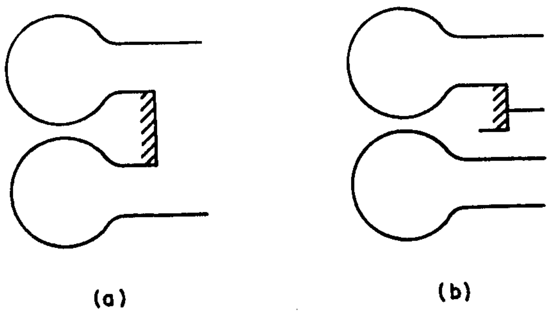

Figure 2: Meeting a right vertical edge

a set of rectangles has a linear number of edges in its outer contour. This is similar to a result in [4], for a set of rectangles without holes in their union.

Theorem 2.1 *Given n isothetic rectangles in the plane, the outer contour of their union contains $O(n)$ edges.*

Proof: The outer contour of the union has as many vertices as edges and each connected component has an outer boundary which is a closed curve. Consider one component consisting of m rectangles, say. Since its outer boundary is a closed curve, the number of reflex vertices is four less than the number of convex vertices. But the number of convex vertices is at most $4m$, since each rectangle can contribute at most four convex vertices and convex vertices are formed in no other way. Thus there are at most $4m - 4$ reflex vertices and therefore at most $8m - 4$ vertices or edges in the component. Clearly over all components there are at most $8n - 4$ vertices or edges, and this bound is achievable, see [4] □

We now prove the needed combinatorial result.

Theorem 2.2 *Given n isothetic rectangles in the plane, the total number of all edges in contour cycles intersecting any vertical or horizontal line is $O(n)$.*

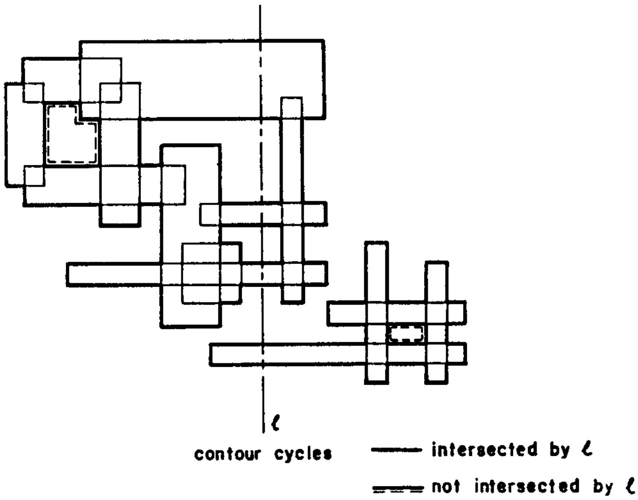

Figure 3: Contour cycles intersected by a line

Proof: Let us argue for each connected component of the set of rectangles separately. Consider a connected component of m rectangles. Cut each rectangle intersected by vertical line l into two rectangles at l, see Figure 3. Now consider the two subsets of rectangles to the left of l and to the right of l separately. Each of these subsets consists of no more than m rectangles. All edges of intersected contour cycles now lie on the outside contour of one of the two subsets. The total number of edges in both outside contours may be even higher than the number of edges in the intersected contour cycles. This is because some of the contour cycle edges may have been split into two, and intersected rectangle edges may have been added to the outside contours. For each subset, the number of edges on the outside contour is linear in the number of rectangles by the above proposition. Hence, for the connected component consisting of m rectangles, we obtain $O(m)$ edges in all intersected contour cycles. As this holds for each connected component, the proof of the theorem is complete. □

Note that the contour cycles such as outer contours and holes are reported by the described plane sweep algorithm from left to right, according to their rightmost point. This means that, in general, the holes belonging to one connected component, and the outer contour of that connected component, are not necessarily reported consecutively. They may be interspersed with other components' holes and outer contours. If it is desired that all contour cycles of a connected component be reported consecutively, it is sufficient to first find all connected components, and then compute the contour cycles for the components separately. Because the connected components can be found in $O(nlogn)$ time (see [1]), optimality is maintained.

Thus we obtain

Theorem 2.3 *Given n isothetic rectangles in the plane, the contour of their union can be computed in $O(nlogn + e)$ time and $O(n)$ space, where e is the number of edges in the contour. Moreover in the same bounds contour cycles corresponding to each connected component can be reported separately such that their outer contour cycle is followed by their hole contour cycles.*

Proof: The first claim follows from the above arguments, so let us turn to the second. First, we use the algorithm of [1] to obtain the connected components of the rectangles. Second, we process each component to obtain its outer contour and, third, we then process each component separately to obtain its inner contour cycles. Since the outer contour cycles require only $O(n)$ space, we can afford to save them for the third step. Hence we output them either before or after their inner contour cycles to obtain the desired result. Clearly these steps can be accomplished in the stated space and time bounds. □

These results are both time-and space-optimal.

Observations similar to the above one can be applied to a number of other problems advantageously. For other examples, see [9]; a more basic treatment of similar arguments can be found in [7].

References

[1] H. Edelsbrunner, J. van Leeuwen, Th. Ottmann, and D. Wood. Computing the connected components of simple rectilinear geometrical objects in d-space. *RAIRO Informatique théoretique*, 18:171–183, 1984.

[2] R.H. Güting. An optimal contour algorithm for iso-oriented rectangles. *Journal of Algorithms*, 5:303–326, 1984.

[3] R.H. Güting. Optimal divide-and-conquer to compute measure and contour for a set of iso-rectangles. *Acta Informatica*, 21:271–291, 1984.

[4] W. Lipski and F.P. Preparata. Finding the contour of the union of a set of iso-oriented rectangles. *Journal of Algorithms*, 1:235–246, 1980.

[5] J. Nievergelt and F.P. Preparata. Plane-sweep algorithms for intersecting geometric figures. *Communications of the ACM*, 25:739–747, 1982.

[6] P. Prusinkiewicz and V.V. Raghavan. *A Simple Space-Optimal Contour Algorithm for a Set of Iso-Rectangles.* Technical Report CS-83-03, University of Regina, Computer Science Dept., 1983.

[7] G.J.E. Rawlins, P. Widmayer, and D. Wood. *Hole Problems for Rectangles in the Plane.* Technical Report, University of Waterloo, Computer Science, 1986.

[8] T.G. Szymanski and C.J. Van Wyk. Space efficient algorithms for VLSI artwork analysis. In *Proceedings of the 20th IEEE design automation conference*, pages 734–739, 1983.

[9] P. Widmayer and D. Wood. *A Time– and Space–Optimal Algorithm for Boolean Mask Operations for Orthogonal Polygons.* Technical Report, Universität Karlsruhe, 1986.

[10] D. Wood. The contour problem for rectilinear polygons. *Information Processing Letters*, 19:229–236, 1984.