JOURNAL OF ALGORITHMS 1, 235–246 (1980)

Finding the Contour of a Union of Iso-Oriented Rectangles*

WITOLD LIPSKI, JR.,† AND FRANCO P. PREPARATA

Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Urbana, Illinois 61801

Received August 13, 1979; revised January 3, 1980

Let $R_1, \ldots, R_m$ be rectangles on the plane with sides parallel to the coordinate axes. An algorithm is described for finding the contour of $F = R_1 \cup \ldots \cup R_m$ in $O(m \log m + p \log(2m^2/p))$ time, where p is the number of edges in the contour. This is $O(m^2)$ in the general case, and $O(m \log m)$ when F is without holes (then $p \leq 8m - 4$); both of these performances are optimal.

1. INTRODUCTION

In this paper we consider the following geometric problem. Let $R_1, \ldots, R_m$ be rectangles in the plane with sides parallel to the coordinate axes. The union $F = R_1 \cup \ldots \cup R_m$ may consist of one or more disjoint connected components, and each component may or may not have internal holes (note that a hole may contain in its interior some connected components of F). The *contour* (boundary) of F consists of a collection of disjoint cycles composed of (alternating) vertical and horizontal edges. By convention, any edge is directed in such a way that we have the figure on the left while traversing the edge; this is equivalent to saying that a cycle is oriented clockwise if it is the boundary of a hole, and counterclockwise if it is an external boundary of a connected component. Given $R_1, \ldots, R_m$, our task is to find the contour of $F = R_1 \cup \ldots \cup R_m$.

There are several applications which involve iso-oriented rectangles, or, without loss of generality, rectangles whose sides are parallel to the coordinate axes. Suffice it to mention its relevance to Very Large Scale

†On leave from Institute of Computer Science, Polish Academy of Sciences, P.O. Box 22, 00-901 Warsaw PKiN, Poland.

*This work was supported in part by the National Science Foundation under Grant MCS 78-13642 and in part by the Joint Services Electronics Program under Contract N00014-79-C-0424.

0196-6774/80/030235–12$02.00/0

Integration [5], to geography and related areas [3], to computer graphics (hidden-line problems, shadows, etc.), and to the organization of data in two-dimensional magnetic bubble memories [6]. There is also considerable theoretical interest in problems related to the one discussed in this paper, such as finding the measure of the union of rectangles [2], testing a set of rectangles for disjointness and reporting all the intersections [4], etc.

The paper is organized as follows. In Section 2, we informally describe the algorithm, with the aid of an example. Then, in Section 3, we give the details of the main data structure which is crucial in an efficient implementation of our algorithm, and we describe the implementation of the basic operations performed on this data structure. Section 4 contains a final description of the whole algorithm, and the analysis of its performance, including a lower bound which proves the optimality of the algorithm when the figure F has no holes.

2. Informal Description of the Algorithm

We informally illustrate the method with the aid of an example. The algorithm consists of two phases. In the first phase we find the set V of vertical edges of the contour (edges 1 through 10 in Fig. 1); in the second phase we link these vertical edges by means of horizontal edges to form the oriented cycles of the contour.

In order to obtain the set V we scan the abscissas corresponding to vertical sides of rectangles from left to right. At a generic abscissa c, the section $\mathcal{S}$ of the vertical line $x = c$ with F is a disjoint union of intervals. This section remains constant between two consecutive vertical sides of the rectangles, and is updated in the scan each time one such side is reached. If s is a vertical side of some rectangle R at abscissa c, the portion of the contour of F contributed by s is given by $s \cap \bar{\mathcal{S}}$, where $\bar{\mathcal{S}}$ is, as usual, the

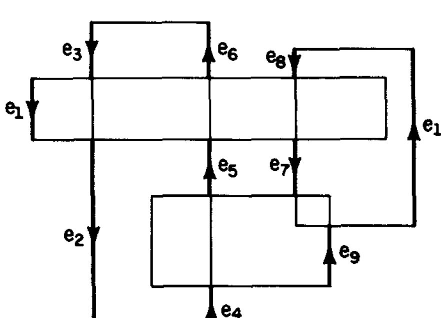

FIG. 1. An instance of the problem.