The Contour Problem for Restricted-Orientation Polygons

Diane L. Souvaine†

Department of Computer Science, Rutgers University, New Brunswick, New Jersey 08903

Abstract

The specifications received by a fabrication house for the design of a VLSI chip consist of a set of polygons, each assigned to a particular layer of the chip. In general, the polygons are far from disjoint. In addition, many edges may intersect in a single point or in a segment. The fabrication house must form the union of the polygons on each layer to determine the contour of the layer. Further processing is performed on the contour. The fabrication houses would prefer to restrict designers to rectilinear polygons for efficiency of processing. The actual chips, however, gain efficiency by minimizing wasted space. Thus designers would prefer the freedom to choose an unlimited number of directions. A compromise of some fixed number of orientations more than two, but definitely less than infinity, may be the solution.

Using two approaches, we study the problem of finding the contour of the union of a collection of polgyons with the following characteristics: 1) all vertices have integer coordinates; 2) the slopes of the sides belong to a finite fixed collection of orientations. First, which algorithms for finding the contour of the union for rectilinear polygons and/or rectangles can be generalized to handle polygons with sides of two or more additional directions while still retaining efficiency? Second, which algorithms for computing the union of arbitrary polygons can be revised to take advantage of the restricted number of orientations, improving robustness, efficiency, or both?

Three rectilinear algorithms (Lipski-Preparata, Guting, and Wood) sweep a vertical line across the plane, stopping only at x-coordinates of the endpoints of the input edges and keeping a representation of the current cross-section in a segment tree or variant thereof. The cross-section of the union of the polygons remains unchanged throughout the interval between adjacent stopping points. These algorithms are robust: all vertices of the contour have integer coordinates and neither multiple collinear edges nor multiple intersections need be problematic. Furthermore, the time complexities depend only on the number of input edges and the number of edges on the final contour, and not the number of intersections between pairs of input edges.

A number of contour algorithms for arbitrary polygons can be created using line sweep and methods for computing line-segment intersections. Unfortunately, the running times of the general algorithms do depend on the number of input edge intersections. Furthermore, robustness becomes an issue: intersection points have arbitrary rational coordinates, subject to round-off error; the algorithms may presume "general position" where in practice numerous edges share an intersection point and many are collinear.

Rectilinear algorithms can be refined to handle more directions, albeit with a penalty in time and space complexity. The restriction to a fixed number of directions can lead to improvements in the robustness and efficiency of general algorithms. The full paper includes both theoretical analysis of these algorithms and empirical observations.

---

†Partially supported by National Science Foundation grant #CCR-88-03549, by a Henry Rutgers Research Fellowship, and by IBM.