The Area Bisectors of a Polygon and Force Equilibria in Programmable Vector Fields*

Karl-Friedrich Böhringer† — Cornell University
Bruce Randall Donald‡ — Cornell University
Dan Halperin§ — Tel Aviv University

Abstract

We consider the family of area bisectors of a polygon (possibly with holes) in the plane. We say that two bisectors of a polygon P are *combinatorially distinct* if they induce different partitionings of the vertices of P. We show that there are simple polygons with n vertices that have $\Omega(n^2)$ combinatorially distinct area bisectors (matching the obvious upper bound), and we present an output-sensitive algorithm for computing an explicit representation of all the bisectors of a given polygon. Our study is motivated by the development of novel, flexible feeding devices for parts positioning and orienting. The question of determining all the bisectors of polygonal parts arises in connection with the development of efficient part positioning strategies when using these devices.

1 Introduction

Let P be a polygon in the plane, possibly with holes, and having n vertices in total. We denote by V the set of vertices of P. For a directed line λ in the plane, we denote by $h_l(\lambda)$ (resp. $h_r(\lambda)$) the open half-plane bounded by λ on the left- (resp. right-) hand-side of λ. The line λ is an area bisector of P if the area of $P \cap h_l(\lambda)$ is equal to the area of $P \cap h_r(\lambda)$.

A line λ partitions V into three sets (two of which may be empty): $V \cap h_l(\lambda)$, $V \cap \lambda$, and $V \cap h_r(\lambda)$. We say that two area bisectors of P are *combinatorially distinct* if the partitioning of V as above induced by the two bisectors is different. We say that two area bisectors of P are *combinatorially equivalent* if they induce the same partitioning of V. We assume that the polygon P is connected, and non-degenerate in the sense that the complement of P has the same boundary as P.

An obvious upper bound on the number of distinct area bisectors of a polygon with n vertices is $O(n^2)$—each combinatorial equivalence class of area bisectors is determined by a pair of vertices of the polygon. In Section 4 we show that a polygon with n vertices can have $\Omega(n^2)$ distinct area bisectors. (Note that the polygon in our construction is *simple.*)

We devise an output-sensitive algorithm for computing an explicit representation of all the area bisectors of a given polygon, by constructing the *bisector curve* β defined in a plane dual to the plane containing the polygon: the curve β is the union of points dual to area bisectors in the primal plane. Our algorithm proceeds by constructing the *zone* of the curve β in an *arrangement of lines* [5] in the dual plane, where each line of the arrangement is the dual of a vertex of the polygon. A sketch of the algorithm is given in Section 5.

Area bisectors were considered by Díaz and O'Rourke [4]. However, their focus is on the continuous version of the *ham-sandwich cut* problem, and of a problem they introduce of *orthogonal four-sections*; see [4] for more details. The problem that we study here can be viewed as a continuous version of the well-known *k-set* problem [5].

2 Motivation: Programmable Vector Fields

Programmable vector fields can be used to control a variety of flexible planar part feeders. These devices often exploit exotic actuation technologies such as arrayed, microfabricated motion pixels [2] or transverse vibrating plates [1]. These new automation designs promise great speed, flexibility, and dexterity—we believe they may be employed to orient, singulate, sort, and feed parts. However, since they have only recently been invented, programming and controlling them for manipulation tasks is challenging.

In [3], we devise a technique for analyzing programmable vector fields called *equilibrium analysis*, lower bounds (i.e., impossibility results) on what the devices *cannot* do, and results on a classification of control strategies yielding design criteria by which well-behaved manipulation strategies may

*Work on this paper by K.-F. Böhringer and B. R. Donald has been supported in part by NSF under grants No. IRI-8802390, IRI-9000532, IRI-9201699, and by a Presidential Young Investigator award to B. Donald, in part by NSF/ARPA Small Grant for Exploratory Research No. IRI-9403903, and in part by the Air Force Office of Sponsored Research, the Mathematical Sciences Institute, Intel Corporation, and AT&T Bell labs. Work on this paper by D. Halperin has been supported by an Alon Fellowship and by ESPRIT IV LTR Project No. 21957 (CGAL).

†Robotics & Vision Lab, Dept. of Comp. Sc., Cornell Univ. Current address: ALPHA Lab, Dept. of Ind. Eng. and Op. Research, Univ. of CA, Berkeley. Email: karl@ieor.berkeley.edu.

‡Robotics & Vision Laboratory, Department of Computer Science, Cornell University. Email: brd@cs.cornell.edu.

§Dept. of Comp. Sc., Tel Aviv Univ., Tel Aviv 69978, ISRAEL. halperin@math.tau.ac.il. Part of the work was carried out while D. H. was at the Robotics Lab, Dept. of Comp. Sc., Stanford University.

13th ACM Symposium on Computational Geometry, Nice, France, June 1997.

be developed. Equilibrium analysis is a fundamentally geometric problem. To illustrate, let us make the following assumptions. Suppose that (1) Each motion pixel is very small relative to the part we wish to manipulate, and hence a dense array of pixels may be modeled as a 2-dimensional vector field. (2) The part to be manipulated is essentially laminar. (3) The part is connected.

A squeeze field is defined by making all the actuators push normally towards a common *squeeze line.* In [3], we show that under certain dynamic and mechanical assumptions, we can compute a sequence of squeeze fields guaranteed to bring any part into unique moment equilibrium. No sensing is required, but a clock is required to switch between fields.

In order to be in equilibrium, the forces and moments must balance. Force equilibrium, under our assumptions, is equivalent to: *the squeeze line must be an area bisector of the polygon.* The number of equilibria critically impacts both the complexity of plan generation, and plan size. Therefore, it is important to bound the number of equilibria for a given part or class of parts. In [3] we show that every convex part has at most $O(n)$ equilibria, and every non-convex part has $O(rn^2)$ equilibria (r is the maximum number of edges of P intersected by any line), under squeeze fields.

In this paper, we show how to analyze the number of area bisectors, and hence bound the number of force equilibria. Our algorithm for computing area bisectors can be used as a preliminary step in designing alignment plans (see Section 6).

3 Properties of Area Bisectors

In this section we state several properties of area bisectors of polygons. The proofs can be found in the full version of the paper.[1]

Lemma 3.1 *Let P be a non-degenerate polygon with n vertices. (1) There exist $O(n^2)$ combinatorially distinct ways in which a line can partition P. (2) Let A and B be the intersections of an area bisector λ with the boundary of the convex hull of P. As the slope of λ varies from $-\infty$ to $+\infty$, A and B progress monotonically counterclockwise on the boundary of the convex hull of P. (3) For every slope $\bar{x}$ there exists a unique bisector λ of P with slope $\bar{x}$.*

Lemma 3.2 *Let P be a polygon with n vertices. Let s be a point in $\mathbb{R}^2$ and let λ be a line that intersects r edges of P. The area bisectors of P that are combinatorially equivalent to λ and pass through s are determined by the roots of a polynomial equation of degree r.*

In the full version of the paper we show that the bisectors of a polygon P can be described by a piece-wise algebraic curve, where each piece is described by a polynomial whose degree depends on the number of edges of P intersected by the corresponding bisectors.

4 Lower Bound

As argued above, a polygon with n vertices can have at most $O(n^2)$ combinatorially distinct area bisectors. In this section we give an example of a simple polygon with n vertices where the bound $\Omega(n^2)$ is attained.

Consider Figure 1. All the vertices v_i, v_i', u_i and u_i' lie on a circle whose center is at c. The vertices w_j lie very close to c on a small circle whose center is c as well, along two convex polygonal chains.

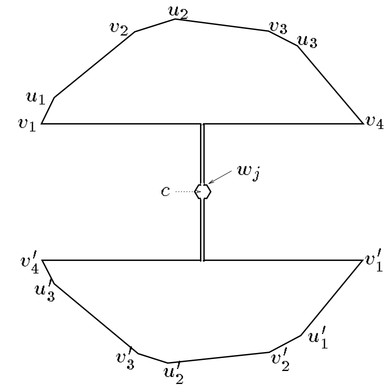

Figure 1: A simple polygon with n vertices that has $\Omega(n^2)$ combinatorially distinct bisectors.

We fix an integer m (that we will determine later; for the polygon in the figure $m = 3$). The distance between the vertices v_i and v_{i+1} is the same for $i = 1, \ldots, m$, and it is the same as the distance between v_i' and v_{i+1}' for $i = 1, \ldots, m$. The area of all triangles $v_i u_i v_{i+1}$ for $i = 1, \ldots, m$ is the same and is equal to the area of all triangles $v_i' u_i' v_{i+1}'$ for $i = 1, \ldots, m$. There are $2m$ vertices w_j near c and they are equally spaced on a small circle centered at c. As can be easily verified, for every pair of vertices v_i and v_i', there is a bisector passing through these points that passes also through the center point c. We next claim that as we rotate the bisector from v_i to v_{i+1} it will move off the center c and sweep m vertices w_j. The reason is that the angle $\angle u_i v_i v_{i+1}$ is larger than the angle $\angle u_i' v_i' v_{i+1}'$. Hence, as the bisector rotates, it will proceed 'faster' on the bottom part of our polygon than on the top part and therefore will sweep half of the vertices w_j on its way. Finally m is chosen such that (roughly) $n = 6m + 8$. The number of distinct area bisectors is evidently $\Omega(m^2) = \Omega(n^2)$.

5 Output-Sensitive Algorithm

It is convenient to study the algorithmic problem in a dual plane: a line $y = 2\bar{x}x - \bar{y}$ in the primal plane is transformed into the point $(\bar{x}, \bar{y})$ in the dual plane. A point (x, y) in the primal plane is transformed into the line $\bar{y} = 2x\bar{x} - y$ in the dual. The dual of an object o will be denoted by o^*. If O is a set of objects in the plane, O^* will denote the set of dual objects.

Let P be a polygon with n vertices as defined in the Introduction, namely connected, non-degenerate and possibly with holes. In the dual plane every vertex v of P is transformed into a line v^* which is the collection of all points dual to lines in the primal plane that pass through v.

For any given direction there is a unique area bisector. We denote the oriented bisector of P that makes an angle θ with the positive x-axis by $B(\theta)$, and (because of symmetry) confine ourselves to the range $[-\pi/2, \pi/2)$ for θ. We denote the collection of points dual to area bisectors of P in that range by β. Note that any θ (besides $-\pi/2$) corresponds to an $\bar{x}$-coordinate in the dual plane.

The curve β is a piece-wise algebraic and $\bar{x}$-monotone curve (this is proved in the full version of the paper). We

[1] Forthcoming as a Technical Report, Computer Science Department, Cornell University.

13th ACM Symposium on Computational Geometry, Nice, France, June 1997.

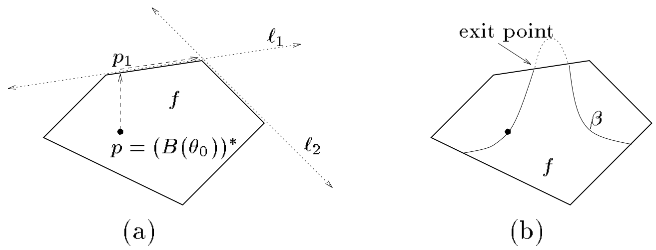

Figure 2: Ray shooting to determine the face f containing p (a), and then finding the maximal pieces of β inside f and its exit points from f (b).

call β the *bisector curve* of P, as it gives a complete specification of all the area bisectors of the polygon P. We denote by κ the number of maximal connected algebraic pieces in β, where the function describing each piece is defined by the fixed set of edges that the corresponding set of bisectors cross. In this section we describe an output-sensitive algorithm to compute β. Since we aim for output-sensitivity, we cannot afford to compute the entire arrangement $\mathcal{A}(V^*)$ whose complexity may be $\Omega(n^2)$. We will discover the maximal pieces of β in their order along β, using two primitive operations: *ray shooting* among the lines V^*, and intersection of an algebraic curve with a straight line.

We choose an arbitrary direction $\theta_0 \in [-\pi/2, \pi/2)$ and look for the area bisector of P in that direction. This requires $O(n \log n)$ time since the polygon may have holes. Next, we obtain the set of edges crossed by $B(\theta_0)$. We denote by $E(\theta)$ the set of edges crossed by $B(\theta)$. The set $E(\theta_0)$ determines a function $\rho := \rho(\theta_0)$ describing the bisector curve β in a neighborhood of θ_0, as long as the set of edges crossed by the bisector does not change. In the dual plane the function ρ describes the curve β as long as we do not leave the face of $\mathcal{A}(V^*)$ which contains the point $p := (B(\theta_0))^*$.

Our next step is to construct the face f that contains the point p in $\mathcal{A}(V^*)$. We describe this procedure assuming f is bounded; the extension to unbounded faces is straightforward. We prepare in advance a data structure $R(V^*)$ that supports efficient ray shooting among the lines V^*. We shoot a ray from p in the upward vertical direction and identify the line $\ell_1 \in V^*$ supporting the edge of f above p. See Figure 2(a). We next proceed in clockwise direction along the boundary of f. From the point $p_1 \in \ell_1$ we shoot a ray in $\mathcal{A}(V^*)$ along ℓ_1, and identify the line ℓ_2 supporting the next edge on the boundary of f, and so on until we have returned to ℓ_1 and thus have identified the entire face f.

Now we determine the maximal connected pieces of $f \cap \beta$. For each edge on boundary of f we compute the intersection of its supporting line v^* with β. This computation is equivalent to finding the bisectors that pass through the vertex v, and intersecting the edges in $E(\theta_0)$. By Lemma 3.2, this reduces to solving a polynomial equation of degree r, where r is the number of edges in $E(\theta_0)$. We denote the time required to find these roots by $\psi(r)$.

We order the resulting intersections along the $\bar{x}$-axis. Since the curve β is $\bar{x}$-monotone, this ordered list of intersections provides a description of the curve β inside f, and indicates what are the neighboring faces that β crosses. We mark each of these additional faces by the point where β crosses out of f. We call each such point an *exit point.* See Figure 2(b).

Since f has already been constructed, we know for each exit point of β the line that contains it. Therefore we can construct each new face using ray shooting queries and proceed as above. We keep a data structure that describes all the faces of $\mathcal{A}(V^*)$ that have already been constructed so that we do not construct the same face twice.

The algorithm stops when we have identified all the intersection points of β with lines in V^*, and so we have also identified the *zone* of β in $\mathcal{A}(V^*)$, namely all the face of $\mathcal{A}(V^*)$ crossed by β.

Further details on the algorithm can be found in the full version of the paper. We summarize the algorithmic result in the following theorem.

Theorem 5.1 *Let P be a non-degenerate polygon (possibly with holes) with n vertices, and such that any line crosses at most r edges of P. For any $\varepsilon > 0$ we can find a complete specification of the area bisectors of P in time $O(\kappa^{2/3} n^{2/3+\varepsilon} + \kappa\alpha(\kappa)\psi(r))$, where κ and $\psi(r)$ are as defined above, and $\alpha()$ is the functional inverse of Ackermann's function. If P is rectilinear, then the algorithms runs in time $O(\kappa^{2/3} n^{2/3+\varepsilon})$. The space required by the algorithm is $O(\kappa^{2/3} n^{2/3+\varepsilon})$.*

6 Moment Equilibria and Alignment Plans

As described in Section 2, bisectors correspond to force equilibria of P in a squeeze field. For *total* equilibrium, the moment acting on the part has to be taken into account as well. In particular, not all of the force equilibrium configurations will be moment equilibria. For each maximal piece b of the bisector curve β there exists only a finite number of moment equilibria (we omit the proof here):

Lemma 6.1 *Let P be a polygon whose interior is connected. Let λ be a bisector of P that intersects r edges of P. There exist $O(r)$ lines λ' that are combinatorially equivalent to λ such that P is in total equilibrium when λ' coincides with the center line of a squeeze field.*

It follows that a squeeze field induces a finite number of total equilibria on a polygonal part P. In [3] we show how to exploit this finiteness property to automatically generate *alignment plans* that bring P into a unique (up to symmetry) orientation, by cascading squeeze fields that systematically reduce the possible orientations of P.

References

[1] K.-F. Böhringer, V. Bhatt, and K. Y. Goldberg. Sensorless manipulation using transverse vibrations of a plate. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1989 – 1996, Nagoya, Japan, May 1995.

[2] K.-F. Böhringer, B. R. Donald, and N. C. MacDonald. Single-crystal silicon actuator arrays for micro manipulation tasks. In *Proc. IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, San Diego, CA, Feb. 1996.

[3] K.-F. Böhringer, B. R. Donald, and N. C. MacDonald. Upper and lower bounds for programmable vector fields with applications to MEMS and vibratory plate parts feeders. In J.-P. Laumond and M. Overmars, editors, *Algorithms for Robotic Motion and Manipulation.* A. K. Peters, Wellesley, MA 02181, 1997.

[4] M. Díaz and J. O'Rourke. Ham-sandwich sectioning of polygons. In *Proc. 2nd Canadian Conference on Computational Geometry*, pages 98–101, Ottawa, 1990.

[5] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science.* Springer Verlag, Heidelberg, Germany, 1987.

13th ACM Symposium on Computational Geometry, Nice, France, June 1997.