Feasibility of the Exact Geometric Computation Paradigm for Largest Empty Anchored Cylinder Computation in the Plane

Stefan Schirra*

Abstract

In the largest empty anchored cylinder problem one is looking for a ray anchored at the origin that maximizes the minimum (weighted) distance to a given set of points. For a set of n points in the plane, Follert et al. presented an optimal $O(n \log n)$ algorithm. We analyze the algebraic degree of the computations involved in their algorithm and show that it is much smaller than it looks at first sight. Indeed, a deliberate implementation of the optimal algorithm surprisingly avoids irrational number computations at all. Thus efficient exact geometric computation becomes feasible for this problem.

1 Introduction

The goal in exact geometric computation [12] is to let the implementation make correct decisions in order to have the same control flow as in its theoretical counterpart published in a research paper. The exact geometric computation approach is appealing because it avoids redesigning an algorithm such that it takes numerical imprecision into account. Furthermore it allows for transferring the correctness proof to the actual code.

Exact geometric computation has been successfully applied to many geometric problems, in particular in the software libraries CGAL [4] and LEDA [9]. Thanks to floating-point filters and related techniques exact geometric computation has been proven to be practical and efficient for many problems in linear geometry where all numerical computations involve rational arithmetic only (see [6, 11, 13] for surveys on such techniques). Exact geometric computation is viable even if irrational algebraic numbers come into play. For example, using constructive separation bounds [2, 10] degeneracies can still be correctly detected [1, 8]. However, the algebraic degree of the numbers involved in a geometric computation has a strong impact on the efficiency of exact geometric computation, because the separation bounds for an arithmetic expression are exponential in the algebraic degree of the expression [2, 10].

In Section 3 we analyze the algebraic degree of largest empty anchored cylinder computation in the plane. In this problem, we are given a set S of n (weighted) points and we are looking for a ray anchored at the origin such that the minimum (weighted) distance to the points is as large as possible, cf. Fig. 1. Follert et al. [5] show that without loss of generality we may assume that all points have unit weights. They present an algorithm that solves the problem in optimal $O(n \log n)$ time. Using the results from Section 3 we show in Section 4 that their algorithm can be implemented carefully such that irrational arithmetic does not arise. In view of the description of the optimal algorithm in [5] this is astonishing.

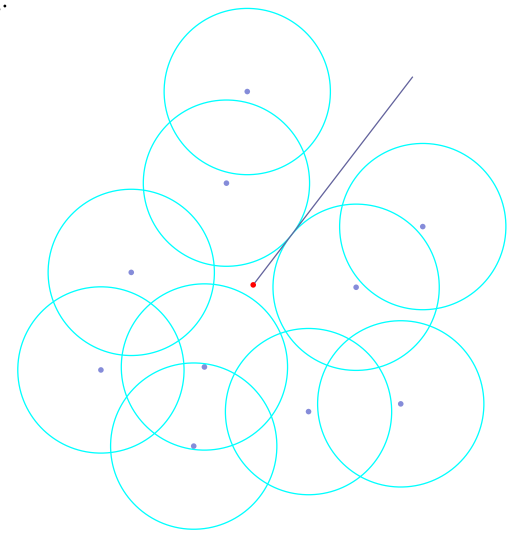

Figure 1: An optimal ray for a set of points with unit weights and circles centered at the points with radius equal to the minimum distance to the ray.

2 The algorithm

Follert et al. [5] separately solve the largest empty anchored cylinder problem for rays emanating to the right and rays emanating to the left and reduce both subproblems to lower envelope computation. In the sequel we consider the subproblem for rays emanating to the right. Let dist(,) denote Euclidean distance and let o denote the origin. Furthermore, let $\delta_{\min} = \min\{\text{dist}(o, p) : p \in S\}$. Follert et al. represent a ray by the angle $\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ between the ray and the positive x-axis. In order to stay within the algebraic computation tree model they identify a ray with

*Department of Simulation and Graphics, Faculty of Computer Science, Otto von Guericke University Magdeburg, Germany. stschirr at isg.cs.uni-magdeburg.de

angle φ with $\sin\varphi \in [-1,1]$. For $\varrho \in [-1,1]$ let $r(\varrho)$ be the corresponding ray.

In [5], for each $p \in S$ the function $l_p : [-1,1] \to \mathbb{R}$ with $\varrho \mapsto l_p(\varrho) = \min(\mathrm{dist}(p, r(\varrho)), \delta_{\min})$ is defined and their lower envelope L_S is considered. Follert et al. show that an optimal ray corresponds to a highest point on L_S. Furthermore, they show that the lower envelope forms a Davenport-Schinzel sequence of order two and hence has size $O(n)$. Follert et al. use a divide-and-conquer approach as described in [7] to compute the lower envelope in $O(n \log n)$ time.

We slightly modify their approach as follows. For $p \in S$ let $D_p \subset [-1,1]$ be the set of values ϱ where $r(\varrho)$ is tangent to the circle with center p and radius $\mathrm{dist}(p, r(\varrho))$. For $p \in S$ and $\varrho \in [-1,1]$ we define $\hat{l}_p(\varrho) = \mathrm{dist}(p, r(\varrho))$ if $\varrho \in D_p$ and $\hat{l}_p(\varrho) = \infty$ otherwise. Furthermore we define $l_{\delta_{\min}}$ with $l_{\delta_{\min}}(\varrho) = \delta_{\min}$ for all $\varrho \in [-1,1]$. Instead of the lower envelope L_S of the functions in $\mathcal{F}(S) = \{l_p : p \in S\}$ we consider the lower envelope $\hat{L}_S$ of the functions in $\hat{\mathcal{F}}(S) = \{\hat{l}_p : p \in S\}$. We have

Lemma 1 *L_S is the lower envelope of $\hat{\mathcal{F}}(S) \cup \{l_{\delta_{\min}}\}$.*

As a direct consequence of Lemma 1 we get

Lemma 2 *Let (ϱ_0, δ_0) be a highest point on $\hat{L}_S$. Then either (ϱ_0, δ_0) or $(\varrho_0, \delta_{\min})$ is a highest point on L_S.*

Thus a highest point on $\hat{L}_S$ corresponds to an optimal ray. Since each pair of functions from $\hat{\mathcal{F}}(S)$ intersects at most twice, $\hat{L}_S$ has size $O(n)$, too, and can be computed by a divide-and-conquer approach in $O(n \log n)$ time as well.

3 The algebraic degree

Follert et al. [5] neither give explicit formulas for the functions in $\mathcal{F}(S)$ nor explicit formulas for their intersection points. Instead of that they give explicit formulas for the endpoints of the subintervals $J_p(\delta)$ of $[-1,1]$ corresponding to rays having distance at most δ to $p = (p_x, p_y) \in S$ for $\delta \in [0, \delta_{\min}]$. They show that $J_p(\delta)$ is

$$\left[\frac{p_y\sqrt{p_x^2+p_y^2-\delta^2} - p_x\delta}{p_x^2+p_y^2}, \frac{p_y\sqrt{p_x^2+p_y^2-\delta^2} + p_x\delta}{p_x^2+p_y^2}\right]$$

if $\delta \le p_x$, that $J_p(\delta)$ is

$$\left[\frac{p_y\sqrt{p_x^2+p_y^2-\delta^2} - p_x\delta}{p_x^2+p_y^2},\ 1\right]$$

if $0 \le p_x \le \delta$ and $p_y \ge 0$, and that $J_p(\delta)$ is

$$\left[-1,\ \frac{p_y\sqrt{p_x^2+p_y^2-\delta^2} + p_x\delta}{p_x^2+p_y^2}\right]$$

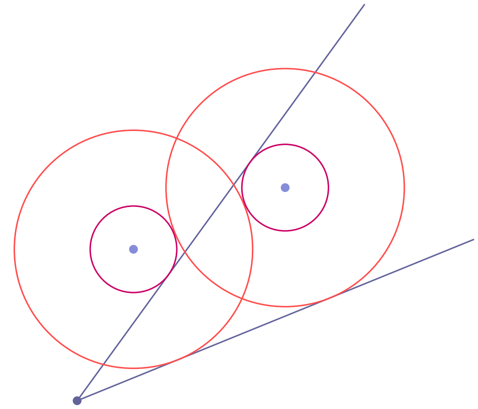

Figure 2: Bitangent rays.

if $0 \le p_x \le \delta$ and $p_y \le 0$. Similar formulas involving square-root operations are given for the case $p_x \le 0$. These formulas from [5] suggest that the intersection points of two functions in $\mathcal{F}(S)$ have irrational, algebraic coordinates in general. However, it is much better than it looks. The rays corresponding to the intersection points of two functions in $\hat{\mathcal{F}}(S)$ are rational! A ray corresponding to an intersection between $\hat{l}_p$ and $\hat{l}_q$ is tangent to circles with the same radius centered at p and q, cf. Fig. 2. We call such a ray a *bitangent* ray. In the system of polynomial equations below, the unknowns X, Y, and Δ represent x- and y- coordinate of the tangency point at the circle centered at p and the squared radius of the circle, respectively. The tangency point at the circle centered at q is $(\lambda X, \lambda Y)$, where λ is another unknown.

$$\begin{aligned}
X^2 - 2\,Xp_x + p_x^2 + Y^2 - 2\,Yp_y + p_y^2 &= \Delta \\
\lambda^2X^2 - 2\,\lambda\,Xq_x + q_x^2 + \lambda^2Y^2 - 2\,\lambda\,Yq_y + q_y^2 &= \Delta \\
X^2 - Xp_x + Y^2 - Yp_y &= 0 \\
\lambda\,X^2 - Xq_x + \lambda\,Y^2 - Yq_y &= 0
\end{aligned}$$

The first and the second equation reflect the fact that the tangency points lie on circles with squared radius Δ and centers p and q respectively. The last two equations reflect the fact that the lines through the circle centers and the tangency points must be orthogonal to the ray. Note that a solution $(X_0, Y_0, \lambda_0, \Delta_0)$ of this polynomial system corresponds to a bitangent ray only if $\lambda_0 \ge 0$. With an appropriate term order, a Gröbner basis computation (cf. [3]) for this system yields a univariate polynomial of degree 2 whose roots are the solutions for X:

$$(p_x^4 + 2\,p_x^2p_y^2 + 2\,q_y^2p_x^2 - 2\,p_x^2q_x^2 - 8\,q_yp_xp_yq_x + p_y^4 + 2\,q_y^2q_x^2 + q_y^4 + q_x^4 - 2\,p_y^2q_y^2 + 2\,p_y^2q_x^2)X^2 + (-2\,p_x^5 - 4\,p_x^3p_y^2 - 2\,q_y^2p_x^3 + 4\,p_x^3q_x^2 - 2\,p_xp_y^4 - 2\,p_xq_x^4 + 2\,p_xp_y^2q_y^2 - 2\,p_xq_y^2q_x^2 + 2\,q_yq_xp_y^3 - 2\,q_y^3q_xp_y - 2\,q_x^3p_yq_y + 10\,p_yq_xq_yp_x^2)X + p_y^2q_x^2q_y^2 - p_y^4q_x^2 + 2\,q_x^3p_yq_yp_x + p_x^6 - 2\,p_yq_xq_yp_x^3 + p_x^2p_y^4 + p_x^2q_x^4 - p_x^2p_y^2q_y^2 - 2\,p_x^2p_y^2q_x^2 + 2\,p_x^4p_y^2 - 2\,p_x^4q_x^2$$

The roots of this quadratic polynomial are rational!

$$X_{1,2} = \frac{(p_x \pm q_x)(p_y^2 \pm p_y q_y + p_x^2 \pm p_x q_x)}{(p_x \pm q_x)^2 + (p_y \pm q_y)^2}$$

Thus the x-coordinate of a tangency point with the circle with center p is rational. Because of symmetry, the y-coordinate is rational as well:

$$Y_{1,2} = \frac{(p_y \pm q_y)(p_y^2 \pm p_y q_y + p_x^2 \pm p_x q_x)}{(p_x \pm q_x)^2 + (p_y \pm q_y)^2}$$

Since the coordinates of the tangency points for p are rational, the squared radii, which are squared distances between tangency point and center p, are rational, too. Because of symmetry again, the same holds with respect to q. The bitangent rays have rational direction vectors (X_1, Y_1) and (X_2, Y_2), respectively. We have

Lemma 3 *The coordinates of the tangency points of bitangent rays are rational.*

That means all bitangent rays have a rational direction vector. Canceling common factors of X_i and Y_i we simply get rational direction vectors $(p_x \pm q_x,\ p_y \pm q_y)$. Alternatively, we get these rational direction vectors by looking at the distance

$$\zeta_p(m) = \frac{|p_y - p_x m|}{\sqrt{1+m^2}}\ ,$$

of p to the non-vertical line $Y = mX$ through o with slope m. An intersection of $\hat{l}_p$ and $\hat{l}_q$ corresponding to a non-vertical bitangent ray corresponds to an intersection of $\zeta_p(m)$ and $\zeta_q(m)$. Observe that both $\zeta_p(m)$ and $\zeta_q(m)$ have factor $(\sqrt{1+m^2})^{-1}$ in common. Thus, in order to compute the slopes at the intersections, it suffices to compute the slopes at the intersections of the remaining piecewise linear factors $|p_y - p_x m|$ and $|q_y - q_x m|$.

Instead of representing a bitangent ray by $\varrho \in [-1, 1]$ as suggested in [5] we represent it by a rational direction vector. In the next section, we show that based on the rationality of direction vectors the lower envelope and highest point computation for $\hat{L}_S$ can be implemented solely using rational arithmetic.

4 The implementation

By Lemma 2 we can find an optimal ray by computing the lower envelope $\hat{L}_S$ and a highest point on it. We represent a lower envelope by a sequence of points $\langle p_1, \ldots, p_m \rangle$ with $p_i \in S \cup \{o\}$, where o represents a section where the lower envelope is ∞ and $p \in S$ represents a section contained in D_p, where $\hat{l}_p$ is on the lower envelope. Each change of representing points either corresponds to a bitangent ray or an endpoint of some D_p. The lower envelope is computed using divide-and-conquer. The essential task is to merge two lower envelopes $L_{S_1} = \langle u_1, \ldots, u_m \rangle$ and $L_{S_2} = \langle v_1, \ldots, v_k \rangle$ into $L_{S_1 \cup S_2} = \langle w_1, \ldots \rangle$. This is done by simultaneously sweeping along the three lower envelopes, similar to the merge step of merge sort.

We start with a few observations. First observe that a ray corresponding to an endpoint of D_p has rational direction orthogonal to the vector $p - o$. Next observe that an intersection between $\hat{l}_p$ and $\hat{l}_q$ is transversal unless p, q and o are collinear. If they are collinear the function of the point further away from o is never below the function of the point closer to o. Therefore, the former function does not contribute to the lower envelope and the corresponding point can be ignored. So we may assume that all intersections between two functions in $\hat{\mathcal{F}}(S)$ are transversal. Furthermore observe that from the relative positions of a bitangent ray r and its defining points p and q and $\mathrm{dist}(p, o)$ and $\mathrm{dist}(q, o)$, we can deduce which of the points p and q comes closer to r, if r is rotated counterclockwise. If the points are on different sides of the line supporting r, the point on the left comes closer; if both points are on the left, the point with larger distance to o becomes the one closer to r, and if both are on the right, the point closer to o comes closer to r by counterclockwise rotation. Since we know a rational point on the bitangent ray, sideness can be determined by an orientation test involving points with rational coordinates only. We can make use of this observation in opposite ways: (1) to elect the bitangent ray if we know which of its defining points comes closer by counterclockwise rotation, and (2) to detect which point comes closer, if we know the bitangent ray. Finally remember, that the squared distance between a bitangent ray and its defining points is rational. Thus we can compare squared (minimum) distances to bitangent rays using rational arithmetic.

Assume we are going to merge $\langle u_1, \ldots, u_m \rangle$ and $\langle v_1, \ldots, v_k \rangle$. Initially we have to find the first point w_1 representing the new lower envelope $\langle w_1, w_2, \ldots \rangle$. If $u_1 = v_1 = o$ we have $w_1 = o$. If exactly one of the points p_1 and u_1 is o, w_1 is the point different from o. If both are different from o, we have to compare their distance to the ray pointing along the negative y-axis. In this case both u_1 and v_1 have non-positive y-coordinate and the distance to the ray is just the absolute value of their x-coordinate. So rational arithmetic suffices to compute the initial point w_1.

Next assume that we are currently sweeping a part labeled u_i on the first lower envelope and a part labeled v_j on the second lower envelope. We must identify the next event of the sweep. This event will correspond to the change from u_i to u_{i+1} on the first envelope or to the change from v_j to v_{j+1} on the second envelope or to an intersection between $\hat{l}_{u_i}$ and $\hat{l}_{v_j}$. The rays corresponding to these events are bitangent rays or correspond to an endpoint of some D_p. We must detect the ray reached first by further counterclockwise rotation. The rays are anchored at o and rightward. For each of the three candidate rays, we know a rational direction vector. Thus we can compare slopes solely with rational arithmetic.

While the actions required at an event are easily identified for separate events – remember that all intersections are transversal – some care must be taken if several events coincide. If we have such a multi-event we know the order before sweeping across the event for at least one pair of functions intersecting at the event. This information can be used to identify the corresponding bitangent ray. Once we know the ray, we can use it to detect the order of functions after processing the multi-event.

We can merge two lower envelopes and finally compute $\hat{L}_S$ solely using predicates that involve just rational arithmetic. With the use of homogeneous coordinates, even integer arithmetic suffices. Since the algebraic degree of the arithmetic expressions involved is low, namely one, the application of floating-point filters and even number types based on expression dags and separation bounds becomes feasible. Such number types are particularly user-friendly since they encapsulate and hide all implementation details of exact computation [1, 8].

Whereas tangency points on bitangent rays have rational coordinates by Lemma 3, the tangency point on a ray tangent to a circle with given squared radius γ centered at $p = (p_x, p_y)$ has irrational coordinates in general. The x-coordinate of the tangency point of such a ray is

$$X_{1,2} = \frac{2\,p_x(p_x^2 + p_y^2 - \gamma) \pm 2\sqrt{\gamma p_y^2(p_x^2 + p_y^2 - \gamma)}}{2\,(p_x^2 + p_y^2)}$$

This holds for $\gamma = \delta_{\min}^2$ in particular. Thus, the tangency point on a ray corresponding to a highest point on L_S might have irrational coordinates. However, Lemma 2 guarantees the existence of an equivalent optimal bitangent ray corresponding to a highest point on $\hat{L}_S$ which has a rational direction vector. Thus, looking at $\hat{L}_S$ instead of L_S is an essential part of our approach.

5 Conclusion

We have shown that the optimal algorithm of [5] for computing a largest empty anchored cylinder in the plane can be implemented using simply rational arithmetic, thereby smoothing the way for efficient exact geometric computation. Unfortunately, for largest empty anchored cylinders in three-dimensions, a very different approach is used in [5] and our results do not immediately extend. Largest empty anchored cylinder computation in the plane is an exemplary geometric problem where a careful implementation and a careful analysis of the algebraic degree of the involved arithmetic expressions reveals an arithmetic demand much smaller than expected.

Acknowledgments: The author would like to thank Michiel Smid for helpful discussions and an anonymous reviewer for useful suggestions.

References

[1] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. Efficient exact geometric computation made easy. In *Symposium on Computational Geometry*, pages 341–350, 1999.

[2] C. Burnikel, S. Funke, K. Mehlhorn, S. Schirra, and S. Schmitt. A separation bound for real algebraic expressions. In *European Symposium on Algorithms*, pages 254–265, 2001.

[3] D. A. Cox, J. B. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer-Verlag, 1998.

[4] CGAL. `http://www.cgal.org`.

[5] F. Follert, E. Schömer, J. Sellen, M. Smid, and C. Thiel. Computing a largest empty anchored cylinder, and related problems. *Internat. J. Comput. Geom. Appl.*, 7:563–580, 1997.

[6] S. Fortune. Progress in computational geometry. In R. Martin, editor, *Directions in Computational Geometry*, pages 81–128. Information Geometers, 1993.

[7] L. J. Guibas and M. Sharir. Combinatorics and algorithms of arrangements. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, pages 9–36. Springer-Verlag, Heidelberg, 1993.

[8] V. Karamcheti, C. Li, I. Pechtchanski, and C. Yap. A core library for robust numeric and geometric computation. In *Symposium on Computational Geometry*, pages 351–359, 1999.

[9] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge, UK, 2000.

[10] S. Pion and C. K. Yap. Constructive root bound for k-ary rational input numbers. In *Symposium on Computational Geometry*, pages 256–263, San Diego, USA, 2003.

[11] S. Schirra. Robustness and precision issues in geometric computation. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 14, pages 597–632. Elsevier Science Publishers, 2000.

[12] C. K. Yap. Towards exact geometric computation. In *Canadian Conference on Computational Geometry*, pages 405–419, 1993.

[13] C. K. Yap. Robust geometric computation. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, 2nd edition. CRC Press LLC, Boca Raton, USA, 2004.