Efficient Randomized Algorithms for Some Geometric Optimization Problems*

Pankaj K. Agarwal[†] Micha Sharir[‡]

October 5, 1995

Abstract

In this paper we first prove the following combinatorial bound, concerning the complexity of the vertical decomposition of the minimization diagram of trivariate functions: Let $\mathcal{F}$ be a collection of n totally- or partially-defined algebraic trivariate functions of constant maximum degree, with the additional property that, for a given pair of functions $f, f' \in \mathcal{F}$, the surface $f(x,y,z) = f'(x,y,z)$ is xy-monotone (actually, we need a somewhat weaker property). We show that the vertical decomposition of the minimization diagram of $\mathcal{F}$ consists of $O(n^{3+\varepsilon})$ cells (each of constant complexity), for any $\varepsilon > 0$. In the second part of the paper we present a general technique that yields faster randomized algorithms for solving a number of geometric optimization problems, including (i) computing the width of a point set in 3-space, (ii) computing the minimum-width annulus enclosing a set of n points in the plane, and (iii) computing the 'biggest stick' inside a simple polygon in the plane. Using the above result on vertical decompositions, we show that the expected running time of all three algorithms is $O(n^{3/2+\varepsilon})$, for any $\varepsilon > 0$. Our algorithm improves and simplifies previous solutions of all three problems.

* Work on this paper by the first author has been supported by NSF Grant CCR-93-01259, an NYI award, and matching funds from Xerox Corporation. Work on this paper by the second author has been supported by NSF Grants CCR-91-22103 and CCR-93-11127, by a Max-Planck Research Award, and by grants from the U.S.-Israeli Binational Science Foundation, the Israel Science Fund administered by the Israeli Academy of Sciences, and the G.I.F., the German-Israeli Foundation for Scientific Research and Development.

† Department of Computer Science, Box 90129, Duke University, Durham, NC 27708-0129

‡ School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel, and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA

1 Introduction

In this paper we present a general technique that yields faster randomized algorithms for the following problems:

1. Computing the width of a set of points in $\mathbb{R}^3$.
2. Computing an annulus of minimum width that contains a given set of points in the plane.
3. Computing a longest segment that can be placed inside a simple polygon in the plane.

In order to achieve a fast implementation of our technique, we use the following combinatorial result, which is derived in the first part of the paper. Let $\mathcal{F}$ be a collection of n totally- or partially-defined algebraic trivariate functions of constant maximum degree, with the following additional *xy-monotonicity* property: For any pair $f, f' \in \mathcal{F}$, the xy-plane can be decomposed into a constant number of faces, each of constant description complexity, such that, for every face c, the surface $f(x,y,z) = f'(x,y,z)$ is the graph of a continuous bivariate function (of x and y) over the interior of c. The *lower envelope* $E_{\mathcal{F}}$ of $\mathcal{F}$ is the pointwise minimum

$$E_{\mathcal{F}}(x,y,z) = \min_{f \in \mathcal{F}} f(x,y,z)\,,$$

and the *minimization diagram* $M_{\mathcal{F}}$ is the projection of the graph of $E_{\mathcal{F}}$ onto $\mathbb{R}^3$. That is, $M_{\mathcal{F}}$ is a decomposition of $\mathbb{R}^3$ into relatively-open connected cells of dimension 0, 1, 2, and 3, so that, over each cell, $E_{\mathcal{F}}$ is attained by a fixed subset of functions of $\mathcal{F}$ (and/or of function boundaries). It is known [27] that $M_{\mathcal{F}}$ has $O(n^{3+\varepsilon})$ cells (of all dimensions).

We prove that the *vertical decomposition* of $M_{\mathcal{F}}$ also consists of $O(n^{3+\varepsilon})$ cells, each of constant description complexity. See below and [8, 10, 14, 28] for the definition of vertical decompositions. Briefly, this is the only known general-purpose technique for decomposing cells of arrangements of low-degree algebraic surfaces in higher dimensions into a reasonably small number of subcells of constant complexity. Such a decomposition is a prerequisite to many randomized incremental or divide-and-conquer algorithms involving arrangements of this kind. Unfortunately, the known upper bounds on the number of resulting subcells are much higher than the actual complexity of the cells being decomposed, and this affects adversely (the upper bounds one can prove on) the complexity of the relevant algorithms. Hence any result, like the one we prove here, which establishes nearly-tight bounds for the size of vertical decompositions, is significant, as indeed will be demonstrated below.

Our bound on the vertical decomposition immediately leads to a data structure, of size $O(n^{3+\varepsilon})$, for efficient point location queries in the region below $E_{\mathcal{F}}$: For a point $\mathbf{x} = (a_1, a_2, a_3, a_4)$ in $\mathbb{R}^4$, we can determine in $O(\log n)$ time whether $a_4 < E_{\mathcal{F}}(a_1, a_2, a_3)$. The technique for constructing this data structure crucially relies on the existence of a vertical decomposition of this region with near-cubic complexity.

We next observe that each of the three optimization problems mentioned above can be reduced to the problem of computing a closest (or farthest) pair between two sets of objects in $\mathbb{R}^d$, under some appropriate (pseudo-)distance function. That is, we define two sets of objects A, B and a function $\delta : A \times B \rightarrow \mathbb{R}^+ \cup \{0\}$, and reduce the original optimization problem to that of computing $\delta^* = \min_{a \in A, b \in B} \delta(a, b)$. Actually, we need to solve several instances of the closest-pair problem, but we show that the overall running time is still bounded by (a polylogarithmic factor times) the time complexity of the algorithm for computing δ^*. We use a randomized divide-and-conquer approach to compute δ^*, which is inspired by the Clarkson–Shor algorithm [11] for computing the diameter of a set of n points in 3-space. The merge and the divide steps of our algorithm require a data structure for point location in the minimization diagram of a set of trivariate functions that satisfy the aforementioned properties. Surprisingly, the xy-monotonicity property, which might be regarded as a somewhat restrictive condition, is also satisfied for each of the three optimization problems under consideration. Using our bounds on the complexity of vertical decompositions, we show that the expected running time of our algorithms, for all three problems, is $O(n^{3/2+\varepsilon})$, for any $\varepsilon > 0$. The previously best known algorithms for these problems are due to Agarwal et al. [1], and are based on Megiddo's *parametric search* technique (see also [3, 7]). The expected running time of the algorithms in [1] is $O(n^{17/11+\varepsilon})$, for any $\varepsilon > 0$.

We consider the main contributions of this paper to be the general algorithmic technique itself and the bound on the size of the vertical decomposition, both of which might be useful in other problems. Even though Megiddo's parametric search technique is a very powerful paradigm, it typically leads to quite complicated algorithms. Recently, there have been several attempts [5, 12, 19, 20, 23] to present simpler and more direct algorithms for some of the problems that have traditionally been solved using parametric searching. Our technique can be viewed as another step in this direction.

The paper is organized as follows. We first establish in Section 2 our bound on the complexity of the vertical decomposition. We then present in Section 3 the general algorithmic technique for computing a closest pair, and exemplify it by applying it to the width problem. We then discuss, more briefly, the minimum width annulus and the biggest stick problems in Sections 4 and 5, respectively.

2 Complexity of the Vertical Decomposition

Let $\mathcal{F}$ be a collection of n totally- or partially-defined algebraic trivariate functions of constant maximum degree b that satisfy the following properties:

(F1) If a function $f \in \mathcal{F}$ is partially defined, then we require that the set of points where f is undefined have measure 0, in the following strong sense: There is a constant number of algebraic arcs of constant maximum degree in the xy-plane (these arcs depend on

f), so that, for each point (x, y) not lying on any of these arcs, f is defined at all points (x, y, z), for any $z \in \mathbb{R}$.

(F2) For any pair of functions $f, f' \in \mathcal{F}$, the surface $f(x, y, z) = f'(x, y, z)$ is xy-monotone, that is, every z-vertical line (not passing through any curve where f or f' is undefined) crosses this surface in exactly one point. Actually, we somewhat relax this assumption, requiring only that, for each such surface σ, the xy-plane can be decomposed into a constant number of regions, each of constant description complexity (i.e., described by a constant number of polynomial equalities and inequalities of constant maximum degree), so that, for each of these regions c, the surface σ is the graph of a continuous bivariate function (of x and y) over the interior of c.

These assumptions are rather restrictive, but, as we will show below, and rather surprisingly, they hold for several of the current main applications of lower envelopes in 4-space, as listed in the introduction and studied recently in [1, 3, 7]. We also assume that the functions in $\mathcal{F}$ are in *general position*, as defined, e.g., in [27]; it is easy to show, using a variant of the argument given in [27], that this assumption does not involve any loss of generality, and that our results also hold for collections not in general position. Under this assumption, for $j = 0, \ldots, 3$, the envelope $E_{\mathcal{F}}$ is attained by at most (or, if the functions in $\mathcal{F}$ are totally defined, exactly) $4 - j$ functions of $\mathcal{F}$ over any j-dimensional cell of $M_{\mathcal{F}}$. We will use the terms vertex, edge, face, and cell to denote, respectively, 0-dimensional, 1-dimensional, 2-dimensional, and 3-dimensional cells of $M_{\mathcal{F}}$. Each vertex, edge, face, or cell c of $M_{\mathcal{F}}$ will be labeled by the corresponding set of functions of $\mathcal{F}$ attaining $E_{\mathcal{F}}$ over c. In particular, the term f-cell will refer to a (3-dimensional) cell of $M_{\mathcal{F}}$ over which $E_{\mathcal{F}}$ is attained by the function f.

The *vertical decomposition* of $M_{\mathcal{F}}$ is defined in the following standard manner. In the first decomposition stage, we erect, for each edge e of $M_{\mathcal{F}}$, a z-vertical wall from e, which is the union of all maximal z-vertical relatively-open segments passing through points of e and not meeting any other vertex, edge or face of $M_{\mathcal{F}}$. The collection of these walls partitions the cells of $M_{\mathcal{F}}$ into subcells, so that each subcell c is bounded from above and from below (in the z-direction) by (portions of) a fixed pair of faces of $M_{\mathcal{F}}$; c may also extend to infinity in either direction. In the second decomposition step, we take each of these subcells c and project it onto the xy-plane. We construct the 2-dimensional vertical decomposition of the projection c^*, by erecting a maximal y-vertical segment, contained in the closure of c^*, from each vertex of c^* and each (locally) x-extremal point on ∂c^*. The collection of these segments partitions c^* into 'pseudo-trapezoidal' subcells. Each of these subcells τ induces a subcell of c, obtained by intersecting c with the vertical cylinder $\tau \times \mathbb{R}$ over τ. The resulting collection of subcells constitutes the vertical decomposition of $M_{\mathcal{F}}$, which we denote by $M^*_{\mathcal{F}}$. Each of these cells has constant description complexity, in the sense that it is defined by a constant number of polynomial equalities and inequalities of constant maximum degree (depending on the maximum degree b of the functions of $\mathcal{F}$). See [8, 10, 28] for more details concerning vertical decompositions.

Theorem 2.1 *If $\mathcal{F}$ is a collection of trivariate functions satisfying the assumptions made above, then the number of subcells of $M^*_{\mathcal{F}}$ is $O(n^{3+\varepsilon})$, for any $\varepsilon > 0$, where the constant of proportionality depends on ε and on the maximum degree b.*

Proof: Let $\mathcal{F}$ be a collection of n trivariate functions satisfying the above assumptions. It is easily seen that, in general, the second vertical decomposition step does not increase the complexity of the decomposition by more than a constant factor, so it suffices to bound the increase in the complexity of $M_{\mathcal{F}}$ caused by the first vertical decomposition step. In other words, we want to count the number of pairs (e, e') of edges of $M_{\mathcal{F}}$, both bounding the same cell c, such that there exists a z-vertical segment connecting a point on e to a point on e' and fully contained in c. (We actually want to count the number of these vertical segments, but, by assumption, this number is larger than the number of pairs (e, e') by only a constant factor, depending on the maximum degree b.) We say that such a pair (e, e') of edges are *vertically visible*. Suppose c is an f_0-cell, for some $f_0 \in \mathcal{F}$ (note that assumption (F1) implies that there is no 3-dimensional cell of $M_{\mathcal{F}}$ over which $E_{\mathcal{F}}$ is undefined). Then e must be either a portion of an intersection curve of the form $f_0 = f_1 = f_2$, for some pair of functions $f_1, f_2 \in \mathcal{F}$, or a portion of the boundary of an xy-monotone piece of a surface $f_0 = f$, for some $f \in \mathcal{F}$. Similarly, e' must also be a portion of an intersection curve or of a boundary curve of the above forms.

We estimate the number of vertically-visible pairs of edges for which the vertical segment connecting the edges crosses an f_0-cell, separately for each fixed $f_0 \in \mathcal{F}$. Recall that, by assumption, each surface $f_0 = f$ can be decomposed into a constant number of xy-monotone pieces, so that the xy-projections of these pieces are pairwise disjoint. Consider the collection $\Sigma(f_0)$ consisting of all these xy-monotone portions of surfaces of the form $f_0 = f$, for $f \in \mathcal{F}$. We regard each such portion as a partially-defined function of x and y. It follows from assumption (F1) that, for each surface $\sigma \in \Sigma(f_0)$ contained in the graph of $f_0 = f$, for some $f \in \mathcal{F}$, either all points lying vertically above σ (in the z-direction) satisfy $f_0 > f$ or all such points satisfy $f_0 < f$ (this property may fail at points lying on the boundary $\partial\sigma$ of σ, but this limit behavior does not affect our analysis). Let $\Sigma^+(f_0)$ (resp. $\Sigma^-(f_0)$) denote the subset of surfaces $\sigma \in \Sigma(f_0)$ for which the corresponding function f satisfies $f_0 > f$ (resp. $f_0 < f$) for all points lying vertically above σ (note that a function f may contribute surfaces to both collections $\Sigma^+(f_0)$, $\Sigma^-(f_0)$, over pairwise-disjoint portions of the xy-plane). It is then clear that the union of all f_0-cells is the same as the region enclosed between the upper envelope of $\Sigma^-(f_0)$ and the lower envelope of $\Sigma^+(f_0)$. It then follows from Theorem 3.2 of [2], concerning the complexity of the region enclosed between two envelopes in 3-space, that the complexity of the vertical decomposition of all f_0-cells is $O(n^{2+\varepsilon})$, for any $\varepsilon > 0$. Repeating this argument over all functions $f_0 \in \mathcal{F}$, we obtain the bound asserted in the theorem. □

Let $C_{\mathcal{F}}$ be the cell in the arrangement of $\mathcal{F}$ lying below $E_{\mathcal{F}}$. The vertical decomposition of $C_{\mathcal{F}}$, denoted as $C^*_{\mathcal{F}}$, can be obtained by lifting each cell $\tau \in M^*_{\mathcal{F}}$ to the cell

$$\hat{\tau} = \{(\mathbf{x}, z) \mid \mathbf{x} \in \tau,\ -\infty \le z \le E_{\mathcal{F}}(\mathbf{x})\}.$$

Since each cell of $M^*_{\mathcal{F}}$ contributes exactly one cell to $C^*_{\mathcal{F}}$, the latter also has $O(n^{3+\varepsilon})$ cells. Similarly, it follows that the vertical decomposition of the cell lying above the graphs of all functions in $\mathcal{F}$ also has $O(n^{3+\varepsilon})$ cells.

Remark 2.2 An obvious open problem is to extend this bound to vertical decompositions of minimization diagrams of more general trivariate functions. Using recent analysis techniques, as those in [2, 27], we can obtain an $O(n^{4+\varepsilon})$ bound for the general case of partially-defined trivariate low- degree algebraic functions, but we conjecture that the correct bound is near-cubic.

3 Width in 3-Space

The *width* of a set S of n points in $\mathbb{R}^3$ is the smallest distance between a pair of parallel planes such that the closed slab between the planes contains S. Although the width of a set of n points in the plane can be computed in $O(n \log n)$ time [18], the problem becomes considerably harder in three dimensions. Houle and Toussaint [18] gave a simple $O(n^2)$-time algorithm for computing the width in $\mathbb{R}^3$, and raised the open problem of obtaining a subquadratic solution. Chazelle et al. [7] presented an $O(n^{8/5+\varepsilon})$-time algorithm, for any $\varepsilon > 0$, which was subsequently improved by Agarwal et al. [1] to $O(n^{17/11+\varepsilon})$. As observed in [7], and further exploited in [1], the problem of computing the width in 3-space can be reduced to the following *bichromatic closest line-pair* problem: Given a set L of m 'red' lines and another set L' of n 'blue' lines in $\mathbb{R}^3$, such that all red lines lie above all blue lines,[1] compute the closest pair of lines in $L \times L'$, where the distance between a pair of lines $\ell, \ell' \in \mathbb{R}^3$ is

$$d(\ell, \ell') = \min_{p \in \ell, q \in \ell'} d(p, q).$$

Let $d(L, L') = \min_{\ell \in L,\ \ell' \in L'} d(\ell, \ell')$ denote the distance between a closest pair in $L \times L'$.

Before presenting the algorithm, we need to describe some geometric transforms, which will be crucial for our algorithm. We can map each line $\ell \in L$, not parallel to the yz-plane, to a point $\psi(\ell) = (a_1, a_2, a_3, a_4)$ in $\mathbb{R}^4$, where $y = a_1 x + a_3$ is the equation of the xy-projection of ℓ, and $z = a_2 x + a_4$ is the equation of the xz-projection of ℓ. For any fixed real parameter $\delta \ge 0$, we can also map a line $\ell' \in L'$ to a surface $\gamma(\ell')$, which is the locus of all points $\psi(\ell)$

[1] For any pair of nonparallel and nonvertical lines $\ell \in L$, $\ell' \in L'$, we say that ℓ lies above ℓ' if the vertical line passing through the intersection point of the xy-projections of ℓ and ℓ' intersects ℓ above ℓ'. It is interesting to note that the requirement that all red lines lie above all blue lines crucially affects the analysis of the complexity of the resulting algorithm.

such that $d(\ell,\ell')=\delta$ and ℓ lies above ℓ'. We refer to the coordinates of this parametric space as ξ_1,ξ_2,ξ_3,ξ_4. Observe that any line parallel to the ξ_4-axis intersects $\gamma=\gamma(\ell')$ in at most one point. If the corresponding lines ℓ in $\mathbb{R}^3$ lie in a vertical plane parallel to ℓ' and not containing ℓ', then the intersection point may not exist. It follows that γ can be partitioned into a constant number of surface patches, each of constant description complexity, such that, for each patch $\tilde{\gamma}$, all points of $\mathbb{R}^4$ lying vertically above $\tilde{\gamma}$ represent lines ℓ in $\mathbb{R}^3$ that lie above ℓ' and $d(\ell,\ell')>\delta$, and all points lying below $\tilde{\gamma}$ represent lines ℓ that either lie below ℓ', pass through ℓ', or lie above ℓ' and $d(\ell,\ell')<\delta$. In other words, $\gamma(\ell')$ is the graph of a partially-defined function $\xi_4=f_{\ell'}(\xi_1,\xi_2,\xi_3)$. For a point $\psi(\ell)=(a_1,a_2,a_3,a_4)$, such that ℓ lies above ℓ', if $f_{\ell'}(a_1,a_2,a_3)$ is defined, then $a_4>f_{\ell'}(a_1,a_2,a_3)$ if and only if $d(\ell,\ell')>\delta$, and $a_4<f_{\ell'}(a_1,a_2,a_3)$ if and only if $d(\ell,\ell')<\delta$. Let $\mathcal{F}$ be the collection $\{f_{\ell'} \mid \ell'\in L'\}$, and let $U_{\mathcal{F}}$ denote the upper envelope of $\mathcal{F}$. For a line $\ell\in L$ with $\psi(\ell)=(a_1,a_2,a_3,a_4)$, we have $a_4\geq U_{\mathcal{F}}(a_1,a_2,a_3)$ if and only if $d(\{\ell\},L')\geq\delta$. It is easily checked that the functions $f_{\ell'}$ are all partially-defined, algebraic functions of constant maximum degree.

Lemma 3.1 *(a) For any line $\ell'\in L'$ and for any fixed ξ_1, ξ_2, so that ξ_1 is not equal to the ξ_1-coordinate of ℓ', the function $f_{\ell'}(\xi_1,\xi_2,\xi_3)$ is defined for all ξ_3.*

(b) For any pair of non-parallel lines ℓ_1', $\ell_2'\in L'$ and for any fixed ξ_1,ξ_2, the equation

$$f_{\ell_1'}(\xi_1,\xi_2,\xi_3)=f_{\ell_2'}(\xi_1,\xi_2,\xi_3) \tag{3.1}$$

has a unique solution ξ_3, except when (ξ_1,ξ_2) lies on a certain critical line $\lambda(\ell_1',\ell_2')$ that depends on ℓ_1' and ℓ_2', or when ξ_1 is equal to the ξ_1-coordinate of ℓ_1' or of ℓ_2'.

Proof: Part (a) is trivial: Let ξ_1,ξ_2,ξ_3 be a triple such that ξ_1 is not equal to the ξ_1-coordinate of ℓ'. Then the spatial orientation of all corresponding lines is fixed and their xy-projection has different orientation than that of ℓ'. Consider the line of the form $y=\xi_1x+\xi_3, z=\xi_2x+\xi_4$ and intersecting ℓ'. If we translate this line in the $+z$-direction (i.e., increase the value of ξ_4), its distance with ℓ' monotonically increases, and therefore there is a unique $\xi_4=\alpha$ such that the line ℓ, with $\varphi(\ell)=(\xi_1,\xi_2,\xi_3,\alpha)$, lies above ℓ' and is at distance δ from ℓ'. Hence, $f_{\ell'}(\xi,\xi_2,\xi_3)$ is defined. (If the xy-projection of these lines has the same orientation as that of ℓ', then $f_{\ell'}(\xi_1,\xi_2,\xi_3)$ is defined only when ξ_2 is equal to the ξ_2-coordinate of ℓ' and when ξ_3 is such that ℓ and ℓ' lie in the same vertical plane.)

As to part (b), let ξ_3 be a solution of (3.1), and let $\xi_4=f_{\ell_1'}(\xi_1,\xi_2,\xi_3)=f_{\ell_2'}(\xi_1,\xi_2,\xi_3)$. The line ℓ^*, parametrized by $(\xi_1,\xi_2,\xi_3,\xi_4)$, thus lies in the vertical plane $\pi(\xi_3)\ :\ y=\xi_1x+\xi_3$, and, as is easily checked, its slope in that plane, with respect to the coordinate frame (u,z), where u is the axis orthogonal to the z-axis, is equal to $\xi_2/\sqrt{1+\xi_1^2}$. Moreover, by definition, ℓ^* is a common upper tangent line to the two cylinders C_1, C_2 of radius δ, whose symmetry axes are the lines ℓ_1', ℓ_2', respectively. Let $K_i=K_i(\xi_3)=C_i\cap\pi(\xi_3)$, for $i=1,2$. The sets K_1 and K_2 are two ellipses, and the line ℓ^* must be a common upper tangent to K_1 and K_2 in the plane $\pi(\xi_3)$ (this holds provided that ξ_1 is not equal to the ξ_1-coordinate of ℓ_1' or of ℓ_2'). As ξ_3 varies, the plane $\pi(\xi_3)$ translates parallel to itself, and the two ellipses $K_i(\xi_3)$

also translate within that plane, so that the positions of their centers are given by two linear functions of ξ_3. Moreover, for $i = 1, 2$, let $w_i = w_i(\xi_3)$ denote the point on $K_i(\xi_3)$ so that the line tangent to K_i at w_i has slope $\xi_2/\sqrt{1+\xi_1^2}$ and lies above K_i. It follows that, as ξ_3 varies, $w_i(\xi_3)$ moves within the plane $\pi(\xi_3)$ as a linear function of ξ_3, for $i = 1, 2$. Thus, ξ_3 solves (3.1) if and only if the line connecting $w_1(\xi_3)$ and $w_2(\xi_3)$ has slope $\xi_2/\sqrt{1+\xi_1^2}$ in $\pi(\xi_3)$. This equation is linear in ξ_3, as easily follows from the above arguments, and so has one solution, no solutions, or infinitely many solutions.

To analyze when this equation has no solution, or has infinitely many solutions, we represent the above geometric reasoning in an algebraic form. It is easily verified that the existence of a unique solution to (3.1) is not affected if we translate ℓ_1' and ℓ_2' by any amounts (such a translation only changes the constant term in the resulting linear equation), so we may assume, with no loss of generality, that both lines pass through the origin. Let (a_1, b_1, c_1), (a_2, b_2, c_2) be two unit vectors lying, respectively, on the lines ℓ_1', ℓ_2'. The intersection $s_1(\xi_3)$ of ℓ_1' with $\pi(\xi_3)$ is a point $(a_1 t, b_1 t, c_1 t)$ that satisfies the equation $b_1 t = \xi_1 a_1 t + \xi_3$, so we have $t = \xi_3/(b_1 - a_1\xi_1)$, which implies that $s_1(\xi_3)$ is the point

$$\left(\frac{a_1\xi_3}{b_1 - a_1\xi_1}, \frac{b_1\xi_3}{b_1 - a_1\xi_1}, \frac{c_1\xi_3}{b_1 - a_1\xi_1}\right),$$

and, similarly, the intersection $s_2(\xi_3)$ of ℓ_2' with $\pi(\xi_3)$ is the point

$$\left(\frac{a_2\xi_3}{b_2 - a_2\xi_1}, \frac{b_2\xi_3}{b_2 - a_2\xi_1}, \frac{c_2\xi_3}{b_2 - a_2\xi_1}\right).$$

(As above, these points are well-defined only when ξ_1 is not equal to the ξ_1-coordinate of ℓ_1' or of ℓ_2'.) For $i = 1, 2$, the point $w_i(\xi_3)$ is a translated copy of $s_i(\xi_3)$ by a fixed vector, independent of ξ_3. The coefficient of ξ_3 in the equation (3.1) is thus easily seen to be (proportional to)

$$\frac{c_2}{b_2 - a_2\xi_1} - \frac{c_1}{b_1 - a_1\xi_1} - \xi_2\left(\frac{a_2}{b_2 - a_2\xi_1} - \frac{a_1}{b_1 - a_1\xi_1}\right).$$

Hence, the equation (3.1) does not have a unique solution only when this expression is 0. That is,

$$\frac{c_2 - a_2\xi_2}{b_2 - a_2\xi_1} = \frac{c_1 - a_1\xi_2}{b_1 - a_1\xi_1},$$

which is easily seen to be a linear equation in ξ_1 and ξ_2 (it does not vanish identically, unless ℓ_1' and ℓ_2' are parallel). This completes the proof of the lemma. □

Lemma 3.1 implies that the collection $\mathcal{F}$ satisfies the assumptions (F1), (F2) of Theorem 2.1. Let $C_{\mathcal{F}}$ denote the cell in the arrangement of $\mathcal{F}$ that lies above the upper envelope of $\mathcal{F}$. In view of Lemma 3.1, Theorem 2.1, the above discussion, and standard point-location techniques (such as those in [8, 9]), we obtain

Corollary 3.2 *The vertical decomposition $C^*_{\mathcal{F}}$ of $C_{\mathcal{F}}$ consists of $O(n^{3+\varepsilon})$ cells, for any $\varepsilon > 0$. Moreover, $C_{\mathcal{F}}$ can be preprocessed in time $O(n^{3+\varepsilon})$ into a data structure of size $O(n^{3+\varepsilon})$, for any $\varepsilon > 0$, so that, for any query point p, we can determine in $O(\log n)$ time whether $p \in C_{\mathcal{F}}$.*

We now describe a divide-and-conquer algorithm for computing $d(L, L')$. The basic idea is as follows: We randomly choose a line $\ell \in L$, and compute the distance δ_0 between ℓ and its nearest neighbor in L'. We discard all those lines of L whose nearest neighbors in L are at distance $\geq \delta_0$. Let $L_1 \subset L$ be the subset of remaining lines. If $L_1 = \emptyset$, we return δ_0; otherwise we recursively compute $d(L_1, L')$. Using a probabilistic argument, one can show that the expected depth of the recursion is $O(\log m)$. The only nontrivial step in the above algorithm is computing the subset L_1. We, however, do not have an efficient procedure for computing L_1, so, after computing δ_0, we proceed in a round about way. We divide the problem of computing $d(L, L')$ into a number of subproblems, of which one is solved recursively and the others are solved using a different algorithm, as described below. We will first present an outline of the algorithm, and then explain each of the nontrivial steps in some detail.

ALGORITHM CLOSEST-PAIR

1. Let n_0 be a sufficiently large constant, whose value will be fixed later. If $n \leq n_0$, then we compute $d(\ell, \ell')$ for every pair $(\ell, \ell') \in L \times L'$, in $O(m)$ time, and return the minimum distance.

2. Assume that $n > n_0$. Randomly choose a line $\ell_0 \in L$ and compute $\delta_0 = d(\{\ell_0\}, L')$, in $O(n)$ time.

3. Set $r = \lceil m^{3/8}/n^{1/8} \rceil$. We partition L into $k+1$ subsets $L_0, L_1, \ldots, L_k$, with the following properties:

   (i) $k = O(r^{3+\varepsilon})$, for any $\varepsilon > 0$;

   (ii) if $r = 1$, then $k = 1$, $L_0 = \emptyset$, $L_1 = L$;

   (iii) for each $1 \leq i \leq k$,
   $$|\{\ell' \in L' \mid d(L_i, \{\ell'\}) < \delta_0\}| \leq \frac{n}{r};$$

   (iv) $L_0 \subseteq \{\ell \in L \mid d(\{\ell\}, L') < \delta_0\}$; L_0 may be empty (as is the case when $r = 1$).

4. For each $1 \leq i \leq k$, we compute a set L'_i of size at most n/r such that
   $$L'_i \supseteq \{\ell' \in L' \mid d(L_i, \{\ell'\}) < \delta_0\};$$
   if $r = 1$ and $k = 1$, we put $L'_1 = L'$. Set $m_i = |L_i|$ and $n_i = |L'_i|$.

5. For each $1 \le i \le k$, we do the following: If $n_i = 0$, we set $d(L_i, L'_i) = +\infty$. Otherwise, we compute $\delta_i^* = d(L_i, L'_i)$ directly, using a different algorithm (detailed below). We then compute $\delta_1 = \min_i \delta_i^*$.

6. If $L_0 \ne \emptyset$, we compute $\delta_2 = d(L_0, L')$ recursively.

7. Return $\min\{\delta_0, \delta_1, \delta_2\}$ as $d(L, L')$.

Next, we explain Steps 3–5 in detail, and analyze their expected running time; the other steps are trivial and need no further explanation. We will then conclude the analysis by proving the correctness of the algorithm.

Steps 3–4: We compute L_i, L'_i, for $1 \le i \le k$, using a divide-and-conquer approach. We construct a tree T, each of whose nodes v is associated with two subsets $L_v \subseteq L$ and $L'_v \subseteq L'$. The root of the tree is associated with L and L' themselves. The subsets associated with the leaves of T will correspond to the sets L_i and L'_i.

If $r = 1$, then T consists of a single node; we set $k = 1$, $L_1 = L$, $L'_1 = L'$, and $L_0 = \emptyset$. Next, assume that $r > 1$. Let s be some sufficiently large constant. We choose a random subset $X \subseteq L'$ of size $c_1 s \log s$, where c_1 is an appropriate constant independent of s, and compute C_X^*, the vertical decomposition of the cell C_X lying above the graphs of all the functions $\{f_{\ell'} \mid \ell' \in X\}$ (defined in terms of the parameter δ_0 computed in Step 2). By Corollary 3.2, C_X^* has $O((s \log s)^{3+\varepsilon})$ cells. For each cell $\tau \in C_X^*$, we compute the set $L'_\tau \subseteq L'$ of lines ℓ' such that $\gamma(\ell')$ intersects τ. By standard ε-net theory [17], we have, with high probability, $|L'_\tau| \le n/s$ for every $\tau \in C_X^*$. If $|L'_\tau| > n/s$ for some $\tau \in C_X^*$, we choose another random subset and repeat the above steps. Otherwise, for each $\tau \in C_X^*$ we compute the subset $L_\tau \subseteq L$ of lines ℓ such that $\psi(\ell) \in \tau$. Set $m_\tau = |L_\tau|$ and $n_\tau = |L'_\tau|$. If $L_\tau \ne \emptyset$, we create a child v_τ of the root corresponding to τ. We associate L_τ, L'_τ with v_τ. If $|L'_\tau| \le n/r$, then v_τ is a leaf. Otherwise, v_τ is an internal node of T, and we expand T further at v_τ by applying the same procedure recursively to L_τ, L'_τ.

By construction, the depth of T is at most $\lceil \log_s r \rceil$. Since each node has at most $O((s \log s)^{3+\varepsilon})$ children, the total number of leaves in T is $k \le c_2 r^{3+\varepsilon'}$, for any $\varepsilon' > \varepsilon$ and for some constant c_2 independent of s and r (but depending on $\varepsilon, \varepsilon'$). We set L_i and L'_i to be the subsets associated with the ith leaf of T, for $i = 1, \ldots, k$. Finally, we set $L_0 = L - \bigcup_{i=1}^k L_i$. Note that a line ℓ is placed in L_0 only when its image $\psi(\ell)$ lies below the upper envelope of some collection $\{f_{\ell'} \mid \ell' \in X\}$, for some $X \subseteq L'$. Hence, by definition, all lines $\ell \in L_0$ satisfy $d(\{\ell\}, L') < \delta_0$. In particular, $\ell_0 \notin L_0$, so $|L_0| < m$, a property that we will use below when proving the correctness of the algorithm. This also shows that L_0 satisfies property (iv) of Step 3.

The sets L_i, for $1 \le i \le k$, are pairwise disjoint, and $|L'_i| \le n/r$, for all $1 \le i \le k$. It thus remains to show that

$$L'_i \supseteq \{\ell' \in L' \mid d(\{\ell'\}, L_i) < \delta_0\}.$$

In fact, the following stronger claim is true, and follows easily by construction.

Lemma 3.3 *For any node v_τ in T,*

$$L'_\tau \supseteq L''_\tau = \{\ell' \in L' \mid d(\{\ell'\}, L_\tau) < \delta_0\,\}.$$

Proof: We prove this by induction on the depth of v_τ in T. The claim obviously holds for the root of T. Suppose it holds for the parent v_ζ of a node v_τ. Since $L_\tau \subseteq L_\zeta$, obviously $L''_\tau \subseteq L''_\zeta$. Let C^*_X be the set of cells that we constructed at v_ζ. Then $\tau \in C^*_X$ and, for every $\ell \in L_\tau$, we have $\psi(\ell) \in \tau$. Let $\ell' \in L''_\tau$ and let ℓ be a line in L_τ satisfying $d(\ell, \ell') < \delta_0$. Then, by definition, the point $\psi(\ell)$ lies below the surface $\gamma(\ell')$. Since τ is unbounded in the $+\xi_4$-direction, it follows that $\gamma(\ell')$ intersects τ. Moreover, by the induction hypothesis, $\ell' \in L'_\zeta$, which implies that $\ell \in L'_\tau$, and thus the claim is true for τ as well. □

Hence, the sets L_i, L'_i, for $1 \le i \le k$, satisfy the desired properties of Steps 3 and 4.

Next, we analyze the expected time spent in computing these subsets. Let $f(a, b)$ denote the maximum expected time spent by the recursive algorithm for Steps 3–4, where expectation is with respect to the choices of random samples by the algorithm, and where the maximum is taken over all sets L, L' of lines, as above, of respective sizes a, b. At each level of recursion, X is chosen, with high probability, only once, and we spend $O((s \log s)^{3+\varepsilon}(a+b))$ time to compute all the sets L_τ, L'_τ. Since C^*_X consists of $O((s \log s)^{3+\varepsilon})$ cells, we obtain the following recurrence:

$$f(a, b) \le \sum_{i=1}^{c(s\log s)^{3+\varepsilon}} f(a_i, b_i) + c'(s \log s)^{3+\varepsilon}(a + b)\,,$$

where $\sum_i a_i \le a,\quad b_i \le b/s$, and c, c' are constants (depending on ε). The recursion stops when $b \le n/r$, so $f(a, b) = O(1)$ for $b \le n/r$.

The solution of the above recurrence is

$$f(a, b) \le A\left(a \log b + \frac{b^{3+\varepsilon'}}{n^2} r^2\right),$$

for any $\varepsilon' > \varepsilon$; here $A = A(\varepsilon')$ is a sufficiently large constant depending on the value of ε.

We prove this by induction on b. The inequality obviously holds for $b \le n/r$. For larger values of b, we obtain, by the induction hypothesis,

$$\begin{aligned} f(a, b) &\le \sum_{i=1}^{c(s\log s)^{3+\varepsilon}} A\left(a_i \log b_i + \frac{b_i^{3+\varepsilon'}}{n^2} r^2\right) + c'(s \log s)^{3+\varepsilon}(a + b) \\ &\le A \sum_{i=1}^{c(s\log s)^{3+\varepsilon}} \left(a_i \log \frac{b}{s} + \left(\frac{b}{s}\right)^{3+\varepsilon'} \cdot \frac{r^2}{n^2}\right) + c'(s \log s)^{3+\varepsilon}(a + b) \\ &\le Aa \log b + a\left[c'(s \log s)^{3+\varepsilon} - A \log s\right] + \end{aligned}$$

$$
\begin{aligned}
& Ab^{3+\varepsilon'}\frac{r^2}{n^2}\left[cs^{\varepsilon-\varepsilon'}\log^{3+\varepsilon} s + \frac{c'}{A}\frac{n^2/r^2}{b^2}(s\log s)^{3+\varepsilon}\right] \\
\leq\; & A\left(a\log b + b^{3+\varepsilon'}\frac{r^2}{n^2}\right),
\end{aligned}
$$

because $b > n/r$, $\varepsilon' > \varepsilon$, and A is chosen sufficiently large. Since $r = \lceil m^{3/8}/n^{1/8}\rceil$, and initially $a = m$ and $b = n$, we obtain

$$f(m,n) = O(m^{3/4+\varepsilon'}n^{3/4+\varepsilon'} + m^{1+\varepsilon'} + n^{1+\varepsilon'}).$$

Step 5: For each $1 \leq i \leq k$, We compute $d(L_i, L_i')$ using a somewhat simpler version of the randomized algorithm described by Agarwal et al. [1]. We give a brief sketch of this variant.

(i) Let n_0 be some sufficiently large constant. If $n_i \leq n_0$, we compute $d(\ell, \ell')$ for all pairs $\ell \in L_i$, $\ell' \in L_i'$, in $O(m_i)$ time, and return the minimum distance.

(ii) Assume $n_i > n_0$. Choose a random subset $A \subseteq L_i'$ of size $\lceil n_i/2\rceil$; each subset of size $\lceil n_i/2\rceil$ is chosen with equal probability.

(ii) Recursively compute $\delta' = d(L_i, A)$.

(iii) Compute the set
$$B = \{\ell' \in L_i' - A \mid d(L_i, \{\ell'\}) < \delta'\}.$$

(iv) Compute $d(\ell, \ell')$ for all pairs $\ell \in L_i$, $\ell' \in B$, and return the minimum distance (or output δ' if B is empty).

The correctness of the algorithm is obvious (see also [1]), so we now analyze its expected running time. For a line $\ell \in L_i$, let

$$B^{(\ell)} = \{\ell' \in L_i' - A \mid d(\ell, \ell') < \delta'\}.$$

Using a standard probabilistic argument, it can be shown that the expected size of $B^{(\ell)}$ is $O(1)$. Since $B = \bigcup_{\ell \in L_i} B^{(\ell)}$, the expected size of B is $O(m_i)$, and the expected running time of Step 5(iv) is $O(m_i^2)$.

By reversing the direction of the z-axis, setting $\delta = \delta'$, and using Corollary 3.2, L_i can be preprocessed into a data structure of size $O(m_i^{3+\varepsilon})$, so that, for each $\ell' \in L_i' - A$, we can determine in $O(\log m_i)$ time whether $\ell' \in B$. The time spent in Step 5(iii) is thus $O(m_i^{3+\varepsilon} + n_i \log m_i)$, which subsumes the expected cost of Step 5(iv). The running time of both steps can be improved, by a standard batching technique, to $O(m_i n_i^{2/3+\varepsilon} + n_i^{1+\varepsilon})$; see [1]. Let $\varphi(m_i, n_i)$ denote the maximum expected running time for computing $d(L_i, L_i')$ by this algorithm, where the maximum is taken over all sets L_i, L'_i of sizes m_i, n_i, respectively. Then we obtain the following recurrence

$$\varphi(m_i, n_i) \le \varphi(m_i, \lceil n_i/2 \rceil) + O(m_i n_i^{2/3+\varepsilon} + n_i^{1+\varepsilon}),$$

whose solution is easily seen to be

$$\varphi(m_i, n_i) = O(m_i n_i^{2/3+\varepsilon} + n_i^{1+\varepsilon}).$$

By the choice of the parameter r in algorithm CLOSEST-PAIR, the expected time spent in Step 5 is thus

$$\begin{aligned}\sum_{i=1}^{k} \varphi(m_i, n_i) &= \sum_{i=1}^{k} O(m_i n_i^{2/3+\varepsilon} + n_i^{1+\varepsilon}) \\ &= O\left(\left(\frac{n}{r}\right)^{2/3+\varepsilon} \sum_{i=1}^{k} m_i + \left(\frac{n}{r}\right)^{1+\varepsilon} \cdot r^{3+\varepsilon'}\right) \\ &= O(m^{3/4+\varepsilon'} n^{3/4+\varepsilon'} + m^{1+\varepsilon'} + n^{1+\varepsilon'}).\end{aligned}$$

The total expected time spent by algorithm CLOSEST-PAIR, excluding the time spent in the recursive call, is thus $O(m^{3/4+\varepsilon'} n^{3/4+\varepsilon'} + m^{1+\varepsilon'} + n^{1+\varepsilon})$.

Recall that all lines $\ell \in L_0$ satisfy $d(\{\ell\}, L') < \delta_0$. Recall also that ℓ_0 was chosen randomly in Step 2. If we sort the lines $\ell \in L$ in the nondecreasing order of their distances $d(\{\ell\}, L')$, then the probability that ℓ is the i^{th} item in this list is $1/m$, and in this case we must have $|L_0| < i$. Let $T(m, n)$ denote the maximum expected time for algorithm CLOSEST-PAIR to compute $d(L, L')$, where the maximum is taken over all sets L, L' of sizes m and n, respectively. The arguments just given imply that

$$T(m,n) \le \begin{cases} c_1 m & \text{for } n \le n_0, \\ c_2 n^{1+\varepsilon} & \text{for } n > n_0, m < n^{1/3}, \\ \dfrac{1}{m} \displaystyle\sum_{i=0}^{m-1} T(i,n) + A(m^{3/4+\varepsilon'} n^{3/4+\varepsilon'} + m^{1+\varepsilon'} + n^{1+\varepsilon'}) & \\ & \text{for } n > n_0, m > n^{1/3}. \end{cases}$$

The solution of the above recurrence is

$$T(m, n) \le B(m^{3/4+\varepsilon'} n^{3/4+\varepsilon'} + m^{1+\varepsilon'} \log n + n^{1+\varepsilon'}),$$

for any $\varepsilon' > \varepsilon$ and for some constant $B = B(\varepsilon')$.

To complete the analysis, we finally show:

Lemma 3.4 *Algorithm* CLOSEST-PAIR *computes the distance* $\delta^* = d(L, L')$ *correctly.*

Proof: We prove the lemma by double induction on m and n. Let L, L' be sets of size m and n, respectively. If $n < n_0$ then the correctness is trivial (see Step 1). If $m \le n^{1/3}$ (i.e., $r = 1$), then the algorithm is also correct, by the analysis of Step 5 given above. Suppose that $m > n^{1/3}$ (so $r > 1$), and that the algorithm computes $\delta^* = d(L, L')$ correctly for all sets of lines L and L' such that $|L| < m$ or $|L| = m$ and $|L'| < n$. Since the algorithm returns the distance between a line of L and a line of L', it always returns a number at least as large as δ^*.

If $\delta^* = \delta_0$, there is nothing to prove, because Step 7 returns the minimum of δ_0, δ_1, and δ_2. Suppose $\delta^* < \delta_0$. Let $\ell \in L$, $\ell' \in L'$ be a pair of lines with $d(\ell, \ell') = \delta^*$. If $\ell \in L_0$, then, $d(L, L') = d(L_0, L') = \delta_2$. Since $|L_0| < m$, as argued above, the algorithm computes $d(L_0, L')$ correctly, by the induction hypothesis, so $d(L, L')$ is also correctly computed.

If $\ell \in L_i$ for some $1 \le i \le k$, then, by construction, $\ell' \in L_i'$, and therefore $d(L, L') = d(L_i, L_i') = \delta_1$. Since $|L_i| \le m$ and $|L_i'| \le n/r < n$, the claim follows again by the induction hypothesis. This completes the proof of the lemma. □

Hence, we obtain the following result.

Theorem 3.5 *Given a set L of m red lines and a set L' of n blue lines in $\mathbb{R}^3$, such that all red lines lie above all blue lines, $d(L, L')$ can be computed by a randomized algorithm in $O(m^{3/4+\varepsilon} n^{3/4+\varepsilon} + m^{1+\varepsilon} + n^{1+\varepsilon})$ expected time, for any $\varepsilon > 0$.*

Remark 3.6 Note that algorithm CLOSEST-PAIR and its analysis do not use the fact that L and L' are sets of lines in $\mathbb{R}^3$, and that $d(.,.)$ is the Euclidean distance function. In fact, Theorem 3.5 holds in a more general setting as long as L, L', and the distance function satisfy the following properties:

(i) Each object ℓ in L is specified by 4 real parameters, so that ℓ can be mapped to a point $\varphi(\ell)$ in $\mathbb{R}^4$.

(ii) For any parameter $\delta > 0$ and any object $\ell' \in L'$, one can define a trivariate polynomial $f_{\ell'}(x_1, x_2, x_3)$ of constant degree such that, for any $\ell \in L$ with $\varphi(\ell) = (a_1, a_2, a_3, a_4)$, $a_4 \le f_{\ell'}(a_1, a_2, a_3)$ if and only if $d(\ell, \ell') \le \delta$.

(iii) For any subset $R \subseteq L'$ of size r and $\delta > 0$, the vertical decomposition of the minimization diagram of the set $\{f_{\ell'} \mid \ell \in R\}$ has $O(r^{3+\varepsilon})$ cells.

Combining this result with the observations in [7], which we omit here, and which show how the width itself can be computed using Algorithm CLOSEST-PAIR as a subroutine, we obtain the main result of this section.

Corollary 3.7 *The width of a set of n points in $\mathbb{R}^3$ can be computed by a randomized algorithm whose expected running time is $O(n^{3/2+\varepsilon})$, for any $\varepsilon > 0$.*

4 Minimum Width Annulus

Given a set $S = \{p_1, \ldots, p_n\}$ of n points in the plane, we want to compute an annulus of the smallest width that contains S. That is, we want to compute two concentric circles, centered at a point ξ, of radii r_1, r_2, respectively, such that $r_2 - r_1$ is minimized, subject to the constraints $r_1 \le d(\xi, p_i) \le r_2$, for each $1 \le i \le n$.

For a given point $\mathbf{x}$, let $\omega(\mathbf{x})$ denote the smallest width of an annulus containing S and centered at $\mathbf{x}$. Let p_c, p_f be the nearest and the farthest neighbors of $\mathbf{x}$ in S, respectively. Then

$$\omega(\mathbf{x}) = d(\mathbf{x}, p_f) - d(\mathbf{x}, p_c) .$$

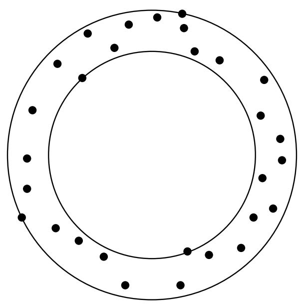

Figure 1: A minimum width annulus

Let $Vor_c(S)$ (resp. $Vor_f(S)$) denote the closest (resp. farthest) point Voronoi diagram of S. Ebara et al. [15] observed that the center of a minimum width annulus containing S is a vertex of $Vor_c(S)$, a vertex of $Vor_f(S)$, or an intersection point of an edge of $Vor_c(S)$ and an edge of $Vor_f(S)$ (see also [29]). Based on this observation, they gave a rather simple $O(n^2)$-time algorithm for computing a minimum width annulus, which was improved by Agarwal et al. [3] to $O(n^{8/5+\varepsilon})$, and subsequently by Agarwal et al. [1] to $O(n^{17/11+\varepsilon})$. Here we obtain a further improved algorithm that runs in $O(n^{3/2+\varepsilon})$ expected time, for any $\varepsilon > 0$, using the machinery developed in the previous section. If one is interested in minimizing the area of the annulus, instead of the width, the problem can be formulated as an instance of linear programming in $\mathbb{R}^4$, and therefore can be solved in $O(n)$ time, using Megiddo's algorithm [25]. See also [21, 30] for other related results.

We first make the following observation, which was missed in the earlier treatments of the problem cited above, and which transforms the problem into a width-like problem in $\mathbb{R}^3$. We lift the points of S to the paraboloid $z = x^2 + y^2$ by the standard lifting map $(x, y) \mapsto (x, y, x^2 + y^2)$.Tthen a circle C of radius r with center (a, b) is mapped to the plane $C^* :\ z = 2ax + 2by + (r^2 - a^2 - b^2)$. A point $p \in \mathbb{R}^2$ lies inside (resp. on, outside) C if and only if its lifted image p^* lies below (resp. on, above) the plane C^*. Thus an annulus with center (a,b) and radii $r_1 < r_2$ and containing S is mapped to a pair of parallel planes

$$z = 2ax + 2by + (r_1^2 - a^2 - b^2)$$

and

$$z = 2ax + 2by + (r_2^2 - a^2 - b^2),$$

such that the image S^* of S is fully contained in the slab bounded between these two planes. In other words, the minimum-width annulus problem reduces to the problem of finding a pair of parallel planes

$$\pi_1 : z = \kappa_1 x + \kappa_2 y + \kappa_3' \quad \text{and} \quad \pi_2 : z = \kappa_1 x + \kappa_2 y + \kappa_3$$

that supports S^*, such that

$$g(\pi_1, \pi_2) = \sqrt{\kappa_3' + \frac{\kappa_1^2 + \kappa_2^2}{4}} - \sqrt{\kappa_3 + \frac{\kappa_1^2 + \kappa_2^2}{4}} \tag{4.1}$$

is minimized.

For a pair of (skew) lines ℓ_1, ℓ_2 in $\mathbb{R}^3$, we define the distance between ℓ_1 and ℓ_2 to be $d(\ell_1, \ell_2) = g(\pi_1, \pi_2)$ where π_1, π_2 are parallel planes containing ℓ_1, ℓ_2, respectively. Using the observation in [7], this width-like problem can also be formulated as the bichromatic closest line-pair problem defined in the beginning of Section 3 (i.e., given a set L of m lines in $\mathbb{R}^3$ and another set L' of n lines in $\mathbb{R}^3$ such that all lines in L lie above all blue lines in L', compute a closest pair between L and L'), except that we use the distance function just described. An explicit expression for $d(\ell, \ell')$ can be obtained as follows. Let the equations defining ℓ, ℓ' be, as in Section 3,

$$\begin{aligned} \ell: &\quad y = a_1 x + a_3, \qquad z = a_2 x + a_4, \\ \ell': &\quad y = b_1 x + b_3, \qquad z = b_2 x + b_4. \end{aligned}$$

The direction of the common normal to both lines is

$$\mathbf{n} = (a_1 b_2 - a_2 b_1,\ a_2 - b_2,\ -(a_1 - b_1)) \ ,$$

and the planes orthogonal to $\mathbf{n}$ and containing ℓ, ℓ', respectively, are

$$(\mathbf{x} - (0, a_3, a_4)) \cdot \mathbf{n} = 0 \quad \text{and} \quad (\mathbf{x} - (0, b_3, b_4)) \cdot \mathbf{n} = 0 \ ,$$

or

$$z = \frac{a_1 b_2 - a_2 b_1}{a_1 - b_1} x + \frac{a_2 - b_2}{a_1 - b_1} y - \frac{a_3(a_2 - b_2) - a_4(a_1 - b_1)}{a_1 - b_1}$$

and

$$z = \frac{a_1 b_2 - a_2 b_1}{a_1 - b_1} x + \frac{a_2 - b_2}{a_1 - b_1} y - \frac{b_3(a_2 - b_2) - b_4(a_1 - b_1)}{a_1 - b_1}.$$

Plugging this into (4.1), we thus can write

$$d(\ell, \ell') = \sqrt{\mu a_3 + a_4 + \nu} - \sqrt{\mu b_3 + b_4 + \nu} \ ,$$

where μ and ν depend only on a_1, a_2, b_1, b_2.

In order to use algorithm CLOSEST-PAIR, we need to prove an appropriate variant of Lemma 3.1. First, we note that if ℓ' is fixed then $d(\ell, \ell')$ is defined whenever the ξ_1-coordinate of ℓ is different than that of ℓ'. In this case $d(\ell, \ell')$ monotonically increases with a_4, which implies that the appropriate variant of the functions $f_{\ell'}$ is well-defined (whenever the ξ_1-coordinates of ℓ and of ℓ' are different). To establish the second part of the lemma, for a fixed parameter δ, let ℓ'_1, ℓ'_2 be two given lines, let ξ_1 and ξ_2 be fixed (with ξ_1 different from the ξ_1-coordinates of ℓ'_1 and of ℓ'_2), and consider the equation

$$f_{\ell'_1}(\xi_1, \xi_2, \xi_3) = f_{\ell'_2}(\xi_1, \xi_2, \xi_3) \ . \tag{4.2}$$

We seek a line ℓ that lies above ℓ'_1 and ℓ'_2 and satisfies $d(\ell, \ell'_1) = d(\ell, \ell'_2) = \delta$. This can be written as

$$\begin{aligned} \sqrt{\mu_1 \xi_3 + \xi_4 + \nu_1} &= \sqrt{\mu_1 b_3^{(1)} + b_4^{(1)} + \nu_1} + \delta, \\ \sqrt{\mu_2 \xi_3 + \xi_4 + \nu_2} &= \sqrt{\mu_2 b_3^{(2)} + b_4^{(2)} + \nu_2} + \delta \ , \end{aligned}$$

where $b^{(1)}$, $b^{(2)}$ are the 4-tuples defining ℓ'_1, ℓ'_2, respectively, and μ_1, ν_1, μ_2, ν_2 are independent of ξ_3, ξ_4. We thus get a linear system of equations in ξ_3, ξ_4, which can be easily reduced to a linear equation in ξ_3 of the form $(\mu_1 - \mu_2)\xi_3 = \alpha$. This shows that (4.2) has a unique solution, unless $\mu_1 = \mu_2$, or

$$\frac{\xi_2 - b_2^{(1)}}{\xi_1 - b_1^{(1)}} = \frac{\xi_2 - b_2^{(2)}}{\xi_1 - b_1^{(2)}} \ ,$$

which is the equation of a line in the $\xi_1\xi_2$-plane.

We have thus shown that conditions (F1) and (F2) of Section 2, and therefore the conditions of Remark 3.6, are satisfied in this case too, so the algorithms and the analysis of the preceding section apply here as well. We thus obtain the following result.

Theorem 4.1 *An annulus of smallest width that contains a given set S of n points in the plane can be computed by a randomized algorithm in $O(n^{3/2+\varepsilon})$ expected time, for any $\varepsilon > 0$.*

5 Biggest Stick in a Polygon

The *biggest stick* problem is to find the longest segment that can be placed inside a simple n-gon P in the plane. Chazelle and Sharir gave an $O(n^{1.99})$-time algorithm [11], which was

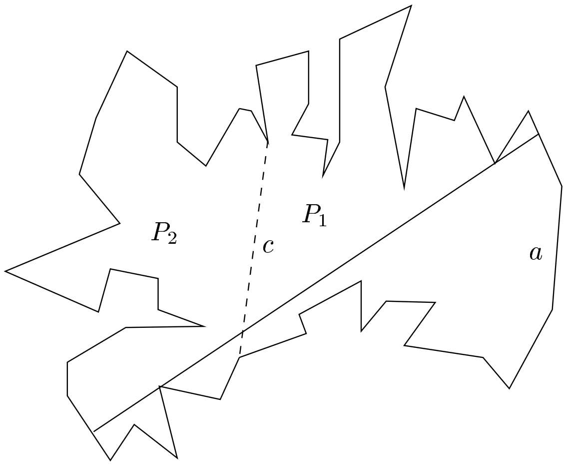

Figure 2: Biggest stick in a polygon

improved by Agarwal et al. [1] to $O(n^{17/11+\varepsilon})$; see also [3] for an intermediate bound. If the endpoints of the segment are restricted to be at vertices of P, the problem becomes considerably easier, and can be solved in $O(n \log^3 n)$ time [4].

Following the same idea as in [3], we use a divide-and-conquer approach. Partition P into two simple polygons P_1, P_2 by a chord c, so that each of P_1, P_2 has at most $2n/3$ vertices. Let l_c be the line passing through c. We recursively compute the longest segment that can be placed within P_1 or within P_2. The merge step requires computing the longest segment having one endpoint in P_1 and the other in P_2. An easy perturbation argument shows that such a longest segment has to touch two vertices of P, say, v_1, v_2. The difficult case is when $v_1 \in P_1$ and $v_2 \in P_2$ (the case where both v_1 and v_2 belong to the same subpolygon is easy, because there can be only $O(n)$ such pairs of vertices — see [3]). Agarwal et al. [3] showed that finding such a segment can be reduced to the following problem: We have a set L_1 of a lines in the plane, where each line $\ell_i \in L_1$ is dual to some vertex v_i of P_1 and has an edge $e_i \in P_1$ associated with it. Similarly, we have another set L_2 of b lines in the plane, where each line $\ell_j \in L_2$ is dual to some vertex v_j of P_2 and has an edge e_j of P_2 associated with it. L_1 and L_2 satisfy the following property: For any pair $\ell_i \in L_1, \ell_j \in L_2$, the segment $g_{ij} = a_i a_j$ lies inside P, where a_i (resp. a_j) is the intersection point of e_i (resp. e_j) with the line passing through v_i and v_j; see Figure 3. The goal is to compute the longest segment g_{ij}, over all pairs $\ell_i \in L_1, \ell_j \in L_2$. If we define the distance function $d(\ell_i, \ell_j)$ as the length of the segment g_{ij}, then the goal is to compute a farthest pair in $L_1 \times L_2$. The reduction of the original subproblem yields many instances of this problem, and the biggest stick that crosses the chord c is the largest of all longest segments g_{ij}, over all instances.

It can be shown that L_1, L_2 can be parametrized by 4 real parameters (two for the

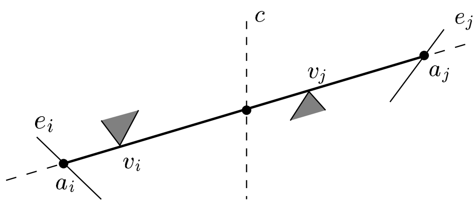

Figure 3: Illustration of the distance function $d(\ell_i, \ell_j)$

associated vertex of P and two for the associated edge), so that the setup of Remark 3.6 arises here as well. In more detail, each line $\ell_i \in L_1$, associated with a vertex v_i and an edge e_i, is mapped to a point $(a_1, a_2, a_3, a_4) \in \mathbb{R}^4$, where a_1, a_2 are the coordinates of the vertex v_i and where the equation of the line containing e_i is $y = a_3x + a_4$ (we assume, without loss of generality, that no such line is vertical). Let w be a real parameter. For each line $\ell_j \in L_2$, associated with a vertex v_j and an edge e_j, we define a trivariate function f_j such that $x_4 = F(x_1, x_2, x_3)$ if and only if the following holds: Let v be the point (x_1, x_2), λ be the line $y = x_3x + x_4$, and a (resp. b) be the intersection point of the line containing e_j (resp. λ) with the line passing through the points v_j and v. Then $d(a, b) = w$.

Lemma 5.1 *The set $\mathcal{F} = \{f_j \mid \ell_j \in L_2\}$ satisfies (F1) and (F2).*

Proof: To establish (F1), we need to show that, for a given line $\ell_j \in L_2$ and for fixed x_1, x_2, the value $f_j(x_1, x_2, x_3)$ is defined for all x_3, except when (x_1, x_2) lies on any of a constant number of algebraic arcs of constant maximum degree. Since the point (x_1, x_2) is fixed, the line λ passing through v_j and (x_1, x_2) is also fixed. If λ is parallel to e_j then $f_j(x_1, x_2, x_3)$ is undefined for all x_3. Otherwise, there exists a unique point p on λ at distance w from the intersection of λ with the line containing e_j and lying to the left of v_j-1-, and for any [1]: Shouldn't it be e? x_3 there is a unique line $y = x_3x + x_4$ passing through p, so $f_j(x_1, x_2, x_3)$ is defined for all x_3. (There is a small technical issue here, in which we would like to exclude the case where $y = x_3x + x_4$ coincides with λ. The analysis of [3] does indeed allow us to ignore this possibility.)

Next, to establish (F2), we need to show that, for two given lines $\ell_j, \ell_k \in L_2$, and for any fixed (x_1, x_2), not lying on any of the arcs where f_j or f_k is undefined, the equation

$$f_j(x_1, x_2, x_3) = f_k(x_1, x_2, x_3)$$

has at most one solution x_3. Assume that both functions are defined over $(x_1, x_2) \times \mathbb{R}$. Suppose that x_3 is a solution of this equation, and that $x_4 = f_j(x_1, x_2, x_3)$. Let v be the fixed point (x_1, x_2). Let s_j (resp. s_k) be the unique segment emanating from a point on e_j (resp. on e_k) and passing through v_j and v (resp. through v_k and v) whose length is w. Let p_j (resp. p_k) be the other endpoint of s_j (resp. s_k) (not the one lying on e_j (resp. on e_k)).

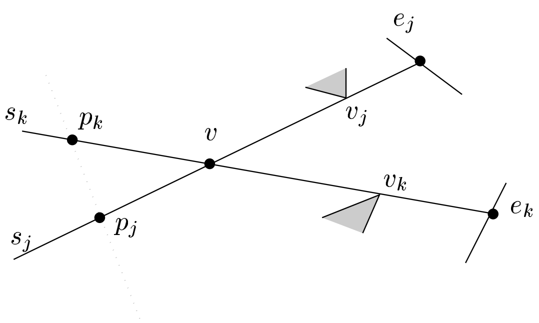

Figure 4: Illustration of the case $f_j(x_1, x_2, x_3) = f_k(x_1, x_2, x_3)$

Then the line $y = x_3x + x_4$ is the unique line connecting p_j and p_k. This is fine as long as both p_j and p_k are different from v. (If $p_j = p_k = v$, then $y = x_3x + x_4$ is undefined; if, say, only $p_j = v$ then $y = x_3x + x_4$ passes through v and through one of the points v_j, v_k, a situation that we wish to exclude in our analysis (as in the preceding paragraph).) In other words, we do not want v to have the property that the length of one of the segments connecting v to e_j through v_j or to e_k through v_k is equal to w. This implies that, as long as v does not lie on one of two corresponding algebraic curves (these curves are of degree 4, and are known as 'conchoids of Nicomedes' [22]), the above equation has a unique solution.

This completes the proof of the lemma. $\square$

Remark 5.2 In the application of Lemma 5.1, the point (x_1, x_2) is restricted to a half-plane bounded by the line containing the dividing chord c, so the functions of $\mathcal{F}$ are defined only on a corresponding halfspace of $\mathbb{R}^4$. We leave it to the reader to verify that the machinery developed in earlier sections is applicable in this case too.

In view of the above lemma, the conditions of Remark 3.6 are satisfied here, so we can apply a variant of algorithm CLOSEST-PAIR to find a farthest pair in L_1 and L_2. Omitting the further, easy and technical details, we obtain the following result.

Theorem 5.3 *The longest segment that can be placed inside a given simple n-gon P can be computed by a randomized algorithm in $O(n^{3/2+\varepsilon})$ expected time, for any $\varepsilon > 0$.*

6 Conclusion

In this paper we presented efficient randomized algorithms for a number of geometric optimization problems. These algorithms are not only asymptotically faster than the previously best known algorithms, but they are simpler as well. Our technique also yields simpler algorithms for many other geometric optimization problems, such as computing a closest line pair in 3-space, etc. [7, 26].

We conclude this paper by mentioning some open problems

(i) Can our algorithms be derandomized without affecting their running time significantly?

(ii) Computing a minimum-width annulus containing a set S of points in the plane is equivalent to fitting a circle C through S, such that the maximum distance between the points of S and C is minimized. However, in many applications, such as computational metrology [31], one is interested in minimizing the sum of squares of distances between C and the points of S. We are not aware of any efficient algorithm for this problem.

(iii) There are various interesting extensions of the biggest stick problem, which are worth pursuing. For example, given a simple polygon P and a triangle T, is there a subquadratic algorithm that determines whether T can be placed inside P, allowing both translations and rotations? A quadratic algorithm for this problem was given by Chazelle [6]. Another related problem is: Given a segment e and a polygon P with holes, determine, in subquadratic time, whether e can be placed inside P, using translations and rotations. A quadratic solution can be found in [28].

References

[1] P. Agarwal, B. Aronov and M. Sharir, Computing lower envelopes in four dimensions with applications, *Proc. 10th Annual Symp. on Computational Geometry*, 1994, 348–358.

[2] P. Agarwal, O. Schwarzkopf and M. Sharir, The overlay of lower envelopes in three dimensions and its applications, *Proc. 11th Annual Symp. on Computational Geometry*, 1995, 182–189.

[3] P. Agarwal, M. Sharir, and S. Toledo, New applications of parametric searching in computational geometry, *J. Algorithms* 17 (1994), 292–318.

[4] A. Aggarwal and S. Suri, The biggest diagonal in a simple polygon, *Inf. Proc. Letters* 35 (1990), 13–18.

[5] H. Brönnimann and B. Chazelle, Optimal slope selection via cuttings, *Proc. 6th Canadian Conf. on Computational Geometry*, 1994, 99–103.

[6] B. Chazelle, The polygon containment problem, in: *Advances in Computing Research, Vol. I: Computational Geometry* (F. Preparata, ed.), JAI Press, Greenwich, CT, 1993, 1–33.

[7] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir, Diameter, width, closest line pair, and parametric searching, *Discrete Comput. Geom.* 10 (1993), 183–196.

[8] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir, A singly exponential stratification scheme for real semi–algebraic varieties and its applications, *Proc. 16th Int. Colloq. on Automata, Languages and Programming*, (1989), 179–192. Lecture Notes in Computer Sciences, vol. 371, Springer-Verlag, Berlin. (Also in *Theoretical Computer Science* 84 (1991), 77–105.)

[9] B. Chazelle and M. Sharir, An algorithm for generalized point location and its applications, *J. Symbolic Computation* 10 (1990), 281–309.

[10] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir and E. Welzl, Combinatorial complexity bounds for arrangements of curves and spheres, *Discrete Comput. Geom.* 5 (1990), 99–160.

[11] K. Clarkson and P. Shor, Applications of random sampling in computational geometry II, *Discrete Comput. Geom.* 4 (1989), 387–421.

[12] M. Dillencourt, D. Mount, and N. Netanyahu, A randomized algorithm for slope selection, *Int. J. Comput. Geom. and Appls* 2 (1992), 1–27.

[13] M. de Berg, K. Dobrindt, and O. Schwarzkopf, On lazy randomized incremental construction, *Proc. 26th Annual ACM Symp. on Theory of Computing,* 1994, 105–114.

[14] M. de Berg, D. Halperin, and L. Guibas, Vertical decomposition for triangles in 3-space, *Proc. 10th Annual Symp. Computational Geometry*, 1994, 1–10.

[15] H. Ebara, N. Fukuyama, H. Nakano, and Y. Nakanishi, Roundness algorithms using the Voronoi diagrams, *First Canadian Conf. on Computational Geometry*, 1989.

[16] D. Halperin and M. Sharir, New bounds for lower envelopes in 3 dimensions, with applications to visibility in terrains, *Discrete Comput. Geom.* 12 (1994), 313–326.

[17] D. Haussler and E. Welzl, ϵ-nets and simplex range queries, *Discrete Comput. Geom.* 2 (1987), 127–151.

[18] M. Houle and G. Toussaint, Computing the width of a set, *IEEE Transactions on Pattern Anal. and Mach. Intell.* 5 (1988), 761–765.

[19] M. Katz and M. Sharir, Optimal slope selection via expanders, *Inf. Proc. Letters* 47 (1993), 115–122.

[20] M. Katz and M. Sharir, An expander-based approach to geometric optimization, *Proc. 9th Annual Symp. on Computational Geometry*, 1993, 198–207.

[21] V. B. Le and D. T. Lee, Out-of-roundness problem revisited, *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-13 (1991), 217–223.

[22] E.H. Lockwood, *A Book of Curves*, Cambridge University Press, Cambridge, 1967.

[23] J. Matoušek, Randomized optimal algorithm for slope selection, *Info. Proc. Letters* 39 (1991), 183–187.

[24] N. Megiddo, Applying parallel computation algorithms in the design of serial algorithms, *J. ACM* 30 (1983), 852–865.

[25] N. Megiddo, Linear programming in linear time when the dimension is fixed, *J. Assoc. Comput. Mach.* 31 (1984), 114–127.

[26] M. Pellegrini, On collision-free placement of simplices and the closest pair of lines in 3-space, *SIAM J. Comp.* 23 (1994), 133–153.

[27] M. Sharir, Almost tight upper bounds for lower envelopes in higher dimensions, *Discrete Comput. Geom.* 12 (1994), 327–345.

[28] M. Sharir and P. Agarwal, *Davenport-Schinzel Sequences and Their Geometric Applications*, Cambridge University Press, Cambridge-New York-Melbourne, 1995.

[29] M. Smid and R. Janardan, On the width and roundness of a set of points in the plane, Tech. Rept. MPI-I-94-111, Max-Planck-Institut für Informatik, 1994.

[30] K. Swanson, An optimal algorithm for roundness determination on convex polygons, *Proc. 3rd Workshop Algorithms Data Structures* 1993, 601–609.

[31] G. Wilfong, private communication.