Almost Tight Bounds for Eliminating Depth Cycles in Three Dimensions*

Boris Aronov[†] Micha Sharir[‡]

December 2, 2015

Abstract

Given n non-vertical lines in 3-space, their vertical depth (above/below) relation can contain cycles. We show that the lines can be cut into $O(n^{3/2} \operatorname{polylog} n)$ pieces, such that the depth relation among these pieces is now a proper partial order. This bound is nearly tight in the worst case. As a consequence, we deduce that the number of *pairwise non-overlapping cycles*, namely, cycles whose xy-projections do not overlap, is $O(n^{3/2} \operatorname{polylog} n)$; this bound too is almost tight in the worst case.

Previous results on this topic could only handle restricted cases of the problem (such as handling only triangular cycles, by Aronov, Koltun, and Sharir, or only cycles in grid-like patterns, by Chazelle et al.), and the bounds were considerably weaker—much closer to quadratic.

Our proof uses a recent variant of the polynomial partitioning technique, due to Guth, and some simple tools from algebraic geometry. It is much more straightforward than the previous "purely combinatorial" methods.

Our technique extends to eliminating all cycles in the depth relation among segments, and of constant-degree algebraic arcs. We hope that a suitable extension of this technique could be used to handle the (much more difficult) case of pairwise-disjoint triangles.

Our results almost completely settle a long-standing (35 years old) open problem in computational geometry, motivated by hidden-surface removal in computer graphics.

*Work on this paper by B.A. has been partially supported by NSF Grants CCF-11-17336, CCF-12-18791, and CCF-15-40656, and by BSF grant 2014/170. Work by M.S. has been supported by Grant 2012/229 from the U.S.-Israel Binational Science Foundation, by Grant 892/13 from the Israel Science Foundation, by the Israeli Centers for Research Excellence (I-CORE) program (center no. 4/11), and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University.

[†]Department of Computer Science and Engineering, Tandon School of Engineering, New York University, New York, NY, USA; *boris.aronov@nyu.edu*.

[‡]Blavatnik School of Computer Science, Tel Aviv University, Tel-Aviv 69978, Israel; *michas@post.tau.ac.il*.

1 Introduction

The problem. Let $\mathcal{L}$ be a collection of n non-vertical lines in $\mathbb{R}^3$ in *general position*. In particular, we assume that no two lines in $\mathcal{L}$ intersect, that the xy-projections of no pair of the lines are parallel, and those of no three of the lines are concurrent. For any pair ℓ, ℓ' of lines in $\mathcal{L}$, we say that ℓ passes *above* ℓ' (equivalently, ℓ' passes *below* ℓ) if the unique vertical line that meets both ℓ and ℓ' intersects ℓ at a point that lies higher than its intersection with ℓ'. We denote this relation as $\ell' \prec \ell$. The relation $\prec$ is total (under our assumptions), but in general it need not be transitive, so it may contain *cycles* of the form $\ell_1 \prec \ell_2 \prec \cdots \prec \ell_k \prec \ell_1$. We call this a *k-cycle*, and refer to k as the *length* of the cycle. Cycles of length three are called *triangular*. See Figure 1.

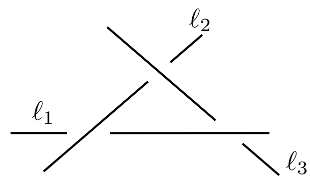

Figure 1: A triangular depth cycle, viewed from above.

If we cut the lines of $\mathcal{L}$ at a finite number of points, we obtain a collection of lines, segments, and rays. We can extend the definition of the relation $\prec$ to the new collection in the obvious manner, except that it is now only a partial relation. Our goal is to cut the lines in such a way that $\prec$ becomes a *partial order*, in which case we call it a *depth order*. We note that it is trivial to construct a depth order with $\Theta(n^2)$ cuts: Simply cut each line near every point whose xy-projection is a crossing point with another projected line. It is desirable though to minimize the number of cuts. A long-standing conjecture, open since 1980, is that one can always construct a depth order with a *subquadratic* number of cuts. In this paper we finally settle this conjecture, in a strong, almost worst-case tight manner; see below for precise details.

Background. The main motivation for studying this problem comes from *hidden surface removal* in computer graphics. A detailed description of this motivation can be found, e.g., in the earlier paper of Aronov et al. [2]. Briefly, a conceptually simple technique for rendering a scene in computer graphics is the so-called Painter's Algorithm, which places the objects in the scene on the screen in a back-to-front manner, painting each new object over the portions of earlier objects that it hides. For this, though, one needs an acyclic depth relation among the objects with respect to the viewing point (which, as we assume in this paper, without loss of generality, lies at $z = -\infty$). When there are cycles in the depth relation, one would like to cut the objects into a small number of pieces, so as to eliminate all cycles, and then paint the pieces in the above manner, obtaining a correct rendering of the scene; see [2, 5] for more details.

The study of cycles in a set of lines in $\mathbb{R}^3$ goes back to Chazelle et al. [8], who have shown that, if the xy-projections of a collection of n segments in 3-space form a "grid" (see

Figure 2), then all cycles defined by this collection can be eliminated with $O(n^{9/5})$ cuts.

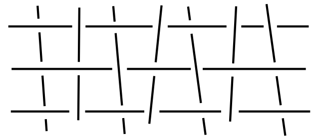

Figure 2: A collection of line segments that forms a grid, viewed from above.

Another significant development is due to Aronov et al. [2], who have considered the problem of triangular cycles, and established the rather weak (albeit subquadratic) $O(n^{2-1/34}\log^{8/17} n)$ upper bound on the number of *elementary triangular* cycles (namely, cycles whose *xy*-projections form triangular faces in the arrangement of the projected lines). They also showed that $O(n^{2-1/69}\log^{16/69} n)$ cuts suffice to eliminate all triangular cycles. Finally, combining this bound with earlier algorithmic techniques of Solan [15] and of Har-Peled and Sharir [11], they have obtained an algorithm that eliminates all triangular cycles by making roughly $O(n^{2-1/138})$ cuts. However, their results did not apply to general (non-triangular) cycles, and, in addition to the very weak bounds just stated, the proof technique was very complicated. Just as the analysis of Chazelle et al. [8], it used, as a major ingredient, the impossibility of certain “weaving patterns” of lines in space, an interesting and intriguing topic in itself. Unfortunately, it appears that arguments based on forbidden weaving patterns lead to fairly weak bounds.

Our contribution. In this paper we settle the general problem and show that *all* cycles in a set of n lines can be eliminated with $O(n^{3/2}\operatorname{polylog} n)$ cuts. A simple and well-known construction, detailed below, yields a scenario where $\Omega(n^{3/2})$ cuts have to be made, implying that ours is the best possible worst-case bound, up to the polylogarithmic factor.

The proof of the new bound is embarrassingly straightforward. It uses tools from algebraic geometry, in the spirit of much recent work that exploits similar ideas; see, e.g., the simple proofs in [12, 14] for the corresponding worst-case tight bound of $\Theta(n^{3/2})$ on the number of so-called *joints* in a collection of n lines in 3-space. At the heart of the construction lies a recent result of Guth [9], which extends the basic *polynomial partitioning* technique of Guth and Katz [10] to higher-dimensional objects (lines or curves in our case).

As a matter of fact, the algebraic approach to this problem is fairly versatile and can be extended to the elimination of cycles involving more general objects. In this paper we also apply it to the cases of line segments (this is in fact a trivial extension) and of constant-degree algebraic arcs. Furthermore, in both cases, by combining our technique with standard tools for constructing output-sensitive cuttings in the plane, we obtain improved bounds on the number of cuts, which depend on the number of intersections among the *xy*-projections of the segments or arcs. See Theorems 4.1 and 4.2.

We note that the practical motivation arising from computer graphics involves data sets consisting of (pairwise openly-disjoint) triangles. However, eliminating cycles in the depth relation of a collection of triangles is a considerably more difficult problem, which so far seems to be out of reach. Still, we hope that the technique presented in this paper could be extended to tackle this case too.

We also note that the problem studied here is different from most of the questions tackled so far by the new algebraic approach, in that they involve *incidences* between points, lines, and other objects. In contrast, in this paper the lines are not incident to one another, and the configurations that we want to capture involve certain spatial (here, "above/below") relationships between them. It is our hope that this study will find applications to additional problems involving relations more general than incidences.

Our proof is constructive, and leads, in principle, to an efficient algorithm for performing the cuts. The only currently missing ingredient is an efficient construction of Guth's partitioning polynomial, a step that we leave as a topic for further research. (The problematic aspects of effectively constructing a partitioning polynomial, for the simpler case of a set of points, and techniques for overcoming these issues, are discussed by Agarwal et al. [1]; one hopes that variants of these techniques could also be used for effectively partitioning lines or curves, and we are presently investigating this question.)

Alternatively, to identify the cuts sufficient to eliminate all cycles, one could also use the earlier algorithms of Har-Peled and Sharir [11] or of Solan [15]. Our analysis implies that they perform $O(n^{7/4} \operatorname{polylog} n)$ cuts (using the algorithm of [11], the one in [15] generates slightly more cuts), in expected time $O(n^{11/6+\varepsilon})$, for any $\varepsilon > 0$, significantly improving previous bounds, but still falling short of the ideal goal of coming close to the worst-case optimal number of cuts.

There is yet another interesting approach to our problem, which is to use the standard greedy algorithm for hitting sets in hypergraphs, and its analysis by Lovász [13]. This analysis requires bounds on the maximum size of so-called "simple k-matchings." In our context, a *simple k-matching* is a collection of cycles whose xy-projections have the property that no portion of any projected line is shared by more than k of them; for the case $k = 1$, we refer to such cycles as *pairwise non-overlapping*. An immediate consequence of our analysis is that the maximum size of a family of pairwise non-overlapping cycles is $O(n^{3/2} \operatorname{polylog} n)$, and a further (easy) extension shows that the maximum size of a simple k-matching of cycles is $O(kn^{3/2} \operatorname{polylog} n)$. Unfortunately, this approach appears not to lead to an immediate sharp bound on the number of cuts, and its naïve implementation runs in exponential time, but it has other merits worth noting (for instance, it can eliminate all k-cycles, for k at most some prescribed constant, in polynomial time, using $O(n^{3/2} \operatorname{polylog} n)$ cuts). We present and discuss this technique in the appendix.

Paper organization. Section 2 presents the main result on the number of cuts sufficient to eliminate all depth cycles among lines in $\mathbb{R}^3$. Section 3 discusses the algorithmic aspects of efficiently finding such a set of cuts. Some of this discussion, involving the hitting-set approach, is deferred to the appendix. Finally, Section 4 discusses the extensions of our technique to the cases of line segments and of constant-degree algebraic arcs.

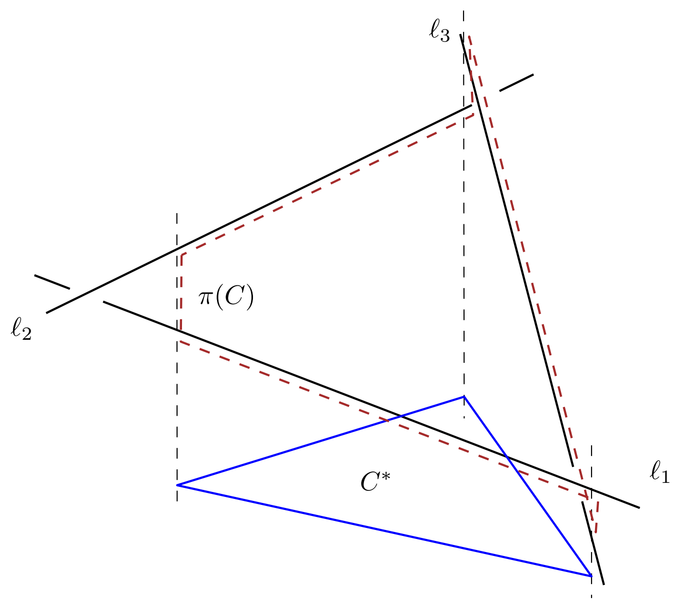

Figure 3: Cycle $C\colon \ell_1 \prec \ell_2 \prec \ell_3 \prec \ell_1$ (thick lines), with the corresponding path $\pi(C)$ (in dashed brown), and its projection C^* (solid blue).

2 Eliminating all cycles

We first introduce a few definitions. Let $\mathcal{L}$ be a collection of n non-vertical lines in $\mathbb{R}^3$ in general position. For each $\ell \in \mathcal{L}$, denote by ℓ^* the xy-projection of ℓ and by $\mathcal{L}^*$ the collection of the n resulting projections. The general position assumption on $\mathcal{L}$ implies that $\mathcal{L}^*$ is also in general position. Consider the planar arrangement $\mathcal{A}(\mathcal{L}^*)$ of $\mathcal{L}^*$.

Recall that k distinct lines $\ell_1, \ldots, \ell_k$ form a *k-cycle* C if $\ell_1 \prec \ell_2 \prec \cdots \prec \ell_k \prec \ell_1$. We can interpret C as a spatial object as follows. For each $i = 1, \ldots, k$, let $v_i^+ \in \ell_i$ and $v_{i+1}^- \in \ell_{i+1}$ (with indices treated $\bmod k$) be the two unique points on these lines that are vertically above each other (informally, C "jumps upwards" from v_i^+ on ℓ_i to v_{i+1}^- on ℓ_{i+1}). Then we associate with C the closed polygonal path

$$\pi(C) := v_1^- v_1^+ v_2^- v_2^+ \cdots v_k^- v_k^+ v_1^- .$$

Let e_i denote the segment $v_i^- v_i^+$ on ℓ_i. Then $\pi(C)$ alternates between the segments e_i and the vertical jumps $v_i^+ v_{i+1}^-$; see Figure 3.

The xy-projection C^* of $\pi(C)$ (or, with a slight abuse of notation, of C) is a closed polygonal path contained in $\cup\mathcal{L}^*$. That is, it is the concatenation of the projections e_i^* of the segments e_i (the vertical segments disappear in the projection).

The path C^* can be fairly arbitrary, non-convex and even self-crossing. Nevertheless, we claim that, for the purposes of eliminating all cycles, it suffices to consider only *simple cycles*,[1] that is, cycles C for which C^* is non-self-crossing. This is because any other cycle C

[1] This is a slight abuse of terminology, as we require the projection C^* of the cycle C to be simple, rather than C itself.

can be shortcut into a cycle C_0, such that (a) C_0 has fewer edges than C, and (b) $C_0^* \subset C^*$. Clearly, any cut that eliminates C_0 also eliminates C. We repeat this reduction until we obtain a simple cycle (in the extreme, we reach the case where C_0 is triangular, and thus simple). Indeed, if C^* is self-crossing, let w be a point where C^* crosses itself, and let ℓ, ℓ' be the lines whose projections cross at w; see Figure 4. Then C^* is split at w into two smaller closed paths, and it is easily checked that one of them is the projection of a cycle C_0 in $\mathcal{L}$ that satisfies the properties claimed above. In what follows we thus restrict ourselves to simple cycles only. Two (simple) cycles C_1, C_2 *overlap* if C_1^* and C_2^* share an edge of $\mathcal{A}(\mathcal{L}^*)$; for example, in Figure 4, cycles C and C_0 overlap.

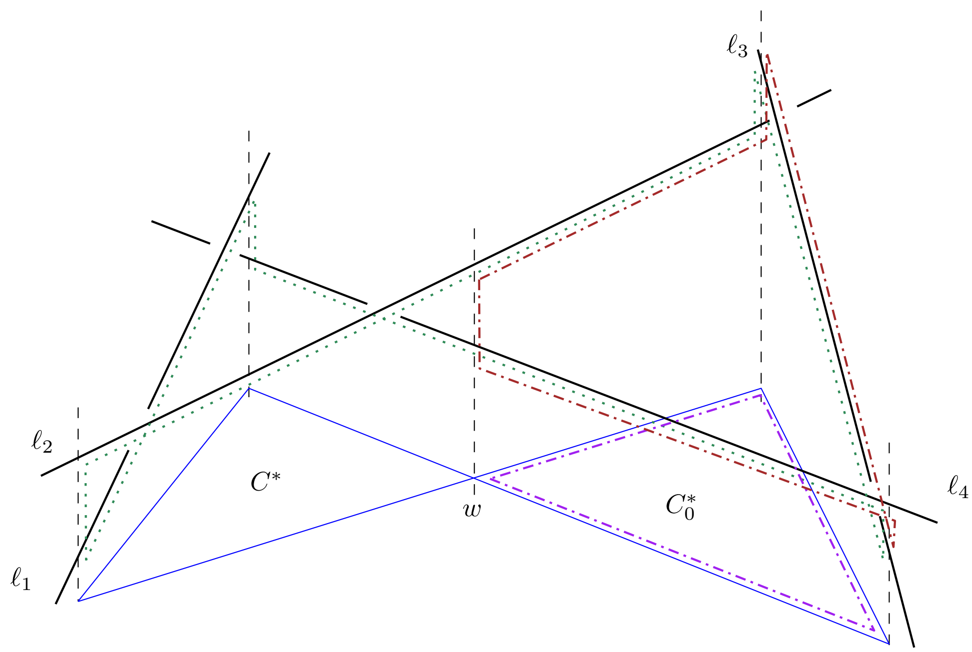

Figure 4: Cycle $C\colon \ell_1 \prec \ell_2 \prec \ell_3 \prec \ell_4 \prec \ell_1$ (thick lines, with corresponding path $\pi(C)$ in dotted green), whose projection C^* (solid blue) crosses itself. We shortcut it to a new triangular cycle $C_0\colon \ell_2 \prec \ell_3 \prec \ell_4 \prec \ell_2$, indicating the corresponding path $\pi(C_0)$ in dash-dotted brown and its projection C_0^* in dash-dotted purple.

The following is the main result of the paper.

Theorem 2.1. *Let $\mathcal{L}$ be a collection of n non-vertical lines in $\mathbb{R}^3$ in general position. Then the lines of $\mathcal{L}$ can be cut at $O(n^{3/2}\,\mathrm{polylog}\,n)$ points so that the depth relation on the resulting pieces (lines, rays, and segments) has no cycles. This bound is almost tight in the worst case.*

Remark. Theorem 2.1 also provides the same upper bound on the maximum size of a family F of pairwise non-overlapping cycles in $\mathcal{L}$, because a distinct cut is required to eliminate each cycle of F.

Proof. As argued above, it suffices to cut all simple cycles. We fix some degree D, which depends on n and will be set below, and construct a trivariate polynomial $f \in \mathbb{R}[x, y, z]$ of degree at most D, such that each of the $s = O(D^3)$ open connected components (*cells*) of $\mathbb{R}^3 \setminus Z(f)$ is intersected by at most cn/D^2 lines of $\mathcal{L}$, where $Z(f)$ denotes the zero set of f, and c is an absolute constant independent of D. By the aforementioned recent result of Guth [9], such a polynomial does exist, for some suitable constant c. (And, as noted, effective and efficient construction of such a polynomial remains to be worked out, and is the only reason this proof is not entirely polynomial-time constructive.) Let $\tau_1, \ldots, \tau_s$ be the cells of $\mathbb{R}^3 \setminus Z(f)$, and, for each i, let $\mathcal{L}_i$ denote the set of lines of $\mathcal{L}$ that intersect τ_i.

In what follows we want to exclude situations in which $Z(f)$ fully contains a vertical segment (and therefore, a line). We can guarantee that this does not happen, by applying a sufficiently small generic "tilting" to the coordinate frame, ensuring that this property holds, and that every simple cycle in $\mathcal{L}$ remains a (simple) cycle.

Define the *level* $\lambda(q)$ of a point $q \in \mathbb{R}^3$ with respect to $Z(f)$ to be the number of intersection points of $Z(f)$ with the downward-directed vertical ray ρ_q emanating from q. Formally, let (x_0, y_0, z_0) be the coordinates of q, and consider the univariate polynomial $F(z) = f(x_0, y_0, z)$. The level $\lambda(q)$ of q is the number of real zeros of F in $(-\infty, z_0)$, counted with multiplicity.

Denote by $\chi(\mathcal{L})$ the minimum number of cuts needed to eliminate all (simple) cycles in the given set $\mathcal{L}$ of lines, and put $\chi(n) := \max_{|\mathcal{L}|=n} \chi(\mathcal{L})$, where the maximum is taken over all collections $\mathcal{L}$ of n non-vertical lines in general position in $\mathbb{R}^3$.

The procedure for cutting the lines. The procedure is recursive, and follows the partitioning induced by $Z(f)$. It consists of the following steps.

(i) We cut each line $\ell \in \mathcal{L}$ not fully contained in $Z(f)$ at all its intersection points with $Z(f)$. The number of such cuts is at most D per line, for a total of $O(nD)$ cuts.

(ii) For each line $\ell \in \mathcal{L}$ not fully contained in $Z(f)$, let $h(\ell)$ be the vertical plane containing ℓ, and let g_ℓ be the bivariate polynomial obtained by restricting f to $h(\ell)$. Technically, parametrize $h(\ell)$ by coordinates (ξ_ℓ, z), where ξ_ℓ is horizontal, and each (ξ_ℓ, z) represents a point $(x_\ell(\xi_\ell), y_\ell(\xi_\ell), z)$ in $h(\ell)$, where $x_\ell(\cdot)$, $y_\ell(\cdot)$ are appropriate linear functions depending on ℓ. Then g_ℓ is given by $g_\ell(\xi_\ell, z) := f(x_\ell(\xi_\ell), y(\xi_\ell), z)$; it is a bivariate polynomial of degree at most D. By removing repeated factors, we may assume that g_ℓ is square-free. We then cut ℓ, in addition to the cuts made in step (i), at each point that lies directly above a singular point, or a point of vertical tangency, of $Z(g_\ell) \subset h(\ell)$. A simple application of Bézout's theorem implies that the number of such points is $O(D^2)$, because each such point is a common zero of g_ℓ and $(g_\ell)_z$. Note that to apply Bézout's theorem, we need to ensure that g_ℓ and $(g_\ell)_z$ do not have a common factor, which is indeed the case since we assume that g_ℓ is square-free. Hence we perform in this step $O(D^2)$ cuts of each line, for a total of $O(nD^2)$ cuts.

(iii) Assume next that $\ell \subset Z(f)$; since $Z(f)$ contains no vertical lines, $h(\ell) \not\subset Z(f)$. Let g_ℓ be the (square-free) bivariate polynomial defined in step (ii). Then $\ell \subset Z(g_\ell)$ is an irreducible component of $Z(g_\ell)$. By removing the linear factor defining ℓ, we replace g_ℓ by another square-free polynomial g_ℓ^0, of degree smaller than D, whose zero set does not fully contain ℓ. We then cut ℓ at each point where it meets $Z(g_\ell^0)$ (this is a variant of step (i)), and at each point that lies directly above a critical point of g_ℓ^0, as defined above (a variant of step (ii)). As before, the number of such cuts of ℓ is $O(D^2)$, for a total of $O(nD^2)$ cuts.

(iv) We now proceed recursively: For each cell τ_i of the partition, we recurse on the corresponding subset $\mathcal{L}_i$ of lines. The bottom of the recursion is at cells τ_i for which $|\mathcal{L}_i| < D^2/c$. For such cells we apply the naïve procedure, noted in the introduction, which cuts the lines of $\mathcal{L}_i$ into $O(|\mathcal{L}_i|^2) = O(D^4)$ pieces, so that all cycles in $\mathcal{L}_i$ are trivially eliminated.

Lemma 2.2. *The procedure described above eliminates all the cycles in $\mathcal{L}$.*

Proof. The proof is by induction on the size of the input. The claim holds at the bottom of recursion, because we make all possible cuts there, thereby eliminating all cycles. Consider then any non-terminal instance of the recursion, involving some subset of lines, which, for convenience, we again call $\mathcal{L}$.

As argued above, it suffices to show that we have cut all simple cycles. Let C be a simple cycle in $\mathcal{L}$, formed by some k lines $\ell_1, \ldots, \ell_k$, with $\ell_1 \prec \ell_2 \prec \cdots \prec \ell_k \prec \ell_1$. Let C^* denote the xy-projection of C, which is a simple polygon with k sides $e_1^*, \ldots, e_k^*$, so that, for $i = 1, \ldots, k$, $e_i^* \subset \ell_i^*$ is the xy-projection of the corresponding segment $e_i \subset \ell_i$ of the path $\pi(C)$.

If $Z(f)$ does not intersect $\pi(C)$ then there exists a cell τ_i of the partition that fully contains $\pi(C)$, so, in particular, all the lines $\ell_1, \ldots, \ell_k$ belong to $\mathcal{L}_i$; that is, they all intersect τ_i. By induction, the cycle C will be eliminated by the recursive call to the procedure with $\mathcal{L}_i$. Assume then, in what follows, that $Z(f)$ intersects $\pi(C)$.

Assume first that $Z(f)$ does not fully contain any of the lines $\ell_1, \ldots, \ell_k$. If $Z(f)$ intersects (but does not contain) one of the segments e_i, for $i = 1, \ldots, k$, then this intersection point, at which we have cut ℓ in step (i), eliminates the cycle C.

Assume next that none of the lines $\ell_1, \ldots, \ell_k$ is fully contained in $Z(f)$, and that none of the segments $e_i \subset \ell_i$ is crossed by $Z(f)$. In this case, the crossing points of $\pi(C)$ with $Z(f)$ must all lie on the vertical edges of $\pi(C)$. Recall that we have ensured that $Z(f)$ does not fully contain any such segment.

Trace $\pi(C)$ in a circular fashion, as in its definition, and keep track of the level $\lambda(q)$ in $Z(f)$ of the point q being traced. By our general position assumption, and by the tilting performed above, the level is well defined, and it can change only at a discrete set of points q, at which the univariate restriction of f to the vertical line through q has a multiple real root; see below for a discussion of this statement). Each time we go up along one of the vertical segments of $\pi(C)$, the level either increases or stays the same, and it strictly increases at least once along the cycle. Since the levels at the beginning and at the end of the tour are the same, the level must go down at least once, as we trace one of the segments e_i, $i = 1, \ldots, k$. Suppose, without loss of generality, that the level goes down as we trace e_1. This must happen at a point $\zeta \in \ell_1$ at which the univariate restriction of f to the vertical line through ζ has a multiple real root. That is, ζ lies vertically above a point at which $g_{\ell_1} = (g_{\ell_1})_z = 0$, where g_{ℓ_1} refers here to the original version of the restriction of f to $h(\ell_1)$. Now if g_{ℓ_1} is square-free, we are done, since, by construction, we have cut ℓ_1 at ζ, and thus C got eliminated by this cut. If g_{ℓ_1} is not square-free, it is possible that $g_{\ell_1} = (g_{\ell_1})_z = 0$ along a one-dimensional curve, so this property holds for an infinity of points ζ on ℓ_1. However, in such a case (i) the multiplicity of the root does not cause the level to change at ζ, and (ii) this vanishing on a one-dimensional curve does not occur for the square-free version of g_{ℓ_1}. This implies that the change in level must occur at a criticality of the square-free version of g_{ℓ_1}, and ℓ_1 has been cut above every such criticality, implying that C has been eliminated in this case as well.

Finally, consider the case where one (or more) of the lines $\ell_1, \ldots, \ell_k$ is fully contained in $Z(f)$; say ℓ_1 is such a line. If one of the edges e_i of C has been cut by steps (i)–(iii), we are done, so assume that this did not happen. As a point q traces $\pi(C)$, as above, the level $\lambda(q)$ goes up at least once, when we go up from v_1^+ to v_2^- (at v_2^- we count ℓ_1 in its level, whereas at v_1^+ we do not). Since the level cannot go down along any of the vertical upward edges of $\pi(C)$, it must go down when q traverses some edge e_i of C. Therefore, arguing as above, q must lie directly above a critical point of the square-free version of the restricted polynomial g_{ℓ_i}, or of its reduced version $g^0_{\ell_i}$ if $\ell_i \subset Z(f)$. In either case, ℓ_i has been cut at q and C has been eliminated.

Having covered all cases, the lemma follows. □

It remains to bound the number of cuts. Collecting the bounds from each step of our construction and maximizing over $\mathcal{L}$ produces the recurrence

$$\chi(n) \le \begin{cases} bD^3\chi(cn/D^2) + O(nD^2), & \text{for } n > D^2/c \\ bD^4, & \text{for } n \le D^2/c, \end{cases}$$

where c is the absolute constant mentioned above, and b is another suitable absolute constant.

Setting $D = c^{1/2}n^{1/4}$, the termination condition $n \le D^2/c$ becomes $n = 1$, in which case no cuts are needed.[2] That is, $\chi(1) = 0$, and for $n > 1$ we have

$$\chi(n) \le b_1 n^{3/4}\chi(n^{1/2}) + O(n^{3/2}),$$

for a suitable absolute constant b_1. Since the depth of recursion is $O(\log\log n)$, its solution is easily seen to be $\chi(n) = O(n^{3/2}\log^{\beta} n)$, where β is a constant that depends only on the absolute constant b_1. This completes the proof of the upper bound.

Lower bound. The near-tightness of the bound follows from the grid-like construction of $\Theta(n^{3/2})$ *joints* (points incident to at least three non-coplanar lines) in a collection of n (or rather $3n$) lines, where the joints are the vertices of the $\sqrt{n} \times \sqrt{n} \times \sqrt{n}$ integer grid, and the lines are the $3n$ axis-parallel lines of the grid; see, e.g., [2, 8]. By slightly perturbing (translating and tilting) each of the lines, and by appropriately tilting the coordinate frame, each joint is mapped to a small elementary triangular cycle in the arrangement of $\Theta(n)$ lines in general position in the plane. As the cycles do not overlap, each requires a separate cut. □

Remark. Setting D to a sufficiently large constant, rather than a function of n, in the above argument allows us to avoid having to work with arbitrarily high-degree polynomials at the expense of slightly weakening the upper bound to $O(n^{3/2+\varepsilon})$, where $\varepsilon = \varepsilon(D) > 0$ depends on

[2] For the algorithmic part of the analysis, it is preferrable to work with constant degree D, which is why the analysis, up to this point, is stated in terms of arbitrary values for D; see a remark below and Section 3 for more details.

the choice of D and can be made arbitrarily small. Of course, the implied constant in the big-Oh grows with D.

3 Algorithmic considerations

In this section we outline and discuss several algorithms for eliminating cycles.

Implementing the procedure in the proof of Theorem 2.1. The most straightforward way to obtain the required cuts would be to implement the mostly-constructive proof of Theorem 2.1, except that we set D to be a sufficiently large constant (see the remark after the proof), in order to control the cost of the algebraic calculations that are needed to determine the cutting points.

However, this would require a constructive (and efficient) way of obtaining the partitioning polynomial of Guth [9], which is not known to be possible at the moment. One may hope that the techniques developed in Agarwal et al. [1] for effective construction of "approximate partitioning polynomials" for sets of points would be helpful here as well. However, the machinery employed by Guth to prove the existence of the said polynomial is sufficiently different to make an extension to this case a serious challenge.

The rest of the algorithm would proceed as in the proof of Theorem 2.1. One needs to assume a suitable model of algebraic computation that supports constant-time execution of each of the various primitive algebraic operations required by the algorithm (such as finding the intersections of a line with $Z(f)$, finding the critical points of the polynomials g_ℓ, etc.) for constant-degree polynomials. See, e.g., Basu et al. [4] for a discussion of the existing machinery for implementing operations of this kind.

The algorithms of Har-Peled and Sharir and of Solan. Alternatively, we can use the algorithms of Har-Peled and Sharir [11] or Solan [15], specifically designed to eliminate cycles in the depth relation. Given a collection $\mathcal{L}$ of n lines (or line segments) in $\mathbb{R}^3$, these algorithms work on the arrangement $\mathcal{A}(\mathcal{L}^*)$ of the xy-projections of the lines, and partition the plane into regions, either by a cutting (as in Solan [15]), or by incrementally refining regions into subregions (as in Har-Peled and Sharir [11]), exploiting the fact that one can efficiently detect the presence of a depth cycle in a collection of line segments in $\mathbb{R}^3$ using an algorithm of De Berg et al. [6]. Both algorithms generate close to $O(n\sqrt{\chi})$ cuts, where χ is the minimum number of cuts required to eliminate all cycles. Concretely, the slightly improved randomized algorithm in [11] makes $O(n\sqrt{\chi}\alpha(n)\log n)$ cuts in expectation (the bound in Solan's algorithm [15] is slightly worse), and runs in expected time $O(n^{4/3+\varepsilon}\chi^{1/3})$, for any $\varepsilon > 0$; see [15, Theorem 2.1] and [11, Theorem 6.1]. Therefore, we may conclude:

Theorem 3.1. *There exists a randomized algorithm that, given a set of n lines in $\mathbb{R}^3$, can find a set of $O(n^{7/4}\,\mathrm{polylog}\,n)$ cuts sufficient to eliminate all cycles in the depth relation among the lines, in expected time $O(n^{11/6+\varepsilon})$, for any $\varepsilon > 0$.*

This significantly improves the best previously known bounds, achievable by a polynomial-time construction, but it is still far from the ideal goal of finding a set of cuts close to the minimum possible size (or just of size close to $n^{3/2}$), as in the (not yet fully polynomial-time) construction in the proof of Theorem 2.1.

Notice that identifying a smallest possible set of cuts for a given family of line segments is known to be NP-complete [3].

The greedy algorithm for hitting sets. As we detail in the appendix, one can view the cycle-cutting question as an instance of the hitting-set problem in hypergraphs. Specifically, we consider the hypergraph $(X, \mathcal{R})$, where X is the set of edges of $\mathcal{A}(\mathcal{L}^*)$, and each hyperedge in $\mathcal{R}$ represents a simple cycle C as the set of edges of $\mathcal{A}(\mathcal{L}^*)$ that comprise C^*. A hitting set in that hypergraph translates in a straightforward manner to a set of cuts that eliminate all cycles.

We can construct a hitting set using the standard greedy algorithm. Its analysis by Lovász [13] provides an upper bound on the size of the constructed set via sharp bounds on the size of so-called simple k-matchings of cycles, as mentioned in the introduction and presented in more detail in the appendix. We present this approach because we feel it is rather elegant and because it can be applied to eliminate all cycles up to any fixed constant length, even though in its current state it yields neither a tight bound on the number of cuts sufficient to eliminate *all* cycles, nor an efficient (polynomial-time) algorithm for identifying them.

4 Extensions to line segments and algebraic arcs

In this section we discuss two extensions of our technique, to sets of line segments and of constant-degree algebraic arcs.

The case of line segments. Consider a non-degenerate set S of n non-vertical line segments in $\mathbb{R}^3$, and let $\mathcal{A}(S^*)$ be the arrangement formed by their xy-projections; as we assume general position, each vertex of $\mathcal{A}(S^*)$ is either the projection of an endpoint of a segment in S, or a proper crossing of two projected segments. Let X denote the number of vertices of the latter kind; we refer to them as *proper* vertices.

Of course, suitably perturbed, S can be extended to a set of lines in general position, and therefore all cycles in S can be eliminated using $O(n^{3/2} \operatorname{polylog} n)$ cuts. We want to refine this bound, to make it depend on X.

Clearly the case $X = 0$ requires no cuts, and if $X \le n$, we cut every segment s near each point projecting to a proper vertex of $\mathcal{A}(S^*)$, thereby making $O(n)$ cuts and eliminating all cycles.

For larger values of X, set $r := n^2/X < n$, and construct a $(1/r)$-cutting of S^* with $O(r + \frac{r^2}{n^2}X) = O(r)$ trapezoids, each crossed by at most n/r segments in S^* [7]. Cut every segment of S at each point lying vertically above the boundary of a trapezoid of the cutting, thereby making $O(r) \cdot (n/r) = O(n)$ cuts. Now apply the bound of Theorem 2.1 over each trapezoid separately, concluding that

$$O\left(n + r(n/r)^{3/2} \operatorname{polylog}(n/r)\right) = O(n^{1/2}X^{1/2} \operatorname{polylog} n)$$

cuts are sufficient to eliminate all cycles. Combining the two cases and using the algorithms of Har-Peled and Sharir [11] or of Solan [15], we conclude:

Theorem 4.1. *The number of cuts sufficient to eliminate all cycles in a family of n non-vertical line segments in general position in $\mathbb{R}^3$ is $O(n + n^{1/2}X^{1/2} \operatorname{polylog} n)$, where X is the number of pairs of segments whose xy-projections cross.*

One can compute $O(n + X^{1/4}n^{3/4} \operatorname{polylog} n)$ cuts that eliminate all cycles, in expected time $O(n^{3/2+\varepsilon}X^{1/6})$, for any $\varepsilon > 0$.

The case of algebraic arcs. Our argument extends, with minor adjustments, to a similar bound on the number of cuts needed to eliminate all cycles in a collection of n constant-degree algebraic curves or arcs, with a suitable general position assumption. While not spelling out all the details of this extension, below we highlight the neccessary modifications.

First, in the case of arcs (or curves), one can have cycles of length 1 (a curve passing above itself) or 2 (two twisted curves, each passing above the other). This however does not affect the argument in any significant manner.

The definition of a cycle and of a simple cycle, the xy-projection of a cycle, and the path associated with a cycle, extend to the case of arcs in an immediate and obvious manner.

Guth's polynomial partitioning technique [9] also applies for constant-degree algebraic arcs, with the same performance parameters (albeit with potentially larger constants of proportionality that depend on the maximum degree of the curves). This allows us to run the same recursive cutting procedure, and use the same reasoning to show that it does indeed eliminate all cycles. It results in a similar recurrence (with different constants), that solves to the same bound $O(n^{3/2} \log^{\beta} n)$, albeit with a larger exponent β which depends on the maximum degree of the arcs.

Finally, to obtain a bound that depends on the number of interecting pairs of arc projections, we can first construct a $(1/r)$-cutting of $\mathcal{A}(\Gamma^*)$, exactly as in the case of segments, and apply the bound on the number of cuts within each pseudo-trapezoid of the cutting separately, resulting in the following summary result.

Theorem 4.2. *The number of cuts sufficient to eliminate all cycles in a family of n constant-degree algebraic curves or arcs in general position in $\mathbb{R}^3$ is $O(n + n^{1/2}X^{1/2} \operatorname{polylog} n)$, where X is the number of pairs of arcs whose xy-projections cross, and where the constant of proportionality depends on the degree of the input arcs.*

Remark. Both the algorithms of Har-Peled and Sharir [11] and of Solan [15] rely on a subroutine for quickly checking if a set of lines or line segments has an acyclic depth relation. Analogous tools would have to be developed in order to yield a somewhat efficient construction of a small set of cuts to eliminate all cycles, for the case of algebraic arcs and/or curves.

References

[1] P.K. Agarwal, J. Matoušek, and M. Sharir, On range searching with semialgebraic sets II, *SIAM J. Comput.* 42 (2013), 2039–2062. Also in arXiv:1208.3384.

[2] B. Aronov, V. Koltun, and M. Sharir, Cutting triangular cycles of lines in space, *Discrete Comput. Geom.* 33 (2005), 231–247.

[3] B. Aronov, M. de Berg, C. Gray, and E. Mumford, Cutting cycles of rods in space: Hardness results and approximation algorithms, *Proc. 19th Annu. ACM-SIAM Sympos. Discr. Alg.*, 2008, 1241–1248.

[4] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, Algorithms and Computation in Mathematics 10, Springer-Verlag, Berlin, 2003.

[5] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd Ed., 2008, Springer-Verlag.

[6] M. de Berg, M. Overmars, and O. Schwarzkopf, Computing and verifying depth orders, *SIAM J. Comput.* 23 (1994), 437–446.

[7] M. de Berg and O. Schwarzkopf, Cuttings and applications, *Int. J. Comput. Geom. Appl.* 5 (1995), 343–355.

[8] B. Chazelle, H. Edelsbrunner, L.J. Guibas, R. Pollack, R. Seidel, M. Sharir and J. Snoeyink, Counting and cutting cycles of lines and rods in space, *Comput. Geom. Theory Appls.* 1 (1992), 305–323.

[9] L. Guth, Polynomial partitioning for a set of varieties, in arXiv:1410.8871.

[10] L. Guth and N.H. Katz, On the Erdős distinct distances problem in the plane, *Annals Math.* 181 (2015), 155–190. Also in arXiv:1011.4105.

[11] S. Har-Peled and M. Sharir, Online point location in planar arrangements and its applications, *Discrete Comput. Geom.* 26 (2001), 19–40.

[12] H. Kaplan, M. Sharir and E. Shustin, On lines and joints, *Discrete Comput. Geom.* 44 (2010), 838–843. Also in arXiv:0906.0558.

[13] L. Lovász, On the ratio of optimal integral and fractional covers, *Discrete Math.* 13 (1975), 383–390.

[14] R. Quilodrán, The joints problem in $\mathbb{R}^n$, *SIAM J. Discrete Math.* 23 (2010), 2211–2213. Also in arXiv:0906.0555.

[15] A. Solan, Cutting cycles of rods in space, *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, 1998, 135–142.

Appendix

A Applying the greedy algorithm for hitting sets

Let L be a set of n non-vertical lines in $\mathbb{R}^3$ in general position. To eliminate all the (simple) cycles in L, we use the standard greedy algorithm for the Hitting Set problem in hypergraphs (see below). For this, we define the set system (that is, hypergraph) $(X, \mathcal{R})$, where X is the set of all edges of $\mathcal{A} := \mathcal{A}(L^*)$, and where each set $s_C \in \mathcal{R}$ corresponds to a simple cycle C in L, and is the set of edges of $\mathcal{A}$ whose union is C^*. Note that $\mathcal{R}$ can be of size exponential in n.

A *hitting set* for $(X, \mathcal{R})$ is a set H of edges of $\mathcal{A}$, such that, for each simple cycle C in L, C^* contains an edge of H. Given such a hitting set H, we take each edge e in H, which is a portion of the xy-projection ℓ^* of some line $\ell \in L$, and cut ℓ at a point that projects to e. Clearly, after these $|H|$ cuts, all cycles in L will be eliminated.

It therefore suffices to derive an upper bound for the size $|H|$ of such a hitting set H. To do so, we apply the standard greedy algorithm, which, at each step, chooses an edge e of $\mathcal{A}$ contained in the maximum number of sets (projected cycles) in $\mathcal{R}$ that have not yet been hit, adds e to H, and removes from $\mathcal{R}$ all the cycles that contain e. Lovász [13] has shown that

$$|H| \le \frac{\nu_1}{1 \cdot 2} + \frac{\nu_2}{2 \cdot 3} + \cdots + \frac{\nu_{d-1}}{(d-1) \cdot d} + \frac{\nu_d}{d}, \tag{1}$$

where d is the maximum *degree* of an edge e of $\mathcal{A}$ (which is the number of cycle projections C^* that contain e, and which can be exponential in n), and where ν_k is the maximum size of a *simple k-matching*, which, in our context, is the maximum size of a family of simple cycles, so that each edge e of $\mathcal{A}$ is contained in the projections of at most k of them. Thus ν_1 is the maximum size of a family of pairwise non-overlapping cycles, and ν_d is the overall number of simple cycles.

To summarize, it suffices to bound the quantities $\nu_1, \ldots, \nu_d$, and substitute these bounds in (1), to obtain an upper bound for $|H|$.

Estimating ν_k. Let us write the bound derived in Theorem 2.1, as done at the end of its proof, as $O(n^{3/2} \log^\beta n)$. As implied by Theorem 2.1, and remarked prior to its proof, ν_1, the maximum size of a family of pairwise non-overlapping simple cycles, is $O(n^{3/2} \log^\beta n)$. Interestingly, bounding ν_k can be done by a simple modification of the proof of the theorem: Given a simple k-matching F, we note that, when we make a cut, it can eliminate at most k members of F. Hence, we must have

$$\nu_k = O(kn^{3/2} \log^\beta n). \tag{2}$$

Substituting the bounds of (2) in (1), we get

$$
\begin{aligned}
|H| &\le \sum_{k=1}^{d-1} \frac{\nu_k}{k(k+1)} + \frac{\nu_d}{d} \\
&= O\left(\sum_{k=1}^{d-1} \frac{n^{3/2}\log^{\beta} n}{k+1} + n^{3/2}\log^{\beta} n \right) \\
&= O\left(n^{3/2}\log^{\beta} n \log d \right). \qquad (3)
\end{aligned}
$$

That is, $O(n^{3/2}\log^{\beta} n \log d)$ cuts suffice to eliminate all cycles in L, where d is the largest number of simple cycle projections in which an edge of $\mathcal{A}$ can participate. Unfortunately, d can be exponential in n, so the factor $\log d$ makes the bound rather weak. However, if we apply this approach only to eliminate cycles of length at most k_0. for any fixed constant k_0, we have $d = O(n^{k_0})$, and $\log d = O(k_0 \log n) = O(\log n)$, making the bound in (3) $O(n^{3/2}\log^{\beta+1} n)$.

A brute-force implementation of the above idea gives an exponential-time algorithm to find the greedy-best set of cuts to eliminate all cycles (which is not guaranteed to be small, to add insult to injury!). For the analogous problem for cycles of length at most k_0, for a constant k_0, the greedy algorithm runs in polynomial time and guarantees $O(k_0 \log n)$-factor approximation to the optimal set of cuts.

Remark. We have presented this approach, in spite of all its weaknesses, for several reasons. First, if one only wishes to eliminate all cycles of at most some constant length, this is the only technique that we know of that produces, in polynomial time, $O(n^{3/2}\,\mathrm{polylog}\, n)$ cuts. In practice, it might well be the case that all cycles are "short", in the above sense, so this could be a viable approach to cycle elimination.

Second, it raises interesting and challenging open questions:

(i) It is not clear whether the bound in (2) on ν_k is tight, and improving it might lead to a tighter bound on the number of cuts. For example, is it true that $\nu_k = O(k^{1-c} n^{3/2}\log^{\beta} n)$, for some absolute constant $c > 0$? This would lead to the bound $|H| = O(n^{3/2}\log^{\beta} n)$, as is easily checked.

(ii) Can one improve the running time of the naïve implementation discussed above?