Eliminating Depth Cycles among Triangles in Three Dimensions*

Boris Aronov[†] Edward Y. Miller[‡] Micha Sharir[§]

July 20, 2016

Abstract

Given n non-vertical pairwise disjoint triangles in 3-space, their vertical depth (above/below) relation may contain cycles. We show that, for any $\varepsilon > 0$, the triangles can be cut into $O(n^{3/2+\varepsilon})$ pieces, where each piece is a connected semi-algebraic set whose description complexity depends only on the choice of ε, such that the depth relation among these pieces is now a proper partial order. This bound is nearly tight in the worst case. We are not aware of any previous study of this problem with a subquadratic bound on the number of pieces.

This work extends the recent study by two of the authors on eliminating depth cycles among lines in 3-space. Our approach is again algebraic, and makes use of a recent variant of the polynomial partitioning technique, due to Guth, which leads to a recursive procedure for cutting the triangles. In contrast to the case of lines, our analysis here is considerably more involved, due to the two-dimensional nature of the objects being cut, so additional tools, from topology and algebra, need to be brought to bear.

Our result essentially settles a 35-year-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, Brooklyn, NY 11201, USA; *boris.aronov@nyu.edu*.

[‡]Department of Mathematics, Tandon School of Engineering, New York University, Brooklyn, NY 11201, USA; *em1613@nyu.edu*.

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

1 Introduction

The problem. Let $\mathcal{T}$ be a collection of n non-vertical pairwise disjoint triangles in $\mathbb{R}^3$ *in general position*. In particular, we assume that the xy-projections of the triangles are in general position, in the sense that no pair of projected edges overlap, no vertex is projected to a point that lies on a projected edge of another triangle, and no three projected edges are concurrent. For any pair Δ, Δ' of triangles in $\mathcal{T}$, we say that Δ passes *above* Δ' (equivalently, Δ' passes *below* Δ) if any vertical line that meets both Δ and Δ' intersects Δ at a point that lies higher than its intersection with Δ'; this property is clearly independent of the choice of the vertical line meeting both triangles. We denote this relation by $\Delta' \prec \Delta$. The relation $\prec$ is a partial relation, and in general it may contain *cycles* of the form $\Delta_1 \prec \Delta_2 \prec \cdots \prec \Delta_k \prec \Delta_1$. We call this a *$k$-cycle*, and refer to k as the *length* of the cycle. Cycles of length three (the minimum possible length) are called *triangular*. See Figure 1.

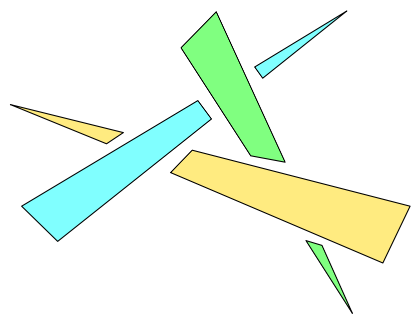

Figure 1: A triangular depth cycle among three triangles.

The problem of cycle elimination is to cut the triangles of $\mathcal{T}$ into a finite number of connected pieces, each being semi-algebraic of constant description complexity (that is, defined by a constant number of polynomial equalities and inequalities, each of degree bounded by some constant), so that the suitably extended depth relation among the new pieces is acyclic, in which case we call it a *depth order*.

The simpler case, in which the triangles are replaced by lines or line segments, has been handled in the recent companion paper [5] (and in earlier papers, cited therein). We note that eliminating cycles in a set of triangles adds, literally, a new dimension to the problem: whereas lines are cut at a discrete set of points, triangles have to be cut into pieces along curves, which makes the analysis considerably more involved—see below. We also observe that the binary space partition (BSP) technique of Paterson and Yao [18] constructs a depth order by cutting the triangles into $\Theta(n^2)$ pieces,[1] but, as in the case of lines, we would like to use fewer cuts, ideally close to the lower bound of $\Omega(n^{3/2})$, which is an immediate extension of a similar lower bound, noted in [5], for the case of lines.

A long-standing conjecture, open since 1980, is that one can indeed always construct a depth order with a *subquadratic* number of cuts. See [8, Chapter 9] for a summary of the

[1] A significant feature of the BSP technique is that the cuts are made by straight lines and therefore the resulting pieces can be taken to be triangular, whereas this is not the case in our construction.

state of affairs circa 1990. In the previous work [5] we have shown that $O(n^{3/2} \operatorname{polylog} n)$ cuts suffice to eliminate all cycles among n lines in space. In this paper we obtain a similar, albeit slightly weaker, bound for the case of triangles, settling this conjecture, in a strong, almost worst-case tight manner (with a few technical reservations, discussed below).

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 an earlier paper of Aronov et al. [3]. 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 we assume here, without loss of generality, to lie 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 (i.e., have an acyclic depth relation among the resulting pieces), and then paint the pieces in the above manner, obtaining a correct rendering of the scene; see [3, 10] for more details. Assuming that the input objects are all given as triangulated polyhedral approximations, as is the case in most practical applications, we face exactly the problem addressed in this paper.[2]

The study of cycles in a set of lines or line segments in $\mathbb{R}^3$ goes back to Chazelle et al. [11], who have shown that, if the xy-projections of a collection of n segments in 3-space form a "grid" (of the form depicted in Figure 2), then all cycles defined by this collection can be

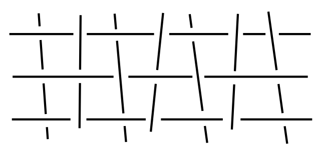

Figure 2: A collection of line segments that forms a grid, viewed from above.

eliminated with $O(n^{9/5})$ cuts. Another significant development is due to Aronov et al. [3], 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. However, their results did not apply to general, non-triangular cycles.

In the recent companion work [5], we essentially settled the case of lines, by showing that $O(n^{3/2} \operatorname{polylog} n)$ cuts are sufficient to eliminate all cycles, which is close to the best possible due to a well known construction requiring $\Omega(n^{3/2})$ cuts.

In contrast, the case of triangles has barely been touched, except for the aforementioned BSP technique in [18] and several subsequent studies of this technique, where improved

[2]That is, aside from our requirement that the triangles be pairwise disjoint and in general position. With some extra care, which we avoid in this version, the analysis will also be able to handle the cases where the triangles share edges, which is the case that arises in practice.

(subquadratic) bounds have been established for several special classes of objects in three dimensions, such as axis-parallel two-dimensional rectangles of bounded aspect ratio [1, 22] or so-called uncluttered scenes [9]; see [12, 22] for surveys of the BSP literature.

Our contribution. In this paper we essentially settle the problem for the case of triangles, and show that *all* cycles in the depth relation in a set of n triangles can be eliminated by cutting the triangles into $O(n^{3/2+\varepsilon})$ connected pieces whose (constant) description complexity depends only on the choice of $\varepsilon > 0$. As noted, our bound is best possible in the worst case, up to the $O(n^{\varepsilon})$ factor.

The proof of this bound follows the high-level approach in the previous analysis for the case of lines [5], which uses the polynomial partitioning technique of Guth [14]. Roughly speaking, this technique spreads the *edges* of the triangles more or less evenly among the cells of the partition, which in turn provides a recursive divide-and-conquer mechanism for performing the cuts. However, the fact that we are dealing here with two-dimensional triangles, rather than with one-dimensional objects like lines or segments, raises substantial technical problems that need to be overcome. The two most significant issues that arise are:

(i) In contrast with the case of lines, where the cuts are made at a discrete set of points, here we need to cut the triangles into two-dimensional regions. Ideally, we would like to cut them into triangular pieces (as does the BSP technique of [18]), but our technique does not achieve this and instead cuts the triangles by a collection of constant-degree algebraic curves into semialgebraic regions of constant description complexity.

(ii) Additional complications arise in controlling the recursive mechanism, to ensure that not too many triangles are passed to a recursive subproblem. As alluded to above, we can control the number of triangles that have an edge that crosses a cell, since the partition is based on the triangle edges, but we do not have a good bound on the number of triangles that "fully slice" through a cell; see below for the precise description. One therefore needs to prune away the triangles that cross a cell in this slicing manner, in order to obtain a recurrence relationship similar to the one in [5] for the case of lines, thereby achieving the desired near-optimal bound for the overall number of cuts; see below for full details.

As in the case of lines, our proof is constructive, and leads, in principle, to an efficient algorithm for performing the cuts (assuming a suitable model of algebraic computation). The only ingredient currently missing is an efficient construction of Guth's partitioning polynomial, a step that we leave as a topic for further research. (As also noted in [5], the problematic aspects of an efficient construction of a partitioning polynomial, for the simpler case of a set of points, and techniques for overcoming these issues, are discussed by Agarwal et al. [2]; one hopes that variants of these techniques could also be used for effectively partitioning space with respect to higher-dimensional objects—the triangle edges in our case.)

Additional issues that arise here and were absent in the case of lines (and are strongly related to item (ii) above) involve the analysis of the topology of the cells of the polynomial partitioning, and the way it is modified by the triangles that slice through the cells. Handling these issues constitutes a novel, non-negligible portion of our analysis.

We note that the previous study [5] proposes two other algorithmic approaches for computing the cuts, one using the algorithms of Har-Peled and Sharir [16] or of Solan [20], and the other using the (slower, albeit polynomial, but sharper) approximation algorithm of Aronov et al. [4]. Unfortunately, neither of these alternative techniques seems (so far) applicable to the case of triangles.

2 Eliminating cycles in a set of triangles

The setup and some notation. Let $\mathcal{O}$ be a collection of pairwise disjoint *objects* in three dimensions, where each object is a path-connected set contained in a non-vertical plane; in our analysis, these will be the triangles or the triangle pieces produced by our construction. Clearly, each object in $\mathcal{O}$ is *xy-monotone*, that is, its intersection with any vertical line is a single point or empty. We define a *depth relation* $(\mathcal{O}, \prec)$ on the objects of $\mathcal{O}$, in the following natural manner: we say that $o_1 \in \mathcal{O}$ *lies* (or *passes) below* $o_2 \in \mathcal{O}$ (in which case we also say that o_2 *passes above* o_1), and write $o_1 \prec o_2$ or $o_2 \succ o_1$, if there exists a vertical line ℓ that meets both o_1 and o_2, and the z-coordinate of its intersection with o_1 is lower than that of its intersection with o_2. For general planar connected regions, this relation need not be well behaved, but, for connected pieces of pairwise disjoint triangles, the relation is well defined, in the sense that it is independent of the choice of the line ℓ.

The final pieces into which the triangles of $\mathcal{T}$ will be cut will have *constant description complexity*, as defined above. However, until the very end of the construction, we will only generate certain constant-degree algebraic curves that are drawn on the respective triangles. Only at the end we will use these curves to construct the desired output collection of constant-complexity pieces with the desired properties.

A *cycle* in $(\mathcal{O}, \prec)$ is a circular sequence of some k objects from $\mathcal{O}$ that satisfy $o_1 \prec o_2 \prec \cdots \prec o_k \prec o_1$. We refer to k as the *length* of the cycle; a cycle of length k is a *k-cycle*. Note that self-loops and 2-cycles are not possible in $\mathcal{O}$ under our assumptions (although they may very well exist for more general objects), so we must have $k \geq 3$.

The problem, restated. We are now ready to formally state the problem: Let $\mathcal{T}$ be a collection of n non-vertical pairwise disjoint triangles in general position in $\mathbb{R}^3$. As already mentioned above and illustrated in Figure 1, $(\mathcal{T}, \prec)$ may contain cycles. Our goal is to cut the triangles of $\mathcal{T}$ into a small number of path-connected pieces of constant description complexity, so that, for the collection $\mathcal{O}$ of the resulting pieces, $(\mathcal{O}, \prec)$ is acyclic—a *depth order*.[3]

There is a straightforward way of achieving this: Project all triangles of $\mathcal{T}$ orthogonally to the xy-plane and form the resulting arrangement of triangles, which consists of at most $O(n^2)$ faces. Extrude each face of this arrangement into an unbounded z-vertical prism, and cut each triangle $\Delta \in \mathcal{T}$ into pieces along the polygonal curve of its intersection with the prism boundary. It is easy to see that the number of resulting pieces is $O(n^3)$, and that the pieces

[3]Technically, we imagine cuts to have non-zero, albeit arbitrarily small, width, in order to keep the resulting objects pairwise disjoint and their depth relation unambiguous. We will not mention hereafter the requirement that cutting the triangles involves leaving small gaps of this sort.

corresponding to a single prism form a linear order under $\prec$, while the pieces from different prisms are unrelated by $\prec$, so indeed there are no cycles. It is moreover easy to refine this decomposition so that the resulting pieces are triangles, with no asymptotic increase in the number of pieces.

The cubic number of pieces obtained by this naive approach is excessive. A better bound on the number of pieces sufficient to eliminate all cycles is provided by the *binary space partition* (BSP) technique of Paterson and Yao [18], which eliminates all cycles by cutting the triangles into $\Theta(n^2)$ (triangular) pieces. In fact, the construction in [18] has a much stronger property: the resulting collection of triangular pieces has no cycles in the depth relation corresponding to *any* viewing direction, or, more generally, to the perspective view from an arbitrary point.

In this paper we show that cycles in the depth relation (for a *fixed* viewing direction, which, as already stated, is taken to be the view from $z = +\infty$) can be eliminated by creating a significantly subquadratic number of pieces, while keeping the complexity of each piece constant. As already mentioned, the number of pieces that our technique yields, which is $O(n^{3/2+\varepsilon})$, for any prespecified $\varepsilon > 0$, is nearly tight in the worst case.

Recall that we have assumed that the triangles of $\mathcal{T}$ are in general position. We will cut the triangles by drawing curves on each triangle; this will be performed in a hierarchical manner, by a recursive procedure. The triangle pieces will be defined implicitly as faces in the induced arrangement of curves in each triangle. We will deliver our promise of constant-description-complexity pieces at the very end of the argument, in Section 2.5. Ideally, we would like the curves to be straight and the pieces to be triangular as in the BSP technique [18], but our argument cannot achieve it, in its current form.

So let $\mathcal{O}$ denote the implicit collection of faces on the triangles. Let C be a cycle $o_1 \prec o_2 \prec \cdots \prec o_k \prec o_1$ in $\mathcal{O}$ (with $k \geq 3$). We associate with C a continuum $\Pi(C) = \Pi(C, \mathcal{O})$ of closed paths (loops), where, informally, each path π in $\Pi(C)$ traces the cycle along the objects. Formally, each such π is defined in terms of k vertical lines $\ell_1, \ldots, \ell_k$, such that, for each i, ℓ_i intersects both o_i and o_{i+1} (where addition of indices is mod k), at respective points[4] v_i^+, v_{i+1}^-, so that v_i^+ lies below v_{i+1}^-. For each i, we connect the two points $v_i^-, v_i^+ \in o_i$ by a Jordan arc $\pi_i \subset o_i$. The path π is then the cyclic concatenation

$$\pi = \pi_1 \parallel v_1^+ v_2^- \parallel \pi_2 \parallel v_2^+ v_3^- \parallel \cdots \parallel v_{k-1}^+ v_k^- \parallel \pi_k \parallel v_k^+ v_1^-, \tag{1}$$

which is an alternation between the arcs π_i along the objects, and the (upward) vertical jumps $v_i^+ v_{i+1}^-$ between them. As already said, there is a continuum of possible paths, representing different choices of the vertical lines (and thus points) at which we decide to jump from object to object, and of the paths along which the "landing" and "take-off" points are connected along each object.[5]

To eliminate all cycles, it suffices to cut all the associated closed paths. The following easy lemma states this precisely.

[4]The superscripts + and − are a bit misleading if interpreted in terms of z-values; they are intended to indicate progress along π, in the sense that v_i^- precedes v_i^+ along o_i.

[5]This is in stark contrast to the case of lines, studied in [5], where each cycle corresponds to a *unique* path of this kind.

Lemma 2.1. *For each $\Delta \in \mathcal{T}$, let Γ_Δ be a collection of curves drawn on Δ, and let $\mathcal{O}_\Delta$ denote the relatively open two-dimensional faces of $\mathcal{A}(\Gamma_\Delta)$; put $\Gamma := \bigcup_\Delta \Gamma_\Delta$ and $\mathcal{O} := \bigcup_\Delta \mathcal{O}_\Delta$. Then, to verify that the depth relation among the pieces in $\mathcal{O}$ is acyclic, it is sufficient to ensure that, for each cycle C in $(\mathcal{T}, \prec)$, and for each path $\pi \in \Pi(C, \mathcal{T})$, one of the arcs π_i of π has been cut by a curve in Γ.*

Remark. Notice that we require that *all* cyclic paths in $\Pi(C, \mathcal{T})$ be cut. For a specific path π, a subpath π_i may be cut in such a way that it first leaves and then reenters the same piece of a triangle, thus keeping the cycle C alive. This would appear to be a problem, as π_i can be replaced by a rerouted subpath π_i' that stays in the same piece. Replacing π_i by π_i' in π, though, produces a *different* cyclic path in $\Pi(C, \mathcal{T})$, which we also require to be cut, and all these cuts will eventually eliminate C. The following proof handles this issue appropriately.

Proof. We proceed by contradiction: Assume that all cyclic paths in $\Pi(C, \mathcal{T})$, for every cycle C in $(\mathcal{T}, \prec)$, have been cut, but nonetheless there remains a cycle $C' \colon o_1 \prec o_2 \prec o_3 \prec \cdots \prec o_k \prec o_1$ in $(\mathcal{O}, \prec)$. In this case the set $\Pi(C', \mathcal{O})$ of paths realizing C' is nonempty, and we pick a path $\pi \in \Pi(C', \mathcal{O})$, having the form (1), where each subpath π_i is contained in the corresponding piece $o_i \in \mathcal{O}$, and each vertical jump $v_{i-1}^+ v_i^-$ moves from o_{i-1} to o_i. Each o_i is contained in some (not necessarily distinct) triangle $\Delta_i \in \mathcal{T}$, so π_i is fully contained in Δ_i, the jump $v_{i-1}^+ v_i^-$ can be viewed as a vertical jump from Δ_{i-1} to Δ_i, and therefore $\Delta_1 \prec \Delta_2 \prec \cdots \Delta_k \prec \Delta_1$ is a cycle in $(\mathcal{T}, \prec)$ with a witness path π that has not been cut, contradicting our assumption. $\square$

Our construction relies on the following result of Guth [14], which extends the earlier polynomial partitioning theorem of Guth and Katz [15]. For a non-zero polynomial $f \in \mathbb{R}[x, y, z]$, of degree D, we let $Z(f) := \{(x, y, z) \mid f(x, y, z) = 0\}$ denote its zero set. Removing $Z(f)$ from $\mathbb{R}^3$ creates $O(D^3)$ open connected *cells* (see, e.g., Warren [24]). The fact stated below is a special instance, tailored to our needs, of the considerably more general result in [14].

Fact 2.2 (Guth [14])**.** *Given a set of N lines in $\mathbb{R}^3$ and an integer $1 \le D \le \sqrt{cN}$, for a suitable absolute constant c, there always exists a non-zero polynomial $f \in \mathbb{R}[x, y, z]$ of degree at most D, so that each cell of $\mathbb{R}^3 \setminus Z(f)$ intersects at most cN/D^2 of the given lines.*

The first step of our construction resembles that of the case of lines in [5]. Specifically, let $\mathcal{E}$ denote the set of the $3n$ edges of the triangles in $\mathcal{T}$. Let f be a non-zero partitioning polynomial, of sufficiently large but *constant* degree D, for the $3n$ lines supporting the segments of $\mathcal{E}$, as provided by Fact 2.2. That is, $\mathbb{R}^3 \setminus Z(f)$ consists of $k = O(D^3)$ open connected cells, each intersected by at most $3cn/D^2$ (lines supporting) segments of $\mathcal{E}$, for an absolute constant $c > 0$. We hereafter assume that $Z(f)$ does not fully contain any vertical line, which we can ensure (as in the case of lines, treated in [5]), by a sufficiently small generic tilting of the coordinate frame; it can be argued (we skip the details) that the general position assumption ensures that such a tilting can be made without destroying any existing cycle[6] in

[6]This will hold if we regard the pieces in $\mathcal{O}$ as relatively open, as we have already indicated. This ensures that if $\Pi(C, \mathcal{O})$ is nonempty before the sufficiently small tilting, it will remain nonempty afterwards.

$\mathcal{T}$. We also assume, without loss of generality, that f is square-free. For each of the k (open) cells σ of $\mathbb{R}^3 \setminus Z(f)$, let $\mathcal{T}_\sigma$ denote the set of triangles of $\mathcal{T}$ that intersect σ.

An overview of the cycle elimination procedure. The general strategy is to construct a partitioning polynomial f as in Fact 2.2, to cut the triangles of $\mathcal{T}$ into pieces, using $Z(f)$ in a manner detailed below, and then to recurse within each cell of the partition. Each of the latter two phases, especially the third one, is more intricate here than in the case of lines, as detailed next.

The setup at each recursive step, at some node ξ of the recursion tree, is as follows. We have a subset $\mathcal{T}_\xi$ of $\mathcal{T}$, and an open cell σ_ξ, which is a connected component of $\mathbb{R}^3 \setminus Z(f_w)$, where f_w is the partitioning polynomial constructed at the parent step w. In fact, we have a hierarchy of cells $\sigma_{\xi_1=root}, \ldots, \sigma_{\xi_k=w}$, each of which arises at some proper ancestor ξ_i of ξ; the cell at the root is the entire 3-space. Note that these cells need not be contained within one another.

The procedure generates, at step ξ, a constant number of constant-degree algebraic curves (where the constants depend on the prespecified ε) on each triangle $\Delta \in \mathcal{T}_\xi$, and clips each curve to the region $\sigma_\xi^{(0)} := \bigcap_{i=1}^{k} \sigma_{\xi_k}$. This may break a curve into several connected arcs; the number of such subarcs, and their pattern of intersection, will be examined in Section 2.4.

We now proceed to describe the process in full detail. As in [5], 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 relatively open downward-directed vertical ray ρ_q emanating from q. Formally, if $q = (x_0, y_0, z_0)$, we consider the univariate polynomial $F(z) = f(x_0, y_0, z)$, and the level $\lambda(q)$ of q is the number of real zeros of F in $(-\infty, z_0)$, counted with multiplicity.

2.1 The procedure for cutting the triangles

The procedure is recursive. At each step of the recursion we have a subset of the triangles, which we also call $\mathcal{T}$, to simplify the notation, and which is processed as follows. (a) We construct a partitioning polynomial f, as in Fact 2.2, for (the lines supporting) the edges of the triangles of $\mathcal{T}$; the degree D of f is a sufficiently large constant that depends on the prespecified ε (in a manner detailed later). (b) We generate curves on the triangles of $\mathcal{T}$. For each $\Delta \in \mathcal{T}$, we generate up to $O(D^3)$ curves of degree D, and one curve of degree $O(D^2)$. (c) We recurse within each cell σ of $\mathbb{R}^3 \setminus Z(f)$ with the subset of those triangles that have at least one edge that crosses σ.

In more detail, the procedure consists of the following steps. In this description, we completely ignore the issue of clipping the curves. It is irrelevant to the main part of the construction, and will be picked up only towards the end, when we construct the pieces into which the triangles are to be cut.

(i) For each triangle $\Delta \in \mathcal{T}$, not fully contained in $Z(f)$, we draw $\Delta \cap Z(f)$ on Δ.

(ii) Consider the set of points $p \in Z(f)$ that are either singular or have a z-vertical tangent line. This set is contained in the common zero set $S(f) := Z\left(f, \frac{\partial f}{\partial z}\right)$ of f and $\frac{\partial f}{\partial z}$, which is one-dimensional since, by assumption, f is square-free. Let $H(f)$ denote the vertical "curtain" spanned by $S(f)$, namely, the union of all z-vertical lines that pass through points of $S(f)$. Since $S(f)$ is an algebraic curve of degree $O(D^2)$ (see, e.g., [13]), $H(f)$ is a two-dimensional variety of the same degree.

We then draw, on each triangle $\Delta \in \mathcal{T}$ not fully contained in $Z(f)$, the curve $\Delta \cap H(f)$.

(iii) If $\Delta \subset Z(f)$, that is, if the plane h_Δ supporting Δ is a component of $Z(f)$, we do not draw any curve on Δ. Note though that other triangles will be cut, in step (i), by h_Δ, which will draw on each such triangle Δ' the segment $\Delta' \cap h_\Delta$ (if nonempty).

(iv) We now want to proceed recursively, within each cell σ of the partition. Before doing so, we first need to draw additional curves on the triangles, as described below.

Let us consider the interaction of the triangles $\Delta \not\subset Z(f)$ with the cells of the partition. By construction, each cell σ meets the edges of at most $O(n/D^2)$ triangles. However, the plane spanned by such a triangle Δ (and therefore the relative interior of Δ) meets $O(D^2)$ of the cells, a consequence of Warren's theorem [24]. Therefore, each cell σ meets $O(n/D^2)$ triangle edges and, on average, $O(n/D)$ triangle interiors; in the worst case, the latter bound is tight.

We say that a triangle Δ *pierces* σ if one or more of its edges intersects σ, and that it *slices* σ if only its relative interior meets σ. A face s of $\Delta \setminus Z(f)$ can be similarly classified as *piercing* if it touches an edge of Δ, and *slicing* if it lies fully in the relative interior of Δ; the cell σ that s pierces or slices varies, in general, with s.

Our plan now is to recurse, for each σ, only on the set $\mathcal{T}_\sigma^{(p)} \subset \mathcal{T}_\sigma$ of triangles that pierce σ (that is, triangles Δ with at least one face s of $\Delta \setminus Z(f)$ piercing σ), and disregard, for the purposes of recursion, the slicing triangles. However, we first want to ensure that the slicing triangles do not participate in any depth cycle within σ, and for this we need to draw additional curves, as described next.

Let $s_1, \ldots, s_t$ be the slicing faces (or *slices*, for short) of all triangles of $\mathcal{T}$. Consider the effect of removing these slices from $\mathbb{R}^3 \setminus Z(f)$, one at a time.[7] The first slice s_1 is contained in some cell τ of $\mathbb{R}^3 \setminus Z(f)$ and cuts it locally in two. However, the removal of s_1 may or may not disconnect τ: $\tau \setminus s_1$ may have either one or two connected components; see Figure 3. Analogously, when we remove s_i from the current cell of $\mathbb{R}^3 \setminus Z(f) \setminus \bigcup_{j=1}^{i-1} s_j$ containing it, this

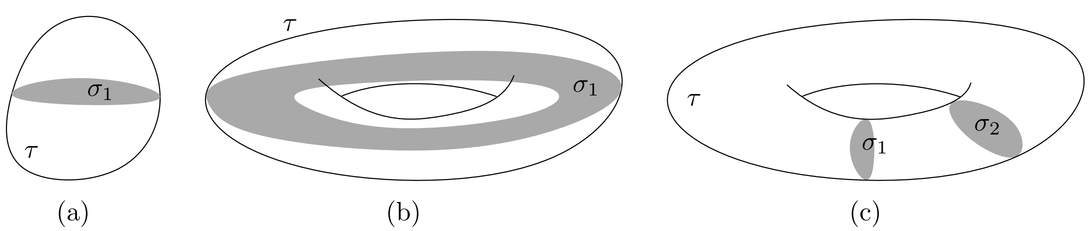

Figure 3: Several different ways the slice σ_1 can cut the cell τ. In (a) and (b) two connected components are produced, while in (c) σ_1 does not separate τ and the resulting cell remains connected; however, the subsequent removal of σ_2 *does* disconnect the cell.

[7] To clarify, removing a slice s from $\mathbb{R}^3 \setminus Z(f)$ means that we *insert* s as a "membrane" which locally separates the two sides of $\mathbb{R}^3 \setminus Z(f)$ near s.

may or may not disconnect the cell. We call s_i a *disconnecting slice* (resp., *non-disconnecting slice*) in the former (resp., latter) case; notice that this classification depends on the ordering of the slices. In Appendix B, we prove the following technical result, which is a crucial ingredient in our analysis and has no analogue in the case of lines:

Lemma 2.3. *There is an ordering of the slices $s_1, \ldots, s_t$, so that the number of non-disconnecting slices is $O(D^3)$.*

Returning to our construction, perform the following operation for every non-disconnecting slice s: Let s^+ denote the unbounded z-vertical cylinder spanned by s. For each triangle $\Delta' \in \mathcal{T}_\sigma$, form the curve $\Delta' \cap \partial s^+$, and add it to the curves already drawn.

(v) We finally apply recursion, within each cell σ of the partition, with the subset $\mathcal{T}_\sigma^{(p)}$ of the piercing triangles of σ. Recall that σ, as a spatial entity, is also crossed by additional slicing triangles, which will not be considered in the recursive subproblem; this should be kept in mind when we address, in the next subsection, the correctness of the procedure.

As in the case of lines, the bottom of the recursion is at cells σ for which $|\mathcal{T}_\sigma^{(p)}| < D^2/(3c)$. For such cells we apply the Paterson-Yao binary space partitioning [18], which cuts the triangles into $O(|\mathcal{T}_\sigma^{(p)}|^2) = O(D^4)$ triangular pieces, whose depth relation does not contain cycles. Following our strategy, we do not really perform the cuts yet, but just add the straight segments that perform these cuts to the collections of curves on the triangles.

2.2 All cycles are eliminated

Let $\Gamma^{(0)}$ denote the set of all curves that have been generated throughout the recursion. As in Lemma 2.1, we write $\Gamma^{(0)}$ as the disjoint union $\bigcup_\Delta \Gamma_\Delta^{(0)}$, where $\Gamma_\Delta^{(0)}$ is the set of curves drawn on Δ, for each $\Delta \in \mathcal{T}$.

Recall that each curve $\gamma \in \Gamma$ is clipped to within the intersection of all the ancestral cells of the cell τ associated with the recursive step at which γ is generated, starting with τ. As already described, we clip each curve $\gamma \in \Gamma^{(0)}$ to within the intersection of all the ancestral cells that lead to the recursive step at which γ has been generated. This clipping is needed to control the number of pieces into which the triangles will eventually be cut.[8] For each $\Delta \in \mathcal{T}$, we let Γ_Δ denote the collection of the clipped portions of the curves in $\Gamma_\Delta^{(0)}$, and let Γ denote the union of these collections.

Lemma 2.4. *The procedure described above eliminates all the depth cycles in $\mathcal{T}$, in the sense that, for each cycle C in $(\mathcal{T}, \prec)$ and for each path $\pi \in \Pi(C, \mathcal{T})$ of the form* (1)*, at least one of the "on-triangle" subpaths π_i of π is crossed by a curve of Γ.*

Proof. The clipping of the curves makes the analysis rather intricate, because whenever we argue that some subpath of π is cut by one of the generated curves, we also need to ensure

[8]Informally, the crucial parameter that controls the number of pieces is the number of intersections between pairs of generated curves. The clipping is performed to ensure that curves generated along unrelated branches of the recursion tree do not cross each other, thus controlling the number of overall intersections; see below for details.

that the cut point lies in all the ancestral cells of the node at which the cutting curve has been generated. Fortunately, these two aspects of the analysis are rather independent of one another, so we prove the lemma in two respective steps. We first suppose that the generated curves are not clipped at all[9] and show that in that case all paths π as in the lemma are cut by the unclipped curves (that is, curves of $\Gamma^{(0)}$). Then we show that each of the cuts produced in the first step is made at a point that belongs to all the ancestral cells, implying that the cut lies on the corresponding clipped curve (in Γ) as well.

Lemma 2.5. *For each C and π as in the premises of Lemma 2.4, one of the subpaths π_i of π is cut by a curve of $\Gamma^{(0)}$.*

Proof. The proof is by a bottom-up induction on the recursion tree. Specifically, let ξ be a node of the tree, let $\mathcal{T}_\xi$ be the set of triangles passed to the recursive step at ξ, let $C\colon \Delta_1 \prec \Delta_2 \prec \cdots \prec \Delta_k \prec \Delta_1$ be a cycle in $(\mathcal{T}_\xi, \prec)$, and let π be a path in $\Pi(C, \mathcal{T}_\xi)$ of the form (1). The inductive claim is that some subpath π_i of π will be cut by an *unclipped* curve generated at ξ or at some descendant thereof.

The claim clearly holds when ξ is a leaf, as the BSP constructed there eliminates all cycles (in the sense of Lemma 2.1). Assume then that the claim holds at all proper descendants of some node ξ, and consider the situation at ξ.

Assume first that π does not intersect $Z(f)$. Then π is fully contained in some cell σ of $\mathbb{R}^3 \setminus Z(f)$, and therefore $\Delta_1, \ldots, \Delta_k \in \mathcal{T}_\sigma$. If each Δ_i pierces σ, then $\Delta_1, \ldots, \Delta_k \in \mathcal{T}_\sigma^{(p)}$, all these triangles are passed to the recursive problem at σ, and the cycle $\Delta_1 \prec \Delta_2 \prec \cdots \prec \Delta_k \prec \Delta_1$, and therefore the cyclic path π too, will be cut, by the inductive hypothesis, in the above sense, by some unclipped curve generated at the recursive call within σ.

Consider then the case where some of the triangles Δ_i are slicing triangles for σ; by definition, for such a triangle Δ_i, every connected component of $\Delta_i \cap \sigma$ is a slice, fully contained in the relative interior of Δ_i. Since π is fully contained in σ and visits each triangle in the cycle, it must visit at least one of these slices. Assume first that it visits at least one non-disconnecting slice, call it s. Then π intersects the (solid) vertical cylinder s^+ erected at step (iv). If it crosses the boundary ∂s^+ of s^+, then any such crossing point q must lie along one of the subpaths π_i (as the vertical jumps along π are vertical and cannot enter or leave s^+), and then π_i will be cut at q, as desired. Otherwise, π is fully contained in the interior of s^+, but then, once it leaves s by a vertical jump upwards, it cannot return to the portion of s^+ below s, unless it crosses s again, from its top side to its bottom side, which contradicts the definition of the paths representing cycles. More specifically, π either follows a triangle, or travels between triangles, which are pairwise disjoint, by vertical jumps *upwards*, which would make it impossible to return from the region above s back to s from below s, all within s^+, to complete the cyclic path π.

Since the above argument can be applied to any non-disconnecting slice belonging to any of the triangles $\Delta_1, \ldots, \Delta_k$, we can now assume that π meets no non-disconnecting slices. Let s be the *first*, necessarily disconnecting, slice that π intersects, in the ordering

[9]We note that the clipping has not been used at all in the analysis up to this point, and in particular it does not affect the recursive construction; it is only needed for the complexity analysis in Section 2.4.

provided by Lemma 2.3; without loss of generality, assume that $s \subset \Delta_1$. Let $\sigma_1 \subseteq \sigma$ be the connected component containing s in the current state of refinement of the cell σ in that order, immediately before inserting s (recall that π meets no slices of any kind that precede s in the insertion order, so $\pi \subset \sigma_1$), and put $\sigma_2 \coloneqq \sigma_1 \setminus s$. As s is a disconnecting slice, σ_2 consists of two connected components, σ^+, lying locally above s, and σ^-, lying locally below it.

Recall that π is fully contained in σ_1. By construction, π enters s from below by an upward jump $v_k^+ v_1^-$, follows it along π_1, and exits it by another upward jump $v_1^+ v_2^-$; so it locally goes from σ^- to s to σ^+. Since π is a closed path, it must now go from σ^+ back to σ^-, within σ_1. But, using exactly the same reasoning as before, this is impossible: the only way to return from σ^+ to σ^- within σ_1 is to cross s again (as it separates σ^- and σ^+), from its upper side to its lower side, which cannot be done by any part of π, as it either follows triangles (which are pairwise disjoint, so none of them meets s, except for Δ_1 itself) or jumps between them in upward direction only. (Note that in general π may revisit the slice s. Then it crosses from below s to above s more than once, and still has no way to return to the bottom side, as previously argued.)

We can thus assume that π is contained within σ and does not use any slicing triangle, so, by the induction hypothesis, it will indeed be cut by the recursive step at σ, as claimed.

This concludes the handling of the case when π avoids $Z(f)$. Hereafter we assume that π intersects $Z(f)$.

Assume first that $Z(f)$ does not fully contain any of the triangles $\Delta_1, \ldots, \Delta_k$. If $Z(f)$ intersects one of the arcs π_i, for $i = 1, \ldots, k$, the arc has been cut and we are done (the exceptional case when π_i is partially or fully contained in $Z(f)$ is addressed in the full version of the paper).

Assume next that none of the triangles $\Delta_1, \ldots, \Delta_k$ is fully contained in $Z(f)$, and that none of the arcs $\pi_i \subset \Delta_i$ is crossed by (or contained in) $Z(f)$. In this case, the crossing points of π with $Z(f)$ must all lie on the vertical edges of π. Recall that we have ensured that $Z(f)$ does not fully contain any such vertical segment.

Trace π 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, $\lambda(q)$ is well defined, and it can change only either at a vertical jump of π, in which case it can only increase, or at a point $q \in H(f)$ (see step (ii) for the definition of $H(f)$). Since the level goes up at least once (at some vertical jump of π, where it crosses $Z(f)$), it must also go down, so at least one arc π_i of π must cross $H(f)$, and the claim holds in this case too.

Finally, consider the case where one (or more) of the triangles $\Delta_1, \ldots, \Delta_k$ is fully contained in $Z(f)$; say Δ_1 is such a triangle. In this case, we have generated, in step (i) (see a comment in step (iii)), for each of the other triangles Δ_i, the straight segment of intersection of Δ_i with the plane h_{Δ_1} (if it exists). The path leaves Δ_1 by a vertical jump into the upper halfspace $h^+_{\Delta_1}$ bounded by h_{Δ_1}, and returns to Δ_1 by a vertical jump from the complementary lower halfspace $h^-_{\Delta_1}$. This however requires π to cross h_{Δ_1} again, from $h^+_{\Delta_1}$ to $h^-_{\Delta_1}$. This cannot be done by a vertical jump, since it may only cross from $h^-_{\Delta_1}$ to $h^+_{\Delta_1}$, as it only moves upward, so it must occur while tracing some subpath $\pi_i \subset \Delta_i$ (with $\Delta_i \neq \Delta_1$). But then π_i must cross the segment $\Delta_i \cap h_{\Delta_1}$, as asserted.

Having covered all cases, the lemma follows. $\square$

In summary, if we stick to the set $\Gamma^{(0)}$ of unclipped curves, the previous lemma shows that every witness path π of any cycle C in $(\mathcal{T}, \prec)$ is cut by a curve of $\Gamma^{(0)}$ along one of its on-triangle subpaths π_i. The following lemma completes the picture by showing that this also holds for the clipped versions of the curves.

Lemma 2.6. *Let C be a cycle in $(\mathcal{T}, \prec)$ and let π be a path in $\Pi(C, \mathcal{T})$. Let ξ be the highest node of the recursion tree at which π has been cut by some unclipped curve of $\Gamma^{(0)}$ that has been generated during the non-recursive processing of ξ, as in Lemma 2.5. Then any such cutting point belongs to all the cells associated with the ancestral nodes of ξ, and thus belongs to the clipped version of the cutting curve.*

Proof. By construction and by the proof of Lemma 2.5, the fact that π has been cut at ξ means that it was fully contained in each of the ancestral cells of ξ. In particular, any point at which π has been cut belongs to all these cells. $\square$

In other words, if we actually draw all the (clipped) curves of Γ on their respective triangles, all cycles will be eliminated. This completes the proof of Lemma 2.4. $\square$

This finishes the proof of correctness of our procedure. We still need to fill three gaps: (i) We need an upper bound on $|\Gamma^{(0)}|$, the number of (unclipped) curves that the procedure generates. (ii) We need to bound the number of connected components of the clipped curves and the number of their intersection points. (iii) We need to cut each triangle into pieces of constant description complexity (and control their number). We now proceed to describe each step in detail.

2.3 Bounding the number of curves

Let $\chi(\mathcal{T})$ denote the maximum number of (unclipped) algebraic curves that our procedure generates on the triangles of $\mathcal{T}$, for the fixed choice of D that we use throughout the recursion. Put $\chi(n) := \max_{|\mathcal{T}|=n} \chi(\mathcal{T})$, where the maximum is taken over all collections $\mathcal{T}$ of n non-vertical pairwise disjoint triangles in general position in $\mathbb{R}^3$. To estimate $\chi(\mathcal{T})$, we collect the bounds from each step of our construction, and obtain the recurrence relation

$$\chi(\mathcal{T}) \leq bD^3\chi(3cn/D^2) + O(nD^3),$$

where b and c are suitable absolute constants; the overhead term $O(nD^3)$ is due to the cuts made by the $O(D^3)$ vertical cylinders erected over non-disconnecting slices, in step (iv) of the construction, and subsumes all other non-recursive cuts made at the present node.

Maximizing over $\mathcal{T}$ produces the recurrence

$$\chi(n) \leq bD^3\chi(3cn/D^2) + O(nD^3),$$

for $n > D^2/(3c)$, and $O(D^4)$ otherwise. The solution of this recurrence is easily seen to be $\chi(n) = O(n^{3/2+\varepsilon})$, where $\varepsilon = \varepsilon(D)$. Specifically, we require ε and D to satisfy the inequality

$$D^{2\varepsilon} \geq 2bc^{3/2+\varepsilon},$$

so $\varepsilon = O(1/\log D)$. Conversely, when ε is prescribed, we need to choose $D = 2^{\Theta(1/\varepsilon)}$, with a suitable constant of proportionality.

2.4 Drawing the curves

At any recursive step, within some cell σ, each triangle $\Delta \in \mathcal{T}_\sigma$ is a piercing triangle in at most $3D$ subcells of σ. Before sending Δ down the recursion, we generate on it one curve of degree D from step (i) and one curve of degree $O(D^2)$ from step (ii), when applicable. In addition, in the preliminary part of step (iv), we generate $O(D^3)$ curves, formed by the cylinders erected over the non-disconnecting slices, each of degree D, and in step (iii), we generate one segment for each of the at most D linear components of $Z(f)$. Altogether, we generate on Δ up to $O(D^3)$ curves of degree at most D, and one curve of degree $O(D^2)$.

We recall though that these drawings are only formed within the intersection $\sigma_\xi^{(0)}$ of all cells associated with the ancestral steps of the current node ξ of the recursion. This is important for controlling the complexity of the arrangement of the curves drawn on each triangle. Concretely, upon termination of the entire recursive process, we take each triangle $\Delta \in \mathcal{T}$, and consider the planar map M_Δ formed on Δ by the hierarchy of drawings constructed on it. That is, we take each curve γ, generated at some recursive node ξ, and draw only its portions that lie in the corresponding intersection cell $\sigma_\xi^{(0)}$.

Each vertex of M_Δ is either (a) an endpoint of a connected component of the clipped portion of some curve γ, or (b) an intersection point between two curves γ, γ', such that either (i) both arcs are generated at the same recursive step, within the same cell σ, or (ii) up to a swap between the arcs, γ, γ' are generated within two respective cells σ, σ', such that the step that generated σ' is an ancestor of the step that generated σ. These properties follow from the hierarchical nature of our drawings. Note that most vertices of type (a) are actually also vertices of type (b), because they are intersections of the full algebraic curve containing γ with the boundary of some ancestral cell, which is an intersection point of this curve with $Z(f)$, for the corresponding partitioning polynomial f. Since we generate (in step (i) of the construction) the curves $\Delta \cap Z(f)$ at each recursive step, the claim follows. The exceptions are endpoints of curves that lie on the edges of the corresponding triangles; we may ignore these vertices, as their overall number is only $O(D|\Gamma^{(0)}|)$, as is easily checked.

It therefore suffices to bound the number of intersection points of arcs with arcs constructed at (proper and improper) ancestral recursive steps. For each arc γ, formed along some triangle Δ, within a cell σ at some recursive step, the number of the ancestral cells of σ is $O(\log_D n)$, and each of them generates on Δ up to $O(D^3)$ curves of degree at most D, and one curve of degree $O(D^2)$. For the present argument, treat these curves as drawn in their entirety—this will only increase the number of intersection points on γ. The number of intersection points of γ with a curve of the former (resp., latter) kind is $O(D^3)$ (resp., $O(D^4)$); this is a consequence of Bézout's theorem, where we make the "worst-case" assumption that the degree of γ is $O(D^2)$. It follows that the number of vertices that can be formed along γ is at most $O(D^6 \log_D n)$, an obvious gross overestimate that we do not bother to optimize.

Multiplying this bound by the number of curves, we get that the overall complexity of the maps M_Δ, over all triangles Δ, is

$$O(D^6 \log_D n) \cdot O(n^{3/2+\varepsilon}),$$

where, as we recall, the prespecified $\varepsilon > 0$ can be chosen arbitrarily small, and where $D = 2^{\Theta(1/\varepsilon)}$, with a suitable constant of proportionality. It then easily follows that, by slightly increasing ε, but keeping it sufficiently small, we can still write the bound as $O(n^{3/2+\varepsilon})$, with $D = 2^{\Theta(1/\varepsilon)}$ and slightly larger constants of proportionality.

2.5 Final decomposition into pseudo-trapezoids

Finally, we take the planar map M_Δ, for each triangle Δ, and decompose it into regions of constant description complexity, by constructing the *trapezoidal decomposition* [10] of M_Δ in some fixed, but arbitrarily (and generically) chosen "vertical" direction within Δ (where the complementary "horizontal" direction is chosen to be perpendicular to the vertical one within Δ). Each resulting piece is a "pseudo-trapezoid," with (at most) two vertical sides, and "top" and "bottom" parts, each consisting of a monotone subarc of one of the curves we have drawn on Δ, and thus having degree at most $O(D^2)$.

The number of trapezoids is proportional to the complexity of M_Δ, which in turn is proportional to the number of its vertices and the number of points on the drawn curves that are "horizontally" extreme (as the curves have bounded complexity, each has only a constant number of such extreme points).

Using the analysis from the preceding subsection, this brings us to the main result of the paper.

Theorem 2.7. *Let $\mathcal{T}$ be a collection of n pairwise disjoint non-vertical triangles in $\mathbb{R}^3$. Then, for any prescribed $\varepsilon > 0$, we can cut the triangles of $\mathcal{T}$ into $O(n^{3/2+\varepsilon})$ pseudo-trapezoids, bounded by algebraic arcs of constant maximum degree δ, whose depth relation is acyclic; here δ, and the constant of proportionality, depend on ε.*

Remark. Our bound is slightly larger than that in [5] due to our choice of a *constant* value, rather than a function of n, for the degree D of the partitioning polynomials.

3 Discussion

In this paper we have essentially settled the long-standing problem of eliminating depth cycles in a set of triangles in $\mathbb{R}^3$. On the positive side, our solution is almost optimal in the worst case, in terms of the number of pieces, as this number is only slightly larger than the worst-case lower bound $\Omega(n^{3/2})$. Less impressively, the cuts are by constant-degree algebraic arcs, rather than, ideally, by straight segments. It is a natural open problem to determine whether a similar bound can be achieved with straight cuts (as in the BSP technique of [18]). Even a weaker bound, as long as it is subquadratic and generally applicable, would be of great significance.

Another direction for further research is to further tighten the bound, removing the ε in the exponent and replacing it by a poly-logarithmic factor, as in [5] (while keeping the shape of the cut pieces simple).

It seems likely that, with some care, our technique can be extended to collections of triangles that are only pairwise openly disjoint but may share edges, and to situations where the triangles are not in general position. Extending the technique to curved objects (e.g., spheres or spherical patches) is also a major challenge.

Finally, as noted above, the BSP partition of [18] has the stronger property that the depth relation of the resulting pieces is acyclic with respect to any viewing point or direction. Our solution does not seem to have this property, so one could ask whether one can cut the triangles into a *subquadratic* number of simple pieces that have this stronger property, or whether $\Omega(n^2)$ pieces are required, in the worst case.

References

[1] P.K. Agarwal, E.F. Grove, T.M. Murali, and J.S. Vitter, Binary space partitions for fat rectangles, *SIAM J. Comput.* 29(5) (2000), 1422-1448.

[2] 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.

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

[4] 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.

[5] B. Aronov and M. Sharir, Almost tight bounds for eliminating depth cycles in three dimensions, *Proc. 48th ACM Sympos. Theory of Computing*, 2016, 1–8. Also in arXiv:1512.00358.

[6] B. Aronov and M. Sharir, On translational motion planning of a convex polyhedron in 3-space, *SIAM J. Comput.* 26(6) (1997), 1785–1803.

[7] S. Basu, R. Pollack, and M.-F. Roy, On the Betti numbers of sign conditions, *Proc. AMS* 133 (2005), 965–974.

[8] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal.* Lecture Notes in Computer Science 703, Springer-Verlag, Berlin, 1993.

[9] M. de Berg. Linear size binary space partitions for uncluttered scenes. *Algorithmica* 28 (2000), 353–366.

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

[11] 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.

[12] A. Dumitrescu, C.D. Tóth, Binary space partitions, in *Encyclopedia of Algorithms*, M.-Y. Kao, Ed., Springer, 2016, 220-223.

[13] W. Fulton, *Introduction to Intersection Theory in Algebraic Geometry*, Expository Lectures from the CBMS Regional Conference Held at George Mason University, June 27–July 1, 1983, Vol. 54. AMS Bookstore, 1984.

[14] L. Guth, Polynomial partitioning for a set of varieties, *Math. Proc. Cambridge Phil. Soc.* 159 (2015), 459–469. Also in arXiv:1410.8871.

[15] 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.

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

[17] A. Hatcher, *Algebraic Topology*, Cambridge University Press, 2001. See also `http://www.math.cornell.edu/~hatcher/AT/ATpage.html`.

[18] M. S. Paterson and F. F. Yao, Efficient binary space partitions for hidden-surface removal and solid modeling, *Discrete Comput. Geom.* 5 (1990), 485–503.

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

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

[21] E. Spanier, *Algebraic Topology*, McGraw Hill, New York, 1966.

[22] C.D. Tóth, Binary space partitions for axis-aligned fat rectangles. *SIAM J. Comput.* 38(1) (2008), 429-447.

[23] C.D. Tóth, Binary space partitions: recent developments, in *Combinatorial and Computational Geometry*, vol. 52 of MSRI Publications, Cambridge University Press, 2005, 529-556.

[24] H. E. Warren, Lower bound for approximation by nonlinear manifolds, *Trans. Amer. Math. Soc.* 133 (1968), 167–178.

Appendix

We will start, in Section A, by proving a technical topological fact, and then use to prove Lemma 2.3 in Section B.

A A topological fact

In this section we prove a technical result, namely Lemma A.1. It is a special case of a more general statement, but we only prove a version sufficient for our purposes.

Consider a finite simplicial complex $K = K_1 \cup K_2 \cup \cdots \cup K_n$ which is a union of non-empty connected subcomplexes K_i. Let $|K|$, $|K_i|$ denote the associated topological spaces.

Consider the simple intersection graph G of $\{K_i\}$. More formally, let $V = \{v_i\}$, $i = 1, \ldots, n$, be a set of distinct vertices (points) of G, with v_i associated with K_i, and let $\{v_i, v_j\}$ be an edge of G whenever $|K_i| \cap |K_j| \neq \emptyset$. Put $E = \{(i,j) \mid 1 \leq i < j \leq n \text{ and } |K_i| \cap |K_j| \neq \emptyset\}$, the set of ordered pairs corresponding to edges of G.

Consider the graph G' obtained by subdividing each edge of G once (G' is directed for convenience, it does not affect the rest of the argument). Formally, let $W \coloneqq \{w_{ij}\}$ be a new set of distinct points w_{ij}, one for each $(i,j) \in E$. The vertex set of G' is $V \cup W$ and the edge set is $\{a_{ij}, b_{ij} \mid (i,j) \in E\}$, where $a_{ij} \coloneqq (v_i, w_{ij})$ and $b_{ij} \coloneqq (w_{ij}, v_j)$.

Since each K_i is connected, we define a continuous mapping $f \colon G' \to |K|$, as follows: For each space $|K_i|$ fix a point $p_i \in |K_i|$ and for each pair $(i,j) \in E$ pick a point $q_{i,j} \in |K_i| \cap |K_j|$. Now let f send v_i to p_i and $w_{i,j}$ to $q_{i,j}$. Viewing G' as a one-dimensional simplicial complex, extend f to all of G' by choosing for each $(i,j) \in E$ a continuous directed path from p_i to $q_{i,j}$ in $|K_i|$ and a continuous directed path from $q_{i,j}$ to p_j in $|K_j|$. Let $(v_i, w_{i,j}) \in G'$ be mapped to $|K_i| \subset |K|$ by the first choice and $(w_{i,j}, v_j)$ to $|K_j| \subset |K|$ by the second choice.

We are now ready to state our topological fact:

Lemma A.1. *Suppose that the simplicial complex K is a union of non-empty connected subcomplexes K_i, $i = 1, \ldots, n$, and that any triple intersection $K_i \cap K_j \cap K_k$ is empty for any choice of pairwise distinct indices i, j, k. Then the induced mapping on first homology*

$$H_1(G; F) = H_1(G'; F) \xrightarrow{f_\star} H_1(|K|; F),$$

with coefficients in any field F, is one to one. In particular, $\operatorname{rank}(H_1(|K|; F)) \geq \operatorname{rank}(H_1(G; F))$.

Proof. The first equality is standard; so it suffices to show that $f_\star$ is one to one. The case $n = 1$ is vacuously true, so we will assume $n > 1$ hereafter.

Our proof proceeds as follows. One embeds $|K|$ into a suitable topological space $M[n]$, by an inclusion $i_n \colon |K| \subset M[n]$, and proves that the composite mapping

$$i_n \circ f \colon G' \xrightarrow{f} |K| \overset{i_n}{\subset} M[n]$$

induces an isomorphism in homology in all dimensions. That is, the composite on homology $(i_n \circ f)_\star = (i_n)_\star \circ f_\star$ induces an isomorphism on homology in each dimension. So in particular, $f_\star$ is one to one for induced mapping on the first homology $H_1(\cdot; F)$ as desired.

Recall that, for a topological space X, the *cone on X with cone point $u \notin X$*, $C_u(X)$, is obtained from the topological product $X \times [0,1] = \{(x,t) \mid x \in X,\ 0 \leq t \leq 1\}$ by identifying the subspace $X \times \{1\}$ to the point u and giving the quotient space the quotient topology. The mapping $x \mapsto (x, 0)$ defines a continuous embedding of X into the cone $C_u(X)$: $X \subset C_u(X)$.

The main proof will proceed by induction. Before defining $M[a]$, for $1 \leq a \leq n$, we construct an intermediate topological space, $L[a]$, as follows. Fix distinct n points $x_1, x_2, \ldots, x_n$ disjoint from $|K|$ and form the identification space

$$L[a] = (\cup_1^a |K_i|) \ \cup \ (\cup_{i=1}^a C_{x_i}(|K_i|) \ / \ \sim,$$

where $|K_i| \subset \cup_1^a |K_i|$ is identified by $\sim$ with $|K_i| \times \{0\} \subset C_{x_i}(|K_i|)$ via $x \mapsto (x, 0)$ for each i. That is, to form $L[a]$, we "cone off" independently the subspace $|K_i|$ of $\cup_1^a |K_i|$ by using the cone point x_i for each $i = 1, \ldots, a$.

Put $E[a] = \{(i,j) \in E \mid 1 \leq i < j \leq a\} \subset E$.

Note that, for each $(i,j) \in E[a]$, two cones on $|K_i| \cap |K_j|$ given by $C_{x_i}(|K_i| \cap |K_j|) \cup C_{x_j}(|K_i| \cap |K_j|)$ are identified along $(|K_i| \cap |K_j|) \times \{0\}$ (this is often called the suspension of $|K_i| \cap |K_j|$) and so includes into $L[a]$ via

$$C_{x_i}(|K_i| \cap |K_j|) \cup C_{x_j}(|K_i| \cap |K_j|)/\sim \ \subset \ C_{x_i}(|K_i|) \cup C_{x_j}(|K_j|)/\sim \ \subset \ L[a].$$

For each pair $(i,j) \in E[n]$, fix a distinct point $y_{i,j} \notin L[n]$ with all these choices distinct. We now form the space $M[a]$, for $a = 1, \ldots, n$, from $L[a]$ by "coning off" independently the suspension subspaces $C_{x_i}(|K_i| \cap |K_j|) \cup C_{x_j}(|K_i| \cap |K_j|)/\sim \ \subset \ L[n]$ by using the cone point $y_{i,j}$ for each $(i,j) \in E[a]$. This process defines $M[a]$ and the inclusions

$$i_a \colon \cup_1^a |K_i| \subset M[a] = (\cup_1^a |K_i|) \cup (\cup_{i=1}^a C_{x_i}(|K_i|) \cup (\cup_{(i,j) \in E[a]} C_{y_{i,j}}(\, C_{x_i}(|K_i| \cap |K_j|) \cup C_{x_j}(|K_i| \cap |K_j|)) \,)/\sim .$$

Now introduce the subgraphs $G'[a]$ of G', for $1 \leq a \leq n$, defined as follows: $G'[n] = G'$; while for $1 \leq a < n$ let $G'[a]$ be the induced subgraph of G' with vertices $\{v_i \mid 1 \leq i \leq a\}$ union $\{w_{i,j} \mid 1 \leq i < j \leq a+1\}$. The edges a_{ij}, b_{ij}, with $1 \leq i < j \leq a$, are mapped by f to the union $\cup_1^a |K_i|$, by construction. The edges $a_{i,a+1}$ for $(i, a+1) \in E$ are also mapped to paths contained in $|K_i|$, by our selection of the points $q_{i,a+1}$ and connecting paths. Hence, restricted to $|G'[a]|$ the map f sends this larger subgraph to the subset $\cup_1^a |K_i|$ also. Let the induced mapping be called f_a:

$$f_a \colon |G'[a]| \to (\cup_1^a |K_i|).$$

The proof that $i_n \circ f \colon G'[n] \to M[n]$ induces an isomorphism on homology is accomplished by induction on a, showing that $i_a \circ f_a \colon G'[a] \to M[a]$ induces an isomorphism on homology for $a = 1, \ldots, n$.

The base of the induction corresponds to $a = 1$. $G'[1] = \{v_1\}$ is a single point, which under f is mapped to $(p_1 \times \{0\}) \in |K_1| \subset M[1] = C_{x_1}(|K_1|)$. This last cone has the homology of a point, so as desired $i_1 \circ f_1$ induces an isomorphism on homology.

Now assume that $i_a \circ f_a$ induces an isomorphism on homology with $1 \le a < n$. Then $G'[a+1] = G'[a] \cup X[a+1]$ where $X[a+1]$ is the subgraph of $G'[a+1]$ induced by vertices $\{v_{a+1}\} \cup \{w_{i,a+1} \mid (i, a+1) \in E[a+1]\}$. $X[a+1]$ is a tree and as such contractible to the point v_{a+1} and has homology of a point.

Now in an analogous fashion, $M[a+1] = M[a] \cup Y[a+1]$, where

$$\begin{aligned} Y[a+1] &= C_{x_{a+1}}(|K_{a+1}|) \cup Z[a+1]/\sim, \text{ where} \\ Z[a+1] &= \bigcup_{(i,a+1)\in E[a+1]} \left[C_{y_{i,a+1}}(C_{x_{a+1}}(|K_i| \cap |K_{a+1}|)) \cup C_{x_i}(|K_i| \cap |K_{a+1}|)\right]/\sim . \end{aligned}$$

Note that by the empty-triple-intersection assumption, the union over pairs $(i, a+1) \in E[a+1]$ is a union of cones which all contain the cone point x_{a+1} and the intersection of any two such is precisely this single point x_{a+1}. Hence, $Z[a+1]$ is a union of cones all meeting at the single point x_{a+1} and has the homology of a point.

From this fact, one may deduce that $Y[a+1]$ has the homology of a point as follows: The exact sequence for pair $(Y[a+1], Z[a+1])$ in reduced homology gives the short exact sequence, $\tilde{H}_\star(Z[a+1]) \to \tilde{H}_\star(Y[a+1]) \to H_\star(Y[a+1], Z[a+1])$ [21]. Now by the above, $\tilde{H}_\star(Z[a+1]) = 0$, so by exactness $\tilde{H}_\star(Y[a+1]) = 0$ will follow if one shows $H_\star(Y[a+1], Z[a+1]) = \tilde{H}_\star(Y[a+1]/Z[a+1]) = 0$. But the quotient space $Y[a+1]/Z[a+1]$ is identified with the cone $C_{x_{a+1}}(|K_{a+1}|/(\cup_{(i,a+1)\in E[a+1]} |K_i| \cap |K_{a+1}|))$ which has vanishing reduced homology. Hence, $H_\star(Y[a+1])$ has vanishing reduced homology; that is, $Y[a+1]$ has the homology of a point.

The mapping $i_{a+1} \circ f_{a+1}$ carries $G'[a]$ to $M[a]$ by $i_a \circ f_a$; it carries $X[a+1]$ which has the homology of a point to $Y[a+1]$ which has the homology of a point, and it carries $G'[a] \cap X[a+1]$ to $M[a] \cap Y[a+1]$.

By the induction hypothesis, the first map $i_a \circ f_a$ induces an isomorphism on homology; by the above the second mapping induces an isomorphism on homology since both terms have the homology of a point. The third mapping is the inclusion of the intersection $G'[a] \cap X[a+1] = \{w_{i,a+1} \mid (i, a+1) \in E[a+1]\}$ into the intersection $M[a] \cap Y[a+1] = \cup_{(i,a+1)\in E[a+1]} C_{x_i}(|K_i| \cap |K_{a+1}|)$. Note that by the triple intersection property these cones are mutually disjoint and f_{a+1} restricted to this intersection decomposes into a disjoint union of inclusions of points into cones, $\{x_i\} \cong \{p_i\} \subset C_{x_i}(|K_i| \cap |K_{a+1}|)$ for each $(i, a+1) \in E[a+1]$. Hence, the induced mapping $G'[a] \cap X[a+1] \to M[a] \cap Y[a+1]$ induces an isomorphism on homology.

Recall [21] that, if A and B are cell complexes with intersection $A \cap B$ a cell subcomplex, the Mayer-Vietoris sequence is a long exact sequence relating the homologies of $A \cap B$, A, B, and $A \cap B$. Moreover, if A' and B' are cell complexes with intersection $A' \cap B'$ a cell subcomplex and $F\colon A \cup B \to A' \cup B'$ is a continuous mapping of topological spaces carrying A to A' and B to B' and so necessarily $A \cap B$ to $A' \cap B'$, then there are associated mappings of the homology groups which are compatible with the Mayer-Vietoris sequences of $A \cup B$ and $A' \cup B'$.

Applied to the union $G'[a+1] = G'[a] \cup X[a+1]$ mapping compatibly by f_{a+1} to the union $M[a+1] = M[a] \cup Y[a+1]$, one obtains the following commutative diagram with rows long exact [21] sequences on homology with field coefficients F:

$$\begin{array}{ccccccccc}
H_\star(\Gamma[a]' \cap X[a+1]) & \longrightarrow & H_\star(\Gamma[a]') \oplus H_\star(X[a+1]) & \longrightarrow & H_\star(\Gamma[a+1]') & \longrightarrow & H_{\star-1}(\Gamma[a]' \cap X[a+1]) & \longrightarrow & H_{\star-1}(\Gamma[a]') \oplus H_{\star-1}(X[a+1]) \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
H_\star(M[a] \cap Y[a+1]) & \longrightarrow & H_\star(M[a]) \oplus H_\star(Y[a+1]) & \longrightarrow & H_\star(M[a+1]) & \longrightarrow & H_{\star-1}(M[a] \cap Y[a+1]) & \longrightarrow & H_{\star-1}(M[a]) \oplus H_{\star-1}(Y[a+1]).
\end{array}$$

By the above discussion and inductive assumption, all the vertical arrows are isomorphims except possibly for the middle mapping $H_\star(\Gamma[a+1]') \to H_\star(M[a+1])$. Therefore, by the standard 5-lemma [21], one concludes that this mapping is also an isomorphism, thereby completing the induction step and the proof of Lemma A.1. $\square$

B The proof of Lemma 2.3

Proof of Lemma 2.3. Our goal is to deduce Lemma 2.3 from Lemma A.1. The high-level idea is the following: Consider the set $\mathbb{R}^3 \setminus Z(f)$ and cut it by all the slices $s_1, \ldots, s_t$ into smaller subcells. Now examine the intersection graph of these subcells and observe that (a) the number of non-disconnecting slices is related to the number of "independent cycles" in this graph (see precise definitions below) and (b) by Lemma A.1, the latter number is upper-bounded by the number of handles in $\mathbb{R}^3 \setminus Z(f)$, which in turn can be estimated based on the fact that f has degree D.

Unfortunately, there are some technical obstacles along this route. We will mention only two: (i) Lemma A.1 is formulated for simplicial complexes, in particular for compact spaces, while $\mathbb{R}^3 \setminus Z(f)$ is open and unbounded. (ii) One has to be careful in how one defines the intersection graph. The easiest way to define the subcells produces open, disjoint subcells, so no intersections are present. Encoding interactions of subcells along the slices that were used to separate them produces a more natural intersection graph, but allows multiple edges and self-loops, which are not permitted in Lemma A.1.

We now supply the somewhat tedious technical details used to finesse the above issues.

We start by observing that our input triangles occupy some bounded region of space, so we can perform all of our reasoning in a sufficiently large bounding box B that fully contains all the triangles in its interior, in order to avoid dealing with unbounded sets (alternatively, we could add some points to $\mathbb{R}^3$ to compactify it).

Secondly, we replace $Z(f)$, which is a two-dimensional algebraic variety, with a "thickened" version $Z \coloneqq \{(x, y, z) \mid |f(x, y, z)| < \delta\}$. It is well known that, for a sufficiently small δ, Z is an arbitrarily good approximation of $Z(f)$, within B. So we replace $B \setminus Z(f)$ by $B \setminus Z$, which is a compact semi-algebraic set and can be triangulated and represented by a simplicial complex that is a subcomplex of a suitable triangulation of B.

We now redefine the *slices* $s_1, \ldots, s_t$, as the faces of $\Delta \setminus Z$ fully contained in the relative interior of Δ, for any $\Delta \in \mathcal{T}$. Refine our triangulation of B so that each slice appears in it as a subcomplex.

Our goal is to apply Lemma A.1 along the following lines: Let K be the complex corresponding to $B \setminus Z$. Cut it into pieces by the slices and let the subcomplexes K_i be (the closures of) the resulting pieces. We will need to work a little harder for the lemma to be applicable, as it requires that (a) the pieces be closed and connected, (b) they must not intersect in triples, and (c) the resulting intersection graph must be simple.

To achieve this, we make some further technical modifications. They can be achieved by a formal transformation of the complex representing $B \setminus Z$, but are easier to visualize if described purely geometrically. Replace each triangle $\Delta \in \mathcal{T}$ by a "puffy triangle" Δ': an object that can be visualized as an arbitrarily thin pillow, which shares vertices and edges with Δ, but has two bounding surfaces: one on top and one on the bottom (to realize this as a simplicial complex, we may need to take a baricentric subdivision; we omit the standard details). The effect of replacing Δ by Δ' locally is that each feature that intersected Δ away from its relative boundary is now duplicated: there is a "top" and a "bottom" version of it. Similarly, now each slice s_i is transformed into a "puffy slice" s_i' which has a top face s_i^+ and a bottom one s_i^- and some (arbitrarily thin) volume in between.

We need another technicality: For reasons we will describe shortly, we cut every single "puffy" slice s_i' into two, by replacing it with two thinner "puffy" slices, s_i^T (top) and s_i^B (bottom), separated by a two-dimensional membrane that is the original s_i. To summarize, s_i' is a three-dimensional "sandwich" of two subcomplexes: s_i^T delimited by s_i on the bottom and s_i^+ on the top, and s_i^B — with s_i^- on the bottom and s_i on the top.

Now let $\sigma_1, \ldots, \sigma_k$ be the connected components ("chunks") of the closure of $B \setminus Z \setminus \bigcup_1^t s_i'$. Consider the subcomplexes corresponding to $\sigma_1, \ldots, \sigma_k, s_1^T, \ldots, s_t^T, s_1^B, \ldots, s_t^B$. Their union is the complex corresponding to $B \setminus Z$. Now we are ready to invoke Lemma A.1. Each chunk is connected by definition. Each half-a-thickened slice is as well (it is a thickened version of a face of $\Delta \setminus Z$ for some $\Delta \in \mathcal{T}$). The graph G is defined as the intersection graph of sets σ_i, s_j^T, and s_ℓ^B. By construction, the bottom of s_j^T touches only the top of s_j^B (along s_j), the top of s_j^T is shared with some chunk σ_ℓ. Similar reasoning applies to the bottom parts s_ℓ^B of the slice "sandwiches." Chunks, by definition, only intersect top faces of top slices, or bottom faces of bottom slices. The graph is in fact tri-partite: objects from the same class do not intersect. Hence the only danger of a triple intersection is one where a chunk, a top slice, and a bottom slice intersect. But a top and a bottom slice only intersect if they are part of a "sandwich" and their intersection is an original slice that is disjoint from any chunk (as it is contained in the interior of a puffy triangle).

To summarize, Lemma A.1 is applicable now: the simplicial complex representing $B \setminus Z$ can be written as a union of connected subcomplexes representing the sets σ_i, s_j^T, and s_ℓ^B, and the triple intersections are empty. The intersection graph G is a proper simple graph, without multiple edges or self-loops (we introduced a separate complex for each slice in order to eliminate self-loops, and had to replace each slice by a sandwich of two thinner slices to eliminate multiple edges in G).

Finally, invoking Lemma A.1 with $F = \mathbb{Z}_2$, we conclude that $\text{rank}(H_1(G, \mathbb{Z}_2)) \leq \text{rank}(H_1(B \setminus Z), \mathbb{Z}_2)$.

As already mentioned, Z approximates $Z(f)$ well for δ sufficiently small, so in particular $\text{rank}(H_1(B \setminus Z)) = \text{rank}(H_1(B \setminus Z(f)))$, which in turn is $O(D^3)$, by the following fact, which is a special instance of the much more general bound of Basu et al. [7, Theorem 1].

Fact B.1. *The first Betti number of* $\mathbb{R}^3 \setminus Z(f)$ *is* $O(D^3)$.

Therefore $\mathrm{rank}(H_1(G, \mathbb{Z}_2)) = O(D^3)$. This quantity is sometimes called the *cyclomatic number* of the graph G; it is the dimension of the vector space with coefficients in $\mathbb{Z}_2$ formed by sets of edges of G representing formal sums of cycles (or subgraphs of even degree, which is the same). In a sense, it is the number of "independent cycles" in G, which we can compute in the following way: let x be the number of connected components in G (it corresponds to the number of connected components of $B \setminus Z$ and therefore of $B \setminus Z(f)$), e be the number of edges of G and v be the number of its vertices. Then $\mathrm{rank}(G, \mathbb{Z}_2) = e - v + x$. Indeed, construct a spanning forest F of G, it will have $v - x$ edges. Intuitively, every additional edge in G (and there are $e - (v - x) = e - v + x$ of them) forms an independent cycle in G.

The above argument is independent of the choice of the forest F. We will constrain it, as follows: Notice that each half-sandwich object s_i^T, s_i^B has degree two in G, with one edge connecting it to the adjacent chunk and one to its twin half-sandwich object; the latter corresponds to an adjacency across an original slice s_i. There always exists a spanning forest F that includes *all* edges of the former type (such edges form an imperfect matching and one can complete it to a spanning forest).

We will now explain why $\mathrm{rank}(G, \mathbb{Z}_2)$ a quantity relevant to us. Order the edges of G in the following manner: First the edges of $G \setminus F$ and then the edges of F. Every edge corresponds to an intersection along (a copy of) a slice. Imagine "deconstructing" $B \setminus Z$, incrementally, one edge of G at a time, in the order given above, by removing the adjacency corresponding to that edge (or, equivalently, cutting along the corresponding copy of a slice). It is easy to check that the edges not in F correspond exactly to non-disconnecting slices: the number of connected components of the graph remains the same, as the spanning forest is undisturbed. Once only forest edges remain, every edge removal leads to a disconnecting cut (note that some of these correspond to adjacencies along a real slice s_i and some along its shifted copies s_i^+ and s_i^-, but as we do not intend to count either type, this does not affect our analysis).

To summarize, the number of non-disconnecting slices is exactly equal to the number of non-spanning-forest edges of G, which is $e - v + x = \beta_1(G, \mathbb{Z}_2) \leq \beta_1(B \setminus Z) = \beta_1(B \setminus Z(f)) = O(D^3)$, completing the proof of Lemma 2.3. $\square$