Cutting Triangular Cycles of Lines in Space*

Boris Aronov† Vladlen Koltun‡ Micha Sharir§

Abstract

We show that a collection of lines in 3-space can be cut into a sub-quadratic number of pieces, such that all depth cycles defined by triples of lines are eliminated. This partially resolves a long-standing open problem in computational geometry, motivated by hidden-surface removal in computer graphics.

1 Introduction

Historical background. The chief goal of most computer graphics applications is to correctly depict ('render') a synthetic 3-dimensional scene onto the computer screen. The geometry of the scene is often represented by a collection of triangles. Correct rendering means, in particular, resolving situations where some object partly occludes another; we want to correctly draw the objects that lie closer to the viewpoint, and avoid drawing the occluded parts.

*Part of the work on this paper has been carried out at the U.S.-Israeli Workshop on Geometric Algorithms, held in Jackson Hole, WY, in the summer of 2002. Work on the paper by Boris Aronov and Micha Sharir has been supported by a joint grant from the U.S.-Israeli Binational Science Foundation. Work by Vladlen Koltun and Micha Sharir has also been supported by a grant from the Israel Science Fund (for a Center of Excellence in Geometric Computing). Work by Boris Aronov was also supported by NSF Grants CCR-99-72568 and ITR CCR-00-81964. Work by Vladlen Koltun was also supported by NSF Grant CCR-01-21555 and by the Rothschild Post-doctoral Fellowship. Work by Micha Sharir was also supported by NSF Grants CCR-97-32101 and CCR-00-98246, and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University.

†Department of Computer and Information Science, Polytechnic University, Brooklyn, NY 11201-3840, USA; *aronov@ziggy.poly.edu*.

‡Computer Science Division, University of California, Berkeley, CA 94720-1776, USA; *vladlen@cs.berkeley.edu*.

§School of Computer Science, Tel Aviv University, Tel-Aviv 69978, Israel, and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA; *michas@post.tau.ac.il*.

The importance of determining which parts of the scene objects are occluded was recognized in computer graphics from its very beginning. Until the 1970s, 'Hidden-Surface Removal' (HSR) was considered one of computer graphics' most important problems, and has received a substantial amount of attention; see [12] for a survey of the ten leading HSR algorithms circa 1974.

A commonly used HSR technique is the *z-buffer* [3], which produces a 'discrete' solution to the problem. Given a computer screen with a specific resolution, the z-buffer heuristically determines for each pixel on the screen the object that is closest to the viewpoint inside the area represented by the pixel. Since the z-buffer yields to efficient implementations in hardware, it is usually the HSR method of choice. It is not, however, applicable in all situations. Since its output consists of a finite number of samples, instead of an analytic description of the visible part of the scene, it does not provide the data necessary for vector-based output devices, and is highly inefficient in terms of memory consumption when dealing for example with high-quality large-scale printing tasks, which require producing images at exceedingly high resolutions. An analytic solution requires little memory and storage space, and can be used to produce images of arbitrary resolution.

These considerations motivated a long study of hidden-surface removal in computational geometry, culminating in the early 1990s with a number of algorithms that provide both conceptual simplicity and satisfactory running-time bounds. See de Berg [2] and Dorward [6] for overviews of these developments, and Overmars and Sharir [9] for a simple HSR algorithm with good theoretical running-time bounds.

A common feature of most HSR algorithms is that they rely on the existence of a consistent *depth order* for the input objects. For example, if object A occludes part of object B, and object B partially occludes object C, it is assumed that C will not occlude any part of A;

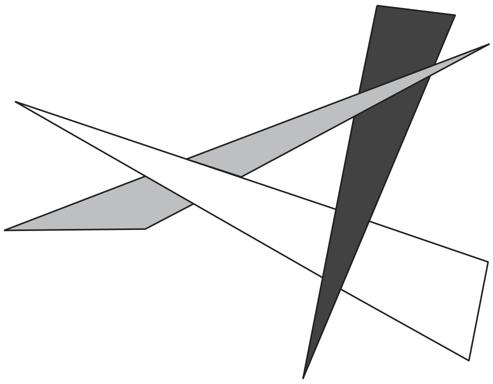

Figure 1: A depth cycle defined by three triangles.

the contrary situation is termed a *depth cycle*, or, simply, a *cycle*; see Figure 1. More precisely, it is assumed that the transitive closure of the relationship $A \prec B$, defined as A occluding part of B, is a partial order. This assumption is not always satisfied in practice, where depth cycles are easily encountered in real-world scenes involving tree branches, industrial pipes, etc. Nevertheless, the reliance on a consistent depth order is crucial to HSR algorithms, most of which begin by sorting the objects either front-to-back (e.g., the Overmars-Sharir algorithm [9]) or back-to-front (e.g., the classical Painter's Algorithm [12]).

A large number of algorithms have been developed for *testing* whether the depth relationship in a collection of triangles contains a cycle with respect to a specific viewpoint; see de Berg [2] and the references therein. However, while these algorithms help detect cycles, they do not provide strategies for dealing with them.

One such strategy is to eliminate all depth cycles, with respect to a specific viewpoint, by cutting the objects into pieces that do not form cycles, and running an HSR algorithm on the resulting collection of pieces. In 1980, Fuchs et al. [7] introduced *Binary Space Partition (BSP) trees*, which can be used to perform the described cutting. However, a BSP tree may force up to a quadratic number of cuts [10], which is problematic in light of the fact that virtually all of the research into hidden-surface removal has concentrated on the development of output-sensitive algorithms that run in subquadratic time whenever possible [2, 6].

It has been open since 1980 whether one can devise an algorithm that, given a specific viewpoint and a collection of n triangles in $\mathbb{R}^3$, removes all depth cycles defined by this collection with respect to the viewpoint using a subquadratic number of cuts. The work of Solan [11] and of Har-Peled and Sharir [8] implies that this is indeed possible, provided a subquadratic number of cuts is known to be sufficient. In particular, these works present algorithms that, given a collection $\mathcal{L}$ of n lines in 3-space, perform close to $O(n\sqrt{C})$ cuts that eliminate all cycles defined by $\mathcal{L}$ as seen from $z = -\infty$, where C is the minimal required number of such cuts.[1] That is, if we can provide a subquadratic bound on the minimum number of cuts that suffice to eliminate all cycles defined by a collection of lines, then the aforementioned algorithms are guaranteed to find a collection of such cuts of (potentially larger but still) subquadratic size.

Such an upper bound has however remained elusive. The only progress in this direction is due to Chazelle et al. [4], who in 1992 have analyzed the following special case of the problem. A collection of line segments in the plane is said to form a *grid* if it can be partitioned into two subcollections of 'red' and 'blue' segments, such that all red (resp., blue) segments are pairwise disjoint, and all red (resp., blue) segments intersect all blue (resp., red) segments in the same order; see Figure 2. Chazelle et al. [4] have shown that if the xy-projections of a collection of n segments in 3-space form a grid, then all cycles defined by this collection (again, as seen from $z = -\infty$) can be eliminated with $O(n^{9/5})$ cuts.

Our contribution. This paper describes the first step towards obtaining subquadratic general upper bounds on the number of cuts that are sufficient to eliminate all cycles defined by a collection of lines in space. Specifically, we

[1] It can be easily shown that stating the problem in terms of collections of lines, instead of the original setting of triangles, does not diminish the problem complexity but does simplify the exposition of the results. Moreover, we can assume without loss of generality that the viewpoint lies at $z = -\infty$, relying on an appropriate transformation of the 3-dimensional space. All previous work on cutting cycles has thus been done with regard to collections of lines or line segments that are viewed from $z = -\infty$ [4, 8, 11]. Since any cycle defined by a collection of line segments is also a cycle in the collection of lines spanned by these segments, we will concentrate on the case of lines.

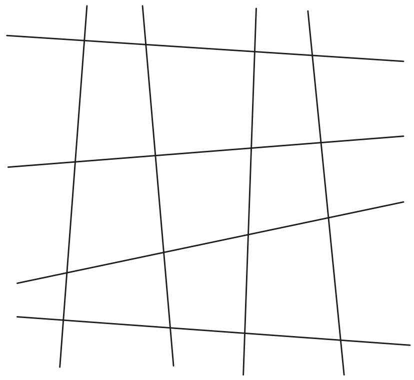

Figure 2: A collection of line segments that forms a grid.

show that all *triangular cycles*, which are cycles formed by triples of lines, can be eliminated with $O(n^{2-1/69+\varepsilon})$ cuts, for an arbitrarily small $\varepsilon > 0$. While this bound is still far from the lower bound $\Omega(n^{3/2})$ that Chazelle et al. [4] have provided for this quantity, and does not immediately apply to cycles defined by an arbitrary number of lines, it is an essential first step towards the complete solution. As the first nontrivial general upper bound for this problem, since the problem's conception more than 20 years ago, we expect it to be generalized and improved, and the techniques we introduce to be extended and simplified. A central component in our proof is a result of independent interest concerning the unrealizability of a certain weaving pattern of lines; see full version for details [1].

2 Cutting Triangular Cycles

Let us provide a formal definition for the problem of cutting cycles. Let $\mathcal{L}$ be a set of n non-vertical lines in 3-space in general position. Define the *depth order* $\prec$ on $\mathcal{L}$ to be such that $\ell \prec \ell'$ if ℓ *passes below* ℓ'; that is, the unique vertical line λ that connects ℓ and ℓ' meets them at two respective points p, p' so that the z-coordinate of p is smaller than that of p'. The relationship $\prec$ can have cycles, and our challenge is to obtain nontrivial bounds on the number of cuts that need to be applied to the lines of $\mathcal{L}$, so that the depth order among the resulting segments and rays (defined in exactly the same manner as for lines) has no cycles.

Let ℓ^* denote the xy-projection of a line ℓ, and let $\mathcal{L}^* = \{\ell^* \mid \ell \in \mathcal{L}\}$ denote the set of the projections of the lines in $\mathcal{L}$. A cycle c in $\mathcal{L}$ of the form $\ell_1 \prec \ell_2 \prec \cdots \prec \ell_j \prec \ell_1$ can be represented as a closed oriented (possibly self-intersecting or even self-overlapping) polygonal path $c^* = p_1 p_2 \ldots p_n p_1$, where p_i is the intersection point of ℓ_i^* and $\ell_{i+1 \pmod j}^*$.

The simplest kind of a cycle in the depth order is a *triangular cycle* defined by three lines ℓ_1, ℓ_2, ℓ_3, satisfying $\ell_1 \prec \ell_2 \prec \ell_3 \prec \ell_1$. We call a triangular cycle c a *clockwise* (resp., *counterclockwise*) cycle if the resulting orientation of c^* (as we trace it in the order $\ell_1^* \to \ell_2^* \to \ell_3^* \to \ell_1^*$) is clockwise (resp., counterclockwise); see Figure 3.

In this paper we confine our study to triangular cycles; thus from now on, the unqualified term 'cycle' will always refer to a triangular cycle. We therefore wish to cut the lines in $\mathcal{L}$ so that all such cycles are eliminated. Here is a simple procedure that achieves this goal. Fix a parameter k to be determined later. For each $\ell \in \mathcal{L}$, cut ℓ at (the points projecting on) every k-th vertex of $\mathcal{A}(\mathcal{L}^*)$ lying on ℓ^*. The total number of cuts is $O(n^2/k)$. It is easy to see that after these cuts are performed, any cycle c that has not been eliminated has the property that c^* is crossed by at most $3k/2$ lines of $\mathcal{L}^*$. Using the probabilistic analysis technique of Clarkson and Shor [5], the overall number of these 'light' triangular cycles is $O(k^3 \nu_0(n/k))$, where $\nu_0(m)$ is the maximum number of triangular cycles c in a collection of m lines in space, such that c^* is a *face* in the arrangement of the projected lines. (We refer to cycles of the latter type as *empty*.) Hence, we can certainly eliminate all triangular cycles in $\mathcal{L}$ using

$$O\left(\frac{n^2}{k} + k^3 \nu_0\left(\frac{n}{k}\right)\right) \qquad (1)$$

cuts.

Let C be a family of triples (ℓ_1, ℓ_2, ℓ_3) of distinct lines of $\mathcal{L}$, such that each triple in C forms a counterclockwise triangular cycle whose xy-projection is a face of $\mathcal{A}(\mathcal{L}^*)$. It suffices to obtain a bound on $|C|$, since the overall number of triangular face cycles is at most twice this bound. Such a bound is given in the following theorem, whose proof constitutes the main technical part of the full version of this paper [1].

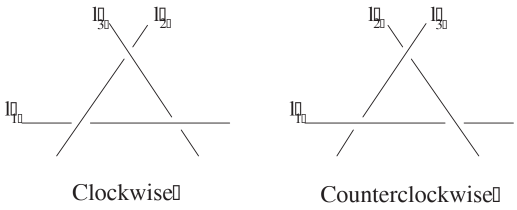

Figure 3: The two kinds of triangular cycles.

Theorem 2.1. *Given a set $\mathcal{L}$ of n nonvertical lines in $\mathbb{R}^3$ in general position, the number of empty triangular counterclockwise cycles defined by $\mathcal{L}$ is $O(n^{2-1/34+\varepsilon})$, for any $\varepsilon > 0$.*

This theorem states that $|C|$ is bounded by $O(n^{2-1/34+\varepsilon})$, for any $\varepsilon > 0$, which implies that $\nu_0(n) = O(n^{2-1/34+\varepsilon})$. Plugging this estimate into (1) we conclude that the number of cuts needed to eliminate all triangular cycles in $\mathcal{L}$ is

$$O\left(\frac{n^2}{k} + k^3\left(\frac{n}{k}\right)^{2-1/34+\varepsilon}\right) = O\left(\frac{n^2}{k} + k^{35/34-\varepsilon}n^{2-1/34+\varepsilon}\right).$$

Choosing $k = n^{1/69}$, and replacing ε by an appropriate multiple, we obtain the main result of this paper.

Theorem 2.2. *A set $\mathcal{L}$ of n nonvertical lines in $\mathbb{R}^3$ in general position can be cut into $O(n^{2-1/69+\varepsilon})$ segments and rays, for any $\varepsilon > 0$, such that no triangular cycles are present in the depth order of these portions of the lines.*

The interested reader is referred to the full version [1] for the missing technical details.

Acknowledgments

The authors wish to express their gratitude to Pankaj Agarwal, Sariel Har-Peled and Shakhar Smorodinsky for insightful suggestions concerning the material presented in this paper.

References

[1] B. Aronov, V. Koltun and M. Sharir. Cutting triangular cycles of lines in space. http://www.cs.berkeley.edu/~vladlen/cycles-conf.zip

[2] M. de Berg, *Ray Shooting, Depth Orders and Hidden Surface Removal*, Lecture Notes Comput. Sci., 703, Springer Verlag, Berlin, 1993.

[3] E. Catmull, *A Subdivision Algorithm for Computer Display of Curved Surfaces*, Ph.D. Thesis, UTEC-CSC-74-133, Dept. Comput. Sci., University of Utah, 1974.

[4] 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.

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

[6] S. E. Dorward, A survey of object-space hidden surface removal, *Internat. J. Comput. Geom. Appl.* 4 (1994), 325–362.

[7] H. Fuchs, Z. M. Kedem and B. Naylor, On visible surface generation by a priori tree structures, *Comput. Graph.* 14 (1980), 124–133.

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

[9] M. H. Overmars and M. Sharir, A simple output-sensitive algorithm for hidden surface removal, *ACM Transactions on Graphics* 11 (1992), 1–11.

[10] 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.

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

[12] I. E. Sutherland, R. F. Sproull and R. A. Schumacker, A characterization of ten hidden-surface algorithms, *ACM Comput. Surv.* 6 (1974), 1–55.