On Triangulating Three-Dimensional Polygons*

Gill Barequet[†] Matthew Dickerson[‡] David Eppstein[§]

Abstract

A three-dimensional polygon is *triangulable* if it has a non-self-intersecting triangulation which defines a simply-connected 2-manifold. We show that the problem of deciding whether a 3-dimensional polygon is triangulable is $\mathcal{NP}$-Complete. We then establish some necessary conditions and some sufficient conditions for a polygon to be triangulable, providing special cases when the decision problem may be answered in polynomial time.

Keywords: Three-dimensions, triangulation.

1 Introduction

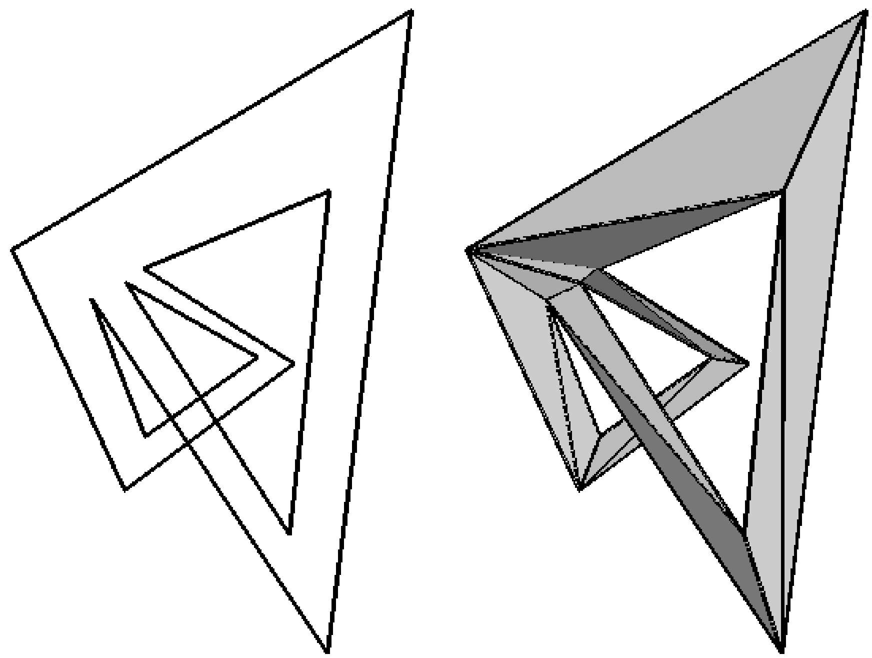
A 3-dimensional polygon is a closed chain of straight segments, where every two successive segments share exactly one point and the intersection of every non-successive pair of segments is empty. A triangulation of a 3-dimensional polygon has the same combinatorial structure as that of a planar polygon, that is, every edge of the polygon appears in exactly one triangle, all the other edges of the triangulation appear in exactly two triangles, and the surface defined by the triangulation is topologically a disk (simply connected). Figure 1(a) shows a complex 3-dimensional polygon. The surface shown in Figure 1(b) is not a valid triangulation of the polygon since it is not simply connected. We require in addition that no two triangles intersect in their interiors, so that a valid triangulation defines a piecewise-linear non-self-intersecting 2-manifold with one boundary (the original polygon). Figure 2(a) shows another 3-dimensional polygon.

*Work on this paper by the first author has been supported by the Israeli Ministry of Science and the Arts under an Eshkol Grant 0562-1-94. Work by the second author has been supported by the funds of the National Science Foundation CCR-9301714. Work by the third author has been supported by NSF grant CCR-9258355 and by matching funds from Xerox Corp.

[†]Dept. of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail: barequet@math.tau.ac.il (current address: Center for Geometric Computing, Dept. of Computer Science, Johns Hopkins University, Baltimore, MD 21218. E-mail: barequet@cs.jhu.edu)

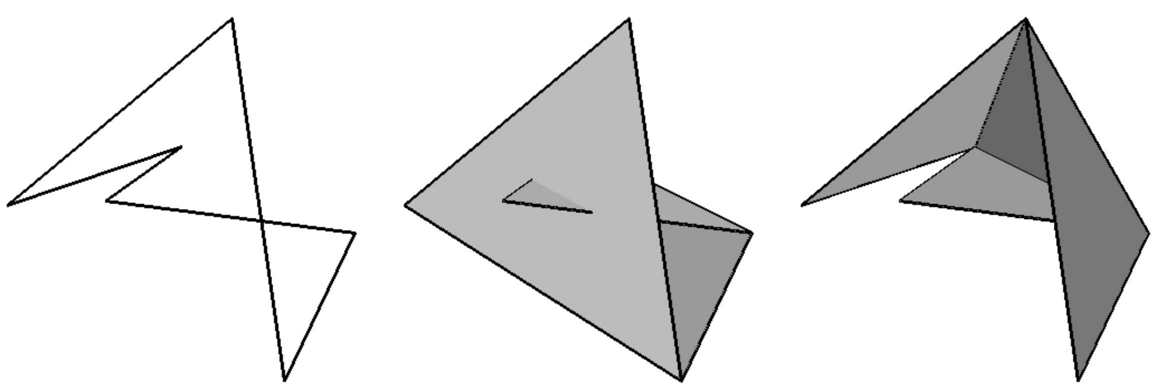
[‡]Dept. of Mathematics and Computer Science, Middlebury College, Middlebury, VT 05753. E-mail: dickerso@middlebury.edu

[§]Dept. of Information and Computer Science, University of California, Irvine, CA 92717. E-mail: eppstein@ics.uci.edu.



(a) Polygon (b) Non-triangulation

Figure 1: An invalid multiply-connected triangulation



(a) Polygon (b) Non-triangulation (c) Triangulation

Figure 2: A valid triangulation does not intersect itself

Figures 2(b,c) show two triangulations of it, the first of which is invalid because it intersects itself. Note that it is clear that every simple planar polygon has a triangulation [12, proposition 30]. A 3-dimensional polygon need not be planar, nor triangulable. In this paper we investigate the triangulability of 3-dimensional polygons.

The triangulation of a planar polygon attracted considerable attention in the literature during the past decade. See, for example, papers by Edelsbrunner [8] or Bern and Eppstein [4] for comprehensive reviews on this subject. The complexity of triangulating a planar polygon was an open problem in computational geometry until Chazelle [7] closed the issue by presenting an ingenious optimal $\Theta(n)$ algorithm. The main tool used in this algorithm is the *visibility map*, which crucially depends on the planarity of the polygon. Unfortunately, for a polygon in three dimensions, the visibility notion becomes irrelevant. (In general position every vertex "sees" every other vertex, thus the visibility-map is not useful any more.) Moreover, the triangulation problem ceases to be decomposable in any simple way (as it is in the plane). A candidate triangle whose vertices are on the polygon does not necessarily split the problem into three independent (smaller) triangulation subproblems; valid solutions of the subproblems may intersect in 3-space, thus their union is not a valid solution of the original problem. The goal is therefore to compute a triangulation of a 3-dimensional polygon if one exists, or to report that no such triangulation exists.

We point out that in three dimensions, the term 'triangulation' usually refers to the tetrahedralization of a 3-dimensional polyhedron or a set of points. We are not aware of previous work on the triangulation of 3-dimensional polygons. However special cases of this problem arise in various contexts, in which the problem is to compute an unknown (polyhedral) surface when only its (polygonal) boundary is known. The surface-reconstruction algorithm of Barequet and Sharir [3] computes (as a subproblem) optimal (minimum-area) triangulations of 3-dimensional biplanar polygons. This algorithm has applications to medical imaging and to the reconstruction of topographic terrains. Several algorithms for repairing CAD objects [2, 5, 14] use 3-dimensional triangulation procedures for filling cracks in polyhedral surfaces.

A more general question is: if a 3-dimensional polygon is triangulable, how many triangulations are there? If we neglect the possible intersections between triangles in 3-space, then the *number* of triangulations of an n-gon identifies with that of a planar convex polygon: $\frac{1}{n-1}\binom{2n-4}{n-2}$ [8, p. 76]. This is, in fact, the $(n-2)$th Catalan number. In general, however, computing the number of valid (non-self-intersecting) triangulations of a 3-dimensional polygon is a non-trivial problem. This leads to another interesting problem which is to compute an *optimal* triangulation for some measure of optimality. This goal is naturally more ambitious than computing any valid triangulation or just determining triangulability. If we allow self-intersecting triangulations (in contrast with our definition), then computing the optimal triangulation can be done (in cubic time) by applying a simple dynamic-programming procedure (see [2, 3]).

In this paper we address several problems related to 3-dimensional polygons. Specifically, we prove that the triangulability of a polygon in 3-space is $\mathcal{NP}$-Complete. We then prove a few necessary and sufficient conditions for the triangulability of a 3-dimensional polygon, and present polynomial-time algorithms for determining whether the sufficient conditions hold. We regard this work as an opening for this issue, with ample space for many more findings.

The paper is organized as follows. In Section 2 we define the triangulability of a 3-dimensional polygon and prove the $\mathcal{NP}$-Completeness of the decision problem. In Section 3 we discuss sufficient and necessary conditions for the 3-dimensional polygon triangulability. In Section 4 we present polynomial-time algorithms for testing the sufficient conditions. We terminate in Section 5 with some open problems.

2 $\mathcal{NP}$-Completeness of 3-Dimensional Polygon Triangulability

We address the following question:

Problem 1 *Given a 3-dimensional polygon, is it triangulable?*

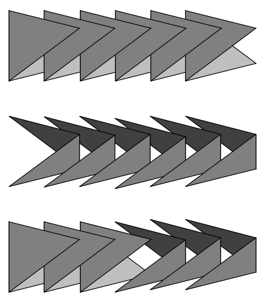
In other words, does a given polygon have a non-self-intersecting triangulation?

We first show that the problem of determining whether a three-dimensional polygon admits a triangulation is $\mathcal{NP}$-Complete. (Tetrahedralization of three-dimensional polyhedra is also known to be $\mathcal{NP}$-Complete [15].) The problem is obviously in $\mathcal{NP}$: testing whether two 3-dimensional triangles (whose vertices have integer coordinates) intersect can be performed in time polynomial in the bit complexity of the coordinates,[1] and checking whether a given candidate solution (a triangulation of a polygon P) does not intersect itself requires $O(n^2)$ such tests, where n is the complexity of P.

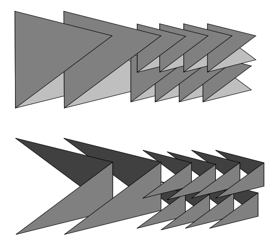
We start with a somewhat simpler proof of a related hardness result. Define the *generalized three-dimensional triangulation* problem to be one of, given a collection of three-dimensional polygons, determine whether one can simultaneously triangulate all polygons in the collection with a collection of triangulations that do not intersect themselves or each other. Our proof that generalized triangulation is hard is via a reduction from 3-SAT. As is usual in this sort of proof, we construct gadgets corresponding to 3-SAT variables, clauses, and connections between them, in such a way that putting together the gadgets corresponding to the objects in a 3-SAT instance results in a collection of polygons, with the collection having a triangulation if and only if the 3-SAT instance is satisfiable.

We construct most of our gadgets using a quadrilateral in which opposite edges are skew. Any quadrilateral has exactly two triangulations. By stacking several such quadrilaterals in a row, we construct gadgets resembling wires (Figure 3(a)). At each

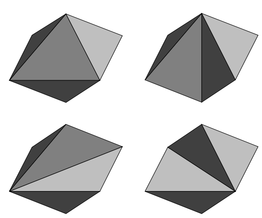
[1]This can be done by expressing the intersection in terms of low degree polynomials—signs of 4×4 determinants. These polynomials can be evaluated exactly by using four times as much precision in the intermediate results as the inputs coordinates.



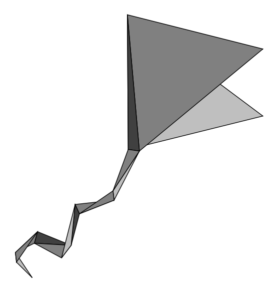
(a) Stacked skew quadrilaterals form a wire, with the presense of a signal indicated by a concavity at the end of the wire.



(b) Gadget for replicating a signal from one wire to two.



(c) Hexagon formed by vertices of an octahedron, triangulated in three ways. Attaching wires to the back of each long diagonal forms a 3-SAT clause.



(d) Knotted connections force each polygon to be triangulated independently.

Figure 3: Components of the $\mathcal{NP}$-Completeness proof

end of a wire, the quadrilateral can have either a convex projecting triangulation or a concave inward triangulation; we think of a wire as carrying a signal (representing the truth of a variable or its negation) when the triangulation is concave. As shown in the figure, a signal can "fade out" when two adjacent quadrilaterals have opposite triangulations, but the reverse transition is not possible: a signal cannot be created from the absence of one without an intersection between triangles in adjacent quadrilaterals.

For each variable in the 3-SAT formula, we then create a "truth-setting" gadget simply consisting of one of these wires (indeed, it could just be a single skew quadrilateral). One end of the wire will correspond to the variable itself, while the other will correspond to the negation. Any such gadget can have at most one signal, either at the end corresponding to the variable (corresponding to assignments in which the variable is true) or at the other end (corresponding to assignments in which the variable is false). It is also possible to have no signal at either end, but this will turn out not to be a problem in our construction (we can show that it is still possible to turn a triangulation into a satisfying assignment in this case; the assignment can be thought of as having a "don't care" state in which either truth value leads to satisfaction).

We will need to create multiple signals for each variable, to be sent to each of the clause gadgets in which the variable is involved. To do this, we use a second type of gadget, a "splitter" (see Figure 3(b)). A wire carrying a signal is fed into one end of this gadget (the left side of the figure), and two wires carrying the same signal come out the other end (the right side of the figure). This splitter has the same property as the wires, that a signal can fade or be propagated but it is not possible to create a new signal from the absence of one.

To complete the construction, we need a clause gadget, that takes three incoming wires and is triangulable if and only if a signal is present on at least one wire. Unlike the other gadgets, it is not possible to build clauses out of quadrilaterals; for if a generalized triangulation instance consisted solely of quadrilaterals it could be translated to a form of 2-SAT and solved in polynomial time. Instead we use a hexagon formed from the vertices of a regular octahedron. Figure 3(c, top-left) shows such an octahedron, with a front face and three other faces visible. The hexagon we wish to use in that figure is the silhouette of the octahedron. This hexagon has fourteen triangulations: one with the front face, one with the other internal triangle, and twelve others each using one of the three main diagonals of the hexagon (none of which contain an internal triangle). In the figure we show four triangulations: the one using the front face, and one for each main diagonal.

To eliminate the possibility of using an internal triangle in a triangulation, we add to our collection of polygons two small triangles intersecting those faces but not interfering with any other possible face of a triangulation. Finally we attach three wires to the gadget, corresponding to the variables in the clause. Each wire is attached by placing it at the back of the gadget shown in Figure 3(c), near one of the three main diagonals, at the end of the diagonal nearest the back face of the octahedron. As can be seen in the figure, the two triangulations using the other two main diagonals use faces well separated from this connection point. If the wire is appropriately placed, it will avoid the third triangulation as well when there is a signal on the wire, but when there is no signal the last quadrilateral of the wire will be triangulated with a triangle that links around the corresponding main diagonal of the hexagon, preventing the use of that main diagonal in any triangulation. Thus this clause gadget will have a triangulation if and only if there is a signal on at least one of the three incoming wires.

This completes the description of the gadgets involved in our $\mathcal{NP}$-Completeness construction for generalized triangulation. To complete the construction, we need only hook the gadgets together. We place them roughly in a common plane, and make them well separated in that plane. The connections necessary between them can be approximated by polygonal chains with $O(1)$ links, leaving and returning to the plane by two long edges. Each such connection can then be filled out by a wire with $O(1)$ quadrilaterals. In this way the total number of vertices needed in the construction is linear in the complexity of the input 3-SAT formula, and explicit integer coordinates for each vertex can easily be constructed in polynomial time.

Theorem 1 *The construction outlined above is a polynomial reduction from 3-SAT to generalized triangulation, and therefore shows that generalized triangulation is $\mathcal{NP}$-Complete.*

Proof: As discussed above, the reduction is polynomial time. It remains to show that satisfying 3-SAT assignments correspond to non-self-intersecting triangulations and vice versa. From any satisfying assignment to the 3-SAT formula, form a triangulation by setting each quadrilateral to represent a signal on wires corresponding to true variables (or to the negations of false variables) and the absence of a signal on all other wires. Then each clause gadget will have at least one incoming signal, so as discussed above it can be triangulated. Conversely suppose one has a triangulation of the collection of polygons. Each clause gadget must use one main diagonal in its triangulation, and must therefore have an incoming signal; because signals cannot be created except at truth-setting gadgets, the assignment coming from the state of each truth-setting gadget must be a satisfying assignment. □

We now describe how to modify this construction to prove $\mathcal{NP}$-Completeness of our original problem, triangulation of a single three-dimensional polygon. The idea is simple: we start with the generalized triangulation problem constructed above, and connect the individual polygons of that problem by narrow “ribbons” to form a single simple polygon (Figure 3(d)). (To connect each ribbon, we split one vertex of the polygon into two vertices very near to each other). For the generalized polygon problem constructed by the reduction above, it is easy to place ribbons in a way that does not interfere with any existing triangulation of the generalized problem (although it is not clear how to do this in general).

However in order to prove $\mathcal{NP}$-Completeness we also need a translation in the other direction, from constructed triangulations to satisfying assignments; therefore we must be careful that in the process of adding ribbons we do not allow extra triangulations not coming from the original generalized triangulation problem. We do this by making three small non-collinear knots near the start of each ribbon (not shown in Figure 3(d)). Any diagonal connecting a vertex of the polygon at the end of the ribbon to another part of the input would pass around ("outside") at least one of these knots, and therefore form a knot with the ribbon boundary, perhaps compounded with other knotting elsewhere. Since a compound of knots is always itself knotted [1, pp. 9 and 104][2] and knots cannot be triangulated (see Section 3.1), such a diagonal cannot be part of any triangulation of the input, so any triangulation of the input must triangulate each polygon independently, in roughly the same shape as a triangulation of the original generalized triangulation problem. Thus we have the following result:

Theorem 2 *Three-dimensional triangulation is $\mathcal{NP}$-Complete.*

3 Triangulability in Special Cases

Since triangulability of a 3-dimensional polygon is $\mathcal{NP}$-Complete, we now focus our attention on some special cases. In this section we present some necessary conditions for triangulability, and some sufficient conditions for triangulability.

3.1 Knotted Polygons

It is fairly easy to prove that:

Theorem 3 *A knotted 3-dimensional polygon does not have a non-self-intersecting triangulation.*

Proof: Assume to the contrary that a knotted 3-dimensional polygon has a non-self-intersecting triangulation. The triangulation is thus outerplanar. Every outerplanar triangulation has an *ear* (a triangle connected to the rest by a single edge) and can be reduced to a single triangle by removing ears one at a time. Consequently, this ear removal process provides a recipe for untangling the triangulation's boundary, contradicting the fact that it is knotted. □

In fact, the last claim is a special (piecewise-linear) case of a more general theorem that states that a closed curve is unknotted if and only if it has a spanning disk (see, e.g., [17, Lemma 1]):

Theorem 4 *A non-self-intersecting simply-connected surface (2D manifold topologically equivalent to a disk) cannot have a knotted boundary.*

[2] See also [13, p. 80, corollary 11]), where Livingstone attributes this result to Schubert [16].

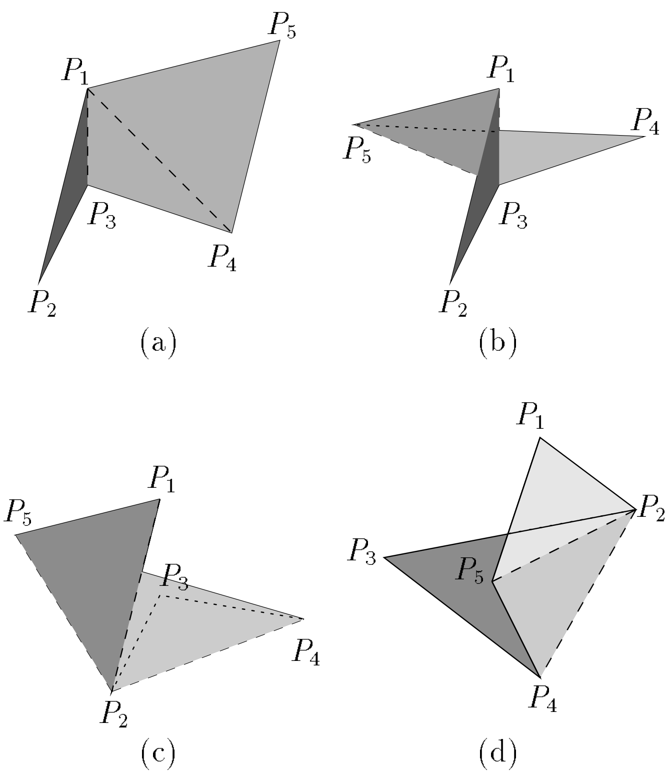


Figure 4: Triangulations of 3-dimensional pentagons

3.2 Unknotted Polygons

A known result in knot theory states that all k-gons with $k \leq 5$ are unknotted (see, e.g., [1, §1.6]). We prove a slightly stronger claim:

Theorem 5 *Every 3-dimensional triangle, quadrilateral, or pentagon, is triangulable.*

Proof: A triangle in 3-space is the triangulation of itself.

Every 3-dimensional quadrilateral has two triangulations formed by its two diagonals, with the single exception where the four vertices lie in the same plane and define a concave quadrilateral. In this case only the triangulation formed by the diagonal that fully lies inside the quadrilateral (within the containing plane) is valid.

Refer now to a 3-dimensional pentagon $(P_1, P_2, P_3, P_4, P_5)$. Assume first that P_4 and P_5 are on the same side of the plane Q defined by the vertices P_1, P_2, and P_3 (e.g., as shown in Figure 4(a)). In this case we initialize the triangulation by the triangle $T_1 = (P_1, P_2, P_3)$. The remaining quadrilateral (P_1, P_3, P_4, P_5) lies on the same side of Q, so its triangulation cannot intersect with T_1.

Assume now that P_4 and P_5 are separated by Q, but the edge P_4P_5 does not intersect with T_1. Here we have three subcases. In the first subcase (shown in Figure 4(b)) the edge P_4P_5 is not on the same side of the segment P_1P_3 (which is not an edge of the

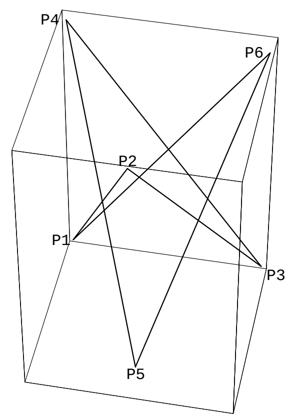


Figure 5: A non-triangulable 3-dimensional biplanar hexagon

polygon) as P_2 (where the side is determined according to whether the intersection point of P_4P_5 with the plane that contains T_1 is on the same side of P_1P_3 as P_2). This means that there is a plane that contains P_1 and P_3, and separates between P_2 to P_4 and P_5. Hence the triangulation of (P_1, P_3, P_4, P_5) does not intersect with T_1. In the other subcases the edge P_4P_5 is on the same side of the segment P_1P_3 as P_2 but still does not intersect with T_1. So P_4P_5 is either (second subcase) not on the same side of P_2P_3 as P_1 (that is, "below" it), or (third subcase) not on the same side of P_1P_2 as P_3 (that is, "above" it). In the second subcase we have the same triangulation as of the first subcase, and in the third subcase we have the triangulation $((P_1, P_4, P_5), (P_1, P_4, P_3), (P_1, P_3, P_2))$.

In the remaining case P_4 and P_5 are separated by Q, but the edge P_4P_5 intersects with T_1 (see Figure 4(c)). For this to happen, P_3 and P_4 have to be on the same side of the plane defined by P_5, P_1, and P_2, and P_1 and P_5 have to be on the same side of the plane defined by P_2, P_3, and P_4. Thus we apply the same argument as in the first case, and obtain the valid triangulation $((P_1, P_2, P_5), (P_2, P_4, P_5), (P_2, P_3, P_4))$.

It is also easy to handle cases where the vertices are not in general position. In case all the five vertices lie in one plane, we have to triangulate a simple pentagon in the plane. In case only four of the vertices lie in the same plane, and the fifth vertex P_5 (with no loss of generality) lies outside that plane (see Figure 4(d)), we construct, again, the triangulation $((P_1, P_2, P_5), (P_2, P_4, P_5), (P_2, P_3, P_4))$. (Note that we could not start with the triangle (P_1, P_4, P_5), since the remaining quadrilateral (P_1, P_2, P_3, P_4) might be, as in this example, a non-simple planar polygon.) □

It is not true that every simple (unknotted) 3-dimensional polygon is triangulable. Consider the 3-dimensional hexagon shown in Figure 5. The six vertices of the hexagon are $P1 = (0, 1.73, 0)$, $P2 = (1, -0.2, 3)$, $P3 = (2, 1.73, 0)$, $P4 = (0, 1.73, 3)$, $P5 = (1, 0.1, 0)$, and $P6 = (2, 1.63, 3)$. It is easy to verify that this hexagon is not a knot. According to the formula given in Section 1, every hexagon has exactly 14 triangulations, which are listed in Table 1. Each triangulation of the hexagon contains at least one pair of intersecting triangles. Thus we have the following:

	Triangulation				
	T_1	T_2	T_3	T_4	Intersecting Triangles
1	(P_6, P_1, P_5)	(P_5, P_1, P_2)	(P_5, P_2, P_4)	(P_4, P_2, P_3)	(T_1, T_4)
2	(P_6, P_1, P_5)	(P_5, P_1, P_2)	(P_5, P_2, P_3)	(P_5, P_3, P_4)	(T_1, T_4)
3	(P_6, P_1, P_5)	(P_5, P_1, P_4)	(P_4, P_1, P_3)	(P_3, P_1, P_2)	(T_1, T_4), (T_2, T_4)
4	(P_6, P_1, P_5)	(P_5, P_1, P_3)	(P_3, P_1, P_2)	(P_5, P_3, P_4)	(T_1, T_3), (T_1, T_4), (T_3, T_4)
5	(P_6, P_1, P_5)	(P_5, P_1, P_4)	(P_4, P_1, P_2)	(P_4, P_2, P_3)	(T_1, T_4)
6	(P_4, P_5, P_6)	(P_4, P_6, P_1)	(P_4, P_1, P_3)	(P_3, P_1, P_2)	(T_1, T_4)
7	(P_4, P_5, P_6)	(P_4, P_6, P_1)	(P_4, P_1, P_2)	(P_4, P_2, P_3)	(T_1, T_4), (T_2, T_4)
8	(P_4, P_5, P_3)	(P_3, P_5, P_6)	(P_3, P_6, P_1)	(P_3, P_1, P_2)	(T_1, T_3), (T_1, T_4), (T_2, T_4)
9	(P_4, P_5, P_6)	(P_4, P_6, P_3)	(P_3, P_6, P_1)	(P_3, P_1, P_2)	(T_1, T_4)
10	(P_6, P_1, P_2)	(P_6, P_2, P_3)	(P_6, P_3, P_5)	(P_5, P_3, P_4)	(T_1, T_4)
11	(P_6, P_1, P_2)	(P_6, P_2, P_3)	(P_6, P_3, P_4)	(P_6, P_4, P_5)	(T_1, T_4)
12	(P_6, P_1, P_2)	(P_6, P_2, P_5)	(P_5, P_2, P_3)	(P_5, P_3, P_4)	(T_1, T_4)
13	(P_6, P_1, P_2)	(P_6, P_2, P_5)	(P_5, P_2, P_4)	(P_4, P_2, P_3)	(T_1, T_3), (T_1, T_4), (T_2, T_4)
14	(P_6, P_1, P_2)	(P_6, P_2, P_4)	(P_6, P_4, P_5)	(P_4, P_2, P_3)	(T_1, T_3), (T_1, T_4), (T_3, T_4)

Table 1: The 14 triangulations of a hexagon

Theorem 6 *There exist unknotted 3-dimensional n-gons (for $n \geq 6$) which are not triangulable.*

Thus unknottedness is a necessary but not sufficient condition for triangulability. Furthermore, efficiently determining knottedness is itself a difficult open question. The question whether a 3-dimensional polygon is knotted is decidable [10], but it is not even known to be in $\mathcal{NP}$ [17].

A *biplanar* polygon is a 3-dimensional polygon whose vertices lie in two parallel planes. The number of edges that connect ("jump") between the two planes is not limited. Computing a triangulation of biplanar polygons appears, for example, in interpolation problems (see, e.g., [3][3]). Unfortunately, biplanarity (even of an unknotted polygon) is not a sufficient condition for triangulability. The hexagon shown in Figure 5 is an unknotted biplanar polygon which is not triangulable.

It is easy to verify that we may still triangulate every unknotted 3-dimensional polygon if we allow the addition of Steiner points in space. Snoeyink [17] shows that the number of such Steiner points may be in the worst case exponential in the complexity of the original polygon.[4]

3.3 Sufficient Conditions for Triangulability

In this section we discuss a few sufficient conditions for triangulability. The common aspect of all the presented conditions is the simplicity of some projection of the polygon. Unfortunately, it is fairly easy to show that none of them is a necessary condition.

[3]Note that the 3-dimensional polygons of [3], so-called *clefts*, always have simple orthogonal projections, hence they are triangulable.

[4]Snoeyink refers to unknotted polygons as "trivial knots".

3.3.1 Simple Orthogonal Projection

Theorem 7 *If the orthogonal projection of a 3-dimensional polygon P onto some plane Q is simple, then P is triangulable.*

Proof: Denote the projection of P onto Q by P'. As noted above, every simple planar polygon has a triangulation. Triangulate P' within Q, and lift the triangulation back to the original vertices of P. Each 3-dimensional triangle is fully contained in a prism perpendicular to Q and bounded by the triangle itself and by the corresponding triangle within Q. No two such prisms intersect; neighboring prisms share a face that corresponds to an edge of the triangulation. Hence, the triangulation of P' within Q induces a non-self-intersecting triangulation of P in 3-space. □

In Section 4 we present an algorithm for computing a plane (if one exists) on which the projection of a given 3-dimensional polygon is simple.

3.3.2 Simple Perspective and Spherical Projections

Perspective and spherical projections have the same triangulability property as orthogonal projections. It is easy to prove the following:

Theorem 8 *If the perspective projection of a 3-dimensional polygon P from some point o onto some plane Q is simple, then P is triangulable.*

Proof: Identical to that of Theorem 7. □

Let S be a sphere in 3-space centered at o. Assume that a 3-dimensional polygon P is fully contained in S and does not pass through o. Each point p on P defines a ray $\vec{r_p}$ that starts at o and passes through p. A spherical projection of P onto S maps every point $p \in P$ to the intersection point of $\vec{r_p}$ with S.

We now show that:

Theorem 9 *If the spherical projection of a 3-dimensional polygon P outward from a point o onto some sphere S (centered at o) is simple, then P is triangulable.*

Proof: Let P' be the spherical projection of P from o onto S. The edges of P' on S are arcs portions of great circles on S. Perform a *central* projection $\mathcal{G}$ [18, pp. 16-18] of P' from o onto a plane tangent to S at any point not in P'.[5] In such mapping every orthodrom (the shortest path between two points) on S is mapped to a straight segment in the plane. Define $P'' = \mathcal{G}(P')$. We compute a triangulation $\mathcal{T}''$ of the planar polygon P'' and map the triangulation back to S: $\mathcal{T}' = \mathcal{G}^{-1}(\mathcal{T}'')$. Every edge of $\mathcal{T}''$ is

[5] In geographic terms this projection is called a *gnomonic* mapping. In fact, it maps only half of the sphere into a plane, so we need to consider two planes when we triangulate the polygon.

mapped back to an arc (portion of a great circle) on S. No two such arcs intersect, for if they did, the corresponding edges of the planar triangulation $\mathcal{T}''$ would also intersect. This triangulation of P' is a partition of one of the two portions of S bounded and separated by P' into spherical triangles. The combinatorial structure of $\mathcal{T}'$ is identical to the planar case: each such triangle is bounded by three arcs; every arc is shared by exactly two triangles, except for arcs that belong to P', which bound only one triangle; and no two such triangles intersect in their interiors. This triangulation of P' induces a "tetrahedralization" of a portion of S. Each "tetrahedron" $T_{t'}$ is defined by o and by the three vertices of a triangle $t' \in \mathcal{T}'$ on S, forming a shape with three planar faces and one spherical face. It is easy to verify that these "tetrahedra" do not intersect in their interiors and share faces which contain the edges of P. Moreover, for every triangle $t' \in \mathcal{T}'$ the triangle $\mathcal{G}^{-1}(t')$ is fully contained by $T_{t'}$. Otherwise the interiors of the "tetrahedra" would not be pairwise disjoint. Thus we can lift back the triangulation $\mathcal{T}'$ on S to 3-space and obtain a triangulation $\mathcal{T} = \mathcal{G}^{-1}(\mathcal{T}')$ of the original 3-dimensional polygon P. □

4 Computing Simple Projections of a 3-Dimensional Polygon

4.1 Computing a Simple Orthogonal Projection

In this section we describe an algorithm for determining whether there exists a simple orthogonal projection of a 3-dimensional polygon. Since this is a sufficient condition (as shown in Section 3.3.1), a positive answer implies the triangulability of the polygon.

Problem 2 *Given a 3-dimensional polygon P, report a plane Q (if one exists) on which the projection of P is simple. (Alternatively, report all the planes with this property.)*

4.1.1 Overview of the Algorithm

The main idea of the proposed algorithm is to exclude all the invalid directions of projections of P. An invalid direction is one that causes the projection of two edges of P to intersect. Thus each pair of edges of P determines a set of invalid directions. Each such set is a 4-sided region on the sphere of directions. First we construct the arrangement of these regions on the sphere; then we look for a depth-0 point in the arrangement. A point on the sphere not contained in any of the 4-sided regions corresponds to a projection in which the image of P is simple.

4.1.2 Computing the Elements of the Arrangement

First we compute all the "forbidden zones" on the sphere of directions. Denote the projection along a direction ℓ onto a plane orthogonal to ℓ by $\mathcal{P}_\ell$. The projected polygon

$\mathcal{P}_\ell(P)$ is non-simple if the projections of any two edges $e_1, e_2 \in P$ intersect. Therefore, all the invalid directions ℓ along which $\mathcal{P}_\ell(e_1)$ and $\mathcal{P}_\ell(e_2)$ intersect are defined by lines passing through points of e_1 and e_2. We represent such direction ℓ by its intersection with the sphere of directions. In this representation, all the invalid directions that correspond to each pair of edges $e_1, e_2 \in P$ define a simply connected 4-sided region on the sphere.

Each vertex of this quadrilateral is the direction that connects between *endpoints* of e_1 and e_2. The 'edges' of the quadrilaterals are arcs: portions of great circles on the sphere of directions. To illustrate this, simply fix a point of the line ℓ to be an endpoint of e_1, and move another point of ℓ continuously from one endpoint of e_2 to its other endpoint. Thus for computing the invalid directions determined by e_1 and e_2, we need only compute the four vertices of the quadrilateral. (Note that if one endpoint of e_1 is collinear with e_2, then the quadrilateral may degenerate into a triangle: a quadrilateral with side of length 0. Similarly, two parallel segments generate a degenerate "quadrilateral" that is only a "segment" on the sphere of directions.)

We repeat this procedure for all the $\binom{n}{2}$ pairs of edges of P and get $\Theta(n^2)$ forbidden zones on the sphere of directions. (Actually each pair of edges defines two such quadrilaterals, but for our purpose we may consider only half of the sphere of directions.) In this representation, the question whether there exists a direction along which the projection of P is simple amounts to determining whether there is a point of depth 0 in the arrangement of the $\binom{n}{2}$ quadrilaterals on the sphere of directions.

4.1.3 Constructing and Investigating the Arrangement

In order to simplify the arrangement, we regard the sphere of directions as a plane. There is an obvious problem with the poles, which is easily resolved by checking whether the poles are contained in any of the quadrilaterals. If not, then we have found a valid direction and may abort the algorithm. If yes, the poles are invalid directions. We thus ignore some small neighborhood of each pole which is fully contained in some quadrilateral. The size of such neighborhood can be computed, with no additional cost, during the pole-quadrilateral containment tests. Now we split the sphere along some longitude, and obtain the desired planar domain. (Splitting along a longitude may require the split of some, possibly all, the quadrilaterals, each into two parts. By doing that we may create pentagons which are further split into triangles.)

Finally we investigate the arrangement of the $\binom{n}{2}$ quadrilaterals, aiming to find whether there exists a point in the plane (clipped to the rectangular image of the sphere) of depth 0, that is, a point which is not covered by any quadrilateral. By splitting all the quadrilaterals and the target rectangle into two triangles each, we obtain the *triangles-cover-triangle* problem which belongs to the collection of so-called 3SUM-Hard problems [11]. Constructing the entire arrangement can be done, for example, by using topological sweeping [9]. The output of the algorithm may be an indication whether such point exists, or a description of the region (or regions) of depth 0.

4.1.4 Complexity Analysis

We measure the complexity of the algorithm as a function of n, the number of edges of the 3-dimensional polygon P.

Computing the forbidden 4-sided regions on the sphere of directions requires $\Theta(n^2)$ time, since each pair of edges contributes a quadrilateral. Testing the poles for being contained in these quadrilaterals and mapping the sphere into a plane also require $\Theta(n^2)$ time.

The main time-consuming step is constructing and investigating the arrangement of the quadrilaterals. Each pair of the $\Theta(n^2)$ quadrilaterals intersects in at most four points. Hence we may have $O(n^4)$ intersection points. A standard plane-sweep procedure requires $O(n^4 \log n)$ time. However, constructing the whole arrangement of $\Theta(n^2)$ quadrilaterals can be done in $O(n^4)$ time by using topological sweeping [9].

To conclude, the whole algorithm runs in $O(n^4)$ time in the worst case.[6] The space complexity of the algorithm is $\Theta(n^2)$.

4.1.5 Remarks

First, we note that the bottleneck of our algorithm is the topological sweeping of the arrangement. We were not able to exploit the fact that the quadrilaterals are not independent. Considering this fact may speed up the sweeping step. As indicated in [11], finding a faster algorithm for the covering problem will speed up a lot of other 3SUM-Hard problems.

Second, the existence of a simple orthogonal projection of P implies only its triangulability. Triangulating the projected polygon and then lifting the triangulation back to 3-space does not guarantee the minimality in any sense (e.g., surface area) of this triangulation.

Finally, this is only a sufficient but not a necessary condition. In other words, the absence of such projection does not rule out the triangulability of P.

4.2 Computing Simple Perspective and Spherical Projections

In this section we propose an algorithm for determining whether there exists a simple perspective (resp., spherical) projection on a plane (resp., sphere) of a 3-dimensional polygon. The idea of this algorithm is almost identical to that of the algorithm described in Section 4.1, so we provide only a sketch of the algorithm.

We use again the idea of "forbidden zones" that correspond to illegal projections. Both types of projections are performed from a single point (source). Each pair of

[6] Recently, Bose et al. [6] presented the same algorithm and pointed out that its time complexity is actually $O(n^2 \log n + k)$, where k is the number of intersections between the quadrilaterals, which is $O(n^4)$ in the worst case.

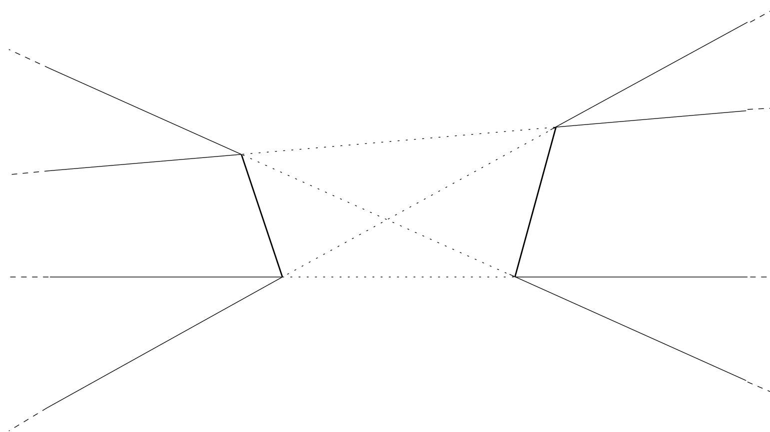


Figure 6: Two prisms containing forbidden sources of perspective and spherical projections

edges defines two such forbidden zones, delimited by the edges and by infinite portions of rays shot from the endpoints of one edge towards the endpoints of the other edge (see Figure 6). Thus each forbidden zone contains all the points that cannot serve as the source of projection. These are simple 3-dimensional (infinite) prisms with (low) constant complexity, and the goal is to determine whether their union covers the whole 3-space.

For this purpose we use a simple plane-sweep algorithm. We have two types of events: 1. $\Theta(n^2)$ vertices of the prisms; and 2. $O(n^6)$ intersection vertices of the prisms. Standard sweeping technique handles each event in $O(\log n)$ time, so the covering question can be answered in $O(n^6 \log n)$ time. Any point that does not belong to the union of the prisms can serve as the source of a perspective projection (if it is outside the convex hull of the polygon) or as the source of a spherical projection (if it is inside the convex hull of the polygon).

5 Conclusion

In this paper we show that the triangulability of a 3-dimensional polygon is an $\mathcal{NP}$-Complete problem. We also establish some necessary conditions and some sufficient conditions for a polygon to be triangulable, and provide algorithms for testing the sufficient conditions.

Many interesting problems related to the triangulability of 3-dimensional polygons remain open, including:

1. What is the complexity of finding the triangulation that optimizes some objective function?

2. Is the decision problem of the triangulability of a biplanar polygon easier than the general problem?

3. Specify more necessary triangulability conditions, other than unknottedness.

4. Specify more sufficient triangulability conditions, other than the projection conditions described above.

5. Can one do better than $O(n^4)$ for finding whether there exists a simple orthogonal projection of a 3-dimensional polygon? (Either by considering the relations between the $O(n^2)$ quadrilaterals on the sphere of directions, which we have totally ignored, or by using another technique.)

Acknowledgment

We wish to thank Joseph O'Rourke and Carole Gitlin for their triangle-triangle intersection program (in 3-space), which verified our visual inspection (using Mathematica) of the non-triangulability of the unknotted 3-dimensional hexagon.

References

[1] C.C. Adams, *The Knot Book: An Elementary Introduction to the Mathematical Theory of Knots*, W.H. Freeman, 1994.

[2] G. Barequet and M. Sharir, Filling gaps in the boundary of a polyhedron, *Computer-Aided Geometric Design*, 12 (1995), 207–229.

[3] G. Barequet and M. Sharir, Piecewise-linear interpolation between polygonal slices, *Computer Vision and Image Understanding*, 63 (1996), 251–272.

[4] M. Bern and D. Eppstein, Mesh generation and optimal triangulation, in: *Computing in Euclidean Geometry* (F.K. Hwang and D.-Z. Du, eds.), World Scientific, 1992.

[5] J.H Bøhn and M.J. Wozny, Automatic CAD-model repair: Shell-closure, *Proc. Symp. on Solid Freeform Fabrication*, Dept. of Mech. Eng., Univ. of Texas at Austin, 1992, 86–94.

[6] P. Bose, F. Gomez, P. Ramos, and G. Toussaint, Drawing nice projections of objects in space, *Proc. Graph Drawing '95, Lecture Notes in Computer Science*, 1027, Springer Verlag, Berlin, 1995, 52–63.

[7] B. Chazelle, Triangulating a simple polygon in linear time, *Discrete & Computational Geometry*, 6 (1991), 485–524.

[8] H. Edelsbrunner, Triangulations, Lecture Notes, CS497, UIUC, Spring 1991.

[9] H. Edelsbrunner and L. Guibas, Topologically sweeping an arrangement, *J. of Computer and Systems Science*, 38 (1989), 165–194.

[10] M. Fellows, Problem corner: 6. The complexity of knot triviality and the Arf invariant, in: *Graphs and Algorithms: Proc. AMS-IMS-SIAM Joint Summer Research Conference, Contemporary Mathematics*, 89, American Mathematical Society, 1989, 189–190.

[11] A. Gajentaan and M.H. Overmars, On a class of $O(n^2)$ problems in computational geometry, *Computational Geometry: Theory and Applications*, 5 (1995), 165–185.

[12] N.J. Lennes, Theorems on the simple finite polygon and polyhedron, *American J. of Mathematics*, 33 (1911), 37–62.

[13] C. Livingstone, *Knot Theory*, The Carus Mathematical Monographs, 24, Mathematical Association of America, 1993.

[14] I. Mäkelä and A. Dolenc, Some efficient procedures for correcting triangulated models, *Proc. Symp. on Solid Freeform Fabrication*, Dept. of Mech. Eng., Univ. of Texas at Austin, 1993.

[15] J. Ruppert and R. Seidel, On the difficulty of triangulating three-dimensional non-convex polyhedra, *Discrete & Computational Geometry*, 7 (1992), 227–253.

[16] H. Schubert, Die eindeutige zerlegbarkeit eines knotens in primknoten (German), Sitzungsber Heidelberg, Akad. Wiss., Math.-Naturw. Kl., 3 (1949), 57–104.

[17] J. Snoeyink, A trivial knot whose spanning disks have exponential size, *Proc. 6th ACM Symp. on Computational Geometry*, 1990, 139–147.

[18] J. Stolfi, *Oriented Projective Geometry: A Framework for Geometric Computations*, Academic Press, 1991.