ELSEVIER

Computational Geometry 6 (1996) 263–276

Computational Geometry
Theory and Applications

Optimal tetrahedralization of the 3D-region "between" a convex polyhedron and a convex polygon ☆

Leonidas Palios [1]

The Geometry Center, University of Minnesota, Minneapolis, MN 55454, USA

Communicated by Mark Keil; submitted 13 April 1995; accepted 18 May 1995

Abstract

Given a convex polyhedron P and a convex polygon Q in $\mathcal{R}^3$ such that Q's supporting plane does not intersect P, we are interested in tetrahedralizing the closure of the difference $convex\_hull(P \cup Q) \setminus P$; since P is convex, this difference is a connected nonconvex subset of $\mathcal{R}^3$ which we call the region "between" P and Q. The problem is motivated by the work of Bern on tetrahedralizing the region between convex polyhedra (Bern, 1993). In this paper, we describe a novel approach that yields an optimal tetrahedralization, that is, $O(n)$ tetrahedra and no Steiner points; the tetrahedralization is compatible with the boundary of the polyhedron P, and can be computed in optimal $O(n)$ time. Our result also implies a simple and optimal algorithm for the *side-by-side case* (Bern, 1993) when Steiner points are allowed: the region "between" two non-intersecting convex polyhedra of total size n can be partitioned into $O(n)$ tetrahedra using $O(n)$ Steiner points; as above, the tetrahedralization is compatible with the boundaries of the two polyhedra, and can be computed in $O(n)$ time. Note that if Steiner points are not allowed, instances of side-by-side convex polyhedra lead to tetrahedralizations quadratic in their sizes.

1. Introduction

A tetrahedralization of a three-dimensional polyhedron is its partitioning into tetrahedra; it is the three-dimensional equivalent of a triangulation of a polygon. A tetrahedralization is characterized as either Steiner or non-Steiner depending on whether vertices other than the vertices of the polyhedron (called *Steiner points*) are allowed in the tetrahedralization. As tetrahedralizations are used in mesh generation, finite element methods, studying the topology of three-manifolds, CAD/CAM applications, etc., the subject has received considerable attention. It turns out that tetrahedralization problems are

☆ This work has been supported by grants from the National Science Foundation (NSF/DMS-8920161), the Department of Energy (DOE/DE-FG02-92ER25137), Minnesota Technology, Inc., and the University of Minnesota.

[1] Current address: 3 Amazonon Street, Palaio Faliro, 17563, Greece.

SSDI 0925-7721(95)00011-9

more intricate than their two-dimensional counterparts. Although convex polyhedra are easy to tetrahedralize, nonconvex ones are not always tetrahedralizable without Steiner points; a typical example is the "twisted" triangular prism due to Schönhardt [14]. In fact, the problem of deciding whether a nonconvex polyhedron can be partitioned into tetrahedra without Steiner points is NP-complete, as shown by Ruppert and Seidel [13]. If Steiner points are allowed, then any n-vertex polyhedron can be partitioned into $O(n^2)$ tetrahedra thanks to an algorithm by Chazelle [5]. Chazelle also proved that $\Omega(n^2)$ pieces are required in the worst case [5]. This lower bound becomes $\Omega(n + r^2)$ if the number r of reflex edges of the polyhedron is taken into account; in this case, a worst-case optimal algorithm (i.e., producing an $O(n + r^2)$-size tetrahedralization) has been described by Chazelle and Palios [6]. Finally, Bajaj and Dey gave improved bounds for the analysis of Chazelle's algorithm and addressed robustness issues [1].

Recently, the focus shifted to tetrahedralizations between polyhedra, starting with the simplest case involving two disjoint convex polyhedra, say P_1 and P_2. Two cases were considered: the *nested* case, where P_2 is entirely contained in P_1 and the goal is to tetrahedralize the difference $P_1 \setminus P_2$, and the *side-by-side* case, where P_1 and P_2 are disjoint and the goal is to tetrahedralize the closure of the difference $convex\_hull(P_1 \cup P_2) \setminus (P_1 \cup P_2)$. In [9], Goodman and Pach showed how both these problems can be solved in arbitrary dimension without introducing Steiner points. In $\mathcal{R}^3$, their algorithm involves projecting the polyhedra onto two appropriate hyperplanes in $\mathcal{R}^4$ and computing the four-dimensional convex hull; the projection of the convex hull back onto $\mathcal{R}^3$ gives an $O(n^2)$-size tetrahedralization (where n is the combined size of the polyhedra). Bern proved that this was worst-case optimal for the side-by-side case by providing a matching lower bound [3]. Moreover, for the nested case, he described an algorithm that produces $O(n \log n)$ tetrahedra: the method involves shrinking the largest of the two polyhedra by plucking off *cups* of its vertices (see [6]) until it coincides with the smallest polyhedron. We remind that these results apply to non-Steiner tetrahedralizations. If Steiner points are allowed, however, both cases can be resolved yielding $O(n)$-size tetrahedralizations; see the algorithm of Chazelle and Shouraboura to tetrahedralize the region between two convex polyhedra, which can be nested, side-by-side, or overlapping [7]. (We should note that for two side-by-side polyhedra P_1 and P_2, their algorithm produces a tetrahedralization of the difference $\mathcal{R}^3 \setminus (P_1 \cup P_2)$. In order to get a solution for the side-by-side case as we defined it above, the resulting tetrahedra need to be clipped with respect to the boundary of the convex hull of $P_1 \cup P_2$ and be refined; in the end, their number is still linear in the combined size of P_1 and P_2.) Chazelle and Shouraboura also proved a $\Theta(n \log n)$ bound in the number of tetrahedra that are needed to partition the region between a convex polyhedron and a disjoint polyhedral terrain.

In this paper, we consider the following problem in $\mathcal{R}^3$: given a convex polyhedron P and a convex polygon Q whose supporting plane does not intersect P, we are interested in tetrahedralizing the *region "between" P and Q*, which in this case is the closure of the difference $convex\_hull(P \cup Q) \setminus P$. The problem is motivated by the work of Bern [3], and in particular an idea of Halperin that yields a subquadratic tetrahedralization for the side-by-side case at the expense of Steiner points. It involves slicing the convex hull of the two side-by-side polyhedra P_1 and P_2 with two parallel planes that do not intersect the polyhedra; this results into partitioning the region between P_1 and P_2 into a cylindrical piece in the center and two end-pieces, each defined by one polyhedron and the intersection of the convex hull and a slicing plane. Since such an intersection is a convex polygon whose supporting plane (the slicing plane) does not intersect the polyhedron, the corresponding subproblem gives rise to precisely the problem that we consider.

The problem can be solved using Bern's algorithm for the nested case; if the sizes of the polyhedron and polygon add up to n, the resulting tetrahedralization has $O(n \log n)$ size. In this paper, we improve on that, achieving an optimal tetrahedralization. We describe a novel approach which yields a partition into no more than $11n - 24$ tetrahedra without introducing Steiner points. The algorithm runs in optimal $O(n)$ time. Another important feature is that the tetrahedralization is compatible with any given triangulation of the boundary of the polyhedron. Finally, the combination of our result with Halperin's idea yields a simple and optimal algorithm for the side-by-side case, if a linear (in the size of the polyhedra) number of Steiner points is allowed; namely, if the overall size of the polyhedra is n, it yields an $O(n)$ size tetrahedralization with $O(n)$ Steiner points in $O(n)$ time.

The paper is structured as follows. Section 2 reviews the basic definitions, and presents two useful lemmas. The key ideas for our algorithm are discussed in Section 3, and the algorithm is described in detail in Section 4. Section 5 summarizes our results and poses some open questions.

2. Definitions and useful lemmas

A *polyhedron* in $\mathcal{R}^3$ is a connected piecewise-linear 3-manifold with connected boundary; the boundary consists of a collection of relatively open sets, the *faces* of the polyhedron, called *vertices*, *edges* and *facets*, if their affine closures have dimension 0, 1 or 2, respectively. A *polygon* is a two-dimensional connected set bounded by a number of closed polygonal lines that do not self-intersect or intersect each other; in turn, its boundary consists of *vertices* and *edges*, relatively open sets whose dimension is 0 and 1, respectively. A polyhedron or polygon is called *convex* if it is a convex set, that is, the line segment connecting two points of the set belongs entirely to the set.

Let us consider a convex polyhedron P and a convex polygon Q (also in $\mathcal{R}^3$) whose supporting plane does not intersect P. Then, the convex hull $\mathrm{ch}(P \cup Q)$ of $P \cup Q$ *properly* contains P, and Q contributes one of $\mathrm{ch}(P \cup Q)$'s facets which is diametrically opposite any facets of $\mathrm{ch}(P \cup Q)$ contributed by P [1]; this is why we refer to the closure of the difference $\mathrm{ch}(P \cup Q) \setminus P$ as the *region "between" P and Q*. Since P is convex, the region "between" P and Q is connected and in general concave. Except for the facets of $\mathrm{ch}(P \cup Q)$ contributed by either P or Q, the remaining facets are incident to vertices of both P and Q, and are called *bridges*, since they "bridge" the polygon and the polyhedron. It is important to observe that the bridges lie on planes *tangent* to both P and Q; recall that the facets of the convex hull of a point set lie on planes tangent to the point set. (For a detailed discussion on convex hulls, see [8,12].) The bridges abut on the boundary of P along a closed connected polygonal line of edges of P, the *horizon*. In the simplest case, the horizon is a simple closed path, but it may collapse into a single vertex of P or a chain of edges of P traversed in both directions; in general, it is a combination of the above cases. The horizon partitions the boundary of P into two polyhedral patches: one of them contains all the *internal* facets of P, that is, the facets that lie in the interior of the convex hull $\mathrm{ch}(P \cup Q)$, the other contains the remaining facets, which are called *external*. Whether a facet f of P is internal or external depends only on the relative position of f and the polygon Q. In particular, if both P and Q lie entirely in the same closed halfspace defined by the plane E_f supporting f, then E_f is obviously tangent to the set $P \cup Q$ and f is an external facet. Otherwise, it is internal; in this case, there is at least one vertex q of Q, such that q and the interior

[1] For simplicity, we assume that the boundaries of P and $\mathrm{ch}(P \cup Q)$ are compatible wherever they coincide.

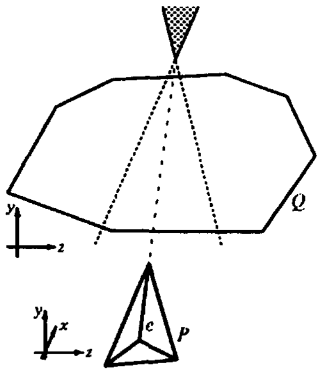

Fig. 1.

of P lie on opposite sides of E_f. Similarly to the internal facets, an edge of P is called *internal* if it lies in the interior of the convex hull $\mathrm{ch}(P \cup Q)$. If an edge lies on the horizon, we call it a *horizon* edge.

Next, we present two lemmas crucial for the discussion to follow. To formalize our description, we define the notions of the in-wedge and out-wedge of an edge e of P: the planes that support the two facets of P incident upon e define four *open* three-dimensional wedges around e; since P is convex, its interior lies entirely in one of them, which we call the *in-wedge* of e. The wedge opposite the in-wedge of e is the *out-wedge* of e. Then, we have Lemma 2.1.

Lemma 2.1. *Let E_Q denote the plane supporting the polygon Q and let e be a horizon edge of P that is* not *parallel to E_Q. Then, the intersection of E_Q and the out-wedge of e is an open two-dimensional wedge that does not intersect Q.*

Proof. (See Fig. 1; the intersection of E_Q and the out-wedge of e is shown shaded.) Since e is a horizon edge of P, it is an edge of the convex hull $\mathrm{ch}(P \cup Q)$, and therefore there exists a plane Π through e that is tangent to $\mathrm{ch}(P \cup Q)$. The plane defines two *open* halfspaces—let them be Π^- and Π^+—and let us assume without loss of generality that the convex hull lies in the closure of Π^-. Although Π contains the line supporting e, it does not intersect the in-wedge of e, otherwise it would intersect P and hence $\mathrm{ch}(P \cup Q)$; then, clearly the in-wedge of e lies entirely in Π^-, since it contains the interior of P. Moreover, since the in-wedge and out-wedge of e are opposite, not only is the out-wedge of e not intersected by Π either, but in fact it lies entirely in the open halfspace Π^+. The lemma follows from the fact that the polygon Q lies entirely in the closure of Π^-, and the intersection of Π^+ and the closure of Π^- is empty. □

The corresponding lemma for internal edges is as follows.

Lemma 2.2. *Let E_Q denote the plane supporting the polygon Q and let e be an internal edge of P that is* not *parallel to E_Q. Then, the intersection of E_Q and the out-wedge of e is an open two-dimensional wedge that intersects Q.*

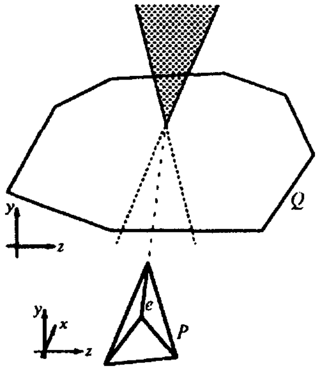

Fig. 2.

Proof. Let W^+ denote the intersection of E_Q and the out-wedge of an internal edge e (in Fig. 2, the intersection W^+ of E_Q and the out-wedge of e is shown shaded). Let us suppose for contradiction that W^+ does not intersect the polygon Q. Then, since both Q and the out-wedge are convex, there exists a line l through the point of intersection of E_Q and the line supporting e, which separates Q from W^+; in other words, W^+ and the polygon Q lie entirely on opposite sides of the line l. Consider the plane Π defined by l and e; Π is well defined, since l and the line supporting e intersect. It is easy to see that Π is tangent to the convex hull $\mathrm{ch}(P \cup Q)$: first, Π contains e; second, because it contains l and e, Π does not intersect the out-wedge of e, and hence it does not intersect the in-wedge of P, which in turn implies that it does not intersect the interior of the polyhedron P. Additionally, the definition of l and the fact that the in-wedge and out-wedge of e are opposite imply that both P and Q lie entirely in the same closed halfspace with respect to Π. Therefore, e belongs to the boundary of the convex hull $\mathrm{ch}(P \cup Q)$, which leads to contradiction since e is an internal edge. □

The definition of the horizon implies that a horizon edge is incident to at least one internal facet. In fact, both facets incident upon a horizon edge may be internal (in Fig. 1, for example, both facets incident upon the edge e are internal). On the other hand, Lemma 2.2 implies that both facets incident upon an internal edge are internal.

Finally, we close this section with some definitions and observations pertaining to tetrahedralizations. A *tetrahedralization* of a closed piecewise-linear subset S of $\mathcal{R}^3$ is a partition of S into tetrahedra, i.e., no two tetrahedra in the partition intersect except at their boundaries, and the union of all the tetrahedra is precisely S. If the intersection of any two tetrahedra is either empty or a face of both tetrahedra, then the tetrahedralization is called a *cell complex*. In some cases, points of S other than its vertices are allowed to become vertices of the pieces in a tetrahedralization of S; such points are called *Steiner points*. Disallowing Steiner points in a tetrahedralization of the region "between" a convex polyhedron P and a convex polygon Q as described above implies that the reported tetrahedra belong to one of the following three classes:

(i) *f–v tetrahedra*: defined by a (triangular) facet of P and a vertex of Q,
(ii) *e–e tetrahedra*: defined by an edge of P and an edge of Q, and
(iii) *v–f tetrahedra*: defined by a vertex of P and a triangle in Q.

It is easy to see that the number of f–v tetrahedra is linear in the size of the polyhedron P; since no two tetrahedra in the tetrahedralization may share interior points, a specific triangular facet of P defines *exactly one* f–v tetrahedron. Similarly, the number of v–f tetrahedra is linear in the size of the polygon Q. Therefore, we get the following observation.

Observation 2.1. Consider a tetrahedralization of the region "between" a convex polyhedron P and a convex polygon Q. If the number of e–e tetrahedra is linear in the combined size n of P and Q, the total size of the tetrahedralization is also linear in n.

3. Shrinking convex hulls and rolling lines

In this section, we introduce the key notion of rolling lines. We start by giving some motivation for their definition and how they relate to the resulting tetrahedralization. To simplify the discussion, we assume that (i) no edge of the polygon Q is parallel to (the plane supporting) any facet of the polyhedron P, and (ii) no edge of P is parallel to the plane supporting Q. These assumptions imply that there exists a unique vertex of P that is closest to (similarly, farthest from) the polygon Q, and that the bridges of the convex hull of $P \cup Q$ are triangles. Finally, we also assume that no two facets of P are coplanar and no two edges of Q are collinear. (It should be noted that none of the above assumptions is needed for our algorithm described in Section 4; ordering by distance from the polygon Q and lexicographic ordering in case of ties are sufficient to relax the assumptions.)

Let us consider a plane Π parallel to the plane supporting the polygon Q, which sweeps the polyhedron P moving in a continuous fashion from the vertex of P farthest away from Q to the vertex of P closest to Q. At any given instant during this sweeping process, we compute the convex hull $\mathrm{ch}(P_\Pi \cup Q)$ of Q and the portion P_Π of P that has not yet been swept by Π. It should be obvious that, for as long as Π intersects external facets of P only, the bridges of $\mathrm{ch}(P_\Pi \cup Q)$ are identical to the bridges of $\mathrm{ch}(P \cup Q)$. However, as soon as Π intersects an internal facet of P and on, the convex hull $\mathrm{ch}(P_\Pi \cup Q)$ starts shrinking and its bridges form a funnel that keeps getting narrower at the end where it touches P; when Π reaches the vertex u of P that is closest to Q, the convex hull $\mathrm{ch}(P_\Pi \cup Q)$ is reduced to a pyramid $C_{u,Q}$ with apex u and base the polygon Q. The continuity of the sweeping implies that, during the convex hull shrinking process the bridges of the convex hull $\mathrm{ch}(P_\Pi \cup Q)$ sweep the closure of $((\mathrm{ch}(P \cup Q) \setminus P) \setminus C_{u,Q})$. In other words, we have Lemma 3.1.

Lemma 3.1. *A tetrahedralization of the region "between" a polyhedron P and a polygon Q can be obtained by tetrahedralizing*
(i) *the pyramid $C_{u,Q}$ defined by Q and the vertex u of P closest to Q, and*
(ii) *the portion of $\mathcal{R}^3$ swept by the bridges of the convex hull* $\mathrm{ch}(P_\Pi \cup Q)$ *during the convex hull shrinking process.*

The former tetrahedralization is easy to obtain: we only need to triangulate Q, and then report tetrahedra defined by u and the triangles in Q. It turns out that the latter tetrahedralization is not particularly hard to obtain either, if we study how the bridges of $\mathrm{ch}(P_\Pi \cup Q)$ change as Π sweeps P. Thanks to the assumptions in the beginning of this section, each bridge is a triangle of one of the following two types:

Type 1: the bridge is defined by a vertex of Q and a line segment on the boundary of P. The line segment is the intersection of Π with an internal facet f of P. In fact, as Π sweeps the facet f, the line segments at which it intersects f define bridges of the shrinking convex hull $\mathrm{ch}(P_\Pi \cup Q)$ with the *same* vertex q of Q; if the interior of P lies below (above, respectively) f, q is the vertex of Q lying above (below, respectively) f that is farthest away from the plane supporting f. This is a simple consequence of the fact that facets of the convex hull of a set of points lie on planes that are tangent to that point set. Then, if all these bridges are stacked one on top of the other, they form a pyramid with apex q and base f.

Type 2: the bridge is defined by an edge of Q and a point on the boundary of P. The point is either a vertex of P or the intersection of Π with an edge e of P; similarly to the previous case, as Π sweeps the edge e, the corresponding point of intersection defines a bridge with the *same* edge e' of Q. (It is important to observe at this point that the definition of the convex hull implies that a line parallel to e' is tangent to P when placed at any point of e.) Again, by stacking these bridges one on top of the other, we get a tetrahedron defined by the edges e and e'.

Therefore, a tetrahedralization of $((\mathrm{ch}(P \cup Q) \setminus P) \setminus C_{u,Q})$ can be obtained by partitioning the pyramids defined by each internal facet of P and its corresponding vertex of Q into tetrahedra, and adding the tetrahedra defined by the corresponding pairs of edges of P and Q.

The above tetrahedralization can be described in a simple way if we view the convex hull shrinking process from a different perspective; this is where the rolling lines come to play. Consider an edge e' of the polygon Q; the boundary of the convex hull $\mathrm{ch}(P \cup Q)$ contains a bridge incident upon e', which according to the assumptions in the beginning of this section is a triangle defined by e' and a vertex, say, w, of P. Then, the *rolling line* associated with e' is a line parallel to e' initially positioned at w; its purpose is to roll on the polyhedral patch that is defined by the closure of all the internal facets of P, so that it always remains parallel to e', is always tangent to P, keeps moving closer to Q, and stops when it reaches the vertex of P that is closest to Q. (Note that the convex hull definition implies that at its initial positioning at w this line is tangent to P.) We call this line the *rolling line* associated with the edge e' of Q. By means of an argument that involves sweeping the polyhedron with a plane parallel to the polygon, it is easy to see that the line rolls along a continuous simple path of internal or horizon edges of P, which we call the corresponding *rolling path*.

Let us now investigate the relationship between the rolling lines and the convex hull shrinking process described earlier. First, the fact that the rolling lines are always parallel to the corresponding polygon edges and tangent to P implies that for any point p in the path of the rolling line corresponding to the edge e' of Q, p and e' define a bridge of the convex hull $\mathrm{ch}(P_\Pi \cup Q)$ when the sweep plane Π is located at p. Another important observation is that the rolling paths may share edges but they do not cross; note that since the rolling lines stay parallel to their corresponding polygon edges and the polyhedron P is convex, their order around a slice of P with a plane parallel to the polygon Q does not change. As the rolling paths extend from vertices on the horizon to the vertex u of P that is closest to Q, they partition the polyhedral patch of all the internal facets of P into subpatches; each subpatch F is characterized by a pair of consecutive edges, say e_1 and e_2, of Q whose associate rolling paths bound F. In fact, if q denotes the common vertex of e_1 and e_2, the facets in F are precisely those internal facets of P that, when intersected by the sweep plane Π, contribute line segments which along with q define bridges of the shrinking convex hull $\mathrm{ch}(P_\Pi \cup Q)$.

In light of the above observations, Lemma 3.1 implies Theorem 3.1.

Theorem 3.1. *The 3D-region "between" a convex polyhedron P and a convex polygon Q in $\mathcal{R}^3$ (whose supporting plane does not intersect P) can be partitioned into tetrahedra as follows:*
1. *We report v–f tetrahedra defined by the vertex of P closest to Q and the triangles in a triangulation of Q;*
2. *For each edge e' of Q, we report e–e tetrahedra defined by e' and each edge of P on the path of the rolling line associated with e';*
3. *For each pair of consecutive edges e_1 and e_2 of Q, we report f–v tetrahedra defined by the vertex of Q incident to both e_1 and e_2 and (a triangulation of) each of the internal facets that lie between the paths of the rolling lines associated with e_1 and e_2.*

The tetrahedralization does not introduce Steiner points and is compatible with the boundary of P and an arbitrary triangulation of Q.

3.1. Merging rolling lines

Unfortunately, the tetrahedralization scheme of Theorem 3.1 is not sufficient to guarantee a number of tetrahedra linear in the total size n of the polyhedron P and the polygon Q; indeed, it is conceivable that $\Theta(n)$ rolling lines roll along a chain of $\Theta(n)$ polyhedron edges, which will result in $\Theta(n^2)$ e–e tetrahedra. If, however, we make sure that no more than a constant number of rolling lines roll along the same edge of the polyhedron P, then in light of Observation 2.1 we will be able to produce an $O(n)$ tetrahedralization. This can be achieved thanks to the idea of "merging" rolling lines.

Consider an edge e of P that is not parallel to the plane E_Q supporting the polygon Q, and let v and w be e's incident vertices, where w is closer to E_Q than v. Additionally, let P_v denote the part of P that lies between the plane E_Q and a plane parallel to E_Q located at v. We distinguish the following three cases that cover all possibilities:

1. *The edge e is an internal edge of P.* In this case, the rolling lines that are about to roll along the edge e correspond to a single chain C of consecutive polygon edges, which is delimited by vertices q_1 and q_2 at which planes parallel to e's incident facets are tangent to the polygon Q; in terms of the shrinking convex hull approach, the above statement is equivalent to saying that the triangles defined by v and the edges in C are bridges of the convex hull $\mathrm{ch}(P_v \cup Q)$ when the sweep plane is located at v. Fig. 3 depicts the situation; the out-wedge of e is not shown, whereas the intersection of E_Q and the in-wedge of e is shown shaded (compare with Fig. 2). Since the

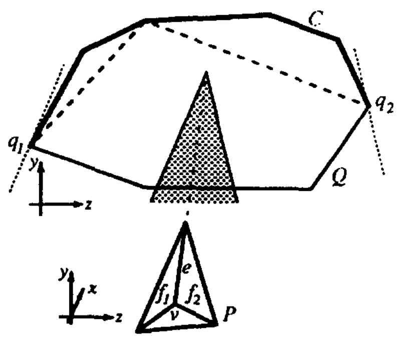

Fig. 3.

closure of the in-wedge of an edge of P entirely contains P, the line segment connecting v with any point in the nonshaded portion of Q does not intersect the polyhedron P. More generally, a tetrahedron defined by v and any triangle in the nonshaded portion of Q does not intersect P; moreover, it lies in $\mathrm{ch}(P_v \cup Q)$ since it lies below the bridges defined by v and the edges in C. These observations suggest that we can "merge" the rolling lines associated with e by (i) finding diagonals in the nonshaded part of Q that will clip parts of Q and "shortcut" subchains of C, (ii) replacing the corresponding rolling lines by rolling lines associated with these diagonals, and (iii) reporting tetrahedra defined by v and the clipped portions of Q. It is not difficult to see that the rolling lines associated with an internal edge e can be "merged" into *at most three* rolling lines that will roll along e; in Fig. 3, we need only two diagonals, and therefore two lines will roll along e.

2. *The edge e is a horizon edge incident to only one internal facet f of P* (Fig. 4). In this case, the rolling lines that are about to roll along the edge e correspond to a single chain C of consecutive polygon edges; in terms of the shrinking convex hull approach, the above statement is equivalent to saying that the triangles defined by v and the edges in C are bridges of the convex hull $\mathrm{ch}(P_v \cup Q)$ when the sweep plane is located at v. It turns out that the entire chain C and the polyhedron P lie on opposite sides of the plane supporting f. Following the reasoning in Case 1, we can then clip the polygon about the diagonal d that separates C from the rest of the polygon, and thus "merge"

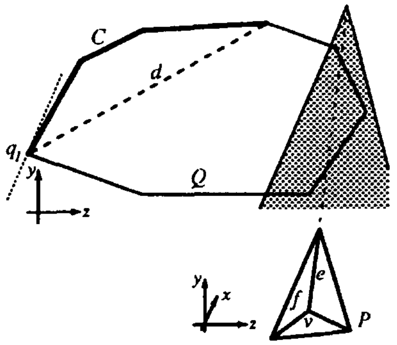

Fig. 4.

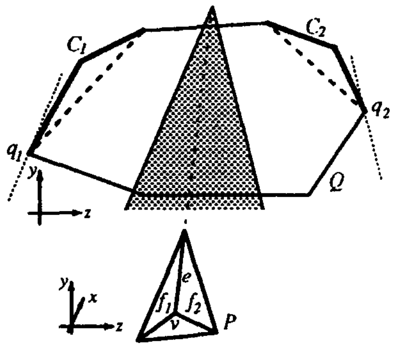

Fig. 5.

all the rolling lines associated with e into a *single* rolling line that corresponds to d. Of course, we also report tetrahedra defined by v and a triangulation of the clipped portion of Q.

3. *The edge e is a horizon edge and both incident facets are internal facets of P* (Fig. 5). This case could be viewed as two copies of Case 2 glued together at e; then, the associated rolling lines correspond to two chains C_1 and C_2 of consecutive polygon edges. In a fashion similar to Case 2, the rolling lines can be "merged" into *two* rolling lines that correspond to the diagonals that clip each of the chains C_1 and C_2 from the rest of the polygon. Again, the merging involves reporting tetrahedra defined by v and triangulations of the clipped parts of the polygon Q.

It is important to observe that, in each of the above three cases, if the polygon Q is clipped into a polygon Q', the tetrahedra reported partition the difference $convex\_hull(P_v \cup Q) \setminus convex\_hull(P_v \cup Q')$. The contribution of the merging process is summarized in the following lemma.

Lemma 3.2. *Thanks to the merging process, the number of rolling lines that end up rolling along an edge of the polyhedron P does not exceed three. In the process, portions of the (convex) polygon Q are clipped about diagonals of Q (thus, the resulting polygon remains convex), and a number of tetrahedra linear in the size of the clipped portions is reported.*

4. The algorithm

We assume, without loss of generality, that the polygon Q lies on the yz-plane, and that the polyhedron P is in the negative x-halfspace. Hence, the lexicographical order of the vertices of P (i.e., sorting them by increasing x-, y-, and then z-coordinate) orders them by decreasing distance from the plane supporting the polygon Q and resolves ties.

The algorithm is based on the sweep-line paradigm: a vertex is processed only when all the rolling lines that roll through it have reached it, and no rolling line rolls past a vertex that has not been processed yet. This necessitates an ordering of the polyhedron vertices; fortunately, it suffices to use the topological ordering of the vertices in the directed acyclic graph G induced by the polyhedron's internal and horizon edges oriented towards their lexicographically largest incident vertex. The ordering does not have to be precomputed; the vertices are processed in order, thanks to a list L that stores the vertices of P whose predecessors in G have all been processed.

The algorithm starts with Steps 1 to 3 that accomplish all the necessary preprocessing. Its main body (Steps 4 to 6) is an iterative procedure processing the vertices of the polyhedron in order. Finally, Step 7 completes the tetrahedralization. In particular, we have the following steps.

Step 1. We input the description of the polygon Q and store it as a doubly connected linked list of edges, so that edges can be inserted or deleted in constant time. We then input the description of the polyhedron P, and we store it using one of the standard representations (see [2,10,11]), so that all the faces incident upon a given face can be located in time linear in their number. Additionally, we orient each edge of P from its lexicographically smallest vertex to the lexicographically largest one; in this way the edge points towards the polygon. The orientation of the edges is used to guide the rolling lines that roll along them.

Step 2. We compute the bridges of the convex hull of $P \cup Q$; this can be done by using the linear-time merging procedure of the divide-and-conquer algorithm to compute the convex hull of a point set in $\mathcal{R}^3$ [8]. The edges of P incident upon the bridges form the horizon. The internal facets of P

can be found easily as well: if a horizon edge e is incident upon two bridges, then both facets of P incident upon e are internal; if e is incident upon a single bridge then the relative position of the bridge and the two facets incident upon e indicates in constant time which facet is internal. In this way, we determine all the internal facets adjacent to the horizon; the remaining internal facets can be found by moving from an already known internal facet to its neighbors without ever crossing the horizon. Last, we determine the starting points for all the rolling lines: for an edge e' of Q, we walk around the bridge incident upon e' (which is not necessarily a triangle) and collect all the vertices of P that we visit; the lexicographically largest vertex among them is the starting point for the rolling line that corresponds to e'.

Step 3. For each vertex, say w, of P incident upon an internal facet, we store at a field *indegree* the number of incident internal or horizon edges of P oriented towards w; if this number is 0, then w is ready to be processed, and we insert it in the list L. (Note that L will contain at least one vertex, the vertex on the horizon that is farthest away from Q.)

Step 4. We remove a vertex, say v, from the list L. If v is the vertex of P that is closest to the polygon Q, then the rolling procedure is complete, and we continue at Step 7. Otherwise, we visit the edges of P emanating from v in order around v, and we match each internal or horizon edge e with those among the rolling lines located at v, if any, that are tangent to P at e. It is important to note that the edges in order around v get matched with rolling lines in the order that the corresponding polygon edges appear around the polygon; therefore this operation involves "chasing" pointers in the lists of polyhedron edges and rolling lines. In case of ties, that is, a rolling line is coplanar with one or more facets and hence it is tangent to more than one edge emanating from v, the rolling line gets matched with the unique edge that lies on the lexicographically largest boundary of the super-facet created when all these facets are merged together. Not only does this rule guarantee consistency, but it also implies that a line will never roll along an edge parallel to itself. Then, we process each edge as described in Step 5.

Step 5. For each internal or horizon edge e out of v, we merge the collected rolling lines as described in Section 3.1[2]: the diagonals along which we clip the polygon Q, and which contribute new rolling lines are easily determined in a traversal of the corresponding polygon edges around Q. Clipping a part R of Q involves reporting v–f tetrahedra defined by the vertex v and a triangulation of R, and replacing the edges of Q that bound R with the diagonal that separates it from the rest of the polygon. The merging process yields at most three rolling lines. Next, these lines roll along e, that is, we report an e–e tetrahedron defined by e and the polygon edge corresponding to each of these rolling lines, and the lines are moved from v to the other polyhedron vertex incident upon e. After that, we associate the internal facets incident upon e with the polygon vertices with which they will define f–v tetrahedra. Referring to the three cases in Section 3.1, we have: In Case 1 (Fig. 3), we associate the facet f_1 to the left of e with the leftmost vertex q_1 of C, and the facet f_2 to the right of e with the rightmost vertex q_2 of C. In Case 2, the internal facet f incident upon e is associated with the leftmost vertex q_1 of C if it is to the left of e (Fig. 4), or the rightmost vertex of C otherwise. In Case 3 (Fig. 5), we associate the facet f_1 to the left of e with the leftmost vertex q_1 of C_1, and the facet f_2 to the right of e with the rightmost vertex q_2 of C_2. In case no lines roll along e, then if e is on the horizon, the

[2] If the edge e is parallel to the polygon Q, we perturb it by pulling its lexicographically largest vertex infinitesimally closer to Q. Note that this perturbation is compatible with the orientation of e, and it enables us to treat the special case in the general framework. Of course, whenever the edge is used to define tetrahedra, the original unperturbed edge is used.

vertex associated with its incident internal facets is determined by the bridges; otherwise, the facet to the right of e gets associated with the same polygon vertex as the facet to the left (note that both facets are internal and that the facet to the left has been updated at a previous step, if the edges out of v are processed from left to right). Finally, if e points from v to w, we decrease the indegree of w by 1; if it becomes equal to 0, we insert w in L.

Step 6. When all the edges emanating from vertex v have been processed, we return to Step 4.

Step 7. Upon reaching this point, the rolling lines have fulfilled their mission, and they are discarded. To complete the tetrahedralization, we need (i) to tetrahedralize the pyramid that is defined by the vertex u of P that is closest to the polygon and the convex polygon Q' which is left from the original polygon as a result of the polygon clipping when rolling lines merge, and (ii) to report the tetrahedra defined by triangles on the boundary of the polyhedron and vertices of the polygon. The former task is done by arbitrarily triangulating Q' and reporting v–f tetrahedra defined by each such triangle and u; if Q' is just a single edge traversed twice, no tetrahedra are reported. Regarding the second task, we triangulate each internal facet f of P in turn, and report f–v tetrahedra defined by the resulting triangles and the polygon vertex associated with f.

The correctness of the algorithm follows from the discussion in Section 3. Additionally, it is not difficult to see that the algorithm runs in time linear in the total number of faces of the polyhedron and the polygon. Three points need to be mentioned: First, that the rolling lines can be sorted by slope in time linear in their number by radix sorting the indices of the corresponding polygons edges (see [15]). Second, note that an edge of the polygon is gone forever after it has been clipped off, and that at most three lines roll along an edge of the polyhedron; thus the merging and the rolling process takes time linear in the size of the clipped portions of the polygon and the degree of the corresponding polygon vertex. Finally, both the polygon Q and the facets of the polyhedron P are convex polygons, whose triangulation takes time linear in their sizes.

Number of tetrahedra produced

Let n_P and n_Q denote the number of vertices of the polyhedron P and the polygon Q, respectively, and m_i and m_h be the number of internal and horizon edges of P. Euler's formula for convex polyhedra in three dimensions implies that a triangulation of the boundary of P will yield $2n_P - 4$ triangles; hence, the total number of f–v tetrahedra is at most $2n_P - 4$. In turn, any triangulation of Q produces $n_Q - 2$ triangles, and thus the number of v–f tetrahedra is precisely $n_Q - 2$. Finally, since at most two lines may roll along a horizon edge and at most three lines may roll along an internal edge, the total number of e–e tetrahedra is no more than $2m_h + 3m_i$.

Summing up the above contributions, we have that the total number of tetrahedra is at most

$$2n_P - 4 + n_Q - 2 + 2m_h + 3m_i \leqslant 2n_P + n_Q + 3(m_i + m_h) - 6. \tag{1}$$

Since an edge of P cannot be both internal and horizon, Euler's formula for the number of edges of a convex polyhedron in terms of the number of its vertices yields

$$m_i + m_h \leqslant 3n_P - 6.$$

Substituting this upper bound into (1), we find that the total number of tetrahedra produced by the above algorithm does not exceed $11n_P + n_Q - 24 \leqslant 11n - 24$, where $n = n_P + n_Q$ is the combined size of P and Q.

5. Conclusions and open problems

Our results are summarized in the following theorem.

Theorem 5.1. *For a convex polyhedron P and a convex polygon Q (whose supporting plane does not intersect the polyhedron) in $\mathcal{R}^3$ of combined size n, we show that one can partition the region "between" P and Q (i.e., the closure of the difference convex_hull$(P \cup Q) \setminus P$) into at most $11n - 24$ tetrahedra without introducing Steiner points. The tetrahedralization can be computed in optimal $\mathrm{O}(n)$ time, and is compatible with any triangulation of the boundary of P.*

It should be noted that the algorithm imposes an appropriate triangulation on the polygon; this should be expected, since tetrahedralizations compatible with *both* the boundary of the polyhedron and a given triangulation of the polygon may be of quadratic size in the worst case (consider Bern's quadratic lower bound construction for the side-by-side case in [3], where one of the polyhedra has been flattened into a convex polygon). If the polygon does not have consecutive edges that are collinear, the resulting tetrahedralization is guaranteed to be a *cell complex*, since our algorithm ensures that no rolling line rolls along a polyhedron edge that is parallel to itself. In case consecutive collinear edges exist, our approach automatically merges the corresponding rolling lines (and the polygon edges) into a single rolling line (edge, respectively), and thus the tetrahedralization is not a cell complex; a cell complex can be obtained however by either refining the produced tetrahedralization (in which case the number of tetrahedra may become as large as $\Theta(n^2)$), or by using ideas similar to those of Eppstein (in order to "protect" edges of polyhedra during tetrahedralizations (see [4])) at the expense of introducing Steiner points. The tetrahedralization can be easily extended to a tetrahedralization of $\mathcal{R}^3$ with the addition of the tetrahedra that partition P and the complement of the convex hull $\mathrm{ch}(P \cup Q)$ of $P \cup Q$; since P and $\mathrm{ch}(P \cup Q)$ are convex, this can be done in $\mathrm{O}(n)$ time yielding $\mathrm{O}(n)$ tetrahedra.

Our result also implies an optimal size tetrahedralization for the side-by-side case if a linear number of Steiner points are allowed in the worst case; namely, the closure of the difference convex_hull$(P_1 \cup P_2) \setminus (P_1 \cup P_2)$ between two non-intersecting convex polyhedra P_1 and P_2 of total size n can be partitioned into $\mathrm{O}(n)$ tetrahedra using $\mathrm{O}(n)$ Steiner points. The tetrahedralization can be computed in $\mathrm{O}(n)$ time. It would be interesting to investigate the question whether an $\mathrm{O}(n)$ size tetrahedralization is possible if $\mathrm{o}(n)$ Steiner points are allowed.

Another important open question is whether the region between the boundaries of two nested convex polyhedra can be partitioned into a linear number of tetrahedra without Steiner points. Can the idea of rolling lines help?

Finally, let us consider two convex polygons Π_1 and Π_2 that lie on parallel planes in $\mathcal{R}^3$; we are interested in tetrahedralizing their convex hull. Although the object to tetrahedralize is convex, Bern's lower bound for the side-by-side case implies that a tetrahedralization that is compatible with arbitrary triangulations of both Π_1 and Π_2 and does not involve Steiner points may be of quadratic size in the worst case. The construction does not work, however, if the two polygons are copies of the same polygon (triangulated differently). In this case, is the size of a compatible tetrahedralization of their convex hull without Steiner points quadratic in the size of the polygons in the worst case?

References

[1] C.L. Bajaj and T.K. Dey, Convex decompositions of polyhedra and robustness, SIAM J. Comput. 21 (1992) 339–364.

[2] B.G. Baumgart, A polyhedron representation for computer vision, Proc. 1975 National Comput. Conf., AFIPS Conf. Proc. 44 (1975) 589–596.

[3] M. Bern, Compatible tetrahedralizations, Proc. 9th ACM Symp. Comput. Geom. (1993) 281–288.

[4] M. Bern and D. Eppstein, Mesh generation and optimal triangulation, Xerox PARC Technical Report CSL-92-1.

[5] B. Chazelle, Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm, SIAM J. Comput. 13 (1984) 488–507.

[6] B. Chazelle and L. Palios, Triangulating a nonconvex polytope, Discrete Comput. Geom. 5 (1990) 505–526.

[7] B. Chazelle and N. Shouraboura, Bounds on the size of tetrahedralizations, Proc. 10th Annual ACM Symp. Comput. Geom. (1994) 231–239.

[8] H. Edelsbrunner, Algorithms in Combinatorial Geometry (Springer, Berlin, 1987).

[9] J.E. Goodman and J. Pach, Cell decomposition of polytopes by bending, Israel J. Math. 64 (1988) 129–138.

[10] L.J. Guibas and J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, ACM Trans. Graphics 4 (1985) 75–123.

[11] D.E. Muller and F.P. Preparata, Finding the intersection of two convex polyhedra, Theor. Comp. Sci. 7 (1978) 217–236.

[12] F.P. Preparata and M.I. Shamos, Computational Geometry, An Introduction (Springer, Berlin, 1985).

[13] J. Ruppert and R. Seidel, On the difficulty of triangulating three-dimensional nonconvex polyhedra, Discrete Comput. Geom. 7 (1992) 227–253.

[14] E. Schönhardt, Über die Zerlegung von Dreieckspolyedern in Tetraeder, Math. Annalen 98 (1928) 309–312.

[15] R. Sedgewick, Algorithms (Addison-Wesley, Reading, MA, 1988).