Optimal Tetrahedralization of the 3d-Region "Between" a Convex Polyhedron and a Convex Polygon †

(Extended Abstract)

Leonidas Palios
The Geometry Center, Univ. of Minnesota

Abstract: In $\mathcal{R}^3$, we consider a convex polyhedron P and a convex polygon Q, whose supporting plane does not intersect P; we are interested in tetrahedralizing the closure of convex_hull$(P \cup Q) \setminus P$, i.e., the portion of $\mathcal{R}^3$ that along with P forms the convex hull of $P \cup Q$. This problem is motivated by the more general problem to tetrahedralize the region "between" two three-dimensional convex polyhedra that do not intersect, called the *side-by-side* case by Bern in [2]. Bern solves the former problem by applying his algorithm to tetrahedralize the region between the boundaries of two *nested* convex polyhedra; if the total number of vertices of both P and Q is n, his approach yields an $O(n \log n)$ size tetrahedralization without introducing *Steiner points*, i.e., new vertices. 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 it can be computed in optimal $O(n)$ time. Our result also implies an improved algorithm for the side-by-side case; 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 it can be computed in $O(n)$ time.

1. Introduction.

Given a convex polyhedron P and a 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$(P \cup Q) \setminus P$). The problem is motivated by the *side-by-side case* (Bern [2]): given two non-intersecting convex polyhedra P_1 and P_2 in $\mathcal{R}^3$, tetrahedralize the closure of the difference (convex_hull$(P_1 \cup P_2) \setminus (P_1 \cup P_2)$). In [6], Goodman and Pach showed how this problem can be solved in arbitrary dimension without introducing Steiner points (i.e., new vertices); in $\mathcal{R}^3$, their algorithm produces a tetrahedralization whose size is quadratic in the combined size of P_1 and P_2; this was proved optimal in the worst case, thanks to a matching lower bound by Bern ([2]). Bern also mentioned an algorithm that yields a subquadratic tetrahedralization at the expense of introducing Steiner points: the idea, due to Halperin, is to slice the convex hull of P_1 and P_2 with two parallel planes that do not intersect the polyhedra; this results into partitioning the hull into a cylindrical piece in the center and two end-pieces, each defined by one polyhedron and a convex polygon (the intersection of the convex hull and a slicing plane). Since the slicing planes do not intersect the polyhedra, the problem of tetrahedralizing each of the end-pieces is precisely the problem that we consider in this paper. To solve it, Bern applies his

† 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 Univ. of Minnesota.

algorithm to tetrahedralize the region between the boundaries of two *nested* convex polyhedra, which produces $O(n \log n)$ tetrahedra (n is the combined size of the two polyhedra) without Steiner points.

We also mention that Chazelle and Shouraboura have recently described an algorithm that partitions the region between two convex polyhedra (that may be disjoint, nested, or overlapping) into a linear number of tetrahedra [4]; their approach, however, introduces Steiner points.

In this paper, we describe a novel approach to tetrahedralize the closure of $(\text{convex\_hull}(P \cup Q) \setminus P)$ for a convex polyhedron P and a convex polygon Q whose supporting plane does not intersect the polyhedron; if the combined size of the polyhedron and the polygon is n, our algorithm produces at most $11n - 24$ tetrahedra and does not introduce Steiner points. Another important feature is that the tetrahedralization is compatible with any triangulation of the boundary of the polyhedron. The algorithm, however, imposes an appropriate triangulation on the polygon; that should be expected, since a simple extension of Bern's quadratic lower bound for the side-by-side case [2] implies that tetrahedralizations compatible with both the boundary of the polyhedron and a triangulation of the polygon may be of quadratic size in the worst case. The tetrahedralization can be computed in $O(n)$ time. Moreover, it can be easily extended to an $O(n)$ tetrahedralization of $\mathcal{R}^3$ without Steiner points by adding the tetrahedra that partition P and the complement of the convex hull of $P \cup Q$. Returning to the side-by-side case, the combination of our result with Halperin's idea yields an $O(n)$ size tetrahedralization with $O(n)$ Steiner points in $O(n)$ time (n is the size of the polyhedra).

The paper is structured as follows: Section 2 reviews the basic definitions, and presents two important lemmas. The key ideas of 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.

A *polyhedron* in $\mathcal{R}^3$ is a connected piecewise-linear 3-manifold with boundary that is connected and 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. 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 $H(P \cup Q)$ of $P \cup Q$ *properly* contains P, and Q contributes one of the hull's facets, which is diametrically opposite any facets of the hull contributed by P; this is why we refer to the closure of the difference $H(P \cup Q) \setminus P$ as the *region "between" P and Q*. Except for $H(P \cup Q)$'s facets contributed by P or Q, the remaining facets are incident to vertices of both P and Q, and are thus called *bridges*. It is crucial to observe that the bridges lie on planes tangent to both P and Q. (For more on convex hulls, see [5], [10].) The bridges abut on the boundary of P along a connected polygonal line along edges of P, the *horizon*. In the simplest case, the horizon is a simple closed path, but it may collapse into a chain of adjacent edges of P traversed in both directions, or a single vertex of P; in general, it is a combination of the above cases (see [5]). The facets of the polyhedron P that lie in the interior of the convex hull $H(P \cup Q)$ are called *internal*; the remaining ones are said to be *external*. By extension, we call the edges of P that lie in the interior of $H(P \cup Q)$ *internal* as well. Both facets of P incident upon an internal edge are internal.

Next, we present two important lemmas. To formalize our description, we define the notions of the in- and out-wedge of an edge e of P: the planes that support the two facets of P incident upon e define four *open* 3d-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. *The intersection of the plane E_Q that supports the polygon Q and the out-wedge of a horizon edge e of P that is not parallel to E_Q is an open two-dimensional wedge that does not intersect Q.*

Sketch of Proof: Since e is a horizon edge of P, it is an edge of the convex hull $H(P \cup Q)$. So, for any plane Π through e that is tangent to $H(P \cup Q)$, both P and Q lie entirely in (the closure of) one of the two halfspaces defined by Π, whereas the out-wedge of e lies in the other. (In Figure 1, the intersection of E_Q and the out-wedge of e is shown shaded.) ∎

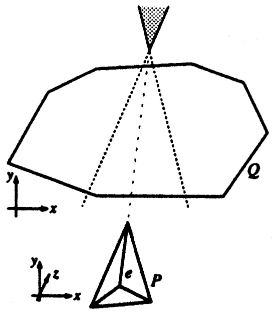

Figure 1

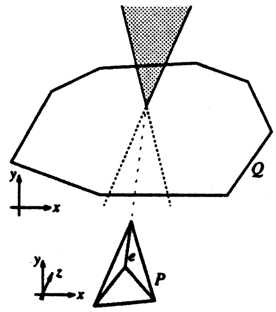

Figure 2

We can prove by contradiction that, for internal edges (Figure 2), we have:

Lemma 2.2. *The intersection of the plane E_Q that supports the polygon Q and the out-wedge of an internal edge e of P that is* **not** *parallel to E_Q is an open two-dimensional wedge that intersects Q.*

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 implies that the reported tetrahedra belong to one of the following three classes:

(i) *v-f tetrahedra* defined by a vertex of P and a triangle in Q,

(ii) *f-v tetrahedra* defined by a triangle on the boundary of P and a vertex of Q, and

(iii) *e-e tetrahedra* defined by an edge of P and an edge of Q.

It is easy to see that the first two classes account for a number of tetrahedra linear in the combined size of P and Q; therefore, making sure that the number of tetrahedra in the third class is also linear in the combined size of P and Q guarantees a linear total size of the tetrahedralization.

3. Rolling Lines.

Consider a convex polyhedron P and a convex polygon Q whose supporting plane does not intersect P; let us orient each edge of P so that it points towards the plane E_Q that supports Q (ties are broken in some arbitrary but consistent way). Then, there exists a *unique* vertex of P such that all incident edges point at it; this is the vertex of P closest to E_Q. Next, we define the notion of rolling lines, which is the key tool in the definition of the tetrahedralization: for each edge e of Q, the corresponding *rolling line* is a line parallel to e that is tangent to the boundary of P; it is initially located at the vertex w of $P \cap closure(b_e)$ closest to e (where b_e is the unique bridge incident to e), and is let free to roll on the boundary of P along internal or horizon edges of P complying with their associated orientations (we stress that the line must always be tangent to P). The line stops when it reaches the vertex of P that is closest to Q. It is important to observe

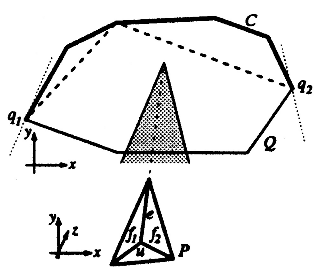

Figure 3

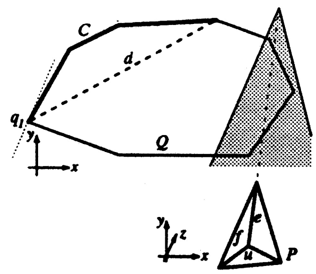

Figure 4

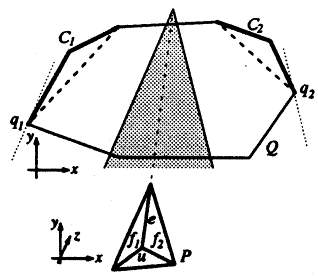

Figure 5

that the line rolls along a simple path of edges of P, which we call the corresponding *rolling path*; the rolling paths may share edges but they do not cross, and their traces around a slice of P (with a plane parallel to Q) are in the same order as the corresponding edges around Q — think in terms of their slopes. In terms of the rolling lines, we define the following tetrahedralization scheme (for proof of correctness, see [9]).

Lemma 3.1. *The region "between" a convex polyhedron P and a convex polygon Q (whose supporting plane does not intersect P) can be tetrahedralized by reporting:*

1. *e-e tetrahedra defined by each edge of Q and the edges in the rolling path of the corresponding rolling line;*
2. *f-v tetrahedra defined by each vertex q of Q and the (internal) facets of P, if any, located between the rolling paths of the rolling lines that correspond to q's incident edges of Q;*
3. *v-f tetrahedra defined by the vertex of P closest to Q and a triangulation of Q.*

3.1. Merging Rolling Lines. Unfortunately, the above tetrahedralization scheme does not guarantee a number of tetrahedra linear in the total size n of P and Q; indeed, it is conceivable that $\Theta(n)$ rolling lines roll along a chain of $\Theta(n)$ polyhedron edges, which will result in a $\Theta(n^2)$ size tetrahedralization. What helps us achieve our goal is the key idea of "merging" rolling lines, so that no more than a constant number of them roll along the same edge of the polyhedron P; it relies on the fact that for an edge e of P (oriented from vertex u to v) that is not parallel to the polygon Q, a tetrahedron defined by u and a triangle in Q that does not intersect the in-wedge of e lies in the closure of $(\text{convex\_hull}(P \cup Q) \setminus P)$.
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 await to roll along the edge e correspond to a single chain C of consecutive edges of Q, which is delimited by the vertices at which planes parallel to e's incident facets are tangent to Q. Figure 3 depicts the situation, where the shaded portion of Q corresponds to the intersection of E_Q with the in-wedge of e (compare with Figure 2). Then, we "merge" the rolling lines associated with e by (i) finding diagonals in the non-shaded part of Q that clip portions of Q and "shortcut" subchains of C, (ii) replacing the corresponding rolling lines by a single rolling line that corresponds to the diagonals, and (iii) reporting tetrahedra defined by u and the clipped portions of Q. It is not difficult to prove that the rolling lines associated with an internal edge e can be "merged" into *at most three* rolling lines that will roll along e (see Figure 3).
2. *The edge e is a horizon edge incident to only one internal facet f of P:* In this case, the rolling lines that await to roll along the edge e correspond to a single chain C of consecutive polygon edges, such that the entire chain C and the polyhedron P lie on opposite sides of the plane supporting f (Figure 4). We can then clip the polygon Q about the diagonal d that separates C from the rest of Q, and thus "merge" all the rolling lines associated with e into a *single* rolling line that corresponds to d. Of course, we also report tetrahedra defined by u 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*: This case is nothing but 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 (Figure 5). 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 u and triangulations of the clipped parts of the polygon 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 e of the polyhedron P is at most three. The process involves clipping portions of the polygon Q about diagonals of Q (thus, the resulting polygon remains convex), and reporting v-f tetrahedra whose number is linear in the size of the clipped portions.*

4. The Algorithm.

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, the topological ordering of the vertices in the directed acyclic graph G induced by the polyhedron's internal and horizon edges oriented towards the polygon is sufficient: to compute the ordering, we maintain a list L of vertices whose predecessors in G have all been processed (the number of unprocessed predecessors is stored in a field *in-degree* in each vertex's record).

The algorithm consists of 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 [1], [7], [8]), 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 so that it points towards the polygon (ties are broken in some arbitrary but consistent way).

Step 2: We compute the *bridges* of the convex hull of $P \cup Q$ (we use the linear-time merging procedure of the divide-and-conquer algorithm to compute the convex hull of a point set in $\mathcal{R}^3$ [5]). The edges of P incident upon the bridges form the *horizon*. Moreover, the *internal* facets of P can be found easily: we first determine the internal facets adjacent to the horizon edges (by using information from the bridges), and then we find the remaining ones by moving from an internal facet to its neighbors, without ever crossing the horizon. Last, we determine the starting points for all the rolling lines as described in Section 3.

Step 3: For each vertex v of P incident upon an internal facet, we store at its field *in-degree* the number of incident internal or horizon edges of P oriented towards v; if this number is 0, v is inserted 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 u, from the list L. If u 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 proceed to Step 5.

Step 5: We process the edges of P emanating from u in order around u. For each such internal or horizon edge e, we select among the rolling lines located at u those, if any, that are tangent to e and would thus roll along e (it is important to note that the edges in order around u get matched with rolling lines in the order that the corresponding edges appear around the polygon). The collected rolling lines are merged as described in Section 3 by finding the appropriate diagonals; in the process, the polygon may be clipped and v-f tetrahedra may be reported. Next, the resulting rolling lines roll along e, that is, we report an e-e tetrahedron defined by e and the polygon edge corresponding to each of these roling lines, and the rolling lines are moved from u to the other polyhedron vertex incident upon e. Finally, the internal facets incident upon e are associated with the polygon vertex with which they are going to define f-v tetrahedra as follows: If lines rolled along e, then the facet that is to the left (right, resp.) of e is associated with the leftmost (rightmost, resp.) vertex incident to the leftmost (rightmost, resp.) polygon edge corresponding to a line that rolled along e (for example, in Figures 3 and 5, f_1 and f_2 are associated with q_1 and q_2 respectively, while in Figure 4, f is associated with q). If no lines roll along e, then if e is on the horizon, the 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 u are processed from left to right). Finally, if e points from u to w, we decrease the in-degree of w by 1; if it becomes equal to 0, we insert w to L.

Step 6: When all the edges emanating from vertex u 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 (i) triangulate what is left of the original polygon Q (due to clipping during the rolling line merging process) and report v-f tetrahedra defined by the vertex of P closest to the polygon Q and the resulting triangles, and (ii) triangulate each facet f of P 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. Moreover, it can be proved that the total time spent by the algorithm is proportional to the size of Q and the total number of faces of P; hence, it is linear in the combined sizes of P and Q (see [9]). Finally, let us count the total number of tetrahedra produced: if the number of vertices of P and Q are n_P and n_Q respectively, and the number of edges of P is e_P, Lemma 3.1 implies that the total number of tetrahedra does not exceed $3e_p + (2n_P - 4) + (n_Q - 2)$; the three terms correspond to the number of e-e tetrahedra, f-v tetrahedra, and v-f tetrahedra respectively. Euler's formula for convex polyhedra in $\mathcal{R}^3$ implies that the total number of edges of P is no more than $3n_P - 6$, which brings the total number of tetrahedra to no more than $11n_P + n_Q - 24 \leq 11n - 24$.

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 P) in $\mathcal{R}^3$ of total combined size n, we show that one can partition the closure of convex_hull$(P \cup Q) \setminus P$ into at most $11n - 24$ tetrahedra without introducing Steiner points. The tetrahedralization is compatible with any triangulation of the boundary of P, and can be computed in $O(n)$ time.*

Unless the polygon has no collinear consecutive edges, the resulting tetrahedralization is guaranteed to be a cell complex. If collinear consecutive edges exist, our approach automatically merges the corresponding rolling lines (and the edges) into a single rolling line (edge resp.), and thus the tetrahedralization is not a cell complex; a cell complex can, however, be obtained by using an idea similar to that of Eppstein (in order to "protect" edges of polyhedra during tetrahedralizations (see [3])) at the expense of introducing Steiner points.

Our result also implies an optimal size tetrahedralization for the side-by-side case (see [2]) if a linear number of Steiner points are allowed; namely, the closure of $(\text{convex\_hull}(P_1 \cup P_2) \setminus (P_1 \cup P_2))$ between two non-intersecting convex polyhedra P_1 and P_2 of total size n can be partitioned into $O(n)$ tetrahedra using $O(n)$ Steiner points. The tetrahedralization is compatible with the boundaries of both P_1 and P_2, and can be computed in $O(n)$ time. It would be of interest to investigate the question whether an $O(n)$ size tetrahedralization is possible with $o(n)$ Steiner points in the worst case.

Moreover, it would be interesting to find out whether the idea of rolling lines can be used to yield a linear size tetrahedralization without Steiner points in the *nested case*, i.e., when tetrahedralizing the region between the boundaries of two nested convex polyhedra.

Finally, let us consider two convex polygons Π_1 and Π_2 that lie on parallel planes in $\mathcal{R}^3$. Clearly, Bern's quadratic lower bound for the side-by-side case implies that a tetrahedralization of their convex hull that is compatible with arbitrary triangulations of Π_1 and Π_2 and does not involve Steiner points may be of quadratic size in the worst case. The method, however, does not work if the two polygons are copies of the same polygon: Is the size of a compatible tetrahedralization of their convex hull in this case quadratic in the size of the polygons in the worst case?

6. References

1. B.G. Baumgart, "A Polyhedron Representation for Computer Vision," *Proc. 1975 National Comput. Conference*, AFIPS Conference Proceedings, **44** (1975), 589–596.
2. M. Bern, "Compatible Tetrahedralizations," *Proc. 9th Annual ACM Symposium on Computational Geometry* (1993), 281–288.
3. M. Bern and D. Eppstein, "Mesh Generation and Optimal Triangulation," *Xerox PARC Technical Report CSL-92-1.*
4. B. Chazelle and N. Shouraboura, "Bounds on the Size of Tetrahedralizations," to appear in *Proc. 10th Annual ACM Symposium on Computational Geometry* (1994).
5. H. Edelsbrunner, *"Algorithms in Combinatorial Geometry"* (1987), Springer-Verlag.
6. J.E. Goodman and J. Pach, "Cell Decomposition of Polytopes by Bending," *Israel Journal of Mathematics* **64** (1988), 129–138.
7. L.J. Guibas and J. Stolfi, "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams," *ACM Transactions on Graphics* 4 (1985), 75–123.
9. L. Palios, "Optimal Tetrahedralization of the 3d-region 'between' a Convex Polyhedron and a Convex Polygon," *Research Report* (1994), The Geometry Center, University of Minnesota.
8. D.E. Muller and F.P. Preparata, "Finding the Intersection of two Convex Polyhedra," *Theoretical Computer Science* **7** (1978), 217–236.
10. F.P. Preparata and M.I. Shamos, *"Computational Geometry, An Introduction"* (1985), Springer-Verlag.