On Polyhedra Induced by Point Sets in Space*

Ferran Hurtado *Godfried T. Toussaint* *Joan Trias*

Abstract

Given a set S of n points in the plane (not all on a line) it is well known that it is always possible to *polygonize* S, i.e., construct a simple polygon P such that the vertices of P are precisely the given points in S. For example, the shortest circuit through S must be such a simple polygon [20]. In 1994 Grünbaum [13] showed that an analogous theorem holds in 3-dimensional space. More precisely, if S is a set of n points in space (not all of which are coplanar) then it is always possible to *polyhedronize* S, i,e., construct a simple (sphere-like) polyhedron P such that the vertices of P are precisely the given points in S. Grünbaum's constructive proof may yield *Schonhardt* polyhedra that cannot be triangulated [**?**]. In this paper we propose several alternative algorithms for constructing such polyhedra induced by a set of points. Our methods yield polyhedra which not only may always be triangulated, but which enjoy several other useful properties as well. Such properties include polyhedra that are star-shaped, have hamiltonian skeletons, and admit efficient point location queries. Furthermore, we show that *polyhedronizations* with a variety of such useful properties can be computed efficiently in $O(n \log n)$ time.

1 Introduction

In 1964 Hugo Steinhaus posed the following problem [25]. There are n points lying in a plane, no three of them lying on the same straight line. Is it always possible to find a closed polygon with n non-intersecting sides whose vertices are these n points? Then he proceeded to give a clever proof by induction that this is true. His proof removes an extreme point of the set and by induction assumes the remaining $n-1$ points admit such a polygon. Then by trial and error he searches for an edge of this polygon that is completely visible from the removed point. A direct implementation of his proof yields an $O(n^3)$ time algorithm for constructing the required polygon.

Independently, in 1966 Michael Gemignani [9] posed the following problem: given n points in the plane, not all lying on the same straight line, are they the vertices of a simple closed polygonal chain and, if so, produce a witness, i.e., construct one. Note that the problem Gemignani posed is more general than the version Steinhaus posed since Gemignani only

*Research supported by NSERC Grant no. OGP0009293 and FCAR Grant no. 93-ER-0291.

assumes not *all* the points are collinear, whereas Steinhaus assumes no three are collinear. Indeed, the induction proof of Steinhaus does not hold if one only assumes that not all points are collinear. Gemignani also conjectures that the *shortest* closed route through the points must be one of these simple polygons. This conjecture had in fact been proved one year earlier by Quintas and Supnick [20]. It has been independently proved elsewhere since then [22] but finding such a shortest circuit is difficult. This is the well known *Euclidean Travelling Salesman Problem* and it is known to be NP-complete [16]. In a later paper Gemignani [8] gives a much simpler proof than the one given in [9] which yields a star-shaped polygon in $O(nlogn)$ time.

In 1994 Branko Grünbaum [13] gives an alternate simple proof of Gemignani's problem that yields a *monotonic* polygon. Furthermore, Grünbaum's proof can also be easily implemented in $O(nlogn)$ time.

Different types of polygonizations are of interest in a variety of disciplines and serve different functions. Clearly for traveling-salesperson-type problems we are interested in polygonizations that have a short if not minimal length. In pattern recognition we are often interested in polygonizations that characterize in a "nice" periosteal manner the boundary of some shape [18], [29]. One may be interested in polygonizations as data structures that afford simple insertions and deletions of points from S [1]. In computational geometry it may also be the case that there exists a simple solution to a problem for polygons that may also be the solution to the problem when the input is the set of vertices of the polygon, i.e., a set of points. If the right kind of polygonization can be found efficiently then a simple solution for polygons may yield a simple solution for point sets. A notable example here is the convex hull problem [11].

Ron Graham in 1972 [11] proposed a simple optimal $O(n \log n)$ time algorithm for computing the convex hull of S that works in two stages. In the first stage he obtains a special polygonization of S and in the second stage he finds the convex hull of the resulting polygon with a linear time backtracking algorithm now well known as the *Graham Scan*. Here we are interested in his polygonization algorithm. First a point inside the convex hull called the origin is found by taking the average (center of gravity) of three of the given points. Then the points in S are sorted by polar angle about the origin. Finally the points in S are connected by edges in the sorted order to yield the polygonization. We shall call this the *star polygonization*.

The star polygonization has the nice property that it is star-shaped and a point in its kernel (namely the origin O) is known. Therefore the polygon can be triangulated with a simple and practical algorithm in linear time [31]. This is an attractive property of a polygonization because a triangulated polygon is very useful for the efficient computation of many geometric properties [28], [30], [14].

The reason a triangulated polygon is so useful is that the dual graph of the triangulation is a tree and this tree can be used to guide efficient search in the polygon. For precisely the same reason, an even more attractive property of a polygon is the admissibility of a good *thin* triangulation [24]. A thin triangulation is one that minimizes the number of nodes of degree three in the dual tree and can be computed in $O(n^3)$ time using $O(n^2)$

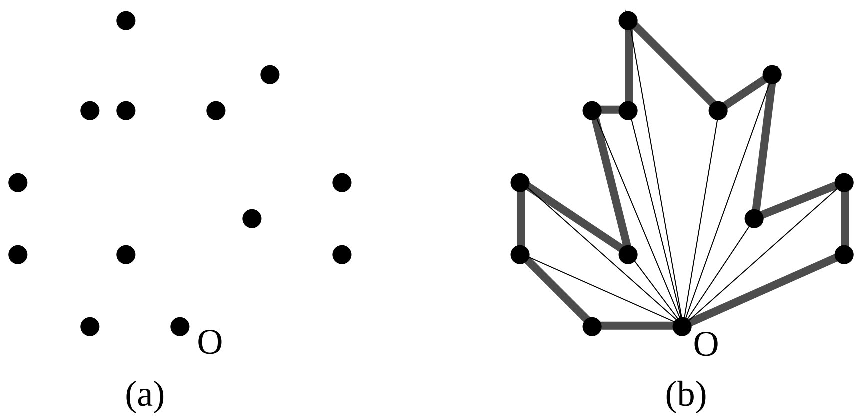

Figure 1: Illustrating the *fan polygonization* of a point set.

space [24]. Clearly an even more attractive property of a polygon is that of admitting a triangulation such that its dual is a chain. Such polygons are called *serpentine*. The disadvantage of the star polygonization is that it may yield a polygon that is not serpentine and therefore it may require $O(n^3)$ time and $O(n^2)$ space to compute a thin triangulation. However, a simple modification of the Graham polygonization not only has this serpentine property but a serpentine triangulation is generated during polygonization at no extra cost. Instead of picking as the origin a point in the interior of the convex hull, we chose it to be a point of S on the convex hull of S such as the point with minimum y coordinate. We call such a polygonization a *fan polygonization* of a point set. A set of points and its fan polygonization is illustrated in Fig. 1. The sorting of the points with respect to O yields the triangulation immediately and therefore no triangulation algorithm is needed. Furthermore, the triangulation thus obtained is clearly serpentine.

Another desirable and useful property of a polygon is its *monotonicity*. Neither the star nor the fan polygonization methods are guaranteed to yield monotonic polygons. However a monotonic polygonization can be easily obtained in $O(n \log n)$ time as follows (refer to Fig. 2. It should be noted that this polygonization was implicitly used in several variations of Graham's convex hull algorithm [2], [3]. First find the points of S with minimum and maximum x-coordinates, say a and b, respectively. Construct a line L through a and b and determine which of the $n-2$ points lie above L (call these S_1) and which below (call these S_2). Sort the points in S_1 by x-coordinate and connect adjacent points by edges. Do the same for points in S_2 . Finally connect the end vertices of the resulting chains to the corresponding extreme points a and b. While such a polygon is clearly monotonic in the x-direction and a simple and practical $O(n)$ time triangulation algorithm exists for monotone polygons [27] this procedure may yield monotonic polygons that are not serpentine and therefore computing a thin triangulation for them may require $O(n^3)$ time and $O(n^2)$ space as in the case of star polygonizations.

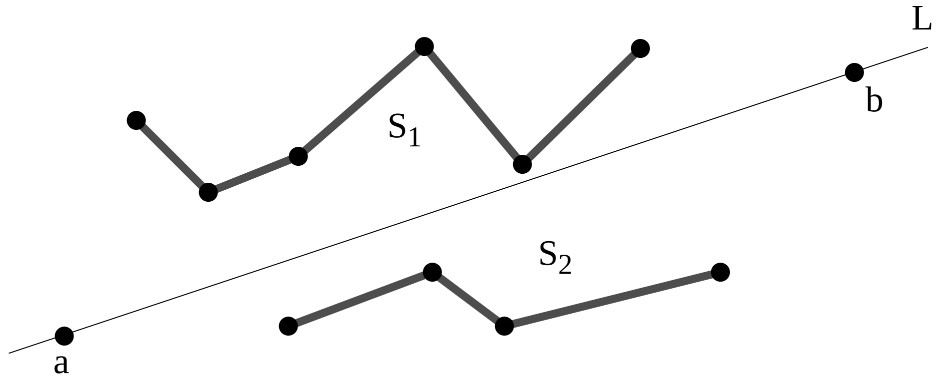

Figure 2: Illustrating a *monotonic* polygonization.

The planar polygonization problem can be generalized in at least two ways to 3-dimensional space. We can ask for a closed polygonal chain that is "simple" in the sense that it is not knotted. We will call this the 3-D-polygonization problem. This problem is trivially solved using the planar polygonization procedures by projecting the points of S onto a plane and then "lifting" the planar polygonization obtained back into space. In the more interesting generalization we can ask for a simple polyhedron the vertices of which are the given point set. We call this problem the *polyhedronization* problem. Surprisingly this problem does not appear to have been studied before in this general setting. However a special case of it is a well known problem in solid modelling and has received much attention in medical applications concerning the reconstruction of solids [5]. In this instance of the problem we are given two simple polygons P and Q of n and m vertices, respectively, each on one of two parallel planes in space, and it is desired to find a simple polyhedron that has the two polygons as faces and whose vertices are precisely the vertices of the two polygons. Clearly, when the two given polygons are *convex* this is always possible as it suffices to compute the convex hull of the union of the two polygons. Furthermore such a polyhedronization can be computed in $O(n + m)$ time by using the "rotating caliper" technique [26]. O'Rourke and Subramanian [19] have shown that such a polyhedronization is not always possible for arbitrary simple polygons. Finally, if a judiciously-placed "Steiner vertex" is permitted then such a polyhedronization always exists [10].

In this paper we study various methods for generating polyhedronizations that have a variety of desirable properties: monotonicity, starshapedness, admitting a tetrahedralization (triangulation), possibly with nice dual structure, possessing a good 1-skeleton from the viewpoint of graph theory and affording fast point location. The 3-D-polygonization problem is solved along the way in that a polyhedronization with the property that it admits a hamiltonian yields a 3-D-polygonization when one of its hamiltonians is reported. Before presenting the 3-D results in section 4, as a way of introducing one of the methods, we first present it in the plane in section 3. This new polygonization method combines the desirable properties of both the monotonic and fan polygonizations yielding in $O(n \log n)$ time

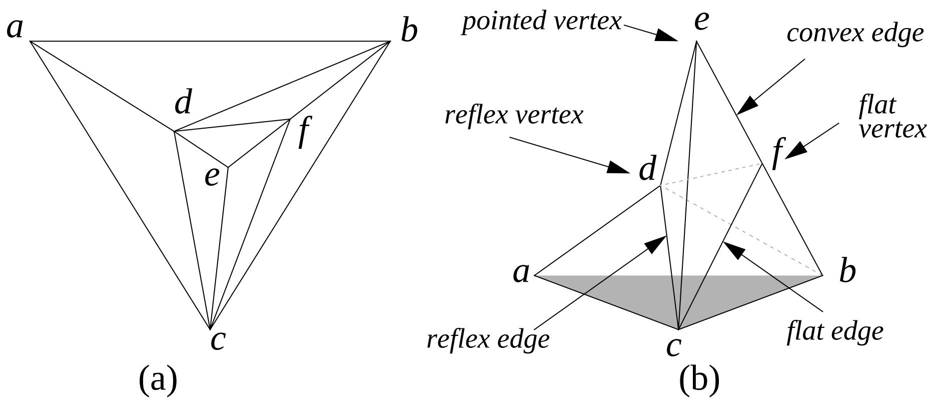

Figure 3: Illustrating the edges and vertices of a terrain.

polygonizations that are: (1) monotonic, (2) serpentine and (3) triangulated in a serpentine manner at no extra cost.

2 Geometric Properties of Polyhedra

Vertices and Edges of Polyhedra

It is helpful to distinguish between a variety of different edges and vertices when speaking about polyhedra. Here we borrow some terminology from Griffiths [12] as well as Chazelle and Palios [6] and introduce some of our own. An edge e of T is said to be *reflex* if the interior dihedral angle formed by its two incident faces is greater than 180 degrees. An edge e of T is said to be *convex* if the interior dihedral angle formed by its two incident faces is less than 180 degrees. An edge e of T is said to be *flat* if the interior dihedral angle formed by its two incident faces is equal to 180 degrees. We say that a vertex is *reflex* if it is incident upon at least one reflex edge, and that it is *flat* if all its incident faces lie in at most two distinct planes. Finally, a vertex is called *pointed* if it is neither flat nor reflex. These definitions are illustrated in Fig. 3 where a polyhedron is shown from the top in Fig. 3(a) and from a perspective view in Fig. 3(b). The polyhedron is constructed as follows. The base (on the xy-plane) is the equilateral triangle abc with sides of length equal to 15 units. Vertex d is located near the center of triangle abc at an elevation of 5 units creating a tetrahedron. Vertex e is located near the center of triangle dbc of the first tetrahedron. Finally a vertex f is created on edge eb and connected to d and c. Then ef is a convex edge, dc is a reflex edge, cf is a flat edge, f is a flat vertex, d is a reflex vertex and e is a pointed vertex.

Tetrahedralizations

It is well known that a simple planar polygon can always be triangulated and the reader is referred to [30] for a survey of this problem and a variety of algorithms for solving it. The problem consists of constructing a *triangulation* of P, i.e., decomposing P into a set of non-overlapping triangles (their interiors do not intersect) without adding new vertices. Mathematicians have been interested in constructive proofs (algorithms) of the existence of triangulations for simple polygons as early as 1911 [17]. The algorithm of Lennes [17] works by recursively inserting diagonals between pairs of vertices of P and runs in $O(n^2)$ time on a polygon with n vertices. This procedure sounds deceptively straight forward. However, since 1911 this type of algorithm has reappeared in a score of papers and text books during the past eighty years very often and surprisingly containing fundamental errors. See the paper by Chung-Wu Ho [15] for a series of counter-examples to published triangulation algorithms of the recursive diagonal insertion kind.

It is less well known that the analogous 3-dimensional triangulation problem of decomposing polyhedra into tetrahedra (also called *tetrahedralization*) is not always possible. Lennes [17] was the first to exhibit such an *indecomposable* polyhedron. Lennes's counter example contained seven vertices. In 1928 Schonhardt [23] strengthened this result by constructing a polyhedron of six vertices that did not admit even a single diagonal and proved that no indecomposable polyhedra existed with less than six vertices. In 1948 Bagemihl [4] generalized Schonhardt's result to polyhedra with any number $n > 6$. These counter-examples open up the intriguing computational question of whether a given polyhedron can be tetrahedralized. In 1992 Ruppert and Seidel [21] showed that the problem of deciding whether a polyhedron can be tetrahedralized is NP-complete. Surprisingly, they showed that the problem remains NP-complete when restricted to the case of star-shaped polyhedra. This raises the interesting question of whether a polyhedral terrain can be tetrahedralized, but first we must define what we mean by this.

First let us define the convex hull of a polyhedral terrain T as the convex hull of its vertex set $V = \{v_1, v_2, \ldots, v_n\}$. We note here that this convex hull, denoted by $CH(T)$, encloses the polyhedral surface defined in the interior of $CH(P)$ on the X-Y plane and can be computed, ignoring the polyhedral structure of T, in $O(n \log n)$ time [7]. We now define the *tetrahedralization* of a terrain T as the tetrahedralization of the region in S (the space strictly above T in E^3) in the interior of $CH(T)$.

An obvious question now is whether every terrain can be tetrahedralized. Next we show that a terrain does not necessarily admit a tetrahedralization. Our construction is based on the example of Schonhardt which we must convert to a terrain. Therefore we first describe Schonhardt's construction (see Fig. 4(a)). The six-vertex polyhedron is constructed by starting with a triangular prism (the top and bottom are shown shaded) and "twisting" the top face in the direction shown by some small amount. The three side faces (rectangles consisting of two triangles each) cannot remain planar and so "buckle" inwards along the diagonals to produce bona-fide triangular faces. It is easy to see that no two non-adjacent vertices of this polyhedron are internally visible to each other. Now we construct an indecomposable terrain. Let the terrain be convex except for a single "pocket" like the crater of a volcano.

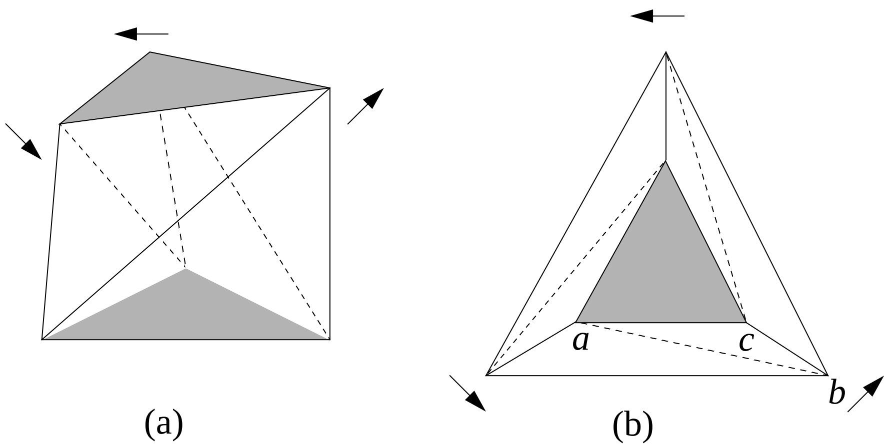

Figure 4: Constructing a polyhedral terrain that cannot be tetrahedralized.

It is sufficient to show that we can build an indecomposable crater. We start by making the crater from Schonhardt's prism first by making the top triangle much bigger than the bottom triangle and then by removing the top face so that it becomes part of the terrain as illustrated in Fig. 4(b) where the crater is viewed from the top. The bottom triangle is shaded. Now we "twist" the top triangle (the rim of the crater) in a counter-clockwise manner by some amount to create the necessary buckling as before and the resulting lack of visibility between non-adjacent pairs of vertices of the crater. However, we must now be careful that during this twisting we do not lose the polyhedral terrain property that we had before twisting. It is easy to show that we can ensure that the twisted crater remains a terrain by rotating through an angle smaller than that required to carry the projection of the diagonal a, b on the X-Y plane to be collinear with the projection of a, c on X-Y plane.

3 On-line Polygonization

We present here a new method of polygonization in the plane which achieves many desirable properties, and that somehow surprisingly does not extend to 3D. The main idea is very simple and consists of sorting all the points along some direction such as the x-axis, creating a triangle from the first three points, and subsequently processing one point at a time in the sorted list which creates a new triangle that is "glued" on to a suitable visible edge of the existing polygonization. We describe next the algorithm more formally.

Algorithm-1

Initialization

(1) Sort lexicographically the points along their $+x$ and $+y$ coordinates to obtain the list $p_1, p_2, \ldots, p_n$.

(2) If p_1, p_2, and p_3 are not collinear, connect them to form triangle T_3, which would also be the initial polygon Q_3 and let us define an initial value ℓ to be $\ell = 4$.

Otherwise let p_j be the first point non-collinear with its preceding points in the list, $p_1, \ldots, p_{j-1}$. We construct now the initial polygon $Q_j = p_1 p_j p_{j-1} \ldots p_2 p_1$, and the triangles $T_3 = p_1 p_j p_2, T_4 = p_2 p_j p_3, \ldots, T_j = p_{j-2} p_j p_{j-1}$, and let the value ℓ be $\ell = j + 1$.

Iteration

for $i = \ell$ to $i = n$ do:

Connect point p_i to a visible edge of triangle T_{i-1} in the polygon constructed thus far, denote this new triangle by T_i and the updated polygon by Q_i.

end for

The correctness of the above procedure follows from the following simple lemma.

Lemma 3.1 *At every step of Algorithm-1 p_i sees completely at least one forward edge of triangle $[p_{i-1}, p_{i-2}, p_{i-3}]$.*

Proof: ■

Theorem 3.2 *A set of points S in the plane admits a* serpentine *polygonization and a triangulated serpentine polygonization can be obtained in $O(n \log n)$ time.*

At first glance it may appear that this on-line algorithm extends to three dimensions by "gluing" a new tetrahedron to one of the three faces incident on the last point of the polyhedron constructed thus far. Unfortunately, it may happen that none of the three faces is completely visible from the new point to be inserted, and therefore the method fails. An example of a set of points for which this procedure fails is shown in Figure Fig. 5.

First consider the six points ordered by increasing x-coordinate: $P_1 = (0, 0, 0)$, $P_2 = (0, 1, 0)$, $P_3 = (0, -3, 1)$, $P_4 = (1, 0, 0)$, $P_5 = (2, -5/2, -3)$ and $P_6 = (3, -4, 1)$. The first, second, and third tetrahedra glued in the construction are given, respectively, by $P_1P_2P_3P_4$, $P_1P_2P_4P_5$, $P_1P_4P_5P_6$.

When viewed from the top ($+z$ direction) the projection of P_5 on the xy plane lies in the interior of the projection of the triangle $P_1P_6P_4$. Therefore the outer normals of faces $P_4P_5P_6$ and $P_1P_5P_6$ are pointing in the negative z direction. Furthermore, any point P_7 above the planes $P_1P_4P_6$, $P_1P_5P_6$ and $P_4P_5P_6$ cannot see faces $P_4P_5P_6$ and $P_1P_5P_6$. Finally, if P_7 is high enough it will not see face $P_1P_4P_6$ either, and if P_7 lies on a nearly vertical line slightly slanted towards the positive x axis, its x-coordinate can be made to be larger than that of P_6, as required.

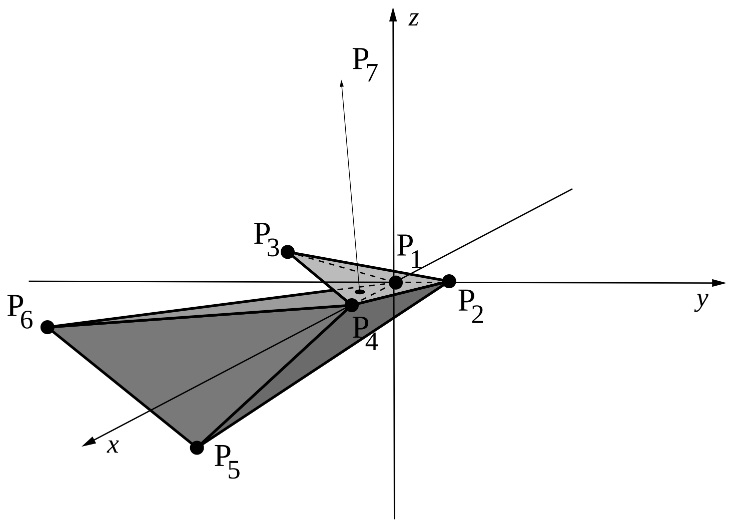

Figure 5: Illustrating the polyhedron constructed from the first three tetrahedra and the line on which the seventh point lies.

4 Polyhedronization of Point Sets in Space

In this section we prove that every set of n points in general position in three dimensional space admits a polyhedronization. We describe several different types of polyhedronizations and analyze their properties as well as algorithms for their computation.

Monotonic Polyhedronizations

In 1994 Grünbaum [13] outlined a constructive proof that a set of points S could always be polyhedronized. As it turns out his idea leads to a monotonic polyhedronization, as defined below. In the following we present a simplification of his approach and show that it can be efficiently computed.

Definition 4.1 *A polyhedron is xy-monotonic provided that its intersection with every line parallel to the z-axis is either empty or a connected set.*

In other words, an xy-monotonic polyhedron is bounded from above and below by terrains. Such polyhedra are ubiquitous in geographic information systems and manufacturing applications, and admit efficient point location queries by deciding whether a query point is above or below the terrains, which can be quickly achieved after performing point location in the projections of the terrains on the xy-plane. Hereafter we use the simpler term *monotonic* to mean xy-monotonic.

Theorem 4.1 *A set of points S in space admits a monotonic polyhedronization that can be obtained in $O(n \log n)$ time.*

Proof: First compute the convex hull $CH(S)$ of S. If no points of S lie inside this convex hull we are done because the poyhedron $CH(S)$ is a monotonic polyhedron. Otherwise, let $CH_L(S)$ and $CH_U(S)$ be the lower and upper convex hull of S, respectively. Let B be the shadow boundary of the convex hull of S, in other words the set of edges common to $CH_L(S)$ and $CH_U(S)$. Let S_U be the subset of points in S which are not vertices of $CH_L(S)$. Triangulate in any way the projection of the set $S_U \cup B$ on the xy-plane, and lift each triangle to the points that projected onto its vertices. By gluing along B this terrain with $CH_L(S)$ we obtain the desired monotonic polyhedron.

Computing the convex hull and constructing a triangulation of the projected points can be done in time $O(n \log n)$ [**?**], which is the overall running time as the lifting step only requires $O(n)$ time. ∎

A drawback of the above construction is that the resulting polyhedronization may not admit a tetrahedralization **(insert here the counterexample or give a pointer)**. We show next an alternative more complicated construction that admits a tetrahedralization. We start with some technical lemmas.

Lemma 4.2 *Given a monotonic tetrahedralized polyhedron, and points that lie above or below the polyhedron such that a vertical line through each of them intersects the boundary of the polyhedron twice, it is possible to enlarge the polyhedron and its tetrahedralization to encompass these points, while remaining monotonic and tetrahedralized.*

Observation. The preceding lemma also applies when we have a convex lower hull instead of a plane polygon P.

Proof: If a triangle abc of the upper part has points above it, translate a copy of it until it meets a point q; then replace the triangle abc with the three triangles qab, qbc and qac. Iterate. ∎

Lemma 4.3 *Let P be a triangulated convex polygon with vertex set V lying on the plane xy, and let S be a point set in 3-space such that every point in S has positive z and projects vertically inside P (strictly). Then it is possible to construct a tetrahedralized monotonic polyhedron with vertex set $V \cup S$ such that its lower terrain is P.*

Theorem 4.4 *A set of points S in space admits a tetrahedralizable monotonic polyhedronization.*

Proof: Compute the convex hull of the shadow boundary. Let B^+ and B^- be its upper and lower part, respectively. Assume that there are points above B^- (otherwise we switch to points below B^+, otherwise we are done). Use the construction from Lemma 4.3 to obtain a first tetrahedralized monotonic polyhedron, then use Lemma 4.2 to incorporate points below B^-, if any. ∎

Star-shaped Polyhedronizations

Definition 4.2 *A polyhedron Q is starshaped from a non-exterior point p provided that for all points $q \in Q$ the line segment pq lies in Q.*

Hinge Polyhedronizations

We give the name of *hinge polyhedronization* to the following construction. Start with any pair of points $x, y \in S$ for which xy is an edge of the convex hull $CH(S)$, consider a plane H that supports S at xy, and let H^* be a halfplane in H bounded by the line $r = xy$. Sort all the remaining points in the order they are encountered when H^* is rotated around r. Connect all these points in sorted order obtaining an open polygonal chain, and finally connect every vertex of this chain to both x and y.

Theorem 4.5 *A hinge polyhedronization can be constructed in $O(n \log n)$ time and has the following properties:*

1. *star-shaped (fan, edge-visible);*
2. *serpentine*
3. *Hamiltonian;*
4. *affords $O(\log n)$ point-location queries.*

Proof: Let $q_1, \ldots, q_{n-2}$ the points in $S \backslash \{x, y\}$ as they appear sorted in the chain. The first two properties follow immediately from the fact that the hinge polyhedronization consists of the union of tetrahedra defined by xy, q_i and q_{i+1}. The path $xq_1 \ldots q_{n-2}yx$ is a Hamiltonian cycle lying on the 1-skeleton of the polyhedron. Angular binary search with the halfplane having its hinge at xy allows point location as claimed.

The selection of an edge from $CH(S)$ can be done in linear time, and sorting the points in $O(n \log n)$ time, which is the dominating step as the final connections are done in linear time. ∎

Orange Polyhedronizations

We describe here *orange polyhedronizations*, which are a slight modification of hinge polyhedronizations. Start with any pair of points $x, y \in S$ for which xy is *not* an edge of the convex hull $CH(S)$, consider a plane H through xy, and let H^* be a halfplane in H bounded by the line $r = xy$. Sort all the remaining points in the cyclic order they are encountered when H^* is rotated around r. Connect all these points in sorted order obtaining a *closed* polygonal chain, and finally connect every vertex of this chain to both x and y.

Theorem 4.6 *An orange polyhedronization can be constructed in $O(n \log n)$ time and has the following properties:*

1. *star-shaped (from a diagonal);*
2. *admits a tetrahedralization whose dual is a cycle*
3. *Hamiltonian;*
4. *has an Eulerian 1-skeleton for even n;*
5. *affords $O(\log n)$ point-location queries.*

Proof: Let $q_1, \ldots, q_{n-2}$ the points in $S \backslash \{x, y\}$ as they appear cyclically sorted in the chain. The first two properties follow immediately from the construction. The path $q_1 x q_2 y q_3 \ldots q_{n-2} q_1$ is a Hamiltonian cycle lying on the 1-skeleton of the polyhedron. The degrees of x and y in the 1-skeleton of the polyhedron are both $n-2$, while any other vertex has degree 4, therefore for n even all the degrees are even and the graph is Eulerian. Angular binary search with the halfplane having its hinge at xy allows point location as claimed.

The computation is done in the same way as for the hinge polyhedronization in $O(n \log n)$ time. ■

Notice that by deleting a "gajo" tetrahedron from an orange polyhedronization, we obtain an alternate construction of a hinge polyhedron, except for the fact that the hinge is not longer a convex hull edge.

Cone Polyhedronizations

A *cone polyhedronization* with apex $v \in CH(S)$ is constructed as follows: let the apex v be any vertex of the convex hull of S. Consider a plane H such that all the points in S except q are strictly in between H and a plane parallel to H that contains q. Let S^* be the perspective projection of $S \backslash \{q\}$ onto the plane H, from the point q. Triangulate S^* and lift every triangle in the triangulation to the corresponding original points in space. Finally, connect to q the points that project to convex hull vertices of $CH(S^*)$

Theorem 4.7 *A cone polyhedronization can be constructed in $O(n \log n)$ time and has the following properties:*

1. *star-shaped (fan);*
2. *admits a tetrahedralization;*
3. *affords $O(\log n)$ point-location queries.*

Proof: The two first properties are obvious from the construction. Given a query point q, let q^* be the intersection of the line pq with the plane H. If q^* is outside $CH(S^*)$ the q is outside of the polyhedron. Otherwise, we determine which triangle contains q^* and test q for inclusion in the tetrahedron corresponding to this triangle. Kirkpatrick's algorithm allows this to be done within the claimed bound [REFERENCE].

There are several methods for picking a vertex v of the convex hull and a suitable plane H in $O(n \log n)$ time, for example by computing the convex hull itself. S^* is then obtained in linear time and triangulated in $O(n \log n)$ time. Finally the triangulation can be preprocessed in $O(n \log n)$ time using linear space to support logarithmic time point location queries, using Kirkpatrick's algorithm. ■

Pyramid Polyhedronizations

A *pyramid polyhedronization* is a slight variation of the cone polyhedronization, in which instead of constructing a triangulation of S^* we obtain a planar polygonization $q_1^* q_2^* \dots q_{n-1}^* q_1^*$ of S^* and lift a triangulation of this polygon. Furthemore, if we use the serpentine triangulation from the preceding section, we obtain a serpentine polyhedronization, which is also Hamiltonian, because $q_1 v q_2 q_3 \dots q_{n-1} q_1$ is a Hamiltonian cycle in its 1-skeleton. As for point location, Kirpatrick's algorithm can also be used in this context, and in fact is even easier. Therefore we have the following theorem:

Theorem 4.8 *A pyramid polyhedronization can be constructed in $O(n \log n)$ time and has the following properties:*

1. *star-shaped (fan);*
2. *admits a serpentine tetrahedralization;*
3. *is Hamiltonian;*
4. *affords $O(\log n)$ point-location queries.*

Proof: The two first properties are obvious from the construction. Given a query point q, let q^* be the intersection of the line pq with the plane H. If q^* is outside $CH(S^*)$ the q is outside of the polyhedron. Otherwise, we determine which triangle contains q^* and test q for inclusion in the tetrahedron corresponding to this triangle. Kirkpatrick's algorithm allows this to be done within the claimed bound [REFERENCE].

There are several methods for picking a vertex v of the convex hull and a suitable plane H in $O(n \log n)$ time, for example by computing the convex hull itself. S^* is then obtained in linear time and triangulated in $O(n \log n)$ time. Finally the triangulation can be preprocessed in $O(n \log n)$ time using linear space to support logarithmic time point location queries, using Kirkpatrick's algorithm. ■

5 Open Problems

1. Find a monotonic polyhedronization which admits tetrahedralization.

2. How fast can it be recognized that a polyhedron is serpentine?
3. For n odd and > 7, do there exist Eulerian polyhedronizations?
4. Combinatorics of polyhedronizations.
5. If we are given a simple polygon on one plane and a set of points on a parallel plane, do they admit a polyhedronization?

NOTE TO OURSELVES: HAMILTONIAN POLYHEDRONIZATIONS GIVE 3D POLYGONIZATIONS AS A SIDE EFFECT, THIS SHOULD BE MENTIONED SOMEWHERE.

6 Conclusions

References

[1] M. Abellanas, J. Garcia, G. Hernandez, F. Hurtado, and O. Serra. Onion polygonizations. In *Proc. 4th Canadian Conference on Computational Geometry*, pages 127–131, St. Johns, Newfoundland, August 10–14 1992.

[2] S. G. Akl and G. T. Toussaint. A fast convex hull algorithm. *Information Processing Letters*, 7(5):219–222, August 1978.

[3] A. M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9(5):216–219, December 1979.

[4] F. Bagemihl. On indecomposable polyhedra. *American Mathematical Monthly*, pages 411–413, September 1948.

[5] J. D. Boissonat. Reconstruction of solids. *Communications of the ACM*, pages 46–54, June 1985.

[6] B. Chazelle and L. Palios. Triangulating a nonconvex polytope. *Discrete & Computational Geometry*, 5:505–526, 1990.

[7] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, 1987.

[8] M. Gemignani. More on finite subsets and simple closed polygonal paths. *Mathematics Magazine*, 39:158–160, 1966.

[9] M. Gemignani. On finite subsets of the plane and simple closed polygonal paths. *Mathematics Magazine*, pages 38–41, Jan.-Feb. 1966.

[10] C. Gitlin, J. O'Rourke, and V. Subramanian. On reconstructing polyhedra from parallel slices. Technical Report 025, Smith College, Northampton, MA., March 12 1993.

[11] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1:132–133, 1972.

[12] H. B. Griffiths. *Surfaces*. Cambridge University Press, 1981.

[13] Branko Grünbaum. Hamiltonian polygons and polyhedra. *Geombinatorics*, 3:83–89, January 1994.

[14] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan. Linear time algorithms for shortest path and visibility problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.

[15] C.-W. Ho. Decomposition of a polygon into triangles. *The Mathematical Gazette*, 59, 1975.

[16] D. S. Johnson. Computational complexity. In Lawler E. L., J. K. Lenstra, A. H. G. Rinnooy, and D. B. Shmoys, editors, *The Traveling Salesman Problem*, pages 37–87. John Wiley, 1985.

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

[18] J. O'Rourke, H. Booth, and R. Washington. Connect-the-dots: a new heuristic. *Computer Vision, Graphics and Image Processing*, 39:258–266, 1987.

[19] J. O'Rourke and V. Subramanian. On reconstructing polyhedra from parallel slices. Technical Report 008, Smith College, June 1991.

[20] L. V. Quintas and F. Supnick. On some properties of shortest hamiltonian circuits. *American Mathematical Monthly*, 72:977–980, 1965.

[21] J. Ruppert and R. Seidel. On the difficulty of triangulating three-dimensional nonconvex polyhedra. *Discrete and Computational Geometry*, 7:227–253, 1992.

[22] D. Sanders. Metric spaces in which minimal circuits cannot self-intersect. In *Proc. American Mathematical Society*, volume 36, pages 383–387, April 1976.

[23] E. Schonhardt. Uber die zerlegung von dreieckspolyedern in tetraeder. *Mathematische Annalen*, 98:309–312, 1928.

[24] T. Shermer. Computing bushy and thin triangulations. *Computational Geometry: Theory and Applications*, 1:115–125, 1991.

[25] Hugo Steinhaus. *One Hundred Problems in Elementary Mathematics*. Dover Publications, Inc., New York, 194.

[26] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE MELECON' 83*, Athens, Greece, May 1983.

[27] G. T. Toussaint. A new linear algorithm for triangulating monotone polygons. *Pattern Recognition Letters*, 2:155–158, 1984.

[28] G. T. Toussaint. A linear-time algorithm for solving the strong hidden-line problem in a simple polygon. *Pattern Recognition Letters*, 4:449–451, December 1986.

[29] G. T. Toussaint, editor. *Computational Morphology.* North-Holland, 1988.

[30] G. T. Toussaint. Efficient triangulation of simple polygons. *The Visual Computer*, 7(5-5):280–295, September 1991.

[31] T. C. Woo and S. Y. Shin. A linear time algorithm for triangulating a point-visible polygon. *ACM Transactions on Graphics*, 4:60–70, 1985.