Algorithmica (1997) 17: 245–265

Algorithmica

Decomposing the Boundary of a Nonconvex Polyhedron

B. Chazelle[1] and L. Palios[2]

Abstract. We show that the boundary of a three-dimensional polyhedron with r reflex angles and arbitrary genus can be subdivided into $O(r)$ connected pieces, each of which lies on the boundary of its convex hull. A remarkable feature of this result is that the number of these convex-like pieces is independent of the number of vertices. Furthermore, it is linear in r, which contrasts with a quadratic worst-case lower bound in the number of convex pieces needed to decompose the polyhedron itself. The number of new vertices introduced in the process is $O(n)$. The decomposition can be computed in $O(n + r \log r)$ time.

1. Introduction. Because simple objects usually lead to simpler and faster algorithms, it is often useful to preprocess an arbitrary object and express it in terms of simpler components. In two dimensions, for example, polygon triangulation is a standard preprocessing step in many algorithms [1], [5], [7], [10], [15], [16]. Similarly, in three dimensions, a polyhedron can be expressed as a collection of convex pieces or tetrahedra in particular (see [2], [4], [6], and [14] for discussions on such decompositions). Of course, the size of the decomposition is critical for the application that uses it. Unfortunately, a convex partition of a polyhedron may be of size quadratic in the description size of the polyhedron in the worst case [4], which makes it unattractive from an efficiency point of view. It would be, therefore, of interest to have partitions into a guaranteed small number of simple components.

In this paper we consider the problem of subdividing the boundary of a nonconvex polyhedron of arbitrary genus into a small number of connected *convex-like* pieces. By convex-like piece, we mean a polyhedral surface which lies entirely on the boundary of it convex hull. Our result is that the boundary of a nonconvex polyhedron that has r reflex angles can be subdivided into no more than $18r - 2$ such pieces. It is interesting to note that the number of pieces is independent of the number of vertices of the polyhedron, and it is linear in r. The algorithm proceeds in two phases. In the first phase we disassemble the boundary of the polyhedron along the polyhedron's reflex edges and along its "ridges" and "keels" (i.e., the edges that contribute local extrema with respect to a fixed direction). This partitioning scheme yields at most $2r + 2$ pieces. The second phase further splits these pieces into smaller ones that are convex-like by clipping them with planes parallel to a fixed plane that go through the endpoints of the polyhedron's reflex edges. The clipping is carried out in such a way that it introduces only $O(n)$ new vertices. It is worth noting that although a convex-like piece may in general be very complex. the convex-

[1] Department of Computer Science, Princeton University, Princeton, NJ 08544, USA.
[2] The Geometry Center, University of Minnesota, Minneapolis, MN 55454, USA. This work was mostly done while the author was a graduate student at Princeton University.

Received December 3, 1994; revised July 25, 1995. Communicated by L. J. Guibas.

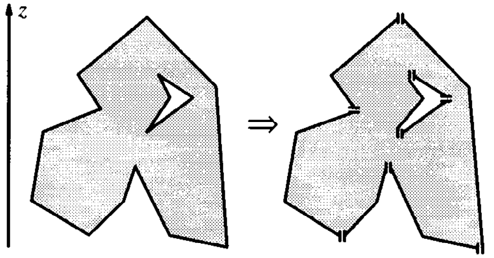

Fig. 1

like pieces that our algorithm produces are simple in shape and well behaved. The entire algorithm runs in linear time, provided that the boundary of the given polyhedron has been triangulated. Boundary triangulation takes $O(n + r \log r)$ time.

In two dimensions the problem is very simple, and admits a linear-time solution that produces the minimum number of polygonal curves into which the boundary of a polygon (possibly with holes) can be cut so that each such curve stays on the boundary of its convex hull. The algorithm first disconnects the boundary of the polygon at its *cusps* (the vertices whose incident edges form an interior angle larger than π), and then breaks each resulting piece in a greedy fashion to enforce the convexity condition. In other words, we start at one end of such a piece and keep walking along it for as long as each encountered edge lies on the convex hull of the subpiece traversed so far, disconnecting it otherwise. The whole process takes linear time. A second algorithm can be obtained by taking the two-dimensional equivalent of the first phase of our algorithm as outlined above. The boundary of the polygon is disassembled at the cusps and at the local extrema with respect to some fixed direction, say, the vertical direction z (Figure 1). It can be proven by induction that a polygon of r cusps has at most $r + 2$ local extrema, which implies that the total number of pieces produced cannot exceed $2r + 2$. This is almost optimal in the worst case, since for any r there is a polygon of r cusps whose boundary cannot be disassembled into fewer than $2r + 1$ pieces each lying on the boundary of its convex hull (Figure 2).

The paper is structured as follows. In Section 2 we introduce our notation and prove a lemma to facilitate the analysis of our algorithm. The algorithm and its complexity analysis are presented in Section 3. Finally, in Section 4 we summarize our results and discuss some open questions.

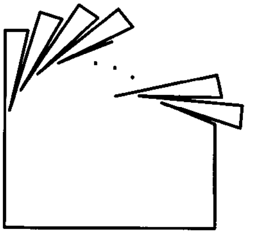

Fig. 2

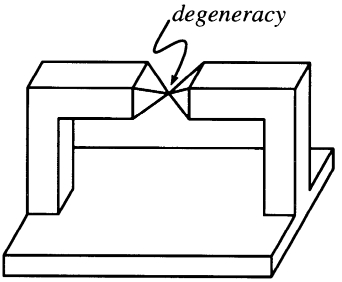

Fig. 3. *Not* a polyhedron.

2. A Geometric Framework. A polyhedron in $\mathfrak{R}^3$ is a connected piecewise-linear 3-manifold with boundary; its boundary is connected and consists of a finite collection of relatively open sets, the *faces* of the polyhedron, which are called *vertices*, *edges*, or *facets*, if their affine closures have dimension 0, 1, or 2, respectively. By virtue of the definition of a polyhedron, no faces can be self-intersecting, dangling, or abutting, and no degeneracies like the one shown in Figure 3 are allowed. An edge e of a polyhedron is said to be *reflex* if the (interior) dihedral angle formed by its two incident facets exceeds π. By extension, we say that a vertex is *reflex* if it is incident upon at least one reflex edge.

A *patch* of a polyhedron P is a collection of facets or subsets of facets of P with their adjoining edges and vertices. The edges of a patch that do not lie on its relative boundary are called *internal*. We try to extend to patches some of the definitions pertaining to polygons. A patch is said to be *connected* if its dual graph is connected; the dual graph of a patch σ has one node for each facet of σ and an edge between a pair of nodes if the corresponding facets of σ are incident on a common edge. Under this definition, neither of the two patches of Figure 4 is considered connected. Unless it consists of a single facet, a connected patch has at least one internal edge. A connected patch is said to be *simple* if it is bounded by a single nonintersecting closed curve; i.e., it does not contain any holes. A patch is called *monotone* with respect to a plane if no two distinct points of the patch project normally to the same point of the plane. Finally, a patch σ is *convex-like* if it lies on the boundary of the convex hull H_σ of its vertices and the interiors of both P and H_σ lie on the same side with respect to each of the facets of σ. The latter

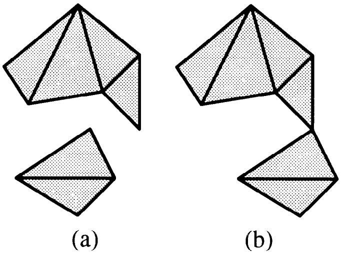

Fig. 4

condition implies that none of the internal edges of a convex-like patch is a reflex edge of the given polyhedron. The following lemma presents sufficient conditions for a patch to be convex-like.

LEMMA 2.1. *Let σ be a patch of a polyhedron P such that none of the internal edges of σ are reflex edges of P. If σ is simple and monotone with respect to a plane Π onto which it projects into a convex polygon, then σ is convex-like.*

PROOF. (The lemma trivially holds if all the facets of σ are coplanar.) We consider the unbounded cylinder whose axis is normal to Π and whose base is the projection of σ onto Π; because σ's projection onto Π is a convex polygon, the cylinder is a convex object. The monotonicity of σ with respect to Π implies that the projection of the boundary of σ onto Π coincides with the boundary of the projection of σ. Hence, the relative boundary of σ lies on the boundary of the cylinder. Moreover, since the patch σ is simple, it separates the cylinder into two unbounded polyhedra, with respect to one of which, say T, the internal edges of σ are nonreflex. Then the interiors of P and T lie on the same side with respect to each of the facets of σ. We need, therefore, only show that σ lies entirely on the boundary of its convex hull, or equivalently that T coincides with its convex hull. The latter can be easily established as follows. Suppose, for contradiction, that T and its convex hull do not coincide. Then there exists an edge of the convex hull of T that lies in the complement of T; let u and v be the vertices of T incident on this edge. Consider the plane that is normal to Π and goes through u and v; we denote it by E. Since σ is simple and none of its edges is a reflex edge of P, the intersection of E and σ is a connected convex chain that goes through u and v. Thus, the intersection of E and T is a convex polygon with u and v as vertices; it therefore contains the line segment connecting u and v, which contradicts the assumption that the edge (of the convex hull of T) connecting u and v lies in the complement of T. □

Finally, we introduce the notion of extrema. A point p of a d-dimensional set S of points is called an *extremum* of S with respect to an oriented line λ, or a λ-*extremum* for short, if S's intersection with a small enough d-ball centered at p lies entirely in one of the two closed half-spaces defined by the hyperplane normal to λ that passes through p. The extrema can be characterized as *negative* or *positive* depending on whether the above intersection lies in the nonnegative or nonpositive half-space, respectively. For the polygon of Figure 5, for instance, the vertices a, b, c, and d are negative λ-extrema,

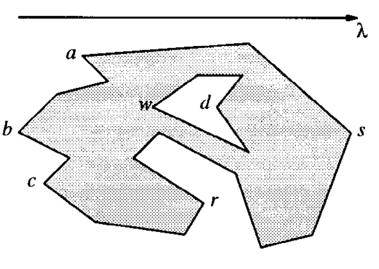

Fig. 5

while r and s are positive ones. The vertex w is *not* a λ-extremum (in fact, it is not a λ'-extremum for any λ'). Clearly, if no edge of a polygon or polyhedron is normal to λ, only vertices can be extrema.

Very often in the following, we consider the intersection of the polyhedron with a plane normal to the x-axis. The intersection consists of several polygons (possibly with holes), and it is referred to as a *yz-cross-section* of the polyhedron.

3. The Decomposition Algorithm. Our goal is to subdivide the boundary of a nonconvex polyhedron P of n vertices and r reflex edges into $O(r)$ connected convex-like patches. The polyhedron is given in any one of the standard representations, e.g., winged-edge [3], doubly-connected-edge-list [12], quad-edge [8], so that all the face incidences either are explicitly stored or can be found in linear time. To simplify the description of the algorithm, we assume that no facet of P is perpendicular to the z-axis, and no edge is normal to the x-axis. These assumptions are not restrictive; they can be checked in linear time, and, if necessary, enforced by rotating the system of reference. We also assume that the boundary of the polyhedron P is triangulated. Boundary triangulation can be achieved in $O(n + r \log r)$ time by employing the polygon triangulation algorithm of Hertel and Mehlhorn [9] on each nontriangular facet of P.

The algorithm consists of two phases; in either phase, patches are split into smaller pieces, starting with the entire boundary of P, which is the initial patch we work on. In the first phase we disassemble the boundary of P into a number of patches by cutting along some of the edges of P; as a result, we get patches whose internal edges are not reflex edges of P, and whose intersection with any plane normal to the x-axis is a collection of chains monotone with respect to the z-axis. In the second phase we further split these patches by clipping them with planes normal to the x-axis that go through the reflex vertices of P. This guarantees that the patches that are finally produced are simple, monotone with respect to the xz-plane, and their projections onto this plane are convex polygons; they are therefore convex-like, by virtue of Lemma 2.1.

Note that throughout the algorithm the patches are orientable; they are subsets of the boundary of the polyhedron, a 2-manifold without boundary, which is orientable.

3.1. *The First Phase.* As mentioned earlier, no internal edge of a convex-like patch can be a reflex edges of P. Therefore, we need to cut along each reflex edge, where cutting along an edge has the effect that its incident facets are no longer considered adjacent. (Note, however, that the internal edges of a patch can all be nonreflex edges of P, and still the patch may not be convex-like. Think of a patch spiraling around several times.) We may then be tempted to embark on the second phase of the algorithm and clip the resulting patches as outlined in the previous paragraph. This, however, will not necessarily produce the desired partition. Consider, for instance, the polyhedron of Figure 6, which is constructed by gluing two tetrahedra along a common facet uvw, where u and v are the vertices with the smallest and largest x-coordinates. Cutting along the reflex edges introduces a cut along the single reflex edge uv, while clipping with planes normal to the x-axis that go through the reflex vertices u and v leads to no additional cut; the boundary of the polyhedron will therefore still form a connected patch, which clearly is not convex-like.

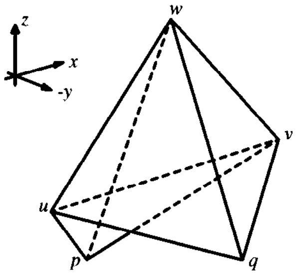

Fig. 6

To rule out such cases, we disassemble the boundary of the polyhedron P along the z-extrema of the yz-cross-sections of P. Since no edge of P is normal to the x-axis and no facet of P is normal to the z-axis, the z-extrema in any yz-cross-section are vertices, the intersections of edges of P with the slicing plane that defines the cross-section. (Figure 14 depicts a typical yz-cross-section of the polyhedron after all the cuts have been introduced.) For the polyhedron of Figure 6, this approach will produce cuts along the edges uw and wv, up and pv, and uq and qv, which in addition to the cut along the reflex edge uv will disassemble the polyhedron's boundary into four patches.

Summarizing, in this phase we cut along the reflex edges of the polyhedron P, and along the edges that contribute z-extrema in a yz-cross-section of P. It is important to observe that determining whether an edge is a reflex edge of P or whether it contributes z-extrema in a yz-cross-section requires only information local to the edge, namely the relative position of its two incident facets. We therefore process the edges in any order; if we need to cut along an edge e we simply mark e as "cut," and from then on we consider that e's incident facets are no longer adjacent. This procedure ensures that any patch σ in the partition of the boundary of P induced by the generated cuts is such that no internal edge of σ is a reflex edge of P, while σ's intersection with any plane normal to the x-axis consists of a number of z-monotone polygonal lines. The entire phase takes time linear in the number of edges, while no more than linear space is needed.

Before estimating the total number of patches produced at the end of this phase, we prove the following lemma.

LEMMA 3.1. *Let σ be a patch at the end of the first phase, whose projection onto the xz-plane is a polygon which we denote by σ_{xz}. Then, if v is a nonreflex vertex of the polyhedron that projects into a point q on the boundary of σ_{xz}, q is not a cusp of σ_{xz}, i.e., q is a vertex of σ_{xz} such that the* (*interior*) *angle formed by the two edges incident on q does not exceed π.*

PROOF. Since v is a nonreflex vertex of the polyhedron, the intersection N_v of the polyhedron with a small enough ball centered at v is a convex object. It is clear that the projection of a convex object onto a plane is also convex; therefore, the projection of N_v onto the xz-plane is a convex set containing q, which implies that q cannot be a cusp. □

Number of Patches Produced. We prove next that the number of patches that are produced at the end of this phase is no more than $2r + 2$. The key observation is that each patch has at least one vertex such that its adjacent vertices that belong to the patch all have larger x-coordinates; the vertex of the patch with the smallest x-coordinate will do, if we take into account the assumption that no edge of the polyhedron is parallel to the yz-plane. In fact, all the negative x-extrema of the patch will do. Similarly, each patch has at least one vertex whose adjacent vertices that belong to the patch all have smaller x-coordinates; any positive x-extremum of the patch satisfies this condition. Therefore, if we compute the number of patches to which a vertex of the polyhedron contributes positive or negative x-extrema, add these numbers over all vertices, and divide by two, we obtain an upper bound in the number of patches that are produced.

The way to compute the number of patches to which a vertex v contributes positive or negative x-extrema can be more easily understood if we consider yz-cross-sections of the polyhedron. First, we consider the yz-cross-section $\mathcal{C}_{yz}(v)$ of the polyhedron at v; recall that $\mathcal{C}_{yz}(v)$ is the intersection of the polyhedron with a slicing plane that is normal to the x-axis and goes through v. In $\mathcal{C}_{yz}(v)$, v is incident on a number of line segments, each being the intersection of the slicing plane with a single polyhedron's facet incident on v (recall the assumption that no edge of the polyhedron is normal to the x-axis); therefore, these line segments correspond to a collection $\mathcal{F}$ of facets of the polyhedron and ultimately to a collection $\mathcal{P}$ of patches that contain these facets. Next, we consider the yz-cross-section $\mathcal{C}^+_{yz}(v)$ infinitesimally away from v toward increasing x-coordinates, and we concentrate on the line segments in this cross-section that correspond to facets incident upon v; if these facets form the set $\mathcal{F}^+$, then $\mathcal{F}^+$ contains $\mathcal{F}$ due to the continuity of the polyhedron's boundary and the assumption that no edge is normal to the x-axis. Let $\mathcal{P}^+$ denote the set of patches that contain the facets in $\mathcal{F}^+$. Then the patches in $\mathcal{P}^+ \backslash \mathcal{P}$ are precisely the patches to which v contributes negative x-extrema; clearly, the number of such patches does not exceed the number of disconnected polygonal chains in $\mathcal{C}^+_{yz}(v)$ (at the end of the first phase) to which the line segments corresponding to the facets in $\mathcal{F}^+ \backslash \mathcal{F}$ belong. For example, in each of the four cases shown in Figure 7, the left-hand side depicts a neighborhood N_v of a vertex v at the cross-section $\mathcal{C}_{yz}(v)$, while the right-hand side depicts what N_r evolves into in C^+_{yz}; the shaded area indicates the interior of the

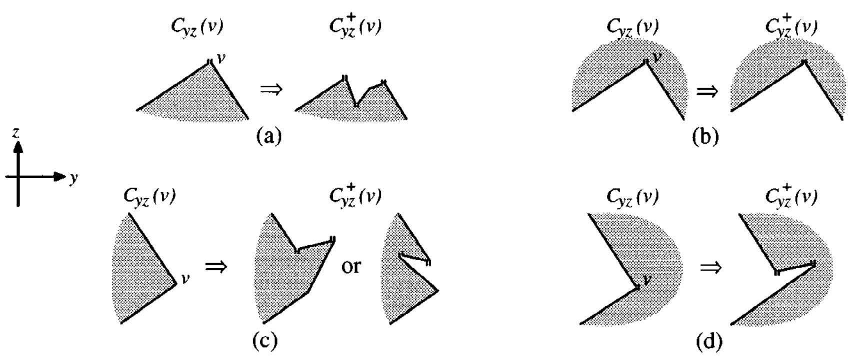

Fig. 7. (a) at most two "new" patches; (b) zero "new" patches; (c), (d) at most one "new" patch.

polyhedron, and the partition of the polyhedron's boundary into patches is exhibited by the openings of the boundary of the shaded area. In Figure 7(a) in particular, the three new line segments that appear in $\mathcal{C}^+_{yz}(v)$ form two disconnected chains, which correspond to at most two patches; therefore, in this case v contributes negative x-extrema to at most two patches. A similar argument applies to the yz-cross-section $\mathcal{C}^-_{yz}(v)$ infinitesimally away from v toward decreasing x-coordinates: the number of patches to which v contributes positive x-extrema does not exceed the number of disconnected polygonal chains in $\mathcal{C}^-_{yz}(v)$ (at the end of the first phase) that correspond to facets incident on v excluding those in $\mathcal{F}$.

It should be expected that the number of patches to which a vertex v contributes positive or negative x-extrema depends on the number r_v of reflex edges incident upon v. More specifically, if the number of reflex edges connecting v to vertices with smaller (resp. larger) x-coordinates is denoted by r_v^- (resp. r_v^+), the number of patches to which v contributes negative x-extrema should depend on r_v^-, while the number of patches to which v contributes positive x-extrema should depend on r_v^+. Figure 7 provides some intuition, by depicting details of $\mathcal{C}_{yz}(v)$ and $\mathcal{C}^+_{yz}(v)$ for a vertex v with $r_v^+ = 1$; the single reflex vertex at the right-hand side figure in each of the four cases shown is the intersection of the reflex edge with the slicing plane. It turns out that one reflex edge may account for no more than two patches having v as an x-extremum, so that a vertex v incident upon r_v reflex edges contributes (positive or negative) x-extrema to at most $2r_v + c$ patches, for some appropriate integer c that depends on the geometry of the neighborhood of v in $\mathcal{C}_{yz}(v)$. We consider the following cases:

1. *The vertex v is of degree* 0 *in $\mathcal{C}_{yz}(v)$ and is a point-polygon of $\mathcal{C}_{yz}(v)$*. In agreement with the definition of a polyhedron, v is an x-extremum of the polyhedron. If v is a negative x-extremum, then, as Figure 8 suggests, it may contribute negative x-extrema to no more than $2r_v^+ + 2$ patches. Moreover, the definition of a negative x-extremum of a polyhedron implies that v cannot possibly contribute positive x-extrema to any patch, as well as that $r_v = r_v^+$. Hence, v contributes x-extrema to at most $2r_v + 2$ patches. In a similar fashion we find that, if v is a positive x-extremum of the polyhedron, it contributes x-extrema to at most $2r_v + 2$ patches as well (the picture in this case is the same as Figure 8 where the pictures for $\mathcal{C}^+_{yz}(v)$ and $\mathcal{C}^-_{yz}(v)$ have been interchanged).
2. *The vertex v is of degree* 0 *in $\mathcal{C}_{yz}(v)$ and is a point-hole in a polygon of $\mathcal{C}_{yz}(v)$*. Figure 9 depicts one of the two basic cases that may arise; the other one stems from Figure 9 after the pictures for $\mathcal{C}^+_{yz}(v)$ and $\mathcal{C}^-_{yz}(v)$ have been interchanged. In either case, $r_v \geq 3$, and v contributes x-extrema to no more than $2r_v - 2$ patches.
3. *The vertex v is of degree* 2 *in $\mathcal{C}_{yz}(v)$*. We consider $\mathcal{C}^+_{yz}(v)$ first. If $r_v^+ = 0$, the two-edge

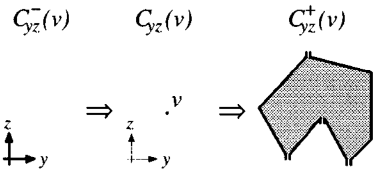

Fig. 8

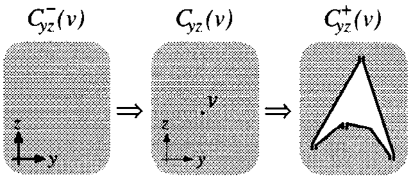

Fig. 9

chain incident on v in $\mathcal{C}_{yz}(v)$ (this is the intersection of the polyhedron's boundary incident on v with the slicing plane) evolves into a convex chain in $\mathcal{C}^+_{yz}(v)$, which implies that no new patches are intersected by the slicing plane in $\mathcal{C}^+_{yz}(v)$. Thus, no patch has v as a negative x-extremum. If, however, r^+_v is positive, the number of such patches does not exceed $2r^+_v$. (Figure 10 depicts the four basic cases for $r^+_v = 2$, where the two cusps at the right-hand side figure of each case are the intersections of the two reflex edges with the slicing plane; compare Figures 7 and 10 to see how cases with larger r^+_v can be produced. In general, for the cases (a)–(d), the bounds are $2r^+_v$, $2r^+_v - 2$, $2r^+_v - 1$, and $2r^+_v - 1$, respectively.) So, for all r^+_v, the number of patches to which v contributes negative x-extrema does not exceed $2r^+_v$.

The above arguments apply without change when we consider $\mathcal{C}^-_{yz}(v)$ implying that the number of patches to which v contributes positive x-extrema is at most $2r^-_v$. The combination of these results yields an upper bound of $2r^+_v + 2r^-_v - 2r_v$ in the number of patches to which v contributes x-extrema.

4. *The vertex v is of degree larger than* 2 *in* $\mathcal{C}_{yz}(v)$. In this case the neighborhood of v in $\mathcal{C}_{yz}(v)$ consists of a number of wedges touching at v; let this number be w_v ($w_v = 4$ in Figure 11). (Note that since the degree of v is larger than 2 in $\mathcal{C}_{yz}(v)$, w_v is always larger than 1.) In general, the situation is a combination of the two cases shown in Figure 12, as well as those obtained from Figure 12 with the pictures corresponding to $\mathcal{C}^-_{yz}(v)$ and $\mathcal{C}^+_{yz}(v)$ interchanged (the dashed curves indicate that some of the wedges may belong to the same polygon of $\mathcal{C}_{yz}(v)$). These wedges may either merge with or get detached from neighboring wedges in $\mathcal{C}^+_{yz}(v)$ and $\mathcal{C}^-_{yz}(v)$. In Figure 11 for instance,

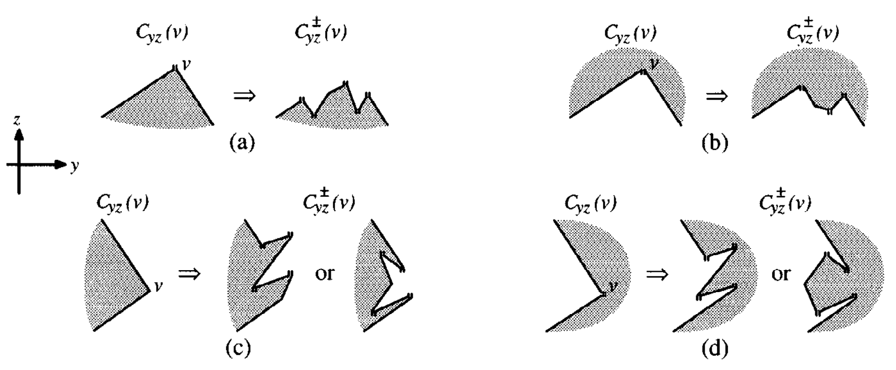

Fig. 10. At most four, two, three and three "new" patches in the cases (a)–(d), respectively, where $r^+_v = 2$.

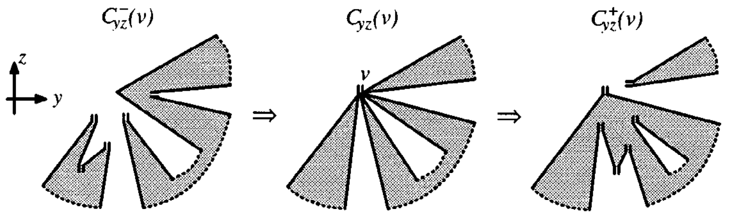

Fig. 11

as $\mathcal{C}_{yz}(v)$ evolves into $\mathcal{C}^+_{yz}(v)$, the three bottom wedges merge into a single wedge cluster, while the top wedge forms a wedge cluster by itself. Let w^+_v and k^+_v (resp. w^-_v and k^-_v) denote the number of such wedge clusters and the number of polygons containing these clusters in $\mathcal{C}^+_{yz}(v)$ (resp. $\mathcal{C}^-_{yz}(v)$) respectively (in Figure 11, we have $w^-_v = 3$, $w^+_v = 2$, and $k^-_v = k^+_v = 2$). Since several wedges may merge into a single wedge cluster and several wedge clusters may belong to the same polygon, then

$$\text{(1)} \qquad w_v \geq w^+_v \geq k^+_v \quad \text{and} \quad w_v \geq w^-_v \geq k^-_v .$$

Moreover, the definition of a polyhedron (see Section 2) implies that the intersection of a small enough neighborhood of v and the interior of the polyhedron P is a connected set; otherwise, we end up with degeneracies like the one exhibited at the object shown in Figure 3. Therefore, the sum of the numbers of wedges that merge in $\mathcal{C}^-_{yz}(v)$ and $\mathcal{C}^+_{yz}(v)$ must be at least equal to $w_v - 1$; in terms of w_v, w^-_v, and w^+_v, this can be expressed as follows:

$$\text{(2)} \qquad (w_v - w^-_v) + (w_v - w^+_v) \geq w_v - 1 \quad \Leftrightarrow \quad w^-_v + w^+_v \leq w_v + 1.$$

(In fact, it is true that $w^-_v + w^+_v = w_v + 1$; this can be proved if we take into account that the closure of the complement of a polyhedron is also a polyhedron.)

Finally, if v is incident upon w_v wedges in $\mathcal{C}_{yz}(v)$, then it is incident upon at least $w_v - 1$ reflex edges: recall that the neighborhood of v in $\mathcal{C}_{yz}(v)$ is a combination of the cases in Figure 12 (or their right-to-left counterparts); in either case shown no new line segments appear in $\mathcal{C}^-_{yz}(v)$ or $\mathcal{C}^+_{yz}(v)$, implying that v does not contribute x-extrema to any of the patches, and yet v is incident upon one reflex edge. Of course, it may be incident upon more such reflex edges, each of which leads to at most two patches to which v contributes an x-extremum, so that such a vertex v accounts for no more than $2(r_v - (w_v - 1))$ patches (see Figure 11).

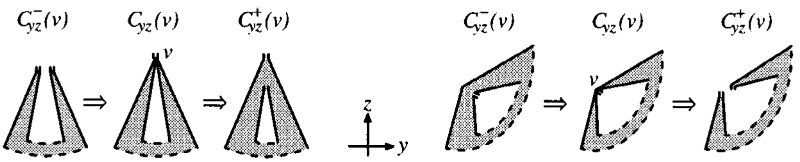

Fig. 12

If we sum all the contributions, we obtain the following bound in the number m of patches, where V_1, V_2, V_3, and V_4 denote the sets of vertices falling in cases 1, 2, 3, and 4, respectively.

$$
\begin{aligned}
(3)\quad m &\leq \tfrac{1}{2}\left(\sum_{v\in V_1}(2r_v+2)+\sum_{v\in V_2}(2r_v-2)+\sum_{v\in V_3}2r_v+\sum_{v\in V_4}2(r_v-(w_v-1))\right)\\
&\leq \sum_{v\in\cup V_i} r_v+|V_1|-\sum_{v\in V_4}(w_v-1)\\
&\leq 2r+|V_1|-\sum_{v\in V_4}(w_v-1),
\end{aligned}
$$

since $\sum_{v\in\cup V_i} r_v=\sum_v r_v=2r$. In light of the following lemma, inequality (3) implies that the number m of patches produced at the end of the first phase satisfies the inequality $m\leq 2r+2$.

LEMMA 3.2. *The total number $|V_1|$ of positive and negative x-extrema of a polyhedron does not exceed $\sum_{v\in V_4}(w_v-1)+2$, where V_1, V_4, and w_v are as defined in the previous paragraphs.*

PROOF. We construct a graph that records the events that mark the history of the yz-cross-section of the polyhedron. Namely, we sweep the polyhedron with a plane normal to the x-axis, and whenever the number of polygons in the cross-section changes (as is the case when positive or negative x-extrema of the polyhedron and vertices in V_4 are encountered), we record the change appropriately (for completeness, we make sure to record all the vertices v in V_4 even those for which $k_v^-=k_v^+$). Note that events where the number of holes in a polygon of a yz-cross-section increases or decreases are not recorded. In particular:

1. At a negative x-extremum (vertex in V_1), we add to the graph two new nodes[3] which we connect by an edge; the first node corresponds to the negative x-extremum, while the second one is a *polygon-node* and corresponds to the series of polygons in the yz-cross-sections to which the negative x-extremum evolves.
2. At a positive x-extremum (vertex in V_1), we add one new node that corresponds to the positive x-extremum and we connect it to the polygon-node that represents the series of polygons in the yz-cross-sections which reduced to this positive x-extremum during the sweeping.
3. At a vertex v in V_4, we add one new node that corresponds to v and edges connecting it to the representatives of the k_v^- polygons in $\mathcal{C}_{yz}^-(v)$. Moreover, k_v^+ polygon-nodes are added, one for each of the k_v^+ polygons in $\mathcal{C}_{yz}^+(v)$, and edges are introduced between them and the node corresponding to v.

Since we are dealing with a single polyhedron, the resulting graph is connected. Figure 13(b) shows the graph that corresponds to the polyhedron of Figure 13(a). The

[3] We use the term *nodes* of a graph instead of vertices to avoid confusion with the vertices of the polyhedron.

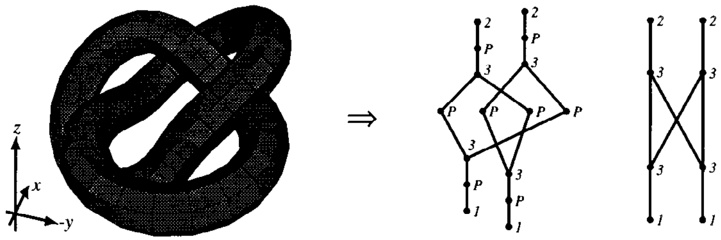

Fig. 13

numbers 1, 2, and 3 denote negative and positive x-extrema, and vertices in V_4, respectively (according to the cases above), while the letter P denotes polygon-nodes. Note that all polygon-nodes are of degree 2, and no two of them are adjacent, while the nodes corresponding to the negative and positive x-extrema of the polyhedron are of degree 1.

To simplify matters, we remove from the graph all polygon-nodes by coalescing their incident edges into a single edge; the resulting graph is a connected multigraph (see Figure 13(c)), whose node set is in one-to-one correspondence with the union of V_1 and V_4. If we denote by $\mathcal{E}$ the total number of edges of the multigraph, we have

$$(4) \qquad 2\mathcal{E} = \sum_{\text{node } v} \text{degree}(v) = \sum_{v \in V_1} 1 + \sum_{v \in V_4} (k_v^- + k_v^+) \le |V_1| + \sum_{v \in V_4} (w_v + 1),$$

since, for any vertex $v \in V_4$, inequalities (1) and (2) imply that $k_v^- + k_v^+ \le w_v^- + w_v^+ \le w_v + 1$. Connectivity, on the other hand, implies that the number of edges is at least equal to one less than the number of nodes of the graph, that is, $\mathcal{E} \ge |V_1| + |V_4| - 1$, which combined with (4) yields

$$2(|V_1| + |V_4| - 1) \ \le\ |V_1| + \sum_{v \in V_4} (w_v + 1) \quad \Rightarrow \quad |V_1| \le 2 + \sum_{v \in V_4} (w_v - 1). \qquad \square$$

3.2. *The Second Phase.* The first phase produces patches that are not necessarily simple and may form spirals around the y-axis. Moreover, although the intersection of such a patch with any plane normal to the x-axis consists of a number of z-monotone polygonal lines, this is not sufficient to ensure that the patch is monotone with respect to the xz-plane; patch monotonicity is guaranteed only if the patch is decomposed into subpatches so that each of these z-monotone polygonal lines belongs to a different subpatch. This will be our goal in this phase, i.e., to decompose each of the patches produced in the previous phase into subpatches, the intersection of each of which with any plane normal to the x-axis is a single z-monotone polygonal line. Then, making sure that the projections of the subpatches on the xz-plane are convex polygons implies that the final patches are convex-like (Lemma 2.1).

The method that we use in order to achieve the desired decomposition parallels the way a nonconvex polygon (that may contain holes) is partitioned into convex pieces by using cuts parallel to a chosen direction to resolve the polygon's cusps. In very general terms, our basic strategy involves splitting each patch by clipping it with planes normal

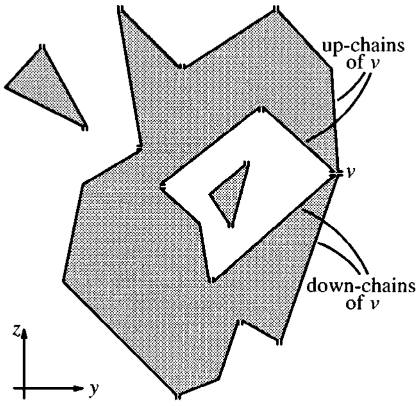

Fig. 14

to the x-axis that pass through the reflex vertices of the given polyhedron. Before giving more details, we introduce the notions of the up-chains and down-chains of a reflex vertex.

We consider the yz-cross-section of the polyhedron at a reflex vertex v after the end of the first phase. Since during the first phase we cut along all the reflex edges and v is incident upon a reflex edge, the boundary of the polyhedron has been cut at v. In general, due to the first phase cuts, the boundaries of the polygons of the cross-section are decomposed into z-monotone polygonal chains. Unless v is a point-hole in one of these polygons or a polygon reduced to a point, v is an endpoint of at least two such chains (Figure 14). If v is the endpoint of a chain with the smallest z-coordinate among the chain's vertices, we refer to the chain as an *up-chain* of v; if v is the endpoint of a chain with the largest z-coordinate among the chain's vertices, we refer to the chain as a *down-chain* of v.

Note that if w wedges touch at v in the corresponding yz-cross-section, then v is incident upon $2w$ up- or down-chains and at least $w - 1$ reflex edges. In other words, if v is incident upon r_v reflex edges, it is incident upon at most $2(r_v + 1)$ chains. Then, if the number of reflex edges of the polyhedron is r, the total number of reflex vertices does not exceed $2r$ and $\sum_v r_v = 2r$; therefore, the number of up- and down-chains of all reflex vertices is at most $2 \times (2r + 2r) = 8r$. This leads to the following lemma:

LEMMA 3.3. *If the number of reflex edges of a polyhedron is r, the total number of the up- and down-chains of all the reflex vertices of the polyhedron does not exceed $8r$.*

In terms of the up- and down-chains, our basic strategy can be expressed as follows: cut along the up- and down-chains of all the reflex vertices of the polyhedron. To simplify matters, we cut along the up- and down-chains in separate passes.

We concentrate on the first pass where we generate cuts along the up-chains of the reflex vertices. Observe that a brute-force approach may cut on the order of n facets every time we process a reflex vertex, which would produce a decomposition of an unacceptably large $\Omega(nr)$ size. To avoid that, we advance the cuts along the up-chains from facet to

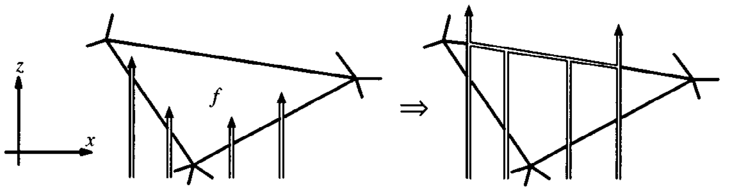

Fig. 15

facet and discontinue some of the cuts that might slice too many facets. The key idea is illustrated in Figure 15. (The facet shown is a triangle, because as previously mentioned the boundary of the polyhedron has been triangulated at a preprocessing step before the first phase of our algorithm.) Assume that we advance four cuts upward through facet f. The plan is to extend only the leftmost and rightmost ones through adjacent facets upward past f; we stop the remaining two cuts at the edge incident upon f with the largest x-extent, and we generate a cut (along that very edge) extending between the intersections of the edge with the rightmost and leftmost cuts. Thus, we maintain the following invariant:

CUT-INVARIANT. An edge of the polyhedron is crossed by at most two cuts which propagate through a facet to an adjacent one.

To enforce the invariant, we process a facet of a patch after its adjacent facet(s) below it (in terms of z-coordinates) in the patch have been processed. Note that, thanks to the first phase cuts along the z-extrema, every patch contains at least one facet without any adjacent facets below it (recall that if a cut has been generated along an edge during the first phase, the two facets incident upon the edge are no longer considered adjacent, although they may still belong to the same patch). Therefore, every patch contains at least one facet that can be processed right away. Figure 16 depicts a patch with these facets shown highlighted.

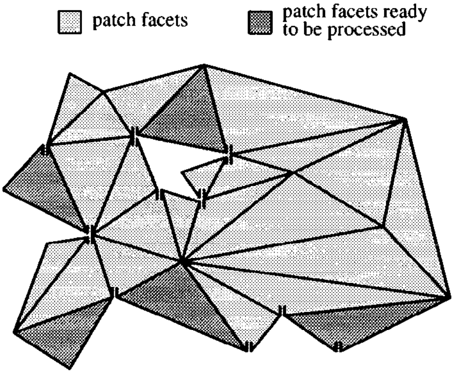

Fig. 16

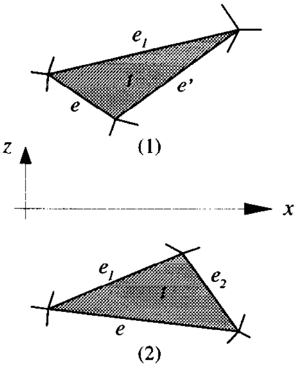

Fig. 17

Here is our method to cut along the up-chains of the reflex vertices in more detail. We process each patch in turn. We start by collecting the facets that can be processed immediately, and we insert them in a queue Q; the purpose of this queue is precisely always to store the facets of the patches that are ready to be processed. Initially, our cut-invariant holds trivially. Then, for each facet t in Q, we iterate on the following procedure: depending on whether t is as shown in Figure 17(1) or 17(2), we execute step 1 or step 2 respectively:

1. The triangular facet t is as shown in Figure 17(1), i.e., the edge e_1 incident upon t with the largest x-extent is "above" the other two edges e and e' incident upon t. Cuts may be propagating in t through the edges e and e', while an additional cut may emanate from the vertex incident to both e and e'. If the total number of cuts proceeding through t is no more than two, then the cuts are simply extended all the way through t and are attached to e_1 ready to advance upward to adjacent facets of the patch. If, however, the number of cuts is larger than two, we apply the idea illustrated in Figure 15. We extend only the leftmost and rightmost cuts all the way through t, and attach them to e_1. All remaining cuts are extended up to e_1 and are stopped there, while a cut is generated along e_1 between the intersection points of e_1 and the leftmost and rightmost cuts. So, our cut-invariant is maintained in this way. Finally, if no cut was made along e_1 during the first phase, we check whether the facet t_1, which in addition to t is incident upon e_1, is a candidate for the queue Q. Namely, if t_1 is as in Figure 18(a), then it is inserted in Q; otherwise, t_1 as in Figure 18(b) and is inserted in Q only if either a cut has been generated during the first phase along the edge $\hat{e}$, or the facet, which in addition to t_1 is incident upon $\hat{e}$, has already been processed.
2. The triangular facet t is as shown in Figure 17(2), i.e., the edge e incident upon t with the largest x-extent is "below" the other two edges e_1 and e_2 incident upon t. If cuts are propagating through e, we extend them through t and we attach them to e_1 or e_2 depending on which edge they intersect. By induction, our cut-invariant is maintained. Next, we test the facets t_1 and t_2, which in addition to t are incident upon e_1 and e_2, respectively, as candidates for the queue Q. (Note that either one or both t_1 and t_2 may not exist as cuts along e_1 or e_2 during the first phase may have

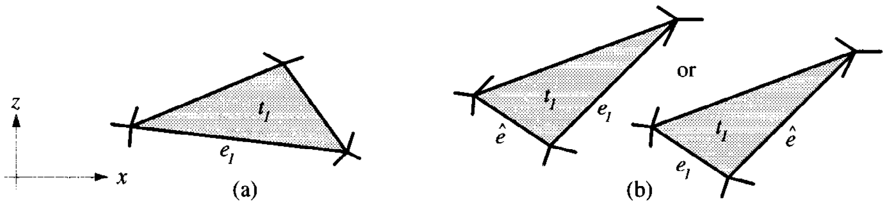

Fig. 18

disconnected the patch at these edges.) The procedure is the same as that involving t_1 in the previous case. Specifically, if t_1 is as shown in Figure 18(a), i.e., e_1 is the edge incident upon t_1 with the largest x-extent, we insert t_1 in Q; if not, in which case t_1 is as in Figure 18(b), we insert t_1 in Q only if either we have generated a cut along the edge $\hat{e}$ during the first phase, or the facet, which in addition to t_1 is incident upon $\hat{e}$, has already been processed. The same test is also applied to t_2, if the facet t_2 exists.

Then t is removed from Q and we proceed with the next facet in Q.

Figure 19 shows a snapshot as the procedure is applied to the patch of Figure 16. It is easy to see that eventually all the facets of the patch enter the queue Q, so that when Q finally empties, cuts along all the up-chains of the patch's reflex vertices have been created. Furthermore, our cut-invariant ensures that at most two cuts are crossing an edge cutting both its incident facets.

This completes the first pass that generates cuts along the up-chains. Next we apply the same procedure with respect to the down-chains. Note that we take into consideration the cuts generated during the previous pass involving the up-chains. Consider, for instance, Figure 20. If a cut c_1 along a down-chain reaches an edge along which we generated a cut during the previous pass, then the cut is stopped there and its processing is considered completed. Moreover, if two cuts c_2 and c_3 along two down-chains reach an edge e that has been cut by an up-chain cut c' and c_2 is between c' and c_3 (with respect to the x-axis), then c_2 is stopped there, and a cut between the intersections of e with c' and c_3 is generated along e. This establishes our cut-invariant for cuts along down-chains.

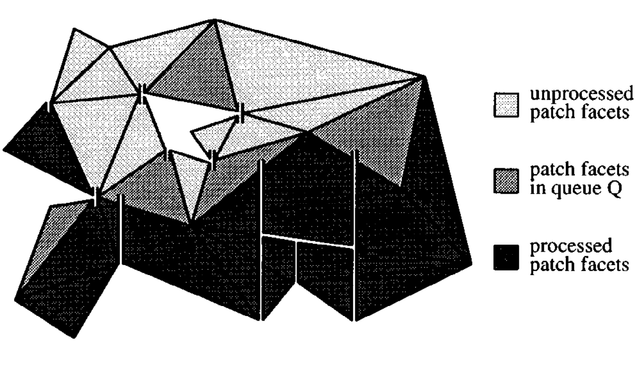

Fig. 19

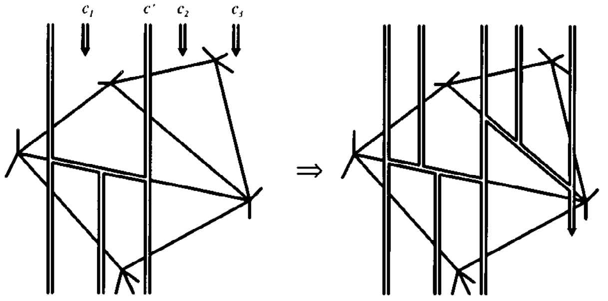

Fig. 20

At the end, pieces of a single facet of the polyhedron may belong to several different patches. Interestingly, however, due to our cut-invariant, no more than four cuts through any given facet proceed to adjacent facets both up and down. These would be the leftmost and rightmost cuts through the facet along both up- and down-chains. Any other cuts stop at the edge with the largest x-extent incident upon the facet. The total time spent in this phase is linear in the number of facets of the polyhedron plus a constant overhead per segment of each cut. As the total number of these segments is proven in Section 3.3 to be linear in the size of polyhedron, so is the total time required for this second phase.

3.3. *Description of the Patches Produced.* Each of the patches that the algorithm produces consists of a portion of the boundary of the polyhedron that is clipped from left and right by two planes normal to the x-axis (cuts along up- or down-chains), and at the top and bottom by either an edge of the polyhedron (a reflex edge, for instance), or a polygonal line consisting of edges that contribute z-extrema in yz-cross-sections of the polyhedron. The following lemma helps us establish that these patches are simple and monotone with respect to the xz-plane.

LEMMA 3.4. *The intersection of any patch after the end of the second phase with a plane normal to the x-axis is a single polygonal line monotone with respect to the z-axis.*

PROOF. Since the intersection of any patch that results from the first phase with any plane normal to the x-axis consists of one or more z-monotone polygonal lines, then so does any patch after the end of the second phase; recall that in the second phase we simply clip the first phase patches.

Suppose, for contradiction, that a plane normal to the x-axis intersects a patch after the end of the second phase into more than one such polygonal line. Then, since the patch is connected, there exists a connected subset σ of the patch with the following property: there exist points p and q of σ that have equal x-coordinates, and are such that a plane normal to the x-axis passing through them intersects σ in a single polygonal line, whereas it intersects σ into two disconnected polygonal lines if it is translated slightly either to the left or to the right along the x-axis (to the right, in the case shown in Figure 21). (Note that the points p and q may coincide.) There exist therefore line segments e_1 and e_2 incident upon p and q, respectively, along which cuts have been generated (e_1 and e_2 may coincide if p and q coincide).

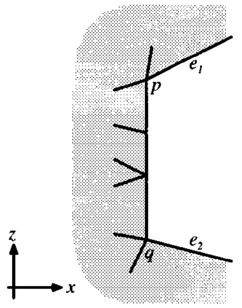

Fig. 21

These cuts do not proceed along up- or down-chains of vertices, and thus e_1 and e_2 lie on edges of the polyhedron. Moreover, as the projections of patches onto the xz-plane exhibit internal angles at most equal to π at the projections of new vertices introduced in the second phase because of the clipping (see Figure 20), p and q must be vertices of the polyhedron. In fact, neither p nor q may be reflex vertices, otherwise a cut along the up-chain of p or the down-chain of q would have split σ. Therefore, both p and q are nonreflex vertices, and e_1 and e_2 contribute z-extrema at the corresponding yz-cross-sections. This is impossible, however, because the continuity of the polyhedron's boundary implies that there would have been an edge e incident on p such that e and e_1 would form an x-monotone polygonal line and e, like e_1, would contribute z-extrema at the corresponding yz-cross-sections; but then a cut would have been generated along e, and σ would have been split. □

Lemma 3.4 directly implies that any patch at the end of the second phase is monotone with respect to the xz-plane. Moreover, it implies that any such patch is also simple; if the patch had a hole, the intersection of the patch with a plane normal to the x-axis that intersects the hole would either be nonmonotone with respect to the z-axis or consist of at least two disconnected pieces. Therefore,

COROLLARY 3.1. *Any patch after the end of the second phase is simple and monotone with respect to the xz-plane.*

In light of Lemma 2.1, Corollary 3.1 and the following lemma establish that the patches produced are convex-like.

LEMMA 3.5. *The projection of any patch on the xz-plane is a convex polygon.*

PROOF. Since the patch is simple, its projection on the xz-plane is a simple polygon too. We now prove that the projection is indeed a convex polygon. Suppose, for contradiction, that there exists a cusp, i.e., a vertex of the polygon such that the (interior) angle formed by its two incident edges exceeds π. The monotonicity of the patch with respect to the xz-plane implies that the cusp is the projection of some vertex v on the boundary of the patch. Since the second phase cuts contribute vertices that are intersections of edges of the polyhedron with the "clipping" planes, no such vertex can project into a cusp on the xz-plane. This implies that v lies on a first phase cut, and in fact it is a nonreflex vertex of the polyhedron. However, then it cannot project into a cusp, according to Lemma 3.1, which leads to contradiction. □

Description Size and Total Number of Patches. Consider a polyhedron P of f facets and e edges, r of which are reflex. We compute the total number of edges of all the patches, where the edges along the cuts are counted twice. The analysis proceeds in an incremental way by taking into account the new edges that each step of our algorithm introduces. The triangulation of the boundary of P does not affect the order of magnitude of the number of edges, so that this number is $O(e)$ before the beginning of the first phase. As the cuts of the first phase proceed along edges of P, the number of edges of all the patches at the end of the phase is at most twice their number after the boundary triangulation. During the second phase, several new edges are introduced in the following three ways: (i) an edge is split into two when a cut crosses it, (ii) a new edge is introduced by cutting through a facet, or (iii) a new edge is introduced along a portion of an edge that is about to be crossed by more than two cuts. We claim that the total number of these new edges does not exceed $16f + 40r$; the claim implies that the total number of edges of all the patches after the end of the second phase is $O(e) + 16f + 40r$, which is linear in the size of the input.

To prove our claim, we charge the new edges created in the second phase to the facets of the polyhedron and the cuts generated in the second phase as follows: each facet is charged with the number of new edges that result from cuts traversing it and advancing to adjacent facets, while each cut is charged with the number of new edges that its traversal through the very last facet causes; in this way, all the new edges are accounted for. We observe that a cut that traverses a facet and advances to adjacent ones leads to the creation of four new edges, some of them shared with other facets; see Figure 22(a). Since at most two such cuts are allowed per facet for each of the two passes of the second phase (cut-invariant), each facet is charged with $4 \times 2 \times 2 = 16$ units. In a similar fashion, each cut leads to the creation of four or five new edges in the facet that the cut traverses last; the latter case corresponds to a cut that is stopped while enforcing the cut-invariant at a facet that is traversed by more than two cuts (see edge pq in Figure 22(b)). Since the cuts of the second phase are generated along up- or down-chains and the total number of such chains does not exceed $8r$ (see Lemma 3.3), the total number of new edges in the second phase does not exceed $16f + 5 \times 8r$ as claimed.

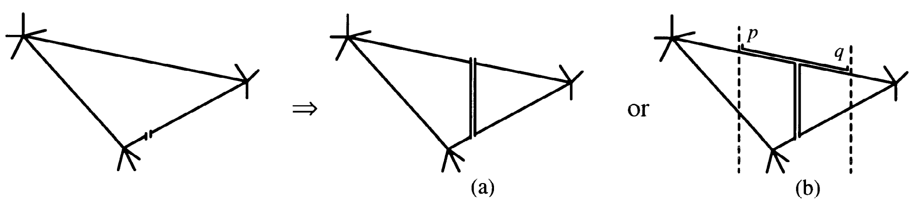

Fig. 22

Next, we estimate the number of patches that are finally produced. As mentioned already, the first phase produces no more than $2r + 2$ patches. We partition the set of cuts generated during the second phase into two classes:

(i) those that extend all the way to the boundary of the corresponding patch produced during the first phase, and
(ii) the remaining ones, that is, those that were stopped at an edge that had been cut at least twice (Figure 22(b)).

Each cut in class (i) increases the number of patches by at most one, whereas each cut in class (ii) increases the number of patches by at most two (see Figure 22). As the total number of cuts is no more than $8r$, and the number of cuts in class (ii) is bounded above by $8r-4$, the total number of patches produced cannot exceed $2r+2+8r+8r-4 = 18r-2$.

4. Conclusions.

Our results are summarized in the following theorem.

THEOREM. *The boundary of a nonconvex polyhedron of n vertices and r reflex edges can be subdivided into $18r - 2$ patches, each of which lies on the boundary of its convex hull. The decomposition can be carried out in $O(n + r \log r)$ time and $O(n)$ space.*

Unfortunately, the cuts performed may pass through facets of P; this has the disadvantage of introducing new vertices into the resulting decomposition. It would be of interest, instead, to achieve a boundary decomposition into a small number of convex-like pieces by means of cuts along edges of the given polyhedron only.

A different question is to find an algorithm that produces the minimum number of convex-like pieces. Is this problem NP-complete, as are many optimization questions in partitions and coverings [11], [13]?

References

[1] E. Arkin, R. Connelly, and J. S. B. Mitchell, On Monotone Paths Among Obstacles, with Applications to Planning Assemblies, *Proc. 5th Annual ACM Symposium on Computational Geometry* (1989) pp. 334–343.
[2] C. L. Bajaj and T. K. Dey, Convex Decompositions of Polyhedra and Robustness, *SIAM Journal on Computing* 21 (1992), 339–364.
[3] B. G. Baumgart, A Polyhedron Representation for Computer Vision, *Proc.* 1975 *National Computer Conference*, AFIPS Conference Proceedings, Vol. 44 (1975), AFIPS Press, Montvale, NJ, pp. 589–596.
[4] B. Chazelle, Convex Partitions of Polyhedra: A Lower Bound and Worst Cast Optimal Algorithm, *SIAM Journal on Computing* 13 (1984), 488–507.
[5] B. Chazelle and L. J. Guibas, Visibility and Intersection Problems in Plane Geometry, *Discrete and Computational Geometry* 4 (1989), 551–581.
[6] B. Chazelle and L. Palios, Triangulating a Nonconvex Polytope, *Discrete and Computational Geometry* 5 (1990), 505–526.
[7] L. J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan, Linear Time Algorithms for Visibility and Shortest Path Problems Inside Triangulated Simple Polygons, *Algorithmica* 2 (1987), 209–233.
[8] 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] S. Hertel and K. Mehlhorn, *Fast Triangulation of a Simple Polygon*, Lecture Notes on Computer Science, Vol. 158 (1983), Springer-Verlag, Berlin, pp. 207–218.

[10] D. G. Kirkpatrick, Optimal Search in Planar Subdivisions, *SIAM Journal on Computing* 12 (1983), 28–35.

[11] A. Lingas, *The Power of Non-Rectilinear Holes*, Lecture Notes in Computer Science, Vol. 140 (1982), Springer-Verlag, Berlin, pp. 369–383.

[12] D. E. Muller and F. P. Preparata, Finding the Intersection of two Convex Polyhedra, *Theoretical Computer Science* 7 (1978), 217–236.

[13] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, Oxford (1987).

[14] J. Ruppert and R. Seidel, On the Difficulty of Tetrahedralizing 3-Dimensional Non-Convex Polyhedra, *Proc*. 5*th Annual ACM Symposium on Computational Geometry* (1989), pp. 380–392.

[15] S. Suri, A Linear Time Algorithm for Minimum Link Paths Inside a Simple Polygon, *Computer Vision*, *Graphics*, *and Image Processing* 35 (1986), 99–110.

[16] G. T. Toussaint, On Separating Two Simple Polygons by a Single Translation, *Discrete and Computational Geometry* 4 (1989), 265–278.