Constructing Convex Hulls of Quadratic Surface Patches

CHAO-KUEI HUNG AND DOUG IERARDI

August 12, 1994

ABSTRACT. We study geometric duality and use modified definitions to obtain good complexity upper bounds on and algorithms dealing with the convex hull of piecewise smooth manifolds in $E^3$. Specifically, we show that the convex hull of a collection of $N$ smooth algebraic surface patches of bounded degree with bounded number of similarly constrained surfaces in $E^3$ has complexity $O(N^2\epsilon)$ for all $\epsilon > 0$ and can be constructed in randomized expected time of the same complexity. The dual construction also produces an $O(N \log N)$ algorithm for constructing the convex hull of an unordered collection of $N$ algebraic curve segments and points in $E^2$.

From a practical point of view, the duality arguments enable us to give a simple characterization of the convex hull of quadratic surface patches bordered by (an arbitrary number of) segments of conic sections. In this case, it is shown how to obtain closed form parameterizations of the new surfaces which arise in the construction of the convex hull. Algebraic overhead is reasonable, involving only numerical computations, and existing simpler algorithms for CSG (Constructive Solid Geometry) intersection may be employed in place of the asymptotically efficient yet more complicated algorithm. Point classification against the hull may be performed in time $O(N \log N \log^* N)$ using a simple algorithm without explicit construction of the hull.

1. Introduction

Computational geometers have exhaustively studied the problem of constructing convex hulls of discrete point sets in various dimensions. Indeed, optimal algorithms for this problem are now known for arbitrary dimension. [8] Recently, however, there has also been growing interest in the design of efficient and practical algorithms for computing convex hulls of polyhedra, and of curved objects. Our interest in the
construction of convex hulls for piecewise smooth objects — objects with curved faces and edges — arises from several geometric problems related to manufacturing processes, initially by efforts to derive efficient manipulation strategies for rigid objects using simple robotic effectors. For example, one may wish to compute the possible stable poses of an object resting on a flat surface, when given a description of its geometry and center of mass (COM). A simple walk over the surface of the convex hull — comparing each face to the relative location of the COM — suffices to compute this information, once the convex hull is constructed. Similarly, information concerning stable grasps of such an object by a simple manipulator, such as the industrial parallel-jaw gripper, is easily recovered from a description of the convex hull. In this setting, it is more the exception than the rule that a part is reliably modeled by a polyhedron. Hence, in these cases, one would like an algorithm to directly and efficiently construct the convex hull of the given piecewise smooth object.

There has been some work on the construction of convex hulls for curved objects in recent years. For example, [32, 2] give asymptotically optimal algorithms for the case of piecewise smooth Jordan curves and regions bounded by algebraic curves in the plane, respectively. The special case of "splinegons" was studied in [12]. For dimension three, Kriegman et al. studied the related problem of computing all stable poses of a single bounded algebraic surface; although the complexity of this algorithm was not analyzed, an implementation is described in [22].

One common approach to constructing convex hulls of discrete point sets proceeds by means of duality. It is well known that if \( K \subseteq E^d \) is an arbitrary set of points, and the origin is contained within the convex hull of \( K \), then the polar set of \( K \) is a convex set, and the double dual of \( K \) — that is, the dual of the dual of \( K \) — is in fact the smallest convex set containing \( K \). [24] One consequence of this observation is that we may construct the convex hull of a finite set of points by first constructing the dual half-space of each point, then computing the polytope that is the intersection of these halfspaces, and finally "inverting" the operation (or dualizing again) to recover the convex hull in the original space. This effectively reduces the problem of constructing convex hulls to one of intersecting half-spaces or, equivalently, of constructing a single cell in an arrangement of hyperplanes. This approach has been exploited in certain randomized incremental algorithms for constructing the convex hulls of finite point sets, as presented in [29] for example.

In this paper, we employ a similar approach for the construction of the convex hull of an arbitrary smooth object in \( E^d \). To do this, we carefully study one form classical geometric duality, the polar set. By constructing a related dual form which acts upon individual faces of the object and is therefore easier to compute, we arrive at a natural algorithm for the construction of the convex hulls of piecewise smooth objects. As in the case of discrete point sets, the problem will be reduced to the construction of a single convex cell in an arrangement of hypersurfaces, and generalizes to arbitrary dimension. For dimension three, we present an asymptotically efficient algorithm that constructs the convex hull of a collection of \( N \) smooth algebraic surface patches with bounded number of subfaces in randomized expected
time $O(N^{2+\varepsilon})$. Since the convex hull of a set of $N$ spheres in $E^3$ can have combinatorial complexity at least $\Omega(N^2)$, this algorithm is in fact nearly optimal [4]. Similarly, optimal algorithms for the related problems in $E^2$ follow from their rectilinear counterpart naturally once we establish the duality results.

However, this approach is not merely of theoretical interest. For objects with surface patches constructed from quadratic surfaces (or conicoids) — which include cones, ellipsoids, and cylinders — bordered by segments of conic sections, the dual surface patches turn out to be quadratic also. The dualizing transform can be computed symbolically once and for all, after which duals can be computed solely by numerical evaluation of certain multivariate polynomials. Hence, existing algorithms used to intersect the primal objects may be utilized to manipulate their duals. Moreover, once the convex hull is constructed, each of its new surfaces has a closed form parametrization. In practice existing or simpler algorithms can be used in place of the above near optimal algorithm for constructions in the dual space.

Finally, for the case of surface patches which arise from conicoids we also consider the simpler problem of classifying a point against their convex hull — that is, the problem of determining whether the point lies inside or outside of the convex hull. For this case, we present a simpler algorithm which uses time $O(N \log N \log^* N)$ without ever constructing this hull.

2. Object Model

Recent research in computational geometry and solid modeling for curves and surfaces seems not yet to have reached a consensus in terminology. Some focus on the algebraic aspects of the problems and deal with a single algebraic surface, or on planar regions bounded by a single algebraic curve, [23, 27, 31] Others deal with arbitrary collections of curve segments in $E^2$ or surface patches and curve segments in $E^3$, all of which may intersect in an almost arbitrary way such as in [11] and [34], and focus more on the combinatorial aspects of the problem. Still others address problems where strong topological constraints on these objects need be enforced, as in the case of the splinesegons of [12] and the piecewise-smooth Jordan curves of [32]. For a more comprehensive discussion on this subject, see Brisson’s article [5].

We adopt the generally agreed-upon view point that the issues at hand are better studied by separating them into the topological and geometrical aspects. To borrow as much as possible from existing literature, we begin with a topological polyhedron (see [7]). Objects under our consideration are their “smooth realizations” in $E^d$. More precisely, we consider homeomorphic images of polyhedral complexes in $E^d$ into $E^d$ where in addition the homeomorphism restricted to each face is also a diffeomorphism. Intuitively, this includes points, smooth compact curved segments, smooth compact surface patches with piecewise smooth boundaries, and all finite unions of such objects in which every non-empty intersection is a face. We shall call such an object a curved polyhedron to emphasize its definition from a topological point of view, and our interest in non-rectilinear objects. This definition reflects the indifference to dimensional homogeneity and to connectedness in our treatment.
since a single curved polyhedron may comprise dangling and/or disconnected pieces of various dimensions.

We shall view a curved polyhedron simply as the point set union of its *faces*. The latter term is used in an analogous way to its use in [28, §3.1] to mean not only surface patches, but also lower dimensional features such as (curved) edges and vertices, which is also consistent with its topological meaning. The boundary $\partial \varphi$ of a face $\varphi$ is not considered a part of it. The definition of a curved polyhedron ensures that, unlike the polyhedron itself, each face $\varphi$ has many of the nice properties that one would desire. These include simple connectedness and openness. The point set union of a face $\varphi$ with its proper subfaces is thus a simply connected manifold with boundary. By abuse of language we shall sometimes write $\overline{\varphi}$ for the closure of a face $\varphi$ when we really mean the collection of its (proper and improper) subfaces. We also call $\overline{\varphi}$ a face, as long as the context makes the distinction clear.

As to the geometric aspect, we focus our attention on the situation where each face lies on a host algebraic variety of bounded degree. Thus each face is a semialgebraic set as well as a manifold.

3. Duality

The central idea is to compute the convex hull by means of duality. This idea underlies several existing convex hull algorithms for finite point sets. For example, in [9] the linear time algorithm for constructing the intersection of two convex polyhedra in $E^3$ immediately gives an asymptotically optimal convex hull algorithm for finite point sets as follows. The points $K = \{p_1, p_2, \ldots, p_N\}$ are first dualized to half spaces. Then $K^*$, the intersection of these half spaces, can be computed by divide and conquer in log $N$ steps each taking $O(N)$ time using the algorithm of Chazelle. The intersection can then be dualized back in time $O(N)$ and the convex hull $\text{conv}(K) = K^{**}$ is found. See also [29] for a randomized incremental algorithm.

We recall that the duality used in such algorithms is one of several classical forms of dual, the *polar set* [24] of $K$

$$K^* \equiv \{x \in E^d : x \cdot y \leq 1 \text{ for all } y \in K\}.$$  

If $\text{conv}(K)$ the convex hull of $K$ contains the origin then $(\text{conv}(K))^* = K^*$. Moreover if $K$ is convex then $K^{**} = K$. It is also well known that $K^{**} = \text{conv}(K)$ [24].

We generalize this same approach to computing convex hull of a curved polyhedron $K$, namely a collection of possibly disconnected smooth faces of various dimensions. In view of the above properties of polar set, our first attempt is to compute the boundary of the polar set $K^*$ so that we may dualize the result again and then recover the convex hull. However in doing so we face difficulties not present in the point set case. The polar set is an operation on the entire set and the definition does not suggest how to actually construct it. By carefully studying the rationales behind the dualizing process, we shall come up with a second form of duality, which is easier to compute and from which the polar set can be recovered via known algorithms.
4. Classification and Representation of Hyperplanes

The following definitions are taken from Lay [24] with slight modifications. Let \( \mathcal{H} \) be the set of hyperplanes in \( E^d \) and \( K \subseteq E^d \). A hyperplane \( \pi \in \mathcal{H} \) is said to strictly bound \( K \) if one of the two closed halfspaces determined by \( \pi \) has an empty intersection with \( K \). It cuts \( K \) if both open halfspaces have non-empty intersections with \( K \). Otherwise it supports \( K \). Clearly \( \mathcal{H} \) is partitioned with respect to \( K \) into the following sets:

\[
\begin{align*}
\mathcal{H}_b(K) & \equiv \{ \pi \in \mathcal{H} : \pi \text{ strictly bounds } K \} \\
\mathcal{H}_s(K) & \equiv \{ \pi \in \mathcal{H} : \pi \text{ supports } K \} \\
\mathcal{H}_c(K) & \equiv \{ \pi \in \mathcal{H} : \pi \text{ cuts } K \}.
\end{align*}
\]

We also write \( \mathcal{H}_{\tan}(K) \) for \( \mathcal{H}_b(K) \cup \mathcal{H}_s(K) \). Our interest in these hyperplanes can be partly justified by the observation that a closed set (which a curved polyhedron is) and its convex hull induce exactly the same partitioning of \( \mathcal{H} \).

The supporting hyperplanes \( \mathcal{H}_s(K) \) of \( K \) are of particular interest in convex hull computations. Yet it may be difficult to compute directly, even when \( K \) is a curved polyhedron consisting of a single maximal face, namely itself. We therefore look at a closely related alternative which is easier to compute. To motivate the following definition we consider a smooth \((d-1)\)-dimensional hypersurface bounding a convex region in \( E^d \) such as a hyperellipsoid. In this case, the supporting hyperplanes are exactly its tangent hyperplanes.

Let \( \varphi \) be a smooth \( k \)-manifold embedded in \( E^d \). The tangent space \( \varphi_p \) at a point \( p \in \varphi \) is also \( k \)-dimensional. Define

\[
\mathcal{H}_{\tan}(\varphi) \equiv \{ \pi \in \mathcal{H} : \varphi_p \subseteq \pi \text{ for some } p \in \varphi \}
\]

In words, \( \mathcal{H}_{\tan}(\varphi) \) consists of hyperplanes having a contact of order at least 1 with \( \varphi \); or equivalently those not transverse to \( \varphi \). [6, §8.13] Naturally \( \mathcal{H}_{\tan}(K) \) for a curved polyhedron \( K \) is defined to be the union of \( \mathcal{H}_{\tan}(\varphi) \) for all faces \( \varphi \) of \( K \). For a closed line segment in \( E^d \), for example, \( \mathcal{H}_{\tan} \) and \( \mathcal{H}_s \) coincide. In the more general case \( \mathcal{H}_{\tan} \) is a good "approximation" of \( \mathcal{H}_s \). More precisely, a supporting hyperplane has to be tangent to some face; and a tangent hyperplane can never strictly bound the object.

Observation 1. Let \( \varphi \in E^d \) be a \( k \)-face of some curved polyhedron, \( k < n \). Then

\[
\mathcal{H}_s(\varphi) \subseteq \mathcal{H}_{\tan}(\varphi) \subseteq \mathcal{H}_s(\varphi) \cup \mathcal{H}_c(\varphi)
\]

Proof. Let \( \pi \) be a supporting hyperplane of \( \varphi \) at \( p \). Consider the face \( \varphi' \) of \( \varphi \) which contains \( p \). By the smoothness of \( \varphi' \) it is clear that \( \pi \) is tangent to \( \varphi' \) at \( p \).

The other inclusion is obvious. \( \square \)
Now that we are interested in $H_s$, $H_{bat}$, and $H_{tan}$ of curved polyhedra, we would like to simplify references to hyperplanes. Let us represent a hyperplane $\pi$ by the inversion of the perpendicular foot dropped from the origin $O$, where the inversion of a position vector $p$ is $p/|p|^2$ [28]. When applied to the three families of hyperplanes classified against a compact convex whose interior contains the origin, this representation produces precisely the polar set.

**Observation 2.** Let $K \subseteq E^d$ be a compact convex set whose interior contains $O$. Then the representations of $H_c(K)$, $H_s(K)$, and $H_{bat}(K)$ coincide with the exterior, boundary, and interior of $K$, respectively.

The proof is straightforward and hence omitted (see [18]). The restriction that $K$ be convex can essentially be dropped as long as the interior of $\text{conv}(K)$ contains $O$ since $(\text{conv}(K))^* = K^*$.

On the other hand, we may also apply the same representation to the set of all tangent hyperplanes of a smooth manifold $\varphi$ to obtain the **tangential dual** of $\varphi$. The tangential dual (or just dual) of a smooth manifold $\varphi$ is denoted $\bar{\varphi}$. The dual of a curved polyhedron is defined to be the union of the duals of its individual faces is usually an arrangement of hypersurfaces in $E^d$. In the special case of a curve in $E^2$ or a surface in $E^3$, it is the inversion of the pedal curve or pedal surface as defined in [6].

Combining Observations 1 and 2 we conclude that:

**Lemma 1.** Let $K \subseteq E^d$ be a convex polyhedron whose convex hull contains $O$ in the interior. Then $K^*$ is the (unique) convex cell in the arrangement of $K$ that contains $O$ in its interior.

**Proof.** By Observation 2, the boundary of the $K^*$ under this representation scheme is a subset of $K$. By Observation 1, the rest of the representing points all fall on the outside of $K^*$. □

We shall generally speak of the space inhabited by the given object $K$ as the **primal space**, and that in which its dual (set of points representing the tangents) exists as the **dual space**.

5. **Tangential Dual**

In this section we give formulas, properties, and examples of tangential duals, some of which we will refer to frequently later on. The derivation of the formulas from definition is straightforward and hence omitted. Instead we give visual examples at the end of this section.

**Lemma 2.** Let $\varphi$ be a smooth $k$-manifold defined by the set of $n - k$ equations

$$f_i(x) = 0, \ i = 1, 2, \ldots, n - k$$

and let

$$a_i(x_0) = \frac{\nabla f_i(x_0)}{\nabla f_i(x_0) \cdot x_0}, \ i = 1, 2, \ldots n - k.$$
Let $A(x_0)$ be the $(n-k-1) \times n$ matrix whose $i$-th row vector is $[a_i(x_0) - a_{n-k}(x_0)]^T$ and let $c_i(x_0)$ for $i = 1, 2, \ldots, k+1$ be a basis for its null space. Then a set of implicit equations in the variables $x$ representing $\mathcal{P}$ is given by eliminating $x_0$ from

$$
(x - a_{n-k}(x_0)) \cdot c_i(x_0) = 0, \quad i = 1, 2, \ldots, k + 1
$$

$$
f_i(x_0) = 0, \quad i = 1, 2, \ldots, m
$$

For example, let $\sigma$ be a surface in $E^3$ defined by:

$$
\sigma = \{(x, y, z) : f(x, y, z) = 0\}.
$$

Then

$$
\tilde{\sigma} = \left\{ \begin{array}{l}
\left( \frac{f_x, f_y, f_z}{x f_x + y f_y + z f_z} \right) : f(x, y, z) = 0
\end{array} \right\}.
$$

For $\varphi$ algebraic, the elimination can be done with Sylvester resultants, as discussed by Hoffman [16]. For parametric manifolds, similar formulas hold:

1. Let $\gamma \subset E^3$ be the parametric curve $\{r(t) : t \in I\}$ where $I$ is an open interval. Then $\tilde{\gamma}$ is the parametric curve:

$$
\left\{ \left( \frac{|r'|^2 r - (r \cdot r') r'}{|r'|^2 - (r \cdot r')^2} (t) : t \in I \right) \right\}.
$$

2. Let $\varphi$ be a parameterized surface in $E^3$ given by $\{x(u, v) : (u, v) \in D\}$ where $D$ is open in $E^3$. Then $\tilde{\varphi}$ is a parameterized manifold in $E^3$ given by

$$
\left\{ \left( \frac{x_u \times x_u}{|x_u \times x_v|} : (u, v) \in D \right) \right\}.
$$

3. Let $\gamma$ be a parameterized curve in $E^3$ given by $\{x(t) : t \in D\}$ where $D$ is an open interval in $R$. Then $\tilde{\gamma}$ is the developable surface

$$
\left\{ \frac{x(t)}{|x(t) \times x'(t)|} + s x(t) \times x'(t) : t \in D \text{ and } s \in R \right\}.
$$

In general, a $k$-manifold dualizes to a $(d-1)$-manifold namely a hypersurface regardless of $k$. The exceptional cases are manifolds having a degenerate family of tangent hyperplanes which can be parameterized by fewer than $(d-1)$ parameters. They dualize to $k'$-manifolds where $k'$ is the number of parameters required to parameterize the family of tangent hyperplanes. For example, in $E^3$ a developable surface other than a plane dualizes to a space curve since it has a 1-parameter family of tangent planes. In particular, (circular, elliptical, or hyperbolic) cones and planar conic curves dualize to each other in $E^3$, an easily proved fact we shall invoke in § 7.1. Linear affine subspaces give a simple but complete set of exceptional examples representative of degeneracy of all levels — a $k$-flat dualizes to a $(d-k-1)$-flat. In $E^3$, a ruled surface dualizes to a ruled surface since tangents to a ruling form
a family of planes sharing a straight line (namely the ruling), which dualize to a straight line.

Dualization also brings certain interesting features on a smooth surface to singularities. For example, a cusp on a planar curve dualizes to an inflection point; a parabolic curve on a surface to a cuspidal edge (Figure 4); and any asymptotic line on a hyperbolic surface to an asymptotic line on the dual hyperbolic surface. Finally, we make the following observation that follows immediately from the definition:

**Observation 3.** The set of common supporting hyperplanes of two manifolds dualize to the intersection of the duals of these manifolds.

Though already known, the following fact is a direct consequence of the above discussion.

**Corollary 1.** In $E^3$, the new surface patches resulting from the construction of the convex hull of a compact set are developable, regardless of the number and dimensions of the surfaces being supported by the new patches.

Let us now look at examples illustrating some of the properties above and Lemma 1. Most of the figures are translated from their "standard positions" so that interesting features in the dual space are more apparent.

Figure 1 shows a cosine curve. The two end points $v_1$ and $v_2$ generate two straight lines $\tilde{v}_1$ and $\tilde{v}_2$ in the dual space. In the dual space we find two intersections on the boundary of the convex cell around the origin, $\tilde{l}_1$ and $\tilde{l}_2$. We thus conclude that their duals $l_1$ and $l_2$ are required in the primal space to complete the convex hull.

Figure 2 shows a clover leaf and its dual. The vertical line in the dual, tangent to the dual at three places, is the image of the intersection of the three branches of the clover leaf. The three corners on the boundary of the polar set are dual images of the three tangents needed to close the clover leaf in forming the convex hull.

Figure 3 shows an S-shaped figure and its dual. The two straight lines are completely outside the cell containing the origin, since they are the images of the two end points $v_1$ and $v_2$, which are inside the convex hull. As is readily seen, two straight lines in the primal space are needed to form the convex hull, which translates in the dual space to two self intersections of the curve.

Figure 4 shows a 3-d example — a “dimpled” object whose boundary is the zero set of

$$(4x^2 + 3y^2)^2 - 4x^2 - 5y^2 + 4z^2 - 1$$

(This equation is taken from [23],) Its dual is presented on the right. The two “umbrellas” turned inside out correspond to tangent planes to points in the two dimples, which do not contribute to the convex hull. The apparent sharp cuspidal edges of the umbrellas are duals to two parabolic curves inside the dimples. The convex hull is not completed by just a single common tangent plane on each side. Correspondingly, the self-intersections of this surface in the dual space consist of a pair of curves, dual to two developables that cover these dimples on the convex hull of the primal surface.
Figure 1. Upper left: Cosine curve at an offset cut at $v_1$ and $v_2$. Right: Its dual, where two intersection points $\tilde{l}_1$ and $\tilde{l}_2$ are marked. Lower left: Its convex hull constructed by dualizing features on the boundary of the polar set. (Origin is marked by an “x”.)
Figure 2. A clover (thick line) leaf and its dual (thin line). The vertical line in the dual, tangent to the dual at three places, is the image of the intersection of the three branches of the clover leaf. The three corners on the boundary of the polar set are dual images of the three tangents needed to close the clover leaf in forming the convex hull.
Figure 3. An S-shaped figure and its dual. The two vertices are inside the hull, and therefore the two dual straight lines are completely outside the polar set. That two lines are needed to complete the hull is manifested by the two intersections in the dual.

Figure 4. A dimpled surface and its dual. The two “umbrellas” turned inside out correspond to tangent planes to points in the two dimples. The apparent sharp cuspidal edges of the umbrellas are duals of two parabolic curves inside the dimples.
6. Asymptotically Efficient Algorithms in Low Dimensions

By lemma 1, the convex hull of a curved polyhedron \( K \) in \( E^3 \) can be constructed in three major steps:

1. Find dual for each face to form an arrangement of hypersurfaces in the dual space.
2. Find the polar set by constructing the cell containing the origin.
3. Dualize the polar set.

The following three subsections are devoted to asymptotically efficient computation of the above tasks. We focus on convex polyhedra in \( E^3 \) whose faces are semialgebraic of bounded degree. For ease of presentation we also assume that the host surfaces and curves are rationally parameterized. In particular the extent of the faces on the host surfaces or curves are assumed to be described by inequalities on the parameters. Less restrictive assumptions are possible, at the cost of messier descriptions and additional computations. Other than these we don't assume any adjacency information in the input since the dual is computed piece by piece.

6.1. Setting Up and Computing \( \tilde{K} \). To be able to apply the duality results, the coordinate system need be such that the origin is in the interior of the desired convex hull. To achieve this, we simply pick 4 non-coplanar points on \( K \), say \( p_1, p_2, p_3 \), and \( p_4 \), find a point in the interior of the tetrahedron \( T = \text{span} \{ p_i : 1 \leq i \leq 4 \} \), and translate the coordinate system. We may do this by submitting the input to a procedure which either picks up enough desired affinely independent points or confirms degeneracy. In the latter case, we recursively invoke the convex hull algorithm on the lower dimensional subspace after rotating the coordinate axes so that some coordinate hyperplane conveniently parallels the affine hull of \( K \).

To avoid dealing with points and pieces at infinity in the dual space, we remove all input maximal faces in the interior of \( T \) in the primal space and trim other faces that intersect it. Note that the trimming down does not grow the complexity of the input by more than a constant factor due to the bounded degree assumption. The extent of the trimmed surface patches or curve segments can still be described by simple inequalities, though not necessarily in terms of the parameters.

The computation of the equation for \( \varphi \) is straightforward as given by equations (1) and (1). The major difficulty is then dealing with singularities: First, as noted above, duals are "usually" hypersurfaces, but they can also be lower dimensional manifolds. Faces dualized to these lower dimensional manifolds do not contribute to the cell in the arrangement. They need be identified and discarded. In \( E^3 \) the only two possible degenerate duals are points and curves. To recognize curves (not those resulting from surface intersections) in the dual space, we need to identify developable surfaces in the primal space. One does this by checking whether the discriminant of the second fundamental form of the input surface is identically zero. This takes constant time for each face, again due to the bound on the degree. The point case is subsumed in the above test. (Note: In higher dimensional spaces degenerate duals can be found by checking whether the Jacobian matrix of their
equations has an identically zero determinant, though this approach is much less efficient even in $E^3$.

Secondly, the remaining dual hypersurfaces need be broken up at singularities such as self-intersections, cuspidal edges, and swallow tails, to ensure smoothness for later processing. Since we have assumed that the input consists of smooth surface patches, such singularities will be absent from the input. However, they may also arise as a result of dualizing. For example, in Figure 4, the original dimpled surface is everywhere smooth; yet the dual exhibits self intersections and cuspidal edges. This problem can be solved with the help of singularity theory [14]. If the inputs are polynomials in integer coefficients, the duals of the surface patches will also be semi-algebraic sets in integer coefficients. Thus algebraic methods such as Gröbner basis or resultants with exact arithmetic may be used. In other cases, numerical criteria and methods can be employed. Here we only note that the lower the degrees of the input surfaces/curves are, the fewer possible singularities we need to worry about. In particular, for quadratic surfaces, the results in § 7.1 assures us that the worst cases are degenerate conicoids such as cones or pairs of planes. In this situation, enumerating all possible cases is straightforward and the duals can be computed numerically with little cost.

In each step above, every face is processed individually.

**Observation 4.** The dual $K$ of a curved polyhedron $K$ with algebraic faces can be computed and broken into smooth pieces in time proportional to the number of faces with a factor determined by the maximum degree.

### 6.2. Constructing Polar Set.

Cells in an arrangement of hypersurfaces can be constructed by any of a variety of well known algorithms. For example, a cylindrical decomposition would suffice for this construction. (This approach is similar to the one taken by Kriegman in [23].) However, more specialized and more efficient algorithms are also available for this particular problem [15].

As we are only interested in one particular cell (which is also convex), we may obtain an (asymptotically) even more efficient algorithm by way of reduction to the computation of the lower envelope, a problem studied in [34]. This randomized algorithm takes as input $\Sigma$ a set of $N$ algebraic surface patches of bounded degree without folds, each with a bounded number of similarly constrained subfaces. It then produces the minimization map. This is basically a planar graph with additional geometric information describing the lower envelope of $\Sigma$, or the set of points right above the plane. The algorithm runs in expected time $O(N^2 + \epsilon)$ for all $\epsilon > 0$.

Let $e$ be the outward-pointing, unit normal of the boundary of an arbitrarily chosen halfspace $H$ with $O$ on the boundary. Consider the following function defined on the interior of $H$:

$$K : x \mapsto a^{-1}x + ae$$

where $a = x \cdot e$. Its effect is to take vectors along the same direction to vectors ending in the same line parallel to $e$, with their order on the original direction preserved. The intuitive interpretation above also serves to show that if $O$ is in the
kernel of a star-shaped cell \( V \) in an arrangement of surfaces (which it is in our case), then half of \( \partial V \cap V \) is mapped to the lower envelope of the new arrangement. Note that \( K \) and its inverse

\[
K^{-1} : y \mapsto \beta y - \beta^2 e
\]

where \( \beta = y \cdot e - 1 \), are both algebraic.

Constructing \( K^* \) the polar set of the curved polyhedron is now straightforward. We first choose a plane \( \pi \) passing \( O \) which transversally intersects, if at all, every face constructed in the previous section so as to avoid annoying singularities in later computations. For each of the two unit normals \( e \) of \( \pi \) we apply \( K \), invoke Sharir’s algorithm, and then compute \( K^{-1} \) of the lower envelope.

Again each face can be processed individually in the above transformation. Hence the complexity of the lower envelop algorithm dominates. We note that the requirement that the number of boundary edges of a 2-face after dualizing and applying \( K \) be bounded can be met by simply requiring this to be the case in the original curved polyhedron. The reason is that each face is transformed individually and that the transformed surfaces corresponding to edges (vertices) in the original curved polyhedron have only 2 (0) subfaces.

**Lemma 3.** The polar set \( K^* \) of an algebraic curved polyhedron whose 2-faces have a bounded number of subfaces has \( O(N^{2+\epsilon}) \) faces for all \( \epsilon > 0 \). It can be computed in randomized expected time \( O(N^{2+\epsilon}) \) for all \( \epsilon > 0 \).

The result on the complexity of the features is of course independent of the cell construction algorithm we choose.

### 6.3. Dualizing the Polar Set

At this point we have a convex set with \( O(N^{2+\epsilon}) \) faces in the dual space. For simplicity, we shall assume that the cell construction algorithm has produced a *Doubly Connected Edge List* of [28] for representing the boundary of a convex curved polyhedron embedded in \( E^3 \) as a connected planar graph \( G \). Most existing general-purpose representations are equivalent to this simple structure as far as planar graphs are concerned, including the one generated by the reduction step described above. We could repeat the same procedure to \( K^* \) in order to find \( K^{**} \), namely the convex hull. However, we really need not compute the dual of the 2-faces of \( K^* \) since they are contained in the input 2-faces. Better yet, we do not even construct the arrangement of the dual of the faces of \( K^* \) since we are already dealing with a convex object. Instead, we just modify \( G \) to obtain the adjacency information of \( K^{**} \) and compute the geometric information as follows.

First a new planar graph \( G' \) is constructed whose vertices correspond to the vertices, edges, and regions of \( G \). The vertices of \( G' \) represent the 2-faces of \( \text{conv}(K) \), the dual of the polar set. An edge is drawn between two vertices of \( G' \) if the two corresponding entities in \( G \) are incident. These represent the 1-faces of \( \text{conv}(K) \). Clearly \( G' \) is planar and the regions of \( G' \) represent the 0-faces of \( \text{conv}(K) \).

Denote by \( v \) the vertex in \( G \) that corresponds to \( v' \in G' \). To recover faces of \( \text{conv}(K) \), we observe that a vertex \( v' \) in \( G' \) represents
(1) part of an input surface patch in the primal space if \( v \) is a region in \( G \). Its geometric information is already present in the input.

(2) a new developable surface patch which bridge two original surfaces if \( v \) is an edge in \( G \). Its geometric information is computed by the formula for duals.

(3) a triple common supporting plane if \( v \) is a vertex in \( G \). Its equation is easily found by computing the dual of the corresponding vertex.

To recover edges and vertices, we observe the following cases according to what kind of incidence in \( G \) an edge \( e' \) in \( G' \) corresponds to.

(1) **Vertex-edge adjacencies.** In this case, \( e' \) represents the ruling in \( \text{conv}(K) \) that marks the transition from a plane \( \pi \) to a developable surface \( \sigma \). Its geometric information is easily computed as the dual of the tangent straight line of \( \sigma \) at \( \overline{\pi} \).

(2) **Edge-region adjacencies.** Here \( e' \) represents a curve in \( \text{conv}(K) \) that marks the transition from a developable surface \( \sigma \) to an input face \( \varphi \). This curve can be considered the intersection of \( \sigma \) and \( \varphi \), and hence the dual of the envelope of the tangent planes of \( \varphi \) along \( \overline{\sigma} \) (that is, their common tangent planes in the dual space). In particular if \( \overline{\sigma} \) can be parameterized then so can this new edge.

(3) **Vertex-region adjacencies.** Now \( e' \) represents a vertex in \( \text{conv}(K) \) that marks the transition point from a plane to an original surface. Its coordinates can be found by dualizing the tangent plane of the region at the vertex.

The size (number of vertices) of \( G' \) is at most six times that of \( G \) since both are planar. Computing the geometric information in each of the above case contributes only a constant factor determined by the maximum degree of the polynomials. Thus the combinatorial complexity of the convex hull and also that of computing \( \text{conv}(K) \) from \( K^{*} \) is the same as the complexity of \( \partial K^{*} \), up to a constant factor. Summarizing all of the preceding results in this section, we conclude with

**Theorem 6.1.** Let \( K \) be a curved polyhedron with \( N \) algebraic faces of bounded degree each with a bounded number of subfaces. Then the convex hull of \( K \) has \( O(N^{2+\epsilon}) \) faces for any \( \epsilon > 0 \). It can be computed in randomized expected time \( O(N^{2+\epsilon}) \) for any \( \epsilon > 0 \) where the constant depends on the maximum degree of the polynomials in the input.

It is known that the complexity of the convex hull of \( N \) spheres can have complexity \( \Theta(N^2) \) [4]. Generally one expects a worst case complexity of \( \Omega(N^2) \) for algebraic surfaces of bounded degrees. It is also conjectured that the complexity is actually \( \Theta(N \lambda_q(N)) \) where \( q \) is a constant depending on the degree of the surface and \( \lambda_q(N) = O(N \log^q N) \) is the length of the maximum Davenport-Schinzel sequence of order \( q \) on \( N \) symbols. See for example [1] for a short summary and further references.

**6.4. Convex Hull in \( E^2 \).** In \( E^2 \) we can define a *simple* curved polyhedron in a way analogous to the rectilinear version of a simple polygon, i.e., being made up of a circular list of curve segments without self intersection. There exist several
combinatorially optimal algorithms running in time \( O(N) \) that deal with similar objects [32, 12, 2]. In view of our result it is more obvious that the strategy given in [28] for the rectilinear problem can directly lead to a combinatorially optimal algorithm with relatively simple logic as shown in [17].

When the curved polygon \( K \subset E^2 \) is algebraic of bounded degree but otherwise general (e.g., disconnected, having dangling edges), we may still obtain an optimal algorithm as follows. After computing \( \hat{\varphi} \) for all faces of \( K \), we break the image of each face at points at infinity and at places with vertical tangent. The result is an arrangement of pseudo lines [11] suitable for processing by a straightforward sweep line algorithm across the dual space along the horizontal axis. This is the dominating step and takes time \( O(N \log N) \), the constant factor depending on the maximum degree of the equations defining the edges. Computing the dual of the constructed cell is then straightforward and takes time \( O(N \lambda_q(N)) \) where \( q \) is determined by the maximum degree of the curves since this is a bound on its size. The algorithm is optimal since an arbitrary set of \( N \) points in \( E^2 \) constitute a curved polygon whose required processing time achieves the upper bound.

7. Quadratic Surface Patches

From a practical point of view we would like to have a simpler algorithm that avoids costly algebraic elimination in the dualization steps and complicated random sampling techniques used in [34]. There may also be occasions when one only needs to classify a few points against the convex hull of a curved polyhedron without really constructing it. In this section we address such issues by restricting our attention to curved polyhedron whose 2-faces are quadratic surface patches and whose 1-faces are (planar) conic sections. On the other hand we do not require that the number of subfaces of each 2-face be bounded. The fact that the duals of a (entire) quadratic surface is again a quadratic surface has simplifying implications on several algebraic aspects of the hull problem.

7.1. Closed Form Solutions for the Convex Hull. We shall show that the dual of a \((d-1)\)-dimensional quadratic hypersurface in \( E^d \) is again a \((d-1)\)-dimensional quadratic hypersurface. Moreover, the coefficients of the dual’s implicit representation can be easily computed from those of the primal’s. The presentation is simplified if we work in the projective space \( P^d \) whose proper points \((x_0, x_1, \ldots, x_d)\) are identified with points \((\frac{x_0}{x_d}, \ldots, \frac{x_{d-1}}{x_d})\) in the embedded Euclidean space \( E^d \).

Let \( \sigma \) be the quadratic hypersurface

\[
\{ \mathbf{x} \in P^d : F(\mathbf{x}) \equiv \mathbf{x}^T A \mathbf{x} = 0 \}
\]

where \( A = (a_{ij}), 0 \leq i, j \leq d \) is an invertible real symmetric matrix. At \( \mathbf{x} = (1, \xi_1, \ldots, \xi_d) \) the direction of the normal vector is the same as that of \( \mathbf{n} = \nabla F = A \mathbf{x} \equiv (\nu_0, \nu_1, \ldots, \nu_d) \). This is also the direction of the position vector of the dual \( \mathbf{x} \equiv (\lambda, \nu_1, \ldots, \nu_d) \) by definition. We ask what \( \lambda \) should be. Returning to \( E^d \) momentarily, we recall that the length of this position vector is to be the reciprocal
of that of the perpendicular foot dropped from the origin to the tangent hyperplane passing $\mathbf{x}$. That is,

$$\left(\frac{\sqrt{\nu_1 + \cdots + \nu_d}}{\lambda}\right) \cdot \left(\frac{\mathbf{x} \cdot \mathbf{n}}{|\mathbf{n}|}\right) = 1$$

(Here we are abusing the symbols $\mathbf{x}$ and $\mathbf{n}$ to mean the vectors in $E^d$ corresponding to their original meanings.) Thus $\lambda = \sum_{i=1}^{d} \xi_i \nu_i = \sum_{i=1}^{d} \xi_i \frac{\partial F}{\partial \xi_i}$. By Euler’s Theorem and by the fact that $F(\mathbf{x}) = 0$, we see that $\lambda = -\frac{\partial F}{\partial \xi_0} = -\nu_0$. Thus

$$\tilde{\mathbf{x}} = J \cdot A \cdot \mathbf{x}$$

or equivalently

$$\mathbf{x} = A^{-1} \cdot J \cdot \tilde{\mathbf{x}}$$

where $J$ is the $(d+1)$ by $(d+1)$ identity matrix with the 1 in the upper left corner replaced by -1.

Since the above holds for all points $\mathbf{x}$ on $\sigma$, the corresponding points $\tilde{\mathbf{x}}$ on $\tilde{\sigma}$ satisfy

$$(\tilde{\mathbf{x}}^T \cdot J \cdot A^{-1}) \cdot (A^{-1} \cdot J \cdot \tilde{\mathbf{x}}) = 0,$$

which shows that

$$\tilde{\sigma} = \{ \tilde{\mathbf{x}} : \mathbf{x}^T B \mathbf{x} = 0 \}$$

where $B$ is $A^{-1}$ with all elements in the first row and first column, except the
element in the upper left corner, negated. To see what happens when $A$ is singular, let us diagonalize $A = U^T (a_i) U$. Let $B = J U^T K U J$ where $K$ is any diagonal matrix whose diagonal elements are either $\frac{1}{a_{ii}}$ if $a_{ii} \neq 0$ or any non-zero constant if $a_{ii} = 0$. It is easy to verify that $\tilde{x}^T B \tilde{x} = 0$. In other words, the dual in this case is the intersection of a family of quadratic surfaces. It is a planar conic curve when $\sigma$ is a cone, cylinder, or a paraboloid; or a pair of points when $\sigma$ is a pair of planes.

7.2. An Example. Figure 6 shows two ellipsoids and part of a cone in space. The contribution of the cone to the convex hull is limited to its vertex. The dual of the cone and that of its base circle are not shown, so as not to clutter the pictures. Curves of intersection have also been omitted, since they too do not contribute to the hull. The ellipsoids have implicit equations

$$
eq 1 \begin{align*}
    e_1 & : 29x^2 + 36y^2 + 180z^2 - 24xy - 48x + 144y - 36 = 0 \\
    e_2 & : 225x^2 + 612y^2 + 388z^2 - 768yz - 2448y + 1536z + 1548 = 0.
\end{align*}
$$

Their duals are found as shown in § 7.1 to be:

$$
eq 2 \begin{align*}
    \tilde{e}_1 & : 36x^2 + 9y^2 + 5z^2 + 24xy - 20y - 5 = 0 \\
    \tilde{e}_2 & : 100x^2 - 3y^2 + 153z^2 + 192yz + 100y - 25 = 0.
\end{align*}
$$

Figure 7 shows their tangential duals and that of the vertex $v_1$. To find the polar set, we trim off the extraneous pieces of the surface patches, leaving alone the innermost
Figure 7. The dual of the surface patches of the object presented in the previous figure. Surfaces which do not support the dual of the convex hull are omitted for clarity.
Figure 8. The polar set of the object, recovered as the cell containing the origin in the arrangement of dual surfaces depicted in Figure 7.
Figure 9. The convex hull of the original object, recovered from the dual of the polar dual (previous figure) by the construction presented in the text. Note that the new surfaces \( f_1, f_3, \) and \( f_4 \) are developable, and arise from corresponding edges of the polar set.
cell in the dual space. It has three 2-faces as shown in Figure 8. Finally the polar set is dualized back to the primal space and the convex hull is obtained as in Figure 9.

7.3. **Point Classification without Explicit Hull Construction.** For $K$ a collection of conicoid patches, point classification against the convex hull can be performed as follows in time $O(\lambda_4(N) \log N)$ even if the hull is not constructed. A point $P$ is on the boundary of the hull if and only if through it there passes exactly one supporting plane of the hull; inside if and only if there is none; and outside if and only if there is more than one. This translates in the dual space to asking whether the intersection of the dual plane $\pi = P$ and the polar set $K^*$ is empty, a singleton, or neither. To find the above intersection, we first intersect each curved halfspace bounded by one dual with $\pi$. The results are $N$ curved convex halfplanes bounded by conic sections. Following the divide-and-conquer paradigm, we form pairwise intersections recursively. The total computation takes said time since the number of boundary segments for an intermediate intersection of $k$ halfplanes is $O(\lambda_4(k))$ and there are at most $\log N$ levels of merging steps. The above discussion assumes that each conicoid is present in the dual space as a whole. If this is not the case, its intersection with $\pi$ may be pieces of a conic curve. Filling the missing parts by “wedges” of ray pairs which are tangent to adjacent pieces at their end points (respectively) will give the same final intersection while the complexity analysis is still valid.

**Lemma 4.** Let $K$ be a curved polyhedron whose 2-faces are conicoid patches and whose 1-faces are segments of conic sections. A point in space can be classified against $\text{conv}(K)$ in time $O(\lambda_4(N) \times \log N)$ without the explicit construction of $\text{conv}(K)$.

7.4. **Other Practical Considerations.** We first mention that the convex hull can be rendered without ever being explicitly constructed. In the dual space we choose a set of nicely spaced directions, for example, ones corresponding to the grid points on the unit sphere with chosen longitudes and latitudes. For each direction we shoot a ray. This ray is intersected with each dual surface to find the intersection point nearest the origin. The tangent plane to this dual surface is found, dualized, and then plotted. It is clear that if these directions form grid points on the boundary of the polar set such as the suggested choice, so will the plotted points in the primal space. Moreover, neighboring plotted points in the primal space have similar normal directions, a good choice for rendering. Note that this technique is applicable to general curved polyhedron such as the dimpled surface in Figure 4, whose duals are too complicated to derive symbolically.

In CAD and CSG applications, being able to represent the original and derived surfaces and curves analytically versus numerically means a significant saving in storage. Better yet, one would usually like to have a piecewise parameterized representation available for rendering and other purposes. Fortunately we can do both quite efficiently and effectively with the duals and convex hulls of conicoids (quadratic surfaces) as shown in § 7.1. The fact that each conicoid is entirely either
elliptic, parabolic, or hyperbolic also allows us to discard all parabolic and hyperbolic surface patches from the very beginning as their duals can never be part of the boundary of a convex cell by themselves.

As pointed out earlier point classification against the hull can be carried out with a simpler algorithm in less time without the explicit construction of the convex hull. In case the hull really needs to be constructed, existing algorithms for converting CSG to boundary representations such as [26, 30] may be used in the dual space in place of the near optimal algorithm for convex cell construction. Although the asymptotic worst case behavior is worse, the similarity of these algorithms to their primal counterparts may argue in favor of their use, especially in those cases where the hulls constructed are not expected to have a high combinatorial complexity.

8. Conclusions

We study duality and modify the classical definitions to obtain algorithms for dealing with the convex hull of piecewise smooth objects in $E^d$. The fact that the new surface patches resulting from the convex hull construction in $E^3$ are developable falls out immediately. In $E^3$ we give an almost tight upper bound on the complexity of the convex hull and give a near optimal algorithm via reduction of the cell construction problem to Sharir’s algorithm. Using duality, combinatorially optimal algorithms in $E^3$ follow painlessly from their rectilinear counterparts.

Motivated by practical considerations, we show that new surface patches on the convex hull of conicoid patches with conic section borders have closed form solutions. Moreover, several simplification techniques are presented for dealing with related problems such as point classification and rendering. The more explicit details of the duals may prove to be a convenient bridge of reduction between seemingly different problems, as they do in the several cases we look at, where existing clever algorithms need not be re-invented in a disguised form.

Acknowledgement

We thank Ken Goldberg for posing the curved convex hull problem to us, which triggered this entire line of research. David Kriegman has kindly sent us updated references on convex hull. We thank Francis Bonahon for pointers to literature in classical duality. Potential applications of a curved convex Hull algorithm are provided by Aristides Requicha. We also wish to thank Boris Aronov, Yui-Bin Chen, Kenneth Clarkson, Ming-Deh Huang, Ram Nevatia, and people on the Internet for help, comments, and references.

The results on conicoids were computed using the software GP/PARI [3]. It was also used, together with GNUPLLOT [21] and RLaB[33], to implement the algorithm and to generate figures.
References


19. Alfred Inselberg, Personal communication.

20. Bill Jones, Herbert Edelsbrunner, and Guibas et al., *On-line bibliography of computational geometry*, anonymous ftp from cs.usask.cain file pub/geometry/geombib.tar.Z.


