An Introduction to Constructive Shell Representations for Free-form Surfaces and Solids

J. P. Menon*

Interactive Geometric Modeling, Computer Science
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

ABSTRACT

Free-form ('sculptured') surfaces are traditionally represented as unions of parametric patches of high implicit degree. Recently low degree algebraic patches have been introduced for representing free-form surfaces. This paper describes the use of algebraic patches in a new representation for free-form surfaces called Constructive Shell Representation (CSR). A CSR is a union of truncated tetrahedra, called *trunctets*, forming a 'thick shell' that contains the free-form surface. One bounding face of each trunctet is an algebraic patch which is a subset of the free-form surface; the other faces are planar.

CSRs for surfaces that are boundaries of free-form solids provide a new, complete hybrid Brep/CSG representation scheme for free-form solids. Properties and applications of this class of CSRs are the main focus of the paper. CSRs may solve some important problems in solid modeling, such as providing means to represent free-form solids in CSG, and hence extending the domain over which Brep → CSG conversion may be done. The paper outlines CSR-based solutions to these problems. Examples computed on an experimental system that exploits the RayCasting Engine (a highly parallel computer for CSG-based solid modeling) are provided.

1 INTRODUCTION

Free-form surfaces (mathematically, 2-D *r-sets* [Req80] embedded in 3-D Euclidean space E^3) are usually modeled as a finite union of patches represented in the traditional parametric or the recently developed algebraic forms. This paper introduces a new representation scheme for free-form surfaces, called Constructive Shell Representation (CSR), that draws on recent research on algebraic patches. CSRs of surfaces that constitute boundaries of solids are found to be very useful for solid modeling, and this class of CSRs is the main focus here.

1.1 Free-form surface representation

Parametric patch technology

Traditionally, each patch in a free-form surface is represented in the *parametric* form as a mapping from 2-D parameter space to 3-D space. The domain is usually rectangular or triangular, resulting in 'tensor-product' or 'triangular' patches respectively. A parametric surface may be *implicitized*, i.e. converted to an implicit form, but the resulting *algebraic* (implicit polynomial) equation $F(x,y,z) = 0$ may have a degree as high as 2mn (or n^2) for a tensor product (triangular) patch with rational functions of degree m, n (degree n) [Sed90]. For example, a bicubic tensor product patch could yield an algebraic equation of degree 18, and a quadratic triangular patch could yield an equation of degree 4.

Although parametric patches are powerful for constructing free-form surfaces, computing on these poses fundamental problems. For example, note that the intersection of a degree-m algebraic surface with a degree-n algebraic surface is a curve of degree mn [Har77]; hence, the intersection of two bicubic patches could result in a space curve of degree 324. Similar problems persist for curve/patch intersection calculations.

Algebraic patch technology

These, and other limitations of parametric patches, led to a recent line of work that seeks to construct free-form surfaces as a collection of *algebraic* patches [Sed85, Dah89, Sed90, LW90, Guo91, Dah92]. Each patch is defined as a low degree implicit polynomial (typically degree 2 or 3) that is 'clipped' by the walls of a tetrahedron, as shown in Figure 1.1a. Control points for the patch are prescribed in the boundary of the tetrahedron, typically by the vertices of the tetrahedron and additional points on the tetrahedral edges. A weight is associated with each control point, and shape control is obtained typically by changing the weights. (See Section 2 for details.)

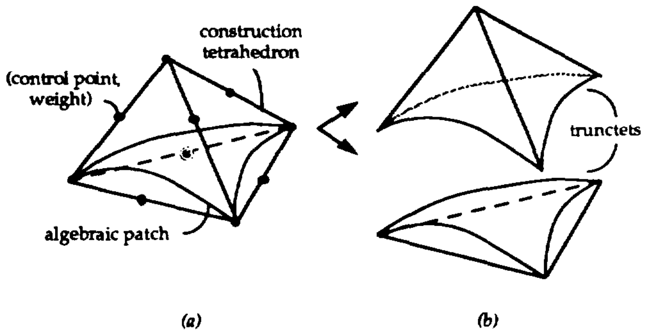


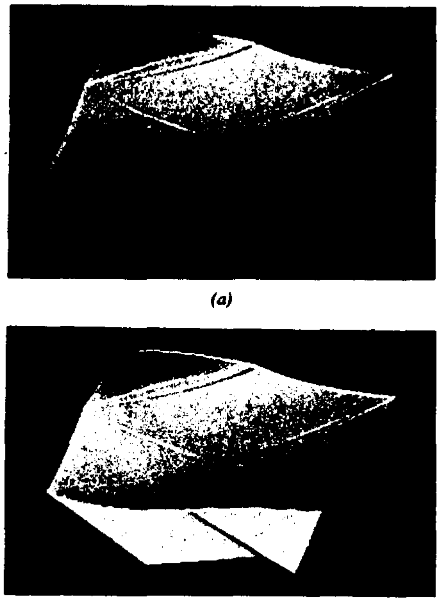
Fig. 1.1: (a) An algebraic patch, and (b) associated trunctet solids.

* *Work done while a Graduate Research Assistant in The Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY.*

2nd ACM Solid Modeling '93-5/93/Montreal, Canada
© 1993 ACM 0-89791-584-4/93/0005/0023...$1.50

A new representation for free-form surfaces

This paper introduces a new representation for free-form surfaces, called *Constructive Shell Representation* (CSR), that exploits the low implicit degrees offered by algebraic patches. A CSR is a union of truncated tetrahedra, called *trunctets*, forming a 'thick shell' that contains the boundary of a free-form surface. One bounding face of each trunctet is an algebraic patch which is a subset of the free-form surface; the other faces are planar, as shown in Figure 1.1b. Figure 1.2a shows an example of a free-form surface, and Figure 1.2b shows an associated 'thick shell' represented by the CSR of the surface.



(a)

(b)

Fig. 1.2: (a) A free-form surface with 26 quadratic algebraic patches, and (b) a 'thick shell' represented by the CSR.

1.2 Free-form solid modeling

We shall focus on CSRs associated with free-form surfaces that constitute boundaries of solids, because they appear to have several important applications in solid modeling.

Known representation schemes in solid modeling

Although six families of complete representation schemes for solids[1] have been known since the late '70s [Req80], only two -- Constructive Solid Geometry (CSG) and Boundary representation (Brep) -- are used commonly to represent solids exactly, i.e. without approximation. CSG represents complicated solids as a binary tree whose internal nodes are Boolean operators (union $\cup_k$, intersection $\cap_k$, and difference $-_k$) 'regularized' in the k-dimensional topology of (usually) halfspace[2] primitives contained in the leaves of the tree [RT78]. The Brep scheme describes a solid as a collection of faces bounding the solid [Req80].

While solids are often defined and edited in these 'primary' (CSG, Brep) schemes, 'secondary' representations -- such as cell decompositions [Req80], spatial enumerations [Req80], and ray representations [EKLTMMV91] -- are often computed from the primary schemes to support various applications.

Brep of free-form solids

Free-form solids are usually represented in the Brep scheme as unions of patches, and traditionally patches have been represented in the parametric form. Two critical problems with these Breps that stem from the fundamental limitations of parametric technology (noted earlier) are:

- patch/patch intersection, which is required for performing Boolean operations on solids [RV85], and
- line/patch intersection, which is used pervasively, e.g. for rendering graphic images of free-form solids.

Because these calculations are difficult to perform reliably and swiftly, Brep systems that support parametric patches suffer from numerical problems. Recent research on algebraic patches provides alternative Breps of free-form solids as unions of algebraic patches, and the associated low implicit degrees alleviate the above problems with parametric patches.

CSG of free-form solids

Nearly all of the research on developing representations for free-form solids has been focused on incorporating parametric patch technology in Breps; little of this work has been proven applicable to CSG representations. The fundamental reason is that halfspaces induced (via implicitization) from the popularly used parametric patches are of high degrees, whereas CSG technology is based on low degree algebraic halfspaces. Consequently, 'separation' and other such problems associated with curved halfspaces are hard to solve in the case of parametric patches [SV91a, Men92].

The little research that has been done on incorporating parametric patching technology into CSG has been based mainly on the following two approaches.

> In the *hybrid* approach, the set of leaf entities of CSG trees is extended to include free-form solids represented by their boundaries (using parametric patch technology), e.g. [Cha87]. The result is a non-homogeneous representation that fails to exploit fully CSG's elegant divide-and-conquer algorithms and often degrades CSG's robustness [Voe92].
> The *Inner Set Outer Set* (ISOS) approach constructs only linear polyhedral *approximations* of the free-form solid that are either contained in the solid or that contain the solid -- the 'inner' or 'outer' sets respectively [DSd89].

Insofar as we know, there has been no work to date on constructing CSG representations of free-form solids bounded by algebraic patches. This paper outlines how exact CSG representations of such solids can be constructed using CSRs.

A new representation for free-form solids

A CSR, that is constructed as a 'thick shell' (the union of trunctets) associated with algebraic patches on the boundary of the solid, provides a new complete hybrid Brep/CSG representation of the free-form solid. A subset of the shell of the free-form solid in Figure 1.3a is shown in Figure 1.3b. The hollow portion inside the CSR solid is a flat-faced polyhedron, called the *core*, as shown in Figure 1.3b.

[1] Solids are assumed to be k-D *r-sets* in E^k, k = 2,3.

[2] A halfspace is defined by the set of points $\{\mathbf{p} \mid F(\mathbf{p}) \leq 0\}$, where F is an algebraic function of low degree (typically 2, 3).

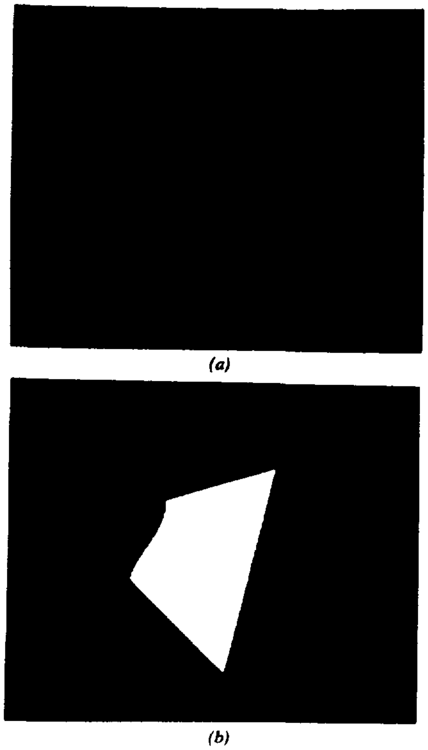


Fig. 1.3: (a) A free-form solid whose boundary is modeled with 32 quadratic algebraic patches, (b) its core and a portion of its CSR.

2 ALGEBRAIC PATCH TECHNOLOGY

Algebraic patch techniques for free-form surfaces describe patches as finite subsets of degree N algebraic surfaces. Two streams of work for constructing free-form surfaces with algebraic patches have emerged.

- *Tetrahedral methods:* Free-form surfaces that interpolate or approximate points and prescribed surface normals are constructed as the union of quadratic and cubic (N = 2, 3 respectively) algebraic patches contained inside tetrahedra [Sed85, Dah89, Sed90, LW90, Guo91, Dah92].
- *Non-tetrahedral methods:* Algebraic surfaces, typically quartics and quintics (N = 4, 5 respectively), not restricted to lie within tetrahedra, are used to interpolate or approximate points, curves and associated surface normals [HH86, War89, BI92a,b].

The paper draws on the tetrahedral methods, and the basic concepts of this recent line of research are summarized below.

2.1 Bernstein-Bezier representation of a patch

Consider a tetrahedron with vertices V_{N000}, V_{0N00}, V_{00N0}, and V_{000N}, where the V's are non-coplanar points in E^3. Let (s,t,u,v) denote the local barycentric coordinates in the tetrahedron. By definition, the barycentric coordinates [Far88] of a point **q** are the values of s, t, u, v such that

$$\mathbf{q} = sV_{N000} + tV_{0N00} + uV_{00N0} + vV_{000N};$$
$$s+t+u+v = 1. \qquad (2.1)$$

Let $a_b(s,t,u,v)$ denote a polynomial scalar function (subscript $_b$ denotes barycentric coordinates). A *contour surface* of the function comprises all points for which $a_b(s,t,u,v)$ is constant. The ***algebraic patch*** p is defined as the *zero contour* of the function that is clipped by the tetrahedron, i.e.

$$p = \{\mathbf{q} \mid \mathbf{q} \in a_b(s,t,u,v) = 0;\ s,t,u,v \geq 0\}, \qquad (2.2)$$

where (s,t,u,v) are the barycentric coordinates of **q**. The ***Bernstein-Bezier*** polynomials provide a convenient basis to control the behavior of the zero contour within the tetrahedron [Sed85]. Specifically, a degree N algebraic patch can be defined by first imposing a lattice of (N+1)(N+2)(N+3)/6 control points c_{ijkl}, such that

$$c_{ijkl} = \frac{i}{N}V_{N000} + \frac{j}{N}V_{0N00} + \frac{k}{N}V_{00N0} + \frac{l}{N}V_{000N};$$
$$i,j,k,l \geq 0;\ i+j+k+l = N. \qquad (2.3)$$

The lattice of control points for a quadric and cubic algebraic patch are shown in Figure 2.1. This lattice defines the control net for the patch, and its convex hull is the tetrahedron itself. Next, a weight w_{ijkl} is assigned to each control point, and the algebraic patch is defined using Bernstein-Bezier basis functions as

$$a_b(s,t,u,v) = \sum_{i,j,k,l \geq 0} w_{ijkl} \frac{N!}{i!\ j!\ k!\ l!} s^i t^j u^k v^l;$$
$$i+j+k+l = N;\ s,\ t,\ u,\ v = 1-s-t-u \geq 0. \qquad (2.4)$$

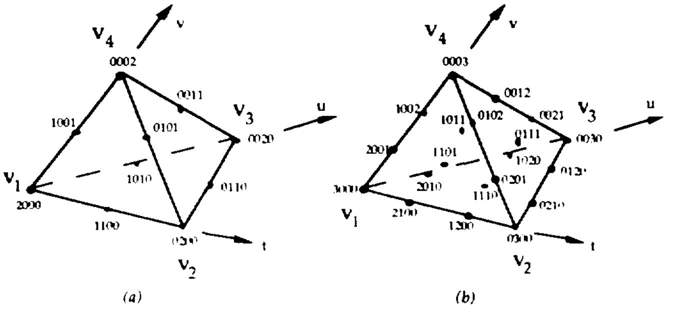


Fig. 2.1: Lattice of control points for (a) quadratic, (b) cubic algebraic patch.

Several properties for controlling the shape of a patch that arise from the above formulation are explained in [Sed85]. An algebraic patch can join smoothly (mesh) with other algebraic patches with specified inter-patch continuity and local support properties; recent approaches for this are summarized below.

2.2 Meshing patches

The C^0 and C^1 continuity conditions of two patches whose tetrahedra share a common face can be expressed easily in terms of the control points of the two patches [SS87, Guo91]. This method cannot be extrapolated in general to create an extended mesh of C^1 continuous algebraic patches. This is because *two-sided* and *three-sided* holes arise during the construction of extended free-form surfaces, and (low degree) algebraic patches do not possess sufficient degrees of freedom to fill up such holes with prescribed continuity.

Consider the three-sided hole problem in Figure 2.2. Three cubic patches are joined at their corners, with the two neighboring patches at each corner having the same tangent plane. It is not possible in general to fill this three-sided hole with a single cubic algebraic patch with C^1 continuity. In this case, the hole-filling patch is split into nine piecewise components to generate sufficient degrees of freedom for C^1 continuity for all adjacent patches [Sed90]. In fact, such a split also yields ten additional degrees of freedom, i.e. ten new control points that may be manipulated to influence the shape of the mesh of patches.

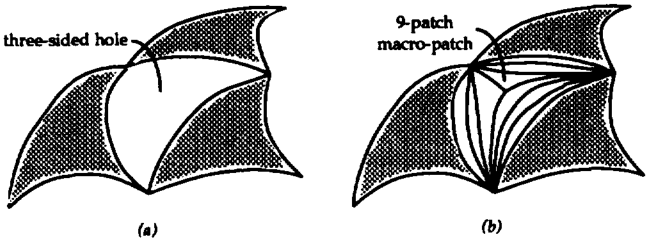


Fig. 2.2: (a) A 3-sided hole, (b) filled with a 9-patch macro-patch.

A group of piecewise algebraic patches obtained as a result of splitting a *three-sided* patch problem, to generate adequate degrees of freedom for satisfying continuity conditions, is called a *macro-patch*. Figure 2.2 illustrated a 9-patch cubic macro-patch.

It is easy to visualize the *two-sided* hole problem by imagining a triangulation of a free-form surface. Associate a construction tetrahedron with each triangle, and fit the surface segment within each tetrahedron with an algebraic patch or with a macro-patch. Clearly surface gaps will appear, as shown in Figure 2.3a. These gaps are filled with additional patches called *blend-patches*, e.g. in Figure 2.3b with two blend-patches for quadratic algebraic patches, that satisfy inter-patch continuity conditions [Guo91].

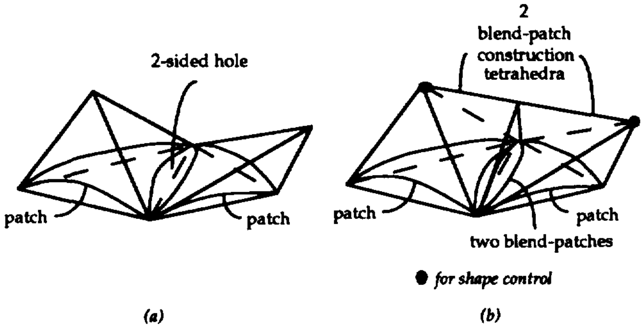


Fig. 2.3: Filling inter-patch gaps and forcing continuity with blend-patches.

2.3 Free-form surface construction

By applying the patch, macro-patch, and blend-patch concepts summarized above, a free-form surface can be constructed in the following three-step fashion, illustrated in Figure 2.4 for a surface that constitutes the boundary of a solid.

1) Specify a triangular faceted input polyhedron P whose vertices and (optional) associated surface normals are to be interpolated by the surface (boundary of a free-form solid for the example in Figure 2.4a).
2) Associate an apex vertex with each triangular facet, hence a tetrahedron, and fill each tetrahedron with a three-sided patch or macro-patch that 'replaces' each triangular facet of P (Figure 2.4b).
3) Fill the resulting two-sided holes with blend-patches (Figure 2.4b).

The resulting free-form surface approximates the shape of polyhedron P, i.e. a 'smoothing' of P is obtained. The methods developed thus far for constructing extended meshes of patches have used quadratic [Dah89, LW90, Guo91, Dah92] and cubic [Sed90, Guo91] algebraic patches with appropriate handles for shape control; see [Men92] for a comparative survey of these methods. For example, Figure 2.5 provides an example of shape control exercised on a spherical surface constructed as a mesh of 32 quadratic patches using Guo's methods [Guo91]. The experimental implementation described in Section 6, and all photographs of free-form surfaces and solids in this paper, use CSRs derived from quadratic patches constructed by Guo's second order technique.

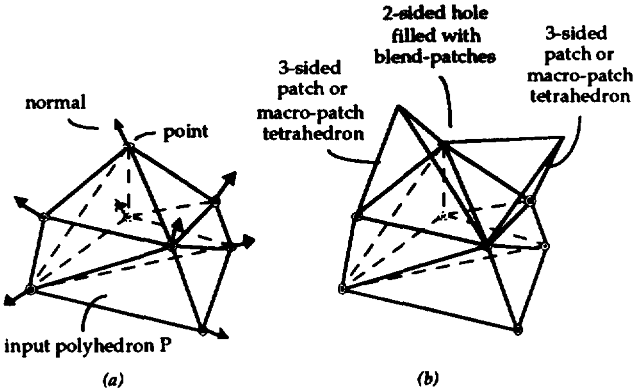


Fig. 2.4: Scheme for constructing a free-form surface: (a) input polyhedron P, and (b) tetrahedral constructs for two faces of P.

An Algebraic Patch Representation (APR) of a free-form surface $\mathcal{S}$ constructed using the steps outlined above can be expressed as the set of n algebraic patches $\{p_i\}$, without distinguishing whether p_i is a patch or a component of a macro- or blend-patch. Each p_i is specified completely by the degree N of the patch, the vertices V_1, V_2, V_3, V_4 of its construction tetrahedron, and the set $\{w_j\}$ of (N+1)(N+2)(N+3)/6 weights associated with the control points (eq. 2.4), i.e.

$$APR(\mathcal{S}) = \{(N, V_1, V_2, V_3, V_4, \{w_j\})_i \mid i = 1, n\}. \qquad (2.5)$$

The following properties of a valid APR (per eq. 2.5) of a free-form surface $\mathcal{S}$ in E^k, k = 2,3, follow directly from the construction rules.

2.1 $\mathcal{S} = \bigcup_i p_i$, and $\mathcal{S} \subseteq \bigcup_i T_i$, where T_i is the construction tetrahedron associated with patch p_i.

2.2 $n = O(f)$, i.e. the number of patches is linear in the number f of triangular facets of P.

2.3 $p_i \subseteq T_i$, i.e. every patch is contained in its tetrahedron, a convex-hull property.

2.4 $p_i \cap_{k-1} p_j = \phi$, $\forall$ i,j, $i \neq j$, all patches are quasi-disjoint.

2.5 Tetrahedra are often quasi-disjoint, i.e. $T_i \cap_k T_j = \phi$, $\forall$ i,j, $i \neq j$. A given set of algebraic patches that satisfies this condition will be called a *normal construction*: for example, the surfaces in Figure 2.6. Sometimes however, tetrahedra may overlap, as can be seen from the example in Figure 2.7. This condition, i.e. $T_i \cap_k T_j \neq \phi$, for some $i \neq j$, shall be called an *abnormal construction*.

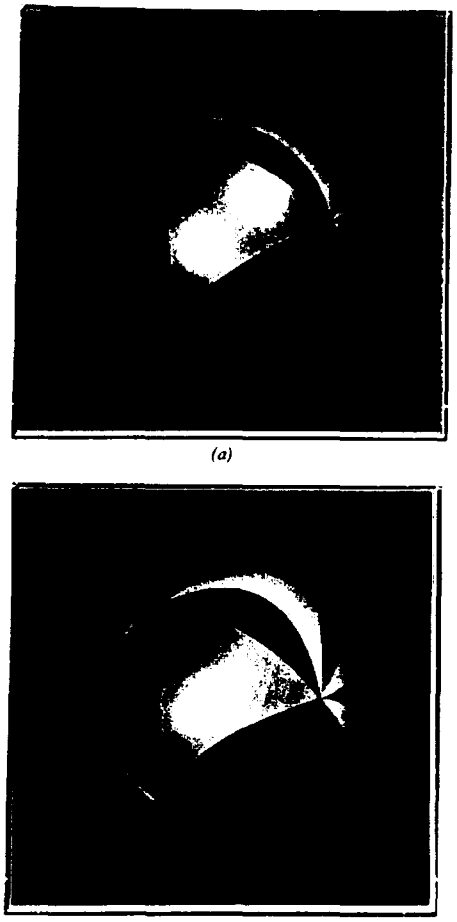


Fig. 2.5: Shape control on a 32-patch mesh of quadratic algebraic patches: (a) sphere, (b) tweaked sphere.

2.4 Brep of a free-form solid

If the surface constructed above is the boundary (∂A) of a free-form solid A, then the collection of algebraic patches APR(∂A) comprises a boundary representation of the solid, i.e. Brep(A). Weights assigned to control points shall be assumed to be in agreement with the material side of the solid, i.e. the material lies where the associated weights are negative. Thus, tetrahedral vertices V_i with negative weight w_i,

$$a_b(\mu_{1i}, \mu_{2i}, \mu_{3i}, \mu_{4i}) = w_i, \quad w_i < 0 \qquad (2.6)$$

give the sense of the inside of the free-form solid. (($\mu_{1i}, \mu_{2i}, \mu_{3i}, \mu_{4i}$) in eq. 2.6 are the barycentric coordinates of vertex V_i.)

Figure 2.6a shows a planar cross section and the construction tetrahedra (now triangles) of a 3-D free-form solid constructed using Guo's technique for quadric patches. Notice that vertices of P (those that lie in the section plane, marked '•') are interpolated by the boundary of the solid, and that pairs of blend-patches are used to fill the two-sided holes. A 2-D free-form solid that interpolates the vertices of P (a polygon in 2-D) is shown in Figure 2.6b. This figure is easier to follow than the cross-section in Figure 2.6a because blend-patch constructs are absent. For the sake of clarity, we shall use illustrations of 2-D solids henceforth, although all arguments hold for 3-D free-form solids constructed per Section 2.3. Thus, Figure 2.7 shows an abnormal construction of a 2-D free-form solid.

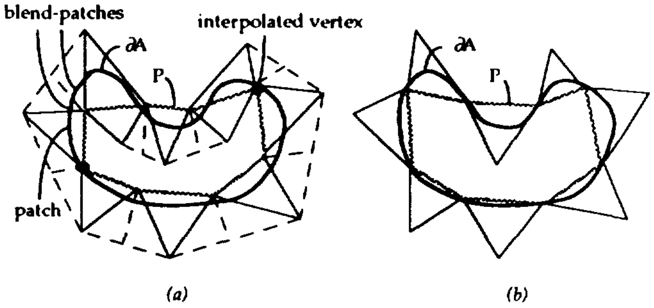


Fig. 2.6: Normal constructions: (a) A planar cross-section of a 3-D free-form solid, (b) a 2-D free-form solid.

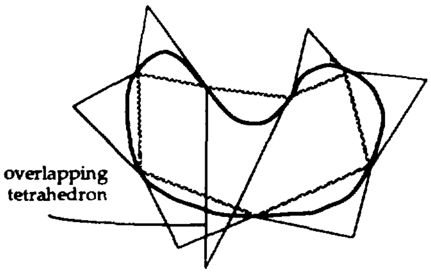


Fig. 2.7: An abnormal construction in 2-D.

3 CONSTRUCTIVE SHELL REPRESENTATION

3.1 Trunctets

Induction of a patch halfspace

Consider a single non-empty patch p in its tetrahedron T. A patch is associated with an extended surface a_b(s,t,u,v) = 0, where a_b(s,t,u,v) is a low degree (typically 2,3) polynomial expressed in the barycentric coordinates of its tetrahedron T (eq. 2.2). The patch representation in barycentric coordinates can be transformed into cartesian coordinates through a suitable linear transformation L [Far88, Men92] such that

$$a(x,y,z) = L(a_b(s,t,u,v)), \qquad (3.1)$$

where a(x,y,z) is a polynomial of equal degree as a_b(s,t,u,v).

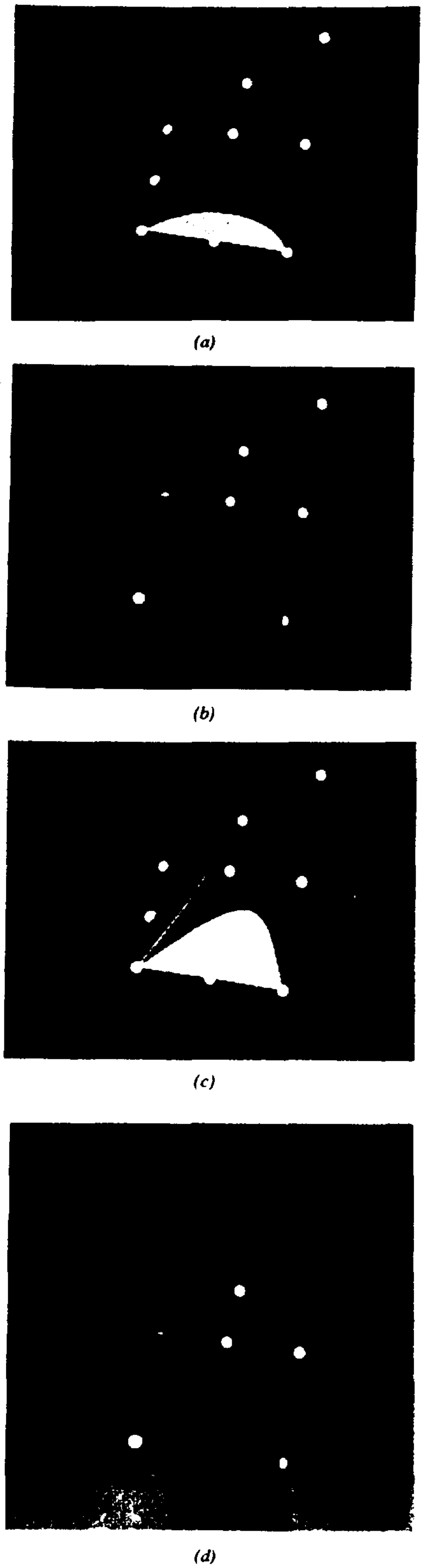
An *algebraic patch halfspace* a(x,y,z) ≤ 0 is induced by setting the sign of a(x,y,z) such that the value of a(x,y,z) evaluated at a tetrahedral vertex V_i is equal to the associated weight w_i (eq. 2.6), i.e. $a(V_i) = w_i$; i = 1,4. For example, the algebraic patch halfspaces for the quadratic patches in Figure 3.1a,c are respectively an ellipsoid and a hyperboloid of two sheets shown in 3.1b,d. The regularized complement of an algebraic patch halfspace, denoted *ca*, is simply a(x,y,z) ≥ 0.

Decomposition of a tetrahedron

Each construction tetrahedron T in E^k, k = 2,3, can be defined as the regularized intersection of k+1 linear halfspaces h_i whose boundaries contain the triangular faces f_i of the tetrahedron ($\partial h_i \supset f_i$), i.e.

$$T = \bigcap_i^k h_i, \quad i = 1,k+1. \qquad (3.2)$$

An algebraic patch halfspace decomposes the tetrahedron into two regions $T \cap_k a$ and $T \cap_k ca$, as shown in Figures 1.1 and



(a)

(b)

(c)

(d)

Fig. 3.1: (a) A quadratic algebraic patch, its trunctet, along with control points, (b) associated halfspace (ellipsoid), (c) tweaked patch and trunctet, and (d) associated halfspace (hyperboloid of 2-sheets; the tetrahedral apex vertex here is behind the top sheet).

3.2. Each region is called a *trunctet* (τ), since it may be viewed as a tetrahedron that is truncated (or capped) by the patch, such as those in Figure 3.1a,c; mathematically,

$$\tau = g \cap_k (\bigcap_i^k \; h_i), \;\; i = 1,k+1, \;\; g \in \{a, ca\}. \qquad (3.3)$$

An *inner trunctet* ${}^I\tau$ is defined as the intersection of the construction tetrahedron and the algebraic patch halfspace (Figure 3.2), i.e.

$${}^I\tau = a \cap_k \mathrm{T}, \qquad (3.4)$$

and an *outer trunctet* ${}^O\tau$ is defined as

$${}^O\tau = ca \cap_k \mathrm{T}. \qquad (3.5)$$

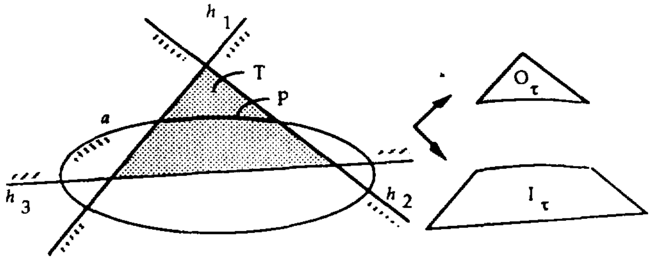


Fig. 3.2: Inner and outer trunctets.

Every patch is associated with an inner and outer trunctet, as shown in Figure 3.2[3]. The distinction between inner and outer trunctets could encapsulate the orientability of a free-form surface. If the free-form surface is ∂A for a solid A, inner trunctets are those regions in T that have the same material 'sense' as the free-form solid. Furthermore, the inner and outer trunctets associated with a single patch are quasi-disjoint, and their union is the construction tetrahedron itself [Men92].

3.2 Shells

A free-form surface $\mathfrak{F}$ can be associated with a thick *shell* S that is the union of trunctets, one associated with each of the n patches, i.e.

$$\mathrm{S} \;=\; \bigcup_i^k \; \tau_i, \;\; i = 1,n \;, \qquad (3.6)$$

and consequently, an inner shell (^{I}S) and an outer shell (^{O}S) can be defined respectively as

$${}^I\mathrm{S} \;=\; \bigcup_i^k \; {}^I\tau_i, \;\; i = 1,n \;, \;\; \text{and} \qquad (3.7)$$

$${}^O\mathrm{S} \;=\; \bigcup_i^k \; {}^O\tau_i, \;\; i = 1,n. \qquad (3.8)$$

For example, the shaded region in Figure 3.3a shows a shell composed as the union of trunctets with inner or outer counterparts chosen at random for every patch. The inner shell and outer shells are shaded in Figures 3.3b and 3.3c respectively. Figure 3.4 provides 3-D examples of inner shells -- 3.4a shows the inner shell for a normal construction, and 3.4b for an abnormal construction.

[3] Rare cases with peculiar assignment of weights to control points may cause the algebraic patch halfspace to result in multiple sheets within T. In such cases, the definitions of inner and outer trunctets can be modified through the introduction of additional 'separating' halfspaces; see Section 5.3 & [Men92].

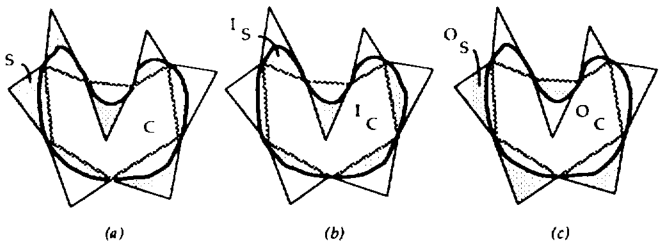
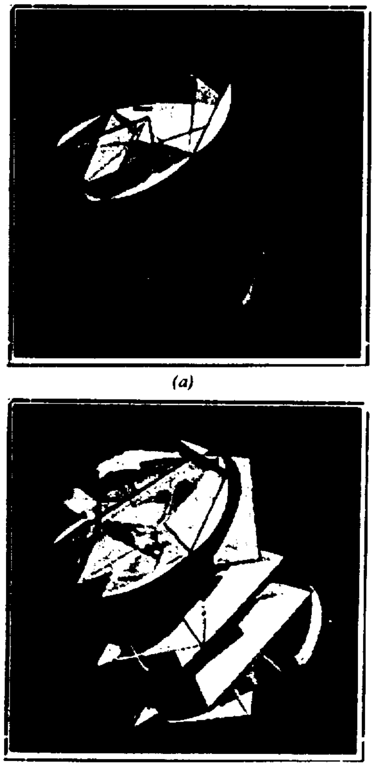


Fig. 3.3: (a) A shell and core, (b) inner shell and core, (c) outer shell and core.


(a)

(b)

Fig. 3.4: An inner shell for a surface with 204 quadratic algebraic patches: (a) normal construction, (b) abnormal construction.

3.3 Definition of CSR

CSR of a free-form surface

A Constructive Shell Representation (CSR) of a free-form surface $\mathfrak{S}$ constructed with algebraic patches (eq. 2.5), is a CSG representation of a shell S associated with $\mathfrak{S}$, i.e.

$$CSR(\mathfrak{S}) \;=\; CSG(S) \;=\; \bigcup_{i}{}_{k}\; CSG(\tau_i),\; i = 1,n, \qquad (3.9)$$

where S could be ${}^{I}S$, ${}^{O}S$, or any shell with randomly chosen inner or outer trunctets. Thus a CSR is a binary tree with union operators for nodes, and trunctet-subtrees for leaves. Each trunctet-subtree in turn has an intersection operator at its root, since trunctets are defined as the intersection of k+2 halfspaces (eqs. 3.2-3.5). Figure 1.2 provides a 3-D example.

CSR of a free-form solid

In an similar manner, a CSR of a solid A (where ∂A is constructed as the union of algebraic patches), is the CSG representation of a shell associated with ∂A (recall the 3-D example in Figure 1.3), defined as

$$CSR(A) \;=\; CSG(S) \;=\; \bigcup_{i}{}_{k}\; CSG(\tau_i),\; i = 1,n. \qquad (3.10)$$

4 PROPERTIES

We shall focus on CSRs of free-form solids since they may be very useful in solid modeling. This section discusses the properties of CSRs of solids, and the next section discusses the applications of CSRs for solving some important problems in solid modeling.

4.1 Cores of a solid

A shell S of a free-form solid A is associated naturally with a *core* C, which is the region of the solid not contained in its shell, i.e.

$$C = A -_k S. \qquad (4.1)$$

Inner (or outer) cores are associated with inner (or outer) shells; thus, ${}^{\alpha}C = A -_k {}^{\alpha}S$, $\alpha \in \{I, O\}$. Two-dimensional examples of cores may be found in Figure 3.3, and a three-dimensional example of the inner core is shown in Figure 1.3.

4.2 Properties of trunctets, shells and cores

This section lists some simple properties of trunctets, shells, and cores that follow from the construction of a valid Brep(A); see [Men92] for more examples.

Basic properties (Figures 2.6, 2.7, 3.3)

4.1 In general, trunctets/shells/cores are not unique for a given Brep(A). But, inner/outer trunctets/shells/cores are unique for a given Brep(A).

4.2 All trunctets, shells and cores are solids (r-sets).

4.3 Since every patch is associated with a trunctet (inner or outer), it follows that $\partial A \subseteq S$, and $\partial A \subseteq {}^{\alpha}S$, $\alpha \in \{I, O\}$.

4.4 A patch p_i, can also be expressed as $p_i = \partial g_i \cap \partial\tau_i$, where τ_i is the inner or outer trunctet associated with the patch and $g_i \in \{a_i, ca_i\}$. (See also Figures 1.1 and 3.2.)

4.5 The outer core ${}^{O}C$ is not a linear polyhedron, as shown in Figure 3.3c.

4.6 The inner core ${}^{I}C$ is always a linear polyhedron, for example, see Figure 3.3b.

4.7 ${}^{O}S \not\subseteq A$, i.e the outer shell is not contained in A: see Figure 3.3c.

Normal constructions (Figures 2.6, 3.3)

4.8 The boundary of a solid is contained in the boundary of its shell, i.e. $\partial A \subseteq \partial S$.

4.9 All trunctets are quasi-disjoint, i.e. $\tau_i \cap_k \tau_j = \phi$, $\forall\, i \neq j$.

4.10 The inner shell is contained in the solid, i.e. ${}^{I}S \subseteq A$.

4.11 The outer shell is contained in the regularized complement (c) of the solid, i.e. ${}^{O}S \subseteq cA$.

Abnormal constructions (Figure 2.7)

Negations of properties for normal constructions hold.

4.12 The solid's boundary is not always contained in the boundary of its shell, i.e. $\partial A \not\subseteq \partial S$.

4.13 Not all trunctets are quasi-disjoint, i.e. $\tau_i \cap_k \tau_j \neq \phi$, for some i,j.

4.14 The inner shell need not be contained in A, i.e. ${}^{I}S \not\subseteq A$.

4.15 The outer shell need not be contained in the regularized complement of A, i.e. ${}^{O}S \not\subseteq cA$.

4.3 Properties of CSRs

Representational completeness

We first establish that CSRs can represent solids unambiguously, providing thereby a new complete representation scheme for free-form solids. Consider a shell S represented by CSR(A). A CSG representation of every trunctet τ_i of the shell S can be obtained trivially from CSR(A). Further, b_i (which is a_i or ca_i) can be found from CSG(τ_i), because it is typically a non-linear halfspace[4]. Using property 4.4, $p_i = \partial b_i \cap \partial\tau_i$, and therefore ∂A is obtained as the union all the p_i's (property 2.1). Now A can be inferred unambiguously from ∂A by using the principle of 'boundary determinism', which states that an r-set is defined unambiguously by its boundary [Req77]. Hence a CSR is a complete representation of a free-form solid.

Property 4.16 CSR(A) is a complete representation of solid A.

A few observations based on the representational completeness of CSRs follow.

- The proof of property 4.16 provides an algorithm for computing Brep(A) from CSR(A) using 'boundary evaluation' (CSG → Brep) operations [RV85] on CSG(τ_i).
- Completeness holds even though $\partial A_j \not\subseteq \partial S_j$ for abnormal constructions, i.e. the boundary of the solid cannot be retrieved from the boundary of its shell (property 4.12).
- Completeness holds even if a patch halfspace a_i results in multiple sheets within its construction tetrahedron T_i, provided CSG(τ_i) represents the desired trunctet (using 'separating' halfspaces, as noted in footnote 3).

Represented point sets

It follows from eq. 3.10 that the point set represented by a CSR of a single free-form solid A_j is its shell S_j, i.e. $[CSR(A_j)] = S_j$[5]. However, the point set represented by a CSR of a Boolean composition of free-form solids, denoted $\mathbf{B}(A_1, ..., A_m)$, does not necessarily correspond to some 'thick shell' S_A of the composed solid A. Note that this does not pose any problems in light of the completeness of CSRs.

Property 4.17 If $A = \mathbf{B}(A_1, ..., A_m)$, m > 1, then $[CSR(A_j)] = S_j$, but $[CSR(A)] \neq S_A$.

For example, Figure 4.1 shows a Boolean intersection of CSRs of two free-form solids. The point sets $[CSR(A_1)]$, $[CSR(A_2)]$, [CSR(A)], and S_A are shaded. Notice how $[CSR(A_j)]$, j = 1,2, correspond to the inner shells of the two solids, but [CSR(A)] does not constitute some shell (e.g. S_A) of A.

Not surprisingly, a 'thick shell' S_A of a Boolean composition of free-form solids does not in general retain a trunctet like character. This is because a Boolean composition of free-form solids could result in patches with non-planar edges -- edges that do not lie in planar faces of tetrahedra. This observation could be of significance if one seeks to model free-form surfaces with non-planar edges using algebraic patches that are associated intrinsically with planar edges [Men92].

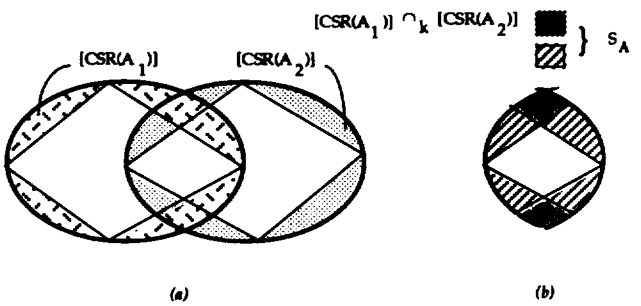


Fig. 4.1: (a) CSRs of two free-form solids, (b) Boolean intersection, represented point sets and some 'shell' of the composed solid.

Non-uniqueness

From property 4.1 it follows that there is no restriction on the kind of shell that a CSR represents. It may represent a shell with randomly chosen inner and outer trunctets as shown in Figure 3.3a, or an inner or outer shell (Figures 3.3b,c).

Property 4.18 The CSRs of a free-form solid may represent non-unique shells.

Non-uniqueness of CSRs does not pose problems as long as the character of the trunctets (inner, outer, or random) in the represented shell is known. A system that constructs CSRs from algebraic patches would keep track of this information, and the choice of trunctets would be governed typically by the application at hand. For example, a CSR representing all inner trunctets is used to produce the displays in Figures 3.4 and 6.2, while suitably chosen CSR-subsets are used for computing exact CSG representations of free-form solids in Section 5.2.

Hybrid Brep/CSG character

A CSR is a hybrid Brep/CSG representation since characteristics of both Brep and CSG schemes are prevalent. Like CSG, it is a binary tree with primitive halfspaces for leaves and regularized Boolean operators for nodes. However, a CSR may also be thought of as representing a 'thick boundary' of the solid, exposing a Brep like character. This was exploited for proving the representational completeness (property 4.16).

This hybrid Brep/CSG character allows algorithmic conveniences of both schemes to be exploited – specifically the execution of boundary traversal algorithms (employed pervasively on Breps [Mil89]) using the divide and conquer paradigm (applied commonly on CSG [Req80]). A new line/solid classification algorithm for computing ray-reps (summarized in Section 5.3) exploits this hybrid character.

5 APPLICATIONS IN SOLID MODELING

5.1 Solid modeling systems

From the mid-1970s until the later 1980s, two generic architectures of solid modeling systems predominated: systems which based all applications on Breps, and systems which maintained two or more consistent representations, usually CSG and Brep [Req80, Mil89, Voe92]. Dual representation <CSG, Brep> systems were deemed promising in 1980 [Req80], but CSG and <CSG, Brep> systems were abandoned commercially for the following two main reasons [Voe92].

[4] In rare cases where planar patches may be used, patch halfspaces may be marked with a distinguishing 'tag'.

[5] We use the notation Xrep(A) to denote a representation of solid A in scheme Xrep, and [Xrep(A)] to denote the represented point set. For example, Brep(A) is a boundary representation of A and [Brep(A)] = ∂A; similarly [CSG(A)] = A.

- CSG technology could accommodate prismatic solids, but not extended domains[6] of free-form solids.
- Even for the restricted domain of prismatic solids, system limitations arose because CSG → Brep conversion methods were available, but Brep → CSG conversion algorithms were not developed until recently [SV91a,b,c].

CSRs may be useful to extend the domain of CSG to represent free-form solids exactly, thereby providing Brep → CSG conversion methods for free-form solids. Thus CSRs could create opportunities to resurrect dual representation <CSG, Brep> systems. An outline of the CSR-based solution to CSG domain extension is presented below, followed by brief discussions on other solid modeling applications of CSRs. Detailed mathematical treatments will be presented in forthcoming papers; see also [Men92].

5.2 Exact CSG representations

Normal constructions

Using a *shell/core* approach, CSG(A) can be constructed as the union of the inner shell and the inner core associated with a free-form solid A, as for example in Figures 1.3 and 5.1a, i.e.

$$\begin{aligned} \text{CSG}(A) &= \text{CSG}({}^{I}S) \cup_k \text{CSG}({}^{I}C) \\ &= \text{CSR}(A) \cup_k \text{CSG}({}^{I}C). \end{aligned} \qquad (5.1)$$

CSG(^{I}S) is A's CSR, which is a Boolean composition of low degree algebraic halfspaces (eq. 3.10). A well known result in Brep → CSG conversion [SV91a] states that a CSG representation of a linear polyhedron can be computed easily from its Brep. This result can be applied directly to CSG(^{I}C), since the inner core is a linear polyhedron (property 4.6).

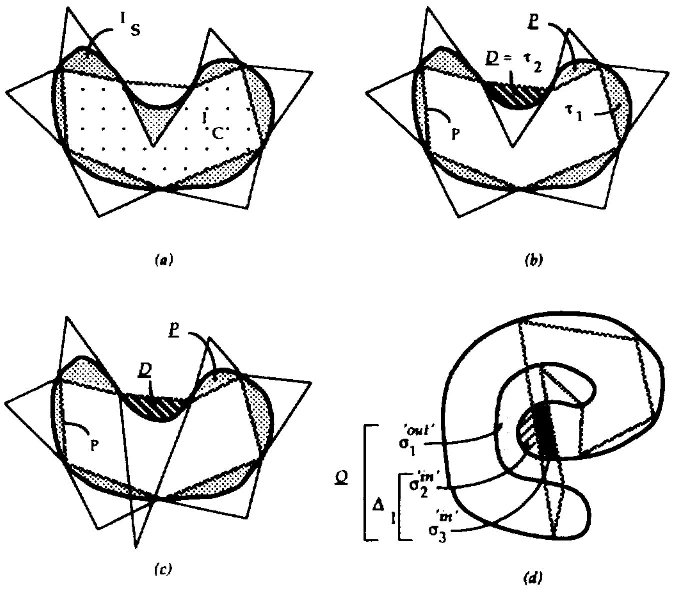


Fig. 5.1: Computing CSG representations of a free-form solid: (a) shell/core method for a normal construction, (b) constructive method for a normal construction, (c) constructive method for an abnormal construction, and (d) segment method for an abnormal construction.

Recall (from Section 2.3) that ∂A is constructed by (loosely) replacing every triangular facet of an input polyhedron P with a patch. A *constructive* approach for computing CSG(A) that mimics this construction procedure is summarized below.

Observe that every patch induces a trunctet that behaves either as a 'protrusion' relative to P (e.g. τ_1 in Figure 5.1b) or as a 'depression' relative to P (e.g. τ_2 in Figure 5.1b). Let $\underline{P}$ denote the union of the protrusion trunctets and let $\underline{D}$ denote the union of all the depression trunctets. CSG(A) can be obtained as the union of the input polyhedron (P) and the protrusions ($\underline{P}$), with the depressions ($\underline{D}$) differenced from this union, i.e.

$$\text{CSG}(A) = (\text{CSG}(P) \cup_k \text{CSG}(\underline{P})) -_k \text{CSG}(\underline{D}), \qquad (5.2)$$

or as the difference of the depressions ($\underline{D}$) from the input polyhedron (P), with the protrusions ($\underline{P}$) unioned to this difference, i.e.

$$\text{CSG}(A) = (\text{CSG}(P) -_k \text{CSG}(\underline{D})) \cup_k \text{CSG}(\underline{P}). \qquad (5.3)$$

CSG($\underline{P}$) and CSG($\underline{D}$) are CSR-subsets that are easy to obtain from specific trunctets. As with CSG(^{I}C) in the shell/core method, CSG(P) can be computed from Brep(P) since P is a linear polyhedron.

Abnormal constructions

Is is easy to see that the above simple methods do not apply in general to abnormal constructions. Specifically (from property 4.14) the shell/core method cannot be applied directly to abnormal constructions (e.g. Figure 5.1c,d). However, the constructive approach can be applied to those abnormal constructions where protrusions do not overlap depressions, i.e. $\underline{Q} = \underline{P} \cap_k \underline{D} = \phi$ (e.g. Figure 5.1c, but not 5.1d). Special treatment is needed for those constructions where $\underline{Q} \neq \phi$, and summarized below is one possible approach: see Figure 5.1d.

We extend the constructive approach described above to cover all classes of abnormal constructions by observing that eqs. 5.2 and 5.3 can be modified respectively as

$$\text{CSG}(A) = ((\text{CSG}(P) \cup_k \text{CSG}(\underline{P})) -_k \text{CSG}(\underline{D})) \cup_k \text{CSG}(\Delta_l) \qquad (5.4)$$

and

$$\text{CSG}(A) = ((\text{CSG}(P) -_k \text{CSG}(\underline{D})) \cup_k \text{CSG}(\underline{P})) -_k \text{CSG}(\Delta_e), \qquad (5.5)$$

where Δ_l (Δ_e) denotes the 'lost' ('excess') region due to protrusion/depression trunctet overlaps. The lost (excess) region is contained in the overlap $\underline{Q}$. Using techniques similar to those employed in Shapiro and Vossler's Brep → CSG conversion (or simply BCSG) algorithm [SV91a,b], CSG(Δ_l) can be obtained as

$$\text{CSG}(\Delta_l) = \bigcup_{i}{}_k \ \text{CSG}(\sigma_i), \qquad (5.6)$$

where σ_i is a *segment* that is represented as a suitable intersection of those trunctets (or their complements) that belong to $\underline{P}$ or $\underline{D}$, such that σ_i classifies as 'in' $\underline{Q}$, as well as 'in' A[7]. Similarly, CSG(Δ_e) is comprised of segments that are 'in' $\underline{Q}$

[6] The domain of a representation scheme is the set of entities representable in the scheme, and thus the domain characterizes the 'descriptive power' of the scheme [Req80].

[7] We use the jargon 'in', 'on', and 'out' as developed in the context of set membership classification [Til80].

but 'out' A. Segment classification can be obtained by classifying a 'characteristic' point in its interior with respect to CSG($\underline{Q}$), and then with respect to Brep(A)[8].

From experience, we observe that in practice abnormal constructions are unlikely to occur (e.g. using methods in [Guo91]). More importantly, even for abnormal constructions, a single test

$$\underline{Q}\ (= \underline{P} \cap_k \underline{D}) \overset{?}{=} \phi \qquad (5.7)$$

determines whether the simple CSG representations in eqs. (5.2-5.3) can be employed or not. Furthermore, the relations

$$[(\mathrm{CSG}(P) \cup_k \mathrm{CSG}(\underline{P})) -_k \mathrm{CSG}(\underline{D})] \subseteq A \qquad (5.8)$$

and

$$[(\mathrm{CSG}(P) -_k \mathrm{CSG}(\underline{D})) \cup_k \mathrm{CSG}(\underline{P})] \supseteq A \qquad (5.9)$$

provide expedient approximate CSG representations of the free-form solid that could be potentially useful in some solid modeling applications.

5.3 Other applications

Extending BCSG technology

Contemporary BCSG technology provides conversions for prismatic solids bounded by 'natural' (planar, cylindrical, spherical, conical) quadric surfaces, and some theoretical results are available for more general algebraic surfaces [SV91c]. The above methods for computing CSG representations extend the BCSG conversion to cover free-form solids as well, provided solid boundaries are represented as collections of algebraic patches. Three main reasons for the effectiveness of CSRs in BCSG are summarized in the following paragraphs.

Algebraic patches induce low degree (typically 2, 3) halfspaces, as opposed to, for example, a degree 18 halfspace associated with a bicubic parametric patch.

The acceptably hard problem of 'separation' encountered for curved surfaces in the BCSG algorithm [SV91a] is simplified, mainly because patches are bounded by planar edges[9]. Specifically, under certain 'monotonicity' conditions (e.g. [Sed85, Guo91]) or with the use of 'nonsplitting macropatches' (NMP) [Guo92], multiple sheets of the zero contour a_b(s,t,u,v) can be avoided entirely, thereby obviating the separation problem; we call such constructions *self-separating* [Men92]. However, if multiple sheets do arise (e.g. since monotonicity conditions are not globally satisfied), the separation problem is localized to the construction tetrahedron for each patch -- a *local separation* problem. Using results from [SV91c], the existence of linear separating halfspaces to solve the local separation problem is shown in [Men92].

In principle, having solved the separation problem, the BCSG algorithm in [SV91a] that 'extends' participating halfspaces, decomposes space into cells, and determines classification of cells, could be employed directly to compute CSG(A). However, the CSR-based CSG constructions described above (Section 5.2) do not extend all halfspaces; instead decompositions are considered within known *finite extents*. Ensuing CSG construction techniques simplify considerably, especially for normal constructions (eqs. 5.1-5.3).

Computing ray-reps

Free-form solids bounded by algebraic patches are constructed in the Brep scheme, and the methods in Section 5.2 provide CSG representations of the solids. Hence a ray-rep of the free-form solid can be computed using well known line/solid classification algorithms on Brep or CSG representations of the solid [Til80, RV85]. Furthermore, given a CSG representation of a free-form solid, a ray-rep can also be computed via massively parallel processing on the RayCasting Engine (RCE) -- a new highly parallel computer for computing ray-reps from CSG representations [EKLTMMV91].

CSRs provide new methods for computing ray-reps of free-form solids through algorithms that exploit their hybrid Brep/CSG character. In essence, the algorithm employs the CSG-like divide and conquer paradigm for line/solid classification with respect to the CSR, but interprets the CSR-based classification results in a Brep fashion to obtain in-solid segments; the main idea is summarized below.

Consider the classification of line l2 with respect to the shaded CSR (obtained using divide and conquer) in Figure 5.2; tetrahedral intercepts are marked 'x', and patch intercepts '•'. It is easy to see that l2 'in' A can be induced by simply ignoring the tetrahedral intercepts on l2, and interpreting patch intercepts as though they were obtained during line/Brep classification[10]. This method can be extended to abnormal constructions as well [Men92]. From a practical standpoint, since the new algorithm retains CSG's divide and conquer character, it is amenable to parallelization, e.g. by mapping it on the RCE [Men92].

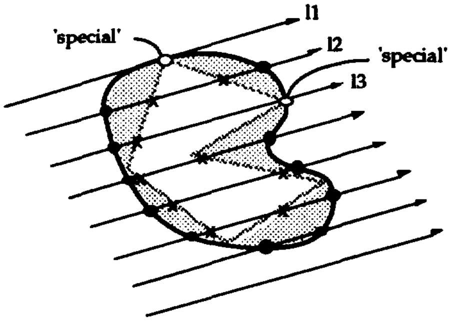


Fig. 5.2: Computing a ray-rep of A from CSR(A).

6 QPATCH EXPERIMENTAL SYSTEM

Figure 6.1 provides a high level overview of the QPATCH experimental system that is built in the environment of a 2304-processor RCE[11] residing in an Adage-3000 frame buffer

[8] An exponential enumeration and examination of segments can be obviated by resorting to geometrical calculations such as those employed in BCSG [SV91a,b], or using an iterative ray-rep based method developed in [Men92].

[9] Separation simplifies considerably because of planar edges in algebraic patches; separation problems that arise from non-planar edges are not currently addressed here.

[10] Lines that pass through interpolated points of P (e.g. l1, l3) require special treatment, since they need to be distinguished for singularity [Men92].

[11] The RCE is being built under collaborative research between Cornell and Duke Universities [EKLTMMV91].

backplane, connected to a DECSystem 5400 host. A free-form solid is constructed in the Brep scheme using techniques developed in [Guo91]. A CSR is constructed from this Brep (shown as conversion BCSR) using the methods developed in Section 3. Now one of three possible routes may be taken.

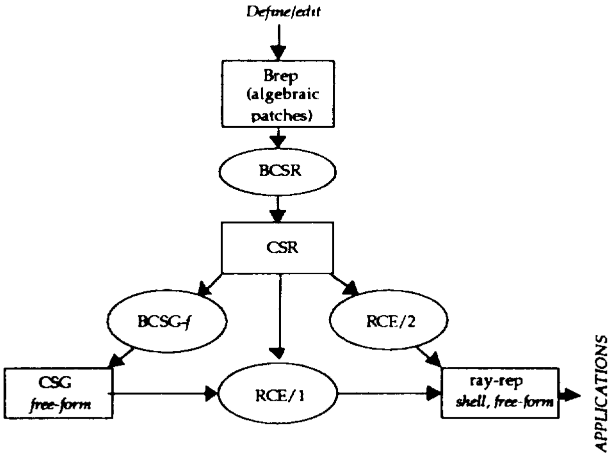


Fig. 6.1: High level overview of the QPATCH experimental system.

1) A CSR may be processed directly on the current RCE (marked RCE/1) to compute a ray-rep of a shell.
2) A CSR may be used to compute a CSG representation of the solid through the BCSG-*f* conversion (BCSG for *f*ree-form solids), and the CSG representation may be processed on RCE/1 to compute a ray-rep of the free-form solid.
3) A CSR may be processed on the modified RCE (marked RCE/2 and currently under design) to compute a ray-rep of (1) a shell, or (2) the free-form solid directly from the CSR (by employing the new hybrid Brep/CSG algorithm for computing a ray-rep from the CSR).

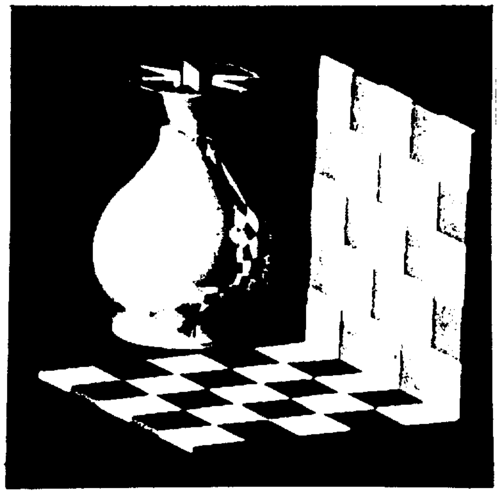


Fig. 6.2: Ray traced vase modeled with 336 quadratic algebraic patches.

Ray-reps computed above are used in various solid modeling applications, such as mass property calculation, sweeping, Minkowski operations, interference detection, ray tracing (Figure 6.2), and NC machining simulation/verification [MMZ92, MR92]. Here are two examples to give an idea of the system performance; grids of 300x300 rays were used here.

- The vase example in Figure 6.2, which is modeled with 336 quadratic algebraic patches, contains 1,680 halfspaces in its CSR; it took 11 seconds to compute a ray-rep of the shell, and subsequent ray tracing took 53 seconds.
- The example in Figure 3.4 contains 204 quadratic algebraic patches that interpolate 30 points obtained from a crushed soda can sampled by a CMM data-acquisition system. The CSR contains 1,020 halfspaces and it took 8 seconds to compute its ray-rep and produce a shaded image.

See [Men92] for system details, and more modeling examples and statistics.

7 CONCLUSIONS AND FUTURE DIRECTIONS

Problems with traditional parametric patching methods for free-form modeling led to recent developments in algebraic patching technology, wherein sculptured surfaces are constructed by piecing together subsets of low degree (typically 2, 3) implicit polynomial surfaces contained inside tetrahedra. The paper exploited this recent line of work on algebraic patches and introduced a new representation scheme -- Constructive Shell Representation (CSR) -- that treats free-form surfaces and solids in terms of 'thick shells'.

We believe that algebraic-patching/CSR technology has tremendous potential for surface and solid modeling. The technology is relatively new, and few areas of emphasis for future work are summarized below.

- *Construction of Surfaces*: Planar edges in tetrahedral constructions are a limitation, but they are also the key reason for simplifying the separation problem in BCSG conversion. Open issues include: 1) the accommodation of non-planar edges such that simplicity of separation is retained, and 2) the automatic conversion of an abnormal construction to a normal construction (since normal constructions provide simple solutions, e.g. for CSG and ray-rep computations).

- *Extension of CSR methods:* Shell/core notions may be extended to free-form surfaces that use patches with planar-edges defined within parallelopipeds [PK89], and similarly for patches with non-planar edges associated with other control nets [BI92a,b]. CSR notions may also be extended to prismatic solids, e.g. wherein Breps of solids may be constructed such that they are amenable to CSG conversion. Open issues surround the determination of conditions under which self/local-separation properties hold for such CSR extensions.

- *Applications in solid modeling:* Apart from those discussed here, we believe that there are other opportunities for exploiting the computationally tractable character of CSRs, e.g. for generating finite element meshes, or for solving boundary value problems, or for computing NC tool paths in sculptured surface machining.

- *Applications in surface modeling:* There are several open problems regarding the use of CSRs in surface modeling, such as supporting non-homogeneous solids, and solids containing 'mixtures' of prismatic and sculptured geometry; see [Men92] for examples.

- *Bilateral parametric ↔ algebraic patch conversion:* It is desirable to convert between parametric and algebraic patch representations, without loss of 'design properties' (e.g.

local continuity, curvature, etc.). While parametric representations of quadratic and cubic algebraic patches can often be computed using known techniques, the conversion from parametric to algebraic patch representations is largely an open problem[12].

ACKNOWLEDGEMENTS

The author would like to thank Professor Herb Voelcker (Cornell University), for his valuable advice and encouragement during the course of this work. Thanks also to Baining Guo (University of Colorado, formerly Cornell), for several discussions, and for providing a definition of the vase. G. Hartquist, R. Marisa, and K. Tokusei (Cornell) helped with implementation issues, and Professor Gershon Kedem's team (Duke University) provided the RCE hardware support. B-O Schneider and A. Crosnier (IBM Research) gave useful feedback on the preliminary draft. Finally, many thanks to Jarek Rossignac (IBM Research) and the reviewers for their constructive comments that helped to sculpt this paper to its final form. This research was supported by the National Science Foundation under grant MIP-90-07501.

REFERENCES

[BI92a] Bajaj, C., and Ihm, I., "Algebraic surface design with Hermite interpolation", *ACM Transactions on Graphics*, vol. 11, no. 1, pp. 61-91, January 1992.

[BI92b] Bajaj, C., and Ihm, I., "Smoothing polyhedra using implicit algebraic splines", Computer Graphics (SIGGRAPH '92), vol. 26, no. 2, pp. 79-88, July 1992.

[Cha87] Chan, K-C, "Solid modelling of parts with quadric and free-from surfaces", Ph.D. Dissertation, Department of Mechanical Engineering, University of Hong Kong, November 1987.

[Dah89] Dahmen, W., "Smooth piecewise quadric surfaces", **Mathematical Methods in CAGD**, eds. T. Lyche and L. Schumaker, Academic Press Inc., pp. 181-193, 1989.

[Dah92] Dahmen, W., "Modeling and visualization with implicit surfaces", Oberwolfach meeting on Curves and Surfaces in Computer Aided Geometric Design, Oberwolfach, June 1992.

[DSd89] Dunnington, D. R., Saia, A., and de Pennington, A., "Constructive solid geometry with sculptured primitives using inner and outer sets", **Theory and Practice of Geometric Modeling**, eds. W. Straber and H. P. Seidel, Springer-Verlag, pp. 127-142, 1989.

[EKLTMMV91] Ellis J. L., Kedem, G., Lyerly, T. C., Thielman, D. G., Marisa, R. J., Menon, J. P., and Voelcker H. B., "The RayCasting Engine and ray representations: a technical summary", *International Journal of Computational Geometry and Applications*, vol. 1, no. 4, pp. 347-380, December 1991.

[Far88] Farin, G., **Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide**, Academic Press, Inc., San Diego, California, 1988.

[Guo91] Guo, B., "Modeling arbitrary smooth objects with algebraic surfaces", Ph.D. Dissertation, Department of Computer Science, Cornell University, Ithaca, NY, June 1991.

[Guo92] Guo, B., "Nonsplitting macro patches for cubic spline surfaces", Computer Science, University of Colorado, December 1992 (in draft).

[Har77] Hartshorne, R., **Algebraic Geometry**, Springer-Verlag, New York, 1977.

[HH86] Hoffmann, C., and Hopcroft, J., "Quadratic blending surfaces", *Computer Aided Geometric Design*, vol. 18, no. 6, pp. 301-306, 1986.

[LW90] Lodha, S., and Warren, J., "Bezier representation for quadric surface patches", *Computer-Aided Design*, vol. 22, no. 9, pp. 574-579, November 1990.

[Men92] Menon, J. P., "Constructive shell representations for free-form surfaces and solids", Ph.D. Dissertation, The Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, August 1992.

[Mil89] Miller, J. R., "Architectural issues in solid modelers", *IEEE Computer Graphics and Applications*, vol. 9, no. 5, pp. 72-87, September 1989.

[MMZ92] Menon, J. P., Marisa, R. J., and Zagajac, J., "More powerful solid modeling through ray representations", Tech. Report CPA92-4, Sibley School of Mechanical Engineering, Cornell University, Ithaca, NY, April 1992.

[MR92] Menon, J. P., and Robinson, D. M., "High performance NC verification via massively parallel raycasting: extensions to new phenomena and geometric domains", *Proc. Symposium on Concurrent Engineering, 1992 ASME Winter Annual Meeting*, Anaheim, California, pp. 179-194, Nov. 8-13, 1992.

[PK89] Patrikalakis, N. M., and Kriezis, G. A., "Representations of piecewise continuous algebraic surfaces in terms of B-splines", *The Visual Computer*, vol. 5, pp. 360-374, 1989.

[Req77] Requicha, A. A. G., "Mathematical models of rigid solid objects", Tech. Memo 28, Production Automation Project, University of Rochester, November 1977. (Available from CPA, Mechanical Engineering, Cornell University, Ithaca, NY.)

[Req80] Requicha, A. A. G., "Representations for rigid solids: theory, methods and systems", *ACM Computing Surveys*, vol. 12, no. 4, pp. 437-464, December 1980.

[RT78] Requicha, A. A. G., and Tilove, R. B., "Mathematical foundations of constructive solid geometry: general topology of closed regular sets", Tech. Memo 27a, Production Automation Project, University of Rochester, June 1978. (Available from CPA, Mechanical Engineering, Cornell University, Ithaca, NY.)

[RV85] Requicha, A. A. G., and Voelcker, H. B., "Boolean operations in solid modeling: boundary evaluation and boundary merging", *Proc. of the IEEE*, vol. 73, no. 1, pp. 30-44, January 1985.

[Sed85] Sederberg, T. W., "Piecewise algebraic surface patches", *Computer Aided Geometric Design*, vol. 2, pp. 53-59, 1985.

[Sed90] Sederberg, T. W., "Techniques for cubic algebraic surfaces: Tutorial part two", *IEEE Computer Graphics and Applications*, vol. 10, no. 5, pp. 12-21, September 1990.

[SS87] Sederberg, T. W., and Snively, J. P., "Parametrization of cubic algebraic surfaces", **The Mathematics of Surfaces II**, ed. R. R. Martin, Clarendon Press, Oxford, pp. 299-319, 1987.

[SV91a] Shapiro, V., and Vossler, D. L., "Construction and optimization of CSG representations", *Computer-Aided Design*, vol. 23, no. 1, pp. 4-19, Jan/Feb 1991.

[SV91b] Shapiro, V., and Vossler, D. L., "Efficient CSG representations of two-dimensional solids", *ASME Journal of Mechanical Design*, vol. 113, pp. 292-305, September 1991.

[SV91c] Shapiro, V., and Vossler, D. L., "B-rep → CSG conversion III: boundary-based separation", Tech. Report CPA91-5, Sibley School of Mechanical Engineering, Cornell University, Ithaca, NY, July 1991.

[Til80] Tilove, R. B., "Set membership classification: a unified approach to geometric intersection problems", *IEEE Transactions on Computers*, vol. C-29, no. 20, pp. 874-883, October 1980.

[Voe92] Voelcker, H. B., "New directions in solid modeling?", *Proc. International Conference on Manufacturing Automation*, ed. N. W. M. Ko and S. T. Tan, University of Hong Kong, pp. 157-168, August 1992.

[War89] Warren, J., "Blending algebraic surfaces", *ACM Transactions on Graphics*, vol. 8, no. 4, pp. 263-278, 1989.

[12] A 'crude' conversion through sampled re-meshing is shown in [Men92].