

# Localized-Hierarchy Surface Splines (LeSS)

Carlos Gonzalez-Ochoa\*  
Department of Computer Science  
Purdue University  
West Lafayette, IN

Jorg Peters†  
Department of Computer and  
Information Science and Engineering  
University of Florida  
Gainesville, FL

## Abstract

An explicit spline representation of smooth free-form surfaces is combined with a hierarchy of meshes to form the basis of an interactive sculpting environment. The environment offers localized hierarchical modeling at different levels of detail, direct surface manipulation, change of connectivity for extrusion and to form holes and bridges, and built-in tangent continuity across the surface where wanted. The free-form surface is represented and can be exported either in NURBS form or as cubic triangular Bézier patches.

Key characteristics of the approach are: (1) mesh pieces and surface pieces are related by strictly local averaging rules; (2) refinement rules depend only on direct, coarser-level ancestors and not on adjacent submeshes or patches; (3) submeshes at different levels look alike. The underlying data structure is a single winged-edge structure with additional pointers to support the hierarchy. Multiply refined regions may be directly adjacent to unrefined regions, and mesh fragments at different levels of refinement can be connected.

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations, Splines; I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types

## 1 Motivation, results and prior work

Conventional modeling systems require precise specification of final features, e.g., distance in datum planes, positioning of control vertices with respect to one another, etc. Besides requiring a certain mindset not always compatible with creative, exploratory design, this is modeling at the highest level of refinement: a change in the overall shape of the model entails modifying all detail, however small, associated with the area of change. Not only is this time-consuming, but repeated detail work puts an undue strain on creative, exploratory design. *Parameterized modeling*, which allows initially precisely instantiated specifications to be varied, has been used in commercial solid modeling systems. A complementary paradigm that supports changes at different levels in the design

history is *hierarchical refinement*. Levels express the designer's intent by encoding the structure and dependence of geometric features. Several recent modeling environments are based on hierarchical modeling [10, 7, 21, 13].

Localized-hierarchy Surface Splines (LeSS) extend this modeling paradigm by combining localized mesh hierarchies with the ability to change topology by adding a handle or punching a hole. LeSS automatically maintains tangent continuity across the surface where wanted and defines the surface in standard form either as bicubic NURBS or cubic triangular Bézier patches. Non-standard properties of LeSS are:

1. The surface is represented explicitly in piecewise polynomial or spline form, and not as surface patches offset from ancestor surfaces. Finer-level patches *replace* coarse-level patches. Changes are propagated from coarse to fine inside an underlying mesh hierarchy: fragments (pieces with boundary) of finer meshes are represented as *offsets* in local coordinate frames of coarser meshes. This has several implications:
  - The evaluation of the spline patches does not require traversal of the hierarchy and is numerically stable.
  - Explicit bounding boxes and convex hulls for the surface are directly available and the moments of the enclosed object can be efficiently computed.
  - The influence of changing coarser levels is transmitted by local reevaluation of the mesh hierarchy.
  - Local topology changes are possible because there is no direct 1-1 correspondence of the coarse to the finer-level surface.
2. LeSS is based on a strictly *localized hierarchy*: the default position of a finer level mesh fragment before local editing is determined by local averaging rules that depend only on those coarser-level nodes from which the mesh nodes of the fragment are offset and not on adjacent mesh or patch information. This simplifies the data structures since no searches for neighbors at different levels of refinement are necessary. Repeatedly-refined patches can be placed directly next to unrefined patches. (Since the surface representation is explicit, the resolution of 'crack problems' where refined and unrefined patches meet can be delegated to the spline display routine.)
3. The representation is *homogeneous*: control structures everywhere, even across levels, look the same. Together with locality this homogeneity allows connecting arbitrary submeshes. This is useful since not all parts of a complex object must enter a strict dependence relation with all other parts – the designer's choice of dependence can be only local and later decisions, say to join parts or create a hole (Figure 9), are not prevented by a rigid, global hierarchy.

---

\* gonzalez@acm.org

† jorg@cise.ufl.edu

‡ Supported by NSF National Young Investigator award 9457806-CCR

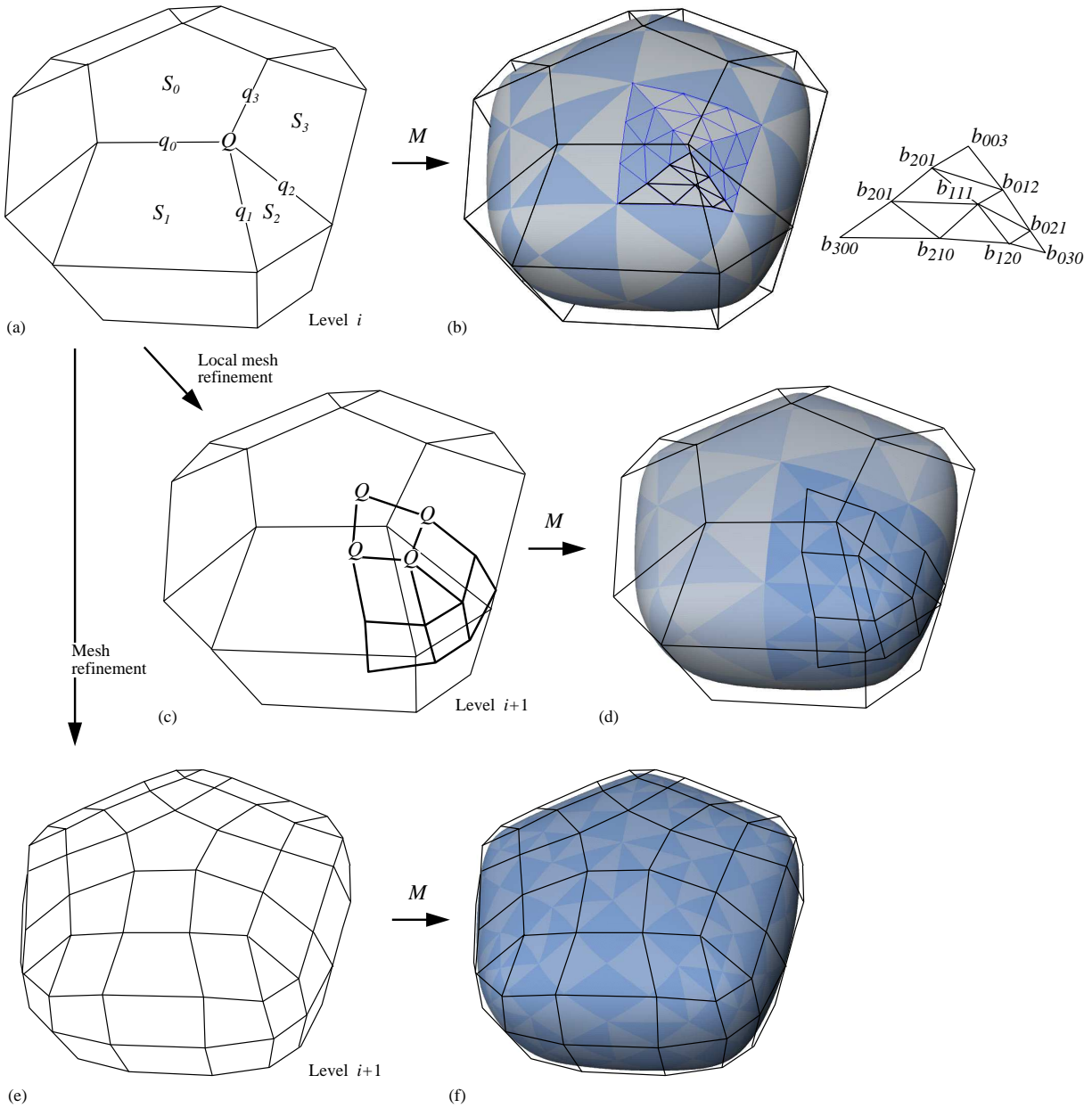


Figure 1: (a) Mesh with property **P**. The 9-point submesh with components  $Q$ ,  $S_i$ , and  $q_i$  defines four triangular patches in (b). To the right, the Bézier net and the Bézier coefficients  $b_{j,k,l}$  of one triangular patch are shown enlarged. The four triangular patches can alternatively be represented as one linearly-trimmed bicubic NURBS patch. Panels (e) and (f) show a globally refined mesh and the associated surface. In panel (c) just a fragment (piece) of the refined mesh is superimposed as a local offset and in (d) the associated, refined surface is seen replacing the coarser section of the surface. When modeling detail, the interior nodes of the refined mesh fragment are moved via direct surface manipulation (see Figure 6).

## 1.1 Prior work

Forsey and Bartels' Hierarchical B-Splines [10, 1, 9, 4] pioneered local refinement of smooth surfaces for computer-based geometric design. A model is locally refined by offsetting with a refined tensor-product B-spline basis; additional interior control points yield degrees of freedom at the refined level. Kraft [14] recently exhibited a basis of the refined space. The tensor-product B-spline basis does not address the problem of smoothly joining more or less than four patches at a mesh point. To extend the arrangement of patches to configurations other than the checkerboard while preserving smooth transitions, geometrically continuous bicubic spline complexes [18, 8] and generalized subdivision algorithms [6, 3, 16, 20] have been used as the underlying surface representation.

Multiresolution hierarchies [7, 21] extend the offset paradigm by making the update almost orthogonal or by distributing updates among coarser and finer levels with intermittent smoothing. For meshes without subdivision connectivity, i.e., where not every generic vertex is surrounded by six triangles, Kobbelt et al. [13] build a localized hierarchy of triangulations from a given triangulation by local mesh simplification, using smoothing as a simplification criterion.

Re-approximation of free-form, organic objects from dense triangle meshes can also be based on the industry standard NURBS representation [15].  $G$ -splines [12] provide smooth surfaces for arbitrary patch layout within the spline paradigm. A particular  $G$ -spline representation is Surface Splines [18] (not to be confused with an archaic name for thin plate splines).  $C^1$ -Surface Splines combine the ability to locally model smooth free-form surfaces with an explicit, least degree polynomial and NURBS representation. The surface representation comes with a shape and bounding box guarantee confining the surface to the local convex hull of the control mesh, i.e., predicting all points on the surface as a convex combination of the mesh nodes 'close by'. LeSS is based on  $C^1$ -Surface Splines.

The *structure of the paper* is as follows. Section 2 shows that certain polyhedral meshes define, by local averaging, smooth surface pieces that form a tangent continuous surface quilt. Both the mesh and the surface pieces can be refined by subdivision (Section 3). Since both refinement and conversion of the mesh to a surface piece are local operations, one can retain fragments (pieces) of the refined meshes as offsets of the coarse meshes. Section 4 describes how the hierarchy of fragments is maintained and converted into a quilt of quadrilateral surface pieces. Section 5 puts the hierarchical mesh and surface representation to work to modify the surface geometrically and topologically.

## 2 Special Meshes and surface quads

The surface representation of LeSS is based on a special class of meshes such that averaging of local mesh nodes defines smooth quadrilateral surface pieces, called *quads* in the following. Quads consist of low degree polynomial or NURBS patches and join automatically to form a smooth surface. Section 2.4 shows how arbitrary meshes are converted into special meshes.

### 2.1 Mesh with property P

A *mesh* is a set of *nodes* connected by line segments. The line segments are called *edges* and shortest circuits of distinct edges are called *facets*. To define meshes with property **P**, the following definitions are convenient. Let  $Q_i \in R^3$ ,  $i = 0, \dots, n-1$  be a sequence of points collected into the  $n \times 3$  matrix  $\mathbf{Q} := [Q_0, Q_1, \dots, Q_{n-1}]^T$ . Let  $\mathbf{1}$  be an  $n \times n$  matrix of ones and  $C$

the  $n \times n$  (rank 2, discrete Fourier transform) matrix with entries  $C(i, j) := 2 \cos(\frac{2\pi}{n}(j-i))$ . Then

$$P := (\mathbf{1} + C)/n$$

is of rank 3 and maps  $\mathbf{Q}$  into a subspace spanned by three of the points; that is, into a common plane, line or point. The map  $P$  is a linear projection since  $PP\mathbf{Q} = P\mathbf{Q}$ .

A mesh has property **P** if:

- (a) at most two facets abut at each edge,
- (b) each interior node  $Q_i$  has exactly four neighbors and
- (c) each  $n$ -sided facet with vertices  $Q_0, \dots, Q_{n-1}$  ( $n > 4$ ) satisfies  $\mathbf{Q} = P\mathbf{Q}$ .

Mesh nodes on a boundary of the mesh are not restricted by (b). Since  $P$  is a projection, property (c) can be enforced by applying  $P$  to the vertices of a non-planar facet (see Section 2.4 for the details). Property (c) implies that all facets with more than four vertices are planar and pass through the facet *centroid*

$$S := \frac{1}{n} \sum Q_j.$$

If a facet has only three vertices it is automatically planar and if  $n = 4$  a plane through  $S$  is defined by the two lines that connect midpoints of opposite edges of the facet even if, as will generally be the case, the facet is not planar.

Section 2.3 shows that facet centroids lie on the surface derived from the mesh. In fact, each centroid is a point where  $n$  quadrilateral surface pieces (quads) meet tangent to the centroid's facet. If all vertices of a facet lie on a line or point, the facet is called *collapsed* and the associated surface can be non-smooth. We collapse facets in our system to model creases and sharp points (Section 5.3, Figure 10).

### 2.2 Spline surface corresponding to a mesh with property P

LeSS *associates* one four-sided surface *quad* with each node  $Q$  of a mesh with property **P** (c.f. Figure 1 (a) and (b)). Let  $F_i$ ,  $i = 0, \dots, 3$  be the facets surrounding  $Q$  in counterclockwise orientation,  $S_i$  the centroid of  $F_i$ ,  $n_i$  the number of edges of  $F_i$ ,  $Q_{i-1}Q$  ( $i$  modulo 4) the edge preceding  $F_i$  in counterclockwise orientation (hence  $Q_{i-1}Q$  is the edge following), and  $q_i := (Q_i + Q)/2$ . The  $S_i$  will become the four corners of the quad.

Then each quad is determined by

- (a) the valences

$$\mathbf{n} := [n_0, \dots, n_3], \text{ and}$$

- (b) the *9-point submesh*

$$\mathbf{q} := [Q, S_0, S_1, S_2, S_3, q_0, q_1, q_2, q_3]^T.$$

For example, in Figure 1(a),  $\mathbf{n} = [5, 4, 3, 4]$ . Even though  $\mathbf{n}$  and  $\mathbf{q}$  depend only on the mesh in a small neighborhood of  $Q$  the quads associated with adjacent nodes join without overlap and form a tangent-continuous surface (except where facets are collapsed).

Each quad is a spline piece. Below we describe the spline as four *cubic triangular patches* in Bernstein-Bézier form. This representation is advantageous for a user wanting full control over graphics and geometry: triangular patches naturally yield triangle strips and have tight bounding structures in the form of a refinable control triangulation. LeSS can convert and output such a quad in the industry standard: by rotating the domain by  $45^\circ$ , basis conversion, and trimming the domain back from a square to a diamond shape, the four cubic patches are exactly represented as one *linearly-trimmed bicubic NURBS patch*. The C-language routine `pcp2nurbs` for converting  $\mathbf{n}, \mathbf{q}$  to a quad in NURBS form is available via *netlib* (see also [19]).

### 2.3 Explicit formulas relating mesh and surface

The quad *associated with*  $Q$  (c.f. Figure 1) consists in the triangular patch description of *four cubic patches in Bernstein-Bézier form*. A triangular cubic patch  $p(u, v)$  with coefficients  $b_{ijk}$ , each a point in 3-space, is in Bernstein-Bézier form if

$$p(u, v) = \sum b_{ijk} B_{ijk}(u, v) \quad \text{where}$$

$$B_{ijk}(u, v) := \frac{3!}{i!j!k!} u^i v^j (1-u-v)^k,$$

$$i + j + k = 3, \quad \text{and } i, j, k \geq 0.$$

We enumerate the coefficients in the order

$$\mathbf{b} := [b_{300}, b_{210}, b_{120}, b_{030}, b_{201}, b_{111}, b_{021}, b_{102}, b_{012}, b_{003}]^T.$$

By treating the indices modulo 4, we need only describe the conversion of  $\mathbf{n}, \mathbf{q}$  to the coefficients  $\mathbf{b}$  of one of the four triangular cubic patches, say the patch  $p_1$  (with edge  $S_1, S_2$ ) shown enlarged to the right of Figure 1(b) – the others are obtained by simultaneous cyclical permutation of the  $S_i$  and  $q_i$  entries of  $\mathbf{q}$ ,  $\alpha_i, \beta_i$  and  $\gamma_{i,i+2}$  entries of  $M$ , and entries of  $\mathbf{n}$ . The coefficients of  $p_1$  are

$$\mathbf{b} := M\mathbf{q},$$

where  $M$  is the matrix

$$M := \frac{1}{12} \begin{bmatrix} 0 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 4 & 4 & 0 & 0 & 0 \\ 4 & 0 & 2\beta_2 & 2\beta_1 & 0 & 0 & 4\alpha_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 4 & 4 & 0 & 0 \\ 4 & \beta_1 & 2\gamma_{0,2} & \beta_1 & 0 & 2\alpha_0 & 2\alpha_1 & 0 & 0 & 0 \\ 4 & 0 & \beta_2 & 2\gamma_{1,3} & \beta_2 & 0 & 2\alpha_1 & 2\alpha_2 & 0 & 0 \\ 4 & \gamma_{1,3} & \gamma_{0,2} & \gamma_{1,3} & \gamma_{0,2} & \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & 0 \end{bmatrix}$$

and

$$c_i := \cos\left(\frac{2\pi}{n_i}\right) \in \left[-\frac{1}{2}, 0\right), \quad \alpha_i := 1 - \frac{c_i + c_{i+1}}{2},$$

$$\beta_i := 1 + c_i, \quad \gamma_{i,i+2} := 1 + \frac{c_i + c_{i+2}}{2}.$$

Application of the matrix  $M$  may be interpreted as a basis conversion to Bézier form. Indeed, where a node  $Q$  is surrounded by exactly four quadrilateral facets, all  $c_i = 0$ , and  $M\mathbf{q}$  are the coefficients of the (degree-raised, cubic) Bézier representation of the *quadratic box spline* whose coefficients are the nodes  $Q$ . Since all entries of  $M$  are non-negative and rows sum to one, it is evident that the surface lies in the convex hull of the 9-point submesh, a fact useful for intersection testing after modifying  $\mathbf{q}$ . The quads join smoothly where the facets are 2-dimensional [18].

### 2.4 Mesh initialization

To get a base mesh with property  $\mathbf{P}$  from a general mesh, we  
(A) insert edges in the general mesh so that no vertex with more than 4 neighbors is attached to a facet with more than 4 edges;  
(B) apply one mesh refinement step (see Section 3.1), and  
(C) apply  $P$  to the vertices of any facet with  $n > 4$  vertices.  
Steps A and B separate large ( $n > 4$ ) facets and thus guarantee that no two projections in step 2 interfere with one another. Instead of step A one could apply mesh refinement twice – but that creates unnecessarily many facets. LeSS does not refer back to the general mesh and considers it discarded after creation of the base mesh. (We are currently evaluating a modified surface representation to eliminate the need for steps A and B and work directly on general meshes.)

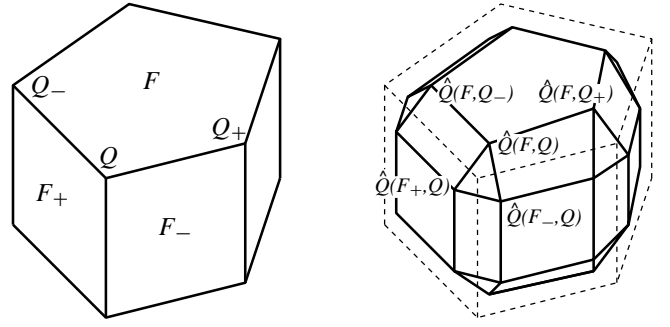


Figure 2: Initialization of the mesh with property  $\mathbf{P}$  (right, solid lines) and nodes  $\hat{Q}$  from a general mesh with nodes  $Q$ .

## 3 Mesh and surface refinement

This section presents two refinement schemes. Both are based on local operations so that we can retain and modify *fragments*, contiguous pieces-with-boundary of the refined entities, to model regions with higher detail (c.f. Section 4). For better exposition, this section describes the refinements in their simpler, unlocalized form. The first scheme, *mesh refinement*, is similar to the Doo-Sabin scheme [6] and creates a finer mesh  $\mathcal{M}_i$  from a coarser mesh  $\mathcal{M}_{i-1}$ . Mesh refinement is the basis of the hierarchy and preserves property  $\mathbf{P}$ . The second, *surface refinement*, subdivides surface patches in Bézier form by de Casteljau's algorithm. The resulting collection of finer triangular cubic patches represents the same surface but at a finer scale. Surface refinement is used in Section 4.2 to complete quads on the boundary of a fragment and ensure smooth transitions across different levels of the hierarchy. Surface refinement is also an efficient way to approximately evaluate the surface for display and intersection testing [17].

### 3.1 Mesh refinement

Let  $F$  be a mesh facet of  $\mathcal{M}_{i-1}$  with  $n$  vertices,  $c := \cos(2\pi/n)$ ,  $Q_-, Q$  and  $Q_+$  three consecutive vertices of  $F$  (c.f. Figure 2 left), and  $F_-$  and  $F_+$  the predecessor facet and successor facet when listing the facets surrounding  $Q$  in counterclockwise order. The refinement of the coarser level mesh  $\mathcal{M}_{i-1}$  to a finer level mesh  $\mathcal{M}_i$  consists of the following averaging and reconnection steps.

1. *Location*: For each pair  $F, Q$  of  $\mathcal{M}_{i-1}$  we create one new node  $\hat{Q}(F, Q)$ , with *default location*,

$$\hat{Q}(F, Q) := (1-2w)Q + wQ_- + wQ_+, \quad w^{-1} := 4(1-c).$$

2. *Connectivity*: Connect  $\hat{Q}(F, Q)$  to the four neighbors  $\hat{Q}(F_-, Q), \hat{Q}(F, Q_+), \hat{Q}(F, Q_-), \hat{Q}(F_+, Q)$  in order.

The new mesh again has property  $\mathbf{P}$ : (a) and (b) are evident and (c) holds since the facets of the refined mesh are either quadrilaterals or agree with facets of the old mesh scaled by  $1/2$ . The rule for connecting the new nodes is identical to the rule for Doo-Sabin subdivision, but differs in the location of the new nodes.

To motivate why the smooth transition between levels requires the work detailed in Section 4.2 we point out that, while surfaces defined by a mesh and its refinement via Section 2.3 (see Figure 1 (b) and (f)) may be visually indistinguishable, they are nevertheless not identical. To see this recall that the formulas of Section 2.3 applied to *regular submeshes*, i.e. a node surrounded by four quadrilaterals, generate *quadratic* quads represented in cubic form; that is,

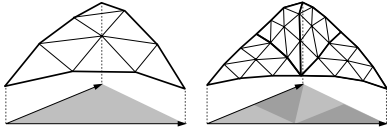


Figure 3: The domain of a cubic triangular Bézier patch and its corresponding control points. Each surface refinement splits each edge at  $1/2$  and the domain into four sub-domains.

if one differentiates the patches twice, the result is constant rather than linear. The surfaces in Figures 1 (b) and (f) differ in the quads corresponding to a coarse-level facet with more or less than four neighbors. Here, the genuinely cubic coarse-level surface quads are replaced by an inner ring of cubic and an outer ring of quadratic quads, since the refined mesh has regular submeshes away from the scaled original facets.

### 3.2 Surface refinement

We use parts of refined surface quads in Section 4.2 to guarantee a smooth transition between different levels of the surface. We refine the triangular patches defined in Section 2.3 by splitting each edge at  $1/2$  (see Figure 3) using De Casteljaeu’s algorithm in two variables (which is efficiently implemented by add-and-shift operations [17]). Since the patch subdivision leaves the surface unchanged but creates Bézier control nets that model the surface more and more closely, we can also use the same routine for approximately evaluating and displaying the surface (see also Section 5.2, picking). In particular, if  $k_0 < k_1$  are the mesh refinement levels of adjacent quads, then applying patch refinement  $k_1 - k_0$  times to the quad associated with the coarser-level node results in a joint Bézier control net without cracks. However, in general, this is less efficient than adaptive evaluation.

## 4 Localized mesh and surface hierarchy

Since refinement and conversion of the mesh to surface quads are local operations, LeSS can retain just *fragments* of meshes in regions where a refined representation is wanted for later manipulation (see Section 5). A mesh fragment is a piece of a refined mesh. Figure 1(c,d) shows a mesh fragment overlaid on the coarser mesh. The fragment nodes are offsets of coarser-level nodes and, by recursion, ultimately of the base mesh.

This section describes how the hierarchy of fragments is maintained and, at the finest-level, converted into a quilt of surface quads. To generate the quadrilateral pieces or quads that make up the surface, the localized hierarchy of mesh fragments is traversed and the mesh offsets added until a mesh node without finer-level descendants is encountered. At such a *terminal node* the 9-point mesh defining the quad is collected either by visiting the immediate neighbors at the same level or accessing information stored at the immediate coarser-level ancestor node.

### 4.1 Data structures

The data structures need to support two operations to define the surface quads associated with the terminal nodes:

- accessing the neighbor nodes and
- traversing the hierarchy, adding offsets (see Section 5.1).

To access neighbors LeSS uses a winged-edge structure [2]. To traverse the hierarchy, each coarser-level node  $Q$  is linked, by pointers that are not part of the winged-edge structure, to the finer-level

nodes  $\hat{Q}(F_i, Q)$  where  $F_i$  is a refined coarser-level facet (see Section 3.1).

(1) *Connectivity within the mesh:* Since the base mesh and all fragments are the result of mesh refinement, each node  $\hat{Q}$  has at most four neighbors  $\hat{Q}_i$  and at least two consecutive neighbors, namely  $\hat{Q}(F, Q_-)$  and  $\hat{Q}(F, Q_+)$ . One or both of the neighbors  $\hat{Q}(F_-, Q)$  and  $\hat{Q}(F_+, Q)$  may not be available because we are at the boundary of a fragment and the adjacent surface has not been refined (is associated with coarser-level nodes). If a node has all neighbors (four nodes and centroids) listed in  $\mathbf{q}$ , it is an *interior* node, otherwise a *boundary* node. When a mesh (fragment) is refined, the neighbors of the corresponding coarse-level nodes are checked for descendants. If descendants exist (because the neighbor is part of a more refined fragment), immediate descendants of both coarser-level nodes are linked so that a boundary node automatically becomes an interior node when all neighbor fragments are equally refined.

(2) *Patch information stored with boundary nodes.* When a new boundary node  $\hat{Q}(F, Q)$  is created as part of a refined fragment the quad associated with  $Q$  is refined according to Section 3.2 and parts of the refined patches are stored with the boundary node: if the  $i$ th neighbor is missing, we store the Bézier coefficients  $b_{ljk}$  for  $k \leq 1$  of patch  $i$  (c.f. Section 2.3). The valence  $n$  of  $F$  is also stored.

### 4.2 Localized hierarchical surface representation

For each interior node of a fragment, the 9-point submesh  $\mathbf{q}$  and hence the associated surface quad is easily assembled from the neighboring nodes and facets. Since boundary nodes lack neighbors, LeSS maintains and inserts additional information to assemble *boundary quads* (Steps 1–3 below) and guarantee a smooth transition across the *border* between coarse and fine. (Recall that in contrast to displacement mapping the surface is not represented as a surface offset from a coarser-level surface. Rather, coarser-level quads are replaced by a collection of finer-level quads.)

Figure 4 illustrates the assembly of boundary quads for a minimal, refined surface region. The associated fragment of the refined mesh is not shown – it consists of just four boundary nodes  $Q_0, \dots, Q_3$  arranged as a quadrilateral and positioned approximately at each meeting point of four small triangular patches. All four boundary nodes are descendants of a coarser-level node  $Q'$  that is positioned close to the intersection of the eight small triangular patches; its coarse-level neighbors are where four large triangular patches meet. Since LeSS keeps boundary nodes fixed, a minimal refined fragment is not terribly useful by itself but serves as a building block for repeated refinement.

*Step 1:* Even for a boundary node some entries of  $\mathbf{q}$  can be collected, and those coefficients  $b_{ijk}$  (o in Figure 4) can be computed whose corresponding rows of  $M$  have non-zero entries only where the entries of  $\mathbf{q}$  are available. In Figure 4 the average  $S_1$  ( $S_2$  after cyclical permutation) of the  $Q_i$  and the average  $q_1$  ( $q_2$ ) of  $Q_i$  and  $Q_{i+1}$  are available and allow computing the Bézier coefficients corresponding to rows 1 and 2 of  $M$  (rows 3 and 4). Since, by continuity, several of the coefficients agree, only nine Bézier coefficients o are visible – at the center of the refined surface.

*Step 2:* For each boundary node of the refined mesh we have already stored the coefficients (• in Figure 4)  $b_{ijk}$  for  $k \leq 1$  of patch  $i$  (see Section 4.1, item (2)). Thus the boundary quad inherits the tangent planes of the quad associated with the coarser-level node ( $Q'$  in the example) along the border and the surface is as smooth as before adding the fragment offset.

*Step 3:* Consider now the transition between two boundary quads associated with  $Q$  and a neighbor  $Q_1$  respectively. We determine

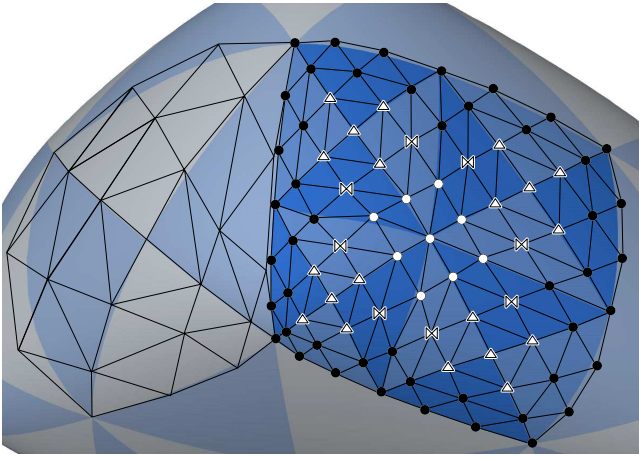


Figure 4: Bézier patches and their control net at the transition between coarse and fine. Filled circles  $\bullet$  represent coefficients derived from patch refinement. Hollow circles  $\circ$  are computed from the refined mesh by applying  $M$ . The position of the bow ties  $\bowtie$  and triangles  $\triangle$  are defined in Section 4.2, Step 3. The nodes of the coarse and fine meshes are not shown. Each would be located approximately at a meeting point of four triangular patches.

the remaining coefficients  $b_{ij}^1$  of patch  $p_1$  (see Section 2.3). For simplicity we drop the superscript 1. The coefficients  $b_{ij}^0$ ,  $k \leq 1$  of  $p_0$  are known from Step 2; so we set  $b_{300} = b_{030}^0$ ,  $b_{201} = b_{021}^0$ ,  $b_{210} = 2b_{021}^0 - b_{120}^0$  and  $b_{030}$ ;  $b_{021}$  and  $b_{120}$  have been computed in Step 1. To compute  $b_{111}$  ( $\bowtie$  in Figure 4) we collect the stored valence  $n$  (Section 4.1(2)),  $c := \cos(\frac{2\pi}{n})$ ,  $S_1 = b_{300}$ ,  $S_2 = b_{030}$  and  $q = (Q + Q_1)/2$ . If  $Q$  and  $Q_1$  have the same ancestor node then

$$b_{111} := \frac{b_{210} + b_{120}}{2} + \frac{Q - q}{3}.$$

Otherwise, we apply row 6 of  $M$ :

$$b_{111} := \frac{S_1 + (2 - c)q + 2Q + (1 + c)S_2}{6}.$$

The remaining coefficients  $\Delta$  in the quad are given by

$$b_{102}^i = \frac{b_{111}^i + b_{111}^{i-1}}{2}, \quad b_{003}^i = \sum_{j=1}^4 \frac{b_{111}^j}{4}, \quad i = 0, \dots, 3.$$

## 5 Local mesh and surface modification

This section uses the machinery developed in the previous three sections to support interactive modeling by direct manipulation.

### 5.1 Offsetting refined nodes

As in approximate displacement mapping [1, 9] the position of each node of a refined mesh fragment is stored as an *offset*  $V_k$  from the default position  $\hat{Q}_k$  prescribed by the refinement process of Section 3.1. Consider consecutive vertices  $Q_{k-1}$ ,  $Q_k$ ,  $Q_{k+1}$  of a facet at the coarse level and a vector  $N$  normal to the facet. If the facet is 2-dimensional then  $N$  is uniquely determined up to scaling. Otherwise, we compute  $N$  as an average of the adjacent facet normals. Then the offset

$$V_k := \mu_0 N + \mu_1(Q_{k-1} - Q_k) + \mu_2(Q_{k+1} - Q_k)$$

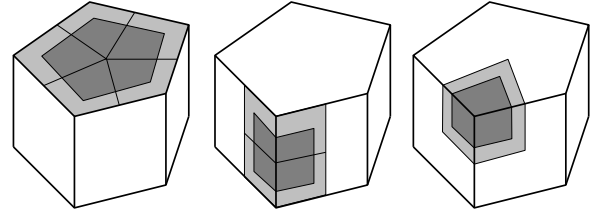


Figure 5: Surface refinement around a face, edge and vertex respectively with boundary (light grey) and interior regions (dark grey) of the refined mesh.

is expressed in a local coordinate frame. When an underlying coarser-level mesh changes, the default position  $\hat{Q}_k$  is recomputed and the offset  $V_k$  added anew. The new location generates a new default position for the descendants and the change percolates to a terminal node at the finest level. There a new local submesh  $\mathbf{q}$  and hence a new surface quad is computed. LeSS keeps boundary nodes fixed and captures finer detail exclusively by modifying (the offset of) interior nodes obtained from refining a ring around a vertex, edge or face (Figure 5).

### 5.2 Direct Manipulation

Parametric surfaces can be manipulated via *control points*, here the nodes  $Q$ . Although their number in LeSS is far smaller than the number of Bézier coefficients, modeling a complex object with such hierarchically arranged control handles is tricky because the meshes obstruct the object. A more intuitive approach is to pick a point on the surface and move it and the surrounding surface directly [10, 11]. Since we display the surface patches as triangle strips resulting from iterated patch refinement, we readily obtain (from the display list and the index into the list returned by OpenGL picking) the patch  $p$  and the parameters  $(u, v)$  of the point  $p(u, v)$  to be moved. To move the point to  $p(u, v) + \Delta$ , LeSS inverts the Surface Spline construction and adjusts the node  $Q$  of the submesh  $q$  that generates  $p$ .

Given  $(u, v)$ , we compute the vector of basis functions each *evaluated* at  $(u, v)$ :

$$\mathbf{B} := [B_{300}, B_{210}, B_{120}, B_{030}, B_{201}, B_{111}, B_{021}, B_{102}, B_{012}, B_{003}].$$

The ordering of the subscripts is the same as for  $\mathbf{b}$  so that  $p(u, v) = \mathbf{B}\mathbf{b}$ . If  $Q$  is moved by  $(1, 1, 1)$  then  $\mathbf{q}$  changes by

$$d := \left[ 1, \frac{1}{n_0}, \frac{1}{n_1}, \frac{1}{n_2}, \frac{1}{n_3}, a_0, a_1, a_2, a_3 \right]^T$$

in each coordinate, where  $n_i$  is the number of vertices of the  $i$ th facet at  $Q$  and

$$a_i := \frac{1}{2} + \begin{cases} \cos(2\pi/n_i) & \text{if } n_i > 4 \\ \cos(2\pi/n_{i+1}) & \text{if } n_{i+1} > 4 \\ 0 & \text{else.} \end{cases}$$

Given the vectors  $\mathbf{B}$  and  $d$ , we compute the scalar

$$\delta := \mathbf{B}M d,$$

the amount by which each coordinate of  $p(u, v)$  moves if  $Q$  moves by 1 in each coordinate. Since  $Md$  depends only on  $\mathbf{n}$  it can be preassembled. The amount  $\delta$  depends on  $(u, v)$  via  $\mathbf{B}$ . Since all



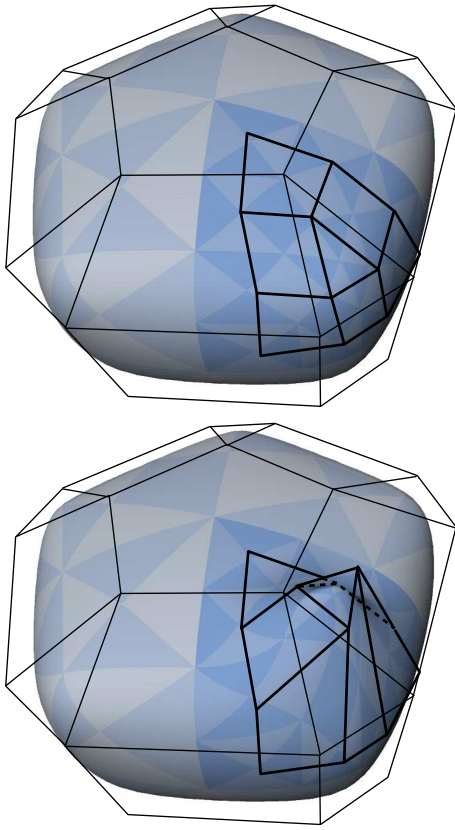


Figure 6: Local surface and mesh fragment update.

entries of  $d$  are strictly positive and  $M$  has no negative and at least on strictly positive entry in each row, the control points  $Md$  are strictly positive and thus  $\delta > 0$ . This gives an explicit algorithm and proves that to interpolate the point  $p(u, v) + \Delta$ , the associated  $Q$  must be offset by  $\Delta/\delta$ . Figure 6 illustrates how the mesh mimics the motion of the surface.

### 5.3 Fillets, blends and semi-sharp creases

LeSS also includes tools to define fillets, blends and semi-sharp creases of controllable sharpness. The underlying ideas have been explained in [18] and more recently in [5]. The tools are based on varying the weight  $w$  in the mesh refinement step 3.1. The default choice of  $w$  as  $1/w = 4(1 - c)$ , where  $c = \cos(\frac{2\pi}{n})$  and  $n$  is the number of vertices of the old, coarse-level facet, creates for each old facet a new facet with the same centroid and orientation but half the diameter. If instead, the tool (interface) sets  $w = \bar{w}$  where  $1/\bar{w} := 2(1 - c)$  then the facet shrinks to a single point resulting in a sharp corner in the surface. If adjacent facets are refined using  $w = \bar{w}$  then a sharp edge is formed. If  $w$  is repeatedly zero, the result is a flat spot. Any modification of  $w$  in  $[0, \bar{w}]$  varies the distribution of curvature.

### 5.4 Extrusions, holes and bridges

We observe that the interior of any mesh fragment can be changed freely: we are not restricted to just modifying offsets, but can modify the connectivity of the interior mesh by any Euler operator. Provided property  $\mathbf{P}$  is preserved, all the machinery of the previous

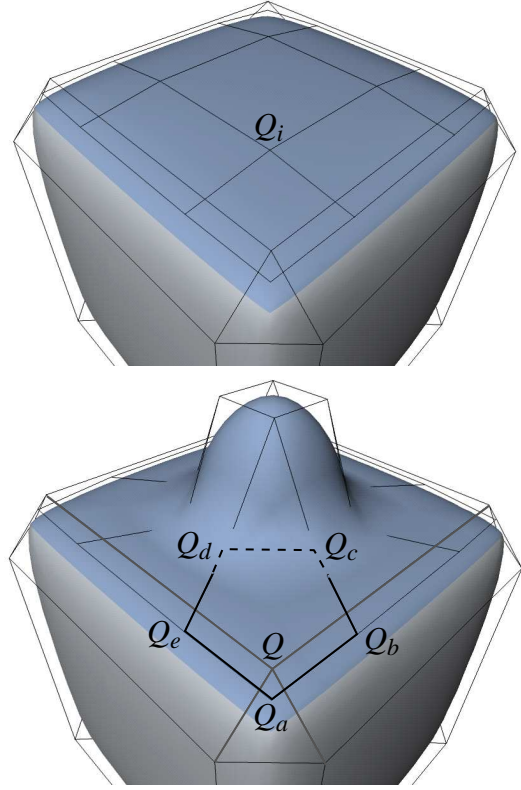


Figure 7: Extrusion of a refined mesh:  $Q_i$  is replaced by  $Q_c$  and  $Q_d$ . Keeping  $Q_a$ ,  $Q_b$  and  $Q_e$  fixed at the transition between coarse and fine levels, the choice of the golden ratio  $\omega = (1 + \sqrt{5})/2$  and  $Q_c = Q_e + \omega(Q_b - Q_a)$ ,  $Q_d = Q_b + \omega(Q_e - Q_a)$  preserve the planarity property.

sections applies to yield a smooth surface where wanted.

Extrusions, holes and bridges are based on mesh connectivity changes (see Figures 7, 8 and 9). A bridge (or a hole) is obtained by connecting two extrusions: just like two spacecraft docking a (4-faced) cylinder is placed between the “top” facets of the extrusions and then the internal facets are removed (Figure 8). As the analogy would indicate when one spacecraft is part of a larger assembly, it does not matter to what refinement level the newly joined meshes belong since all operations are local (see Figure 9).

Thus it suffices to discuss extrusions. Extrusions can be built into the mesh (Figure 8 top) or result from local refinement: Figure 7 shows how the interior node  $Q_i$  of a quadrilateral of the refined mesh is replaced by two nodes  $Q_c$  and  $Q_d$ . The two new nodes are placed subject only to the planarity condition which is satisfied if the new nodes are

$$\begin{aligned} Q_c &= Q_e + \omega(Q_b - Q_a), \\ Q_d &= Q_b + \omega(Q_e - Q_a), \end{aligned}$$

and  $\omega$  is the golden ratio  $(1 + \sqrt{5})/2$ . The two new nodes are added as descendants of the parent node  $Q$ . (In Figure 7 this replacement of one point by two has been carried out four times).

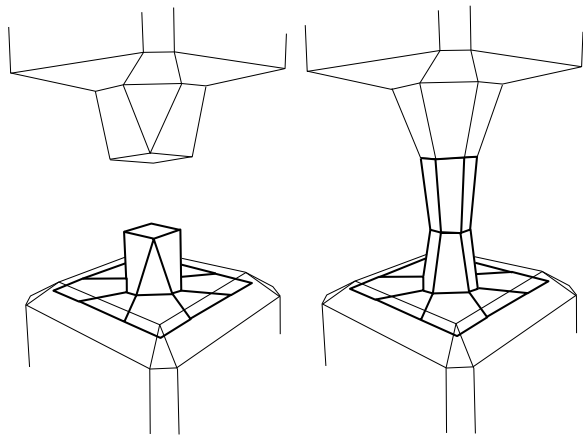


Figure 8: Connecting two extrusions to form a bridge. The lower extrusion is generated by refinement as in Figure 7, while the upper is an original mesh. The bridge mesh retains property **P**.

## 6 Implementation and Examples

Written in C++, LeSS uses Open Inventor for its 3D GUI. It has been ported to SGI and PC platforms. LeSS was developed as a custom modeler to test different surface representations. Presently, a quartic representation is being tested to see whether the removal of the planarity condition warrants a higher degree. With the availability of succinct higher-order smooth surface representations,  $C^1$ -Surface Splines can be replaced or complemented.

The figures were created on a SGI Indigo 2 High Impact. This machine configuration allows interactive modeling at 8 fps or better of objects consisting of 5000 patches. Figure 10 illustrates a modeling session of a complex character.

*Acknowledgment:* Our thanks to John Hughes and the referees for valuable pointers on how to improve the presentation.

## References

- [1] C. Barghiel, R. Bartels, and D. Forsey. Pasting spline surfaces. In P. Schröder, editor, *Subdivision for modeling and animation*, number 36 in Course notes of 25th International conference on Computer Graphics and Interactive Techniques. ACM Siggraph, July 1998.
- [2] Bruce G. Baumgart. A polyhedral representation for computer vision. *Proceedings National Computer Conference*, 1975.
- [3] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10, September 1978.
- [4] Leith Kin Yip Chan, Stephen Mann, and Richard Bartels. World space surface pasting. In Wayne Davis, Marilyn Mantel, and Victor Klassen, editors, *Graphics Interface*, May 1997.
- [5] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, July 1998.
- [6] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10:356–360, September 1978.
- [7] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, August 1995.
- [8] Matthias Eck and Hugues Hoppe. Automatic reconstruction of B-Spline surfaces of arbitrary topological type. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, August 1996.
- [9] David Forsey and Richard H. Bartels. Surface fitting with hierarchical splines. *ACM Transactions on Graphics*, 14(2), April 1995.
- [10] David R. Forsey and Richard H. Bartels. Hierarchical B-spline refinement. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, August 1988.
- [11] Barry Fowler. Geometric manipulation of tensor product surfaces. In David Zeltzer, editor, *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, volume 25, March 1992.
- [12] K. Höllig and H. Mögerle. G-splines. *Computer Aided Geometric Design*, 7(1-4), 1990.
- [13] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH 98 Conference Proceedings*. ACM SIGGRAPH, Addison Wesley, July 1998.
- [14] Rainer Kraft. *Adaptive und linear unabhängige multilevel B-splines und ihre Anwendungen*. PhD thesis, Math Inst A, University of Stuttgart, 1998.
- [15] Venkat Krishnamurthy and Marc Levoy. Fitting smooth surfaces to dense polygon meshes. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, August 1996.
- [16] Charles Loop and Tony D. DeRose. A multisided generalization of Bezier surfaces. *ACM Transactions on Graphics*, 8(3), July 1989.
- [17] J. Peters. Evaluation and approximate evaluation of the multivariate Bernstein-Bézier form on a regularly partitioned simplex. *ACM Transactions on Mathematical Software*, 20(4), December 1994.
- [18] J. Peters.  $C^1$ -surface splines. *SIAM Journal on Numerical Analysis*, 32(2), 1995.
- [19] J. Peters. Algorithm 783: Pcp2Nurb — smooth free-form surfacing with linearly-trimmed bicubic B-splines. *ACM Transactions on Mathematical Software*, 24:261–267, 1998.
- [20] J. Peters and U. Reif. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics*, 16(4), October 1997.
- [21] Denis Zorin, Peter Schröder, and Wim Sweldens. Interactive multiresolution mesh editing. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, August 1997.



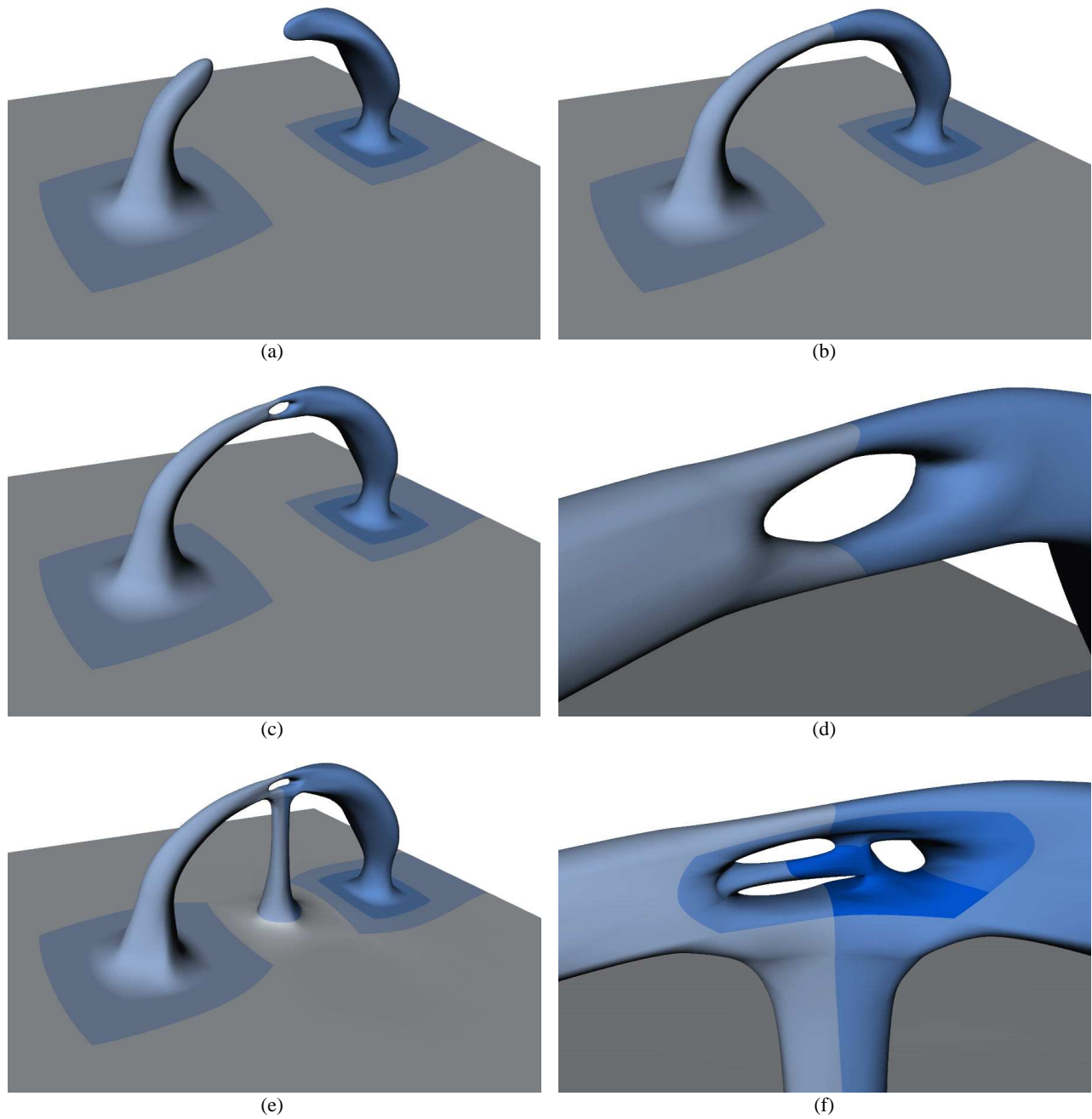


Figure 9: Finer regions in the hierarchy correspond to darker areas. The original plane has been (a) extruded and is reconnected to form a bridge (b). (c,d) A hole is created in a similar fashion. Note that the hole is created at the join of different levels of the hierarchy. (e) Further extrusion and attachment to the base level and (f) playful refinements and attachments within different levels.

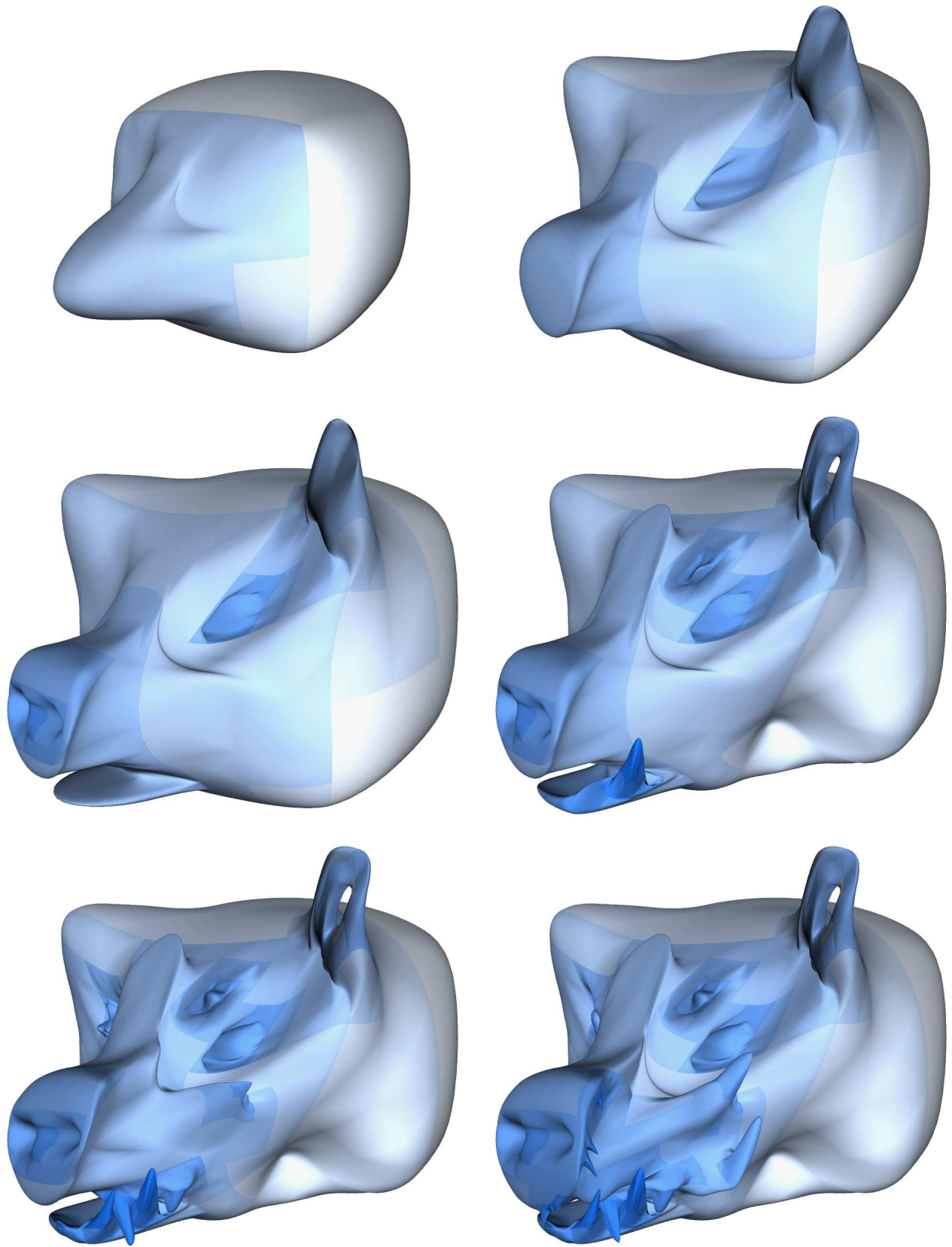


Figure 10: Modeling steps of a complex model that started from a simple cube. Finer regions in the hierarchy correspond to darker areas. Note the hole in the ear and the small details.