Multiresolution Analysis for Surfaces of Arbitrary Topological Type

Tony D. DeRose Michael Lounsbery Joe Warren[1]
Department of Computer Science and Engineering, FR-35
University of Washington
Seattle, WA. 98195

Technical Report Number 93-10-05

October 29, 1993

Abstract

Multiresolution analysis provides a useful and efficient tool for representing shape and analyzing features at multiple levels of detail. Although the technique has met with considerable success when applied to univariate functions, images, and more generally to functions defined on $\mathbb{R}^n$, to our knowledge it has not been extended to functions defined on surfaces of arbitrary genus.

In this report, we demonstrate that multiresolution analysis can be extended to surfaces of arbitrary genus using techniques from subdivision surfaces. We envision many applications for this work, including automatic level-of-detail control in high-performance graphics rendering, compression of CAD models, and acceleration of global illumination algorithms. We briefly sketch one of these applications, that of automatic level-of-detail control of polyhedral surfaces.

1 Introduction

Multiresolution analysis and wavelets have received considerable attention in recent years, fueled largely by the diverse collection of problems that benefit from their use. The basic idea behind multiresolution analysis, or equivalently, the discrete wavelet transform, is to represent a complicated function as a comparatively simple low resolution function, together with a collection of perturbations — at increasing levels of detail — necessary to recover the original function.

In the simplest context, multiresolution analysis considers the representation of scalar-valued functions defined on the entire real line (that is, functions $f : \mathbb{R} \rightarrow \mathbb{R}$). In many applications, however, the function of interest is defined only on a portion of the real line, leading researchers

[1] Work done while on sabbatical from Rice University

to extend the theory to functions defined on the unit interval [11]. Parametric curves $Q(t) = (x(t), y(t)), t \in [0, 1]$ can then be represented by applying the analysis independently to each of the coordinate functions $x(t)$ and $y(t)$. Multiresolution analysis of parametric curves has a number of important applications, including: compression of curves, adaptive data fitting, flexible editing [12], and acceleration of the tiling subproblem in computing surfaces from contours [10].

The extension of multiresolution analysis to functions defined on the unit square is relatively straightforward, and has numerous applications. Among these are the compression of images and operations on tensor product surfaces, such as flexible editing and compression. To achieve similar benefits for surfaces of arbitrary topological type, multiresolution analysis must be extended to functions defined on 2-dimensional domains of the same topological type. For instance, a parametric surface shaped like a deformed sphere can be thought of as a vector-valued function $S(p) = (x(p), y(p), z(p))$ where p varies on a domain (such as an octahedron) which is homeomorphic to a sphere. By analogy to the case of curves, it is reasonable to expect that the shape defined by $S(p)$ could be effectively analyzed and manipulated if it were possible to apply multiresolution analysis to $x(p)$, $y(p)$, and $z(p)$.

In this report, we show that by using techniques from subdivision surfaces, multiresolution analysis can be extended to functions defined on domains of arbitrary topological type. Such an extension is far from trivial, as several challenges must be overcome. In particular, multiresolution analysis, as usually formulated, relies on the notions of dilation and translation of functions. Both properties are difficult to extend to arbitrary topological domains. We show that the use of subdivision surfaces allows a reformulation of multiresolution analysis in a way that avoids the use of dilation and translation.

We believe that multiresolution analysis for surfaces of arbitrary topological type is well suited to a wide variety of applications. These include:

- Automatic level-of-detail control. When a complex shape is rendered in an animation, a fully detailed representation of the shape contains much more detail than is required for all but the closest view. Using a compressed wavelet representation of complex objects, it is possible to greatly reduce the number of polygons in a scene without significantly impacting the visible detail. This has the potential to greatly speed up the rendering of a scene. We briefly present this application in Section 4.

- Shape compression. Multiresolution representations for surfaces can be used as a basis for lossy shape compression, in complete analogy to lossy image compression techniques also based on wavelets.

- Radiosity and global illumination. For a scene rendered with radiosity techniques, the interaction of every polygon with every other polygon must be considered. Because of this, the processing time dramatically increases with the number of polygons used to model the surfaces being rendered. For many types of scenes it is unnecessary to consider all pairs of polygonal interactions. For instance, if two highly detailed objects are separated by a large distance, it is clearly overkill to consider the interaction of all pairs of polygons on each object. By storing a multiresolution representation for each of the objects, it should be possible to greatly reduce the number of significant interactions. In a sense, this use of multiresolution representations can be thought of as a clustering mechanism for use in hierarchical radiosity [7].

Our goal in this report is to describe the theory underlying multiresolution analysis for surfaces of arbitrary topological type. Although one application of the theory – level-of-detail control for polyhedral models – is briefly described in Section 4, implementation details and further applications will be treated in subsequent papers.

In Section 2, we briefly survey the basics of multiresolution analysis and subdivision surfaces. To avoid undue abstraction, we restrict our discussion to subdivision surfaces defined over triangles. In general, however, the subsequent theory can be applied to other types of subdivision surfaces, such as those of Doo-Sabin and Catmull-Clark, defined over quadrilaterals. In Section 3, we show how subdivision surfaces can be used as a basis for multiresolution analysis. In Section 4, we briefly present the results of our theory applied to a complicated polyhedral subdivision surface. Finally, in Section 5, we summarize our results and list several avenues of future work.

2 Background

2.1 Multiresolution analysis

The theory of *multiresolution analysis* was originally developed by Mallat [9] to represent functions defined over $\mathbb{R}^n$. In the following brief survey, we deviate somewhat from Mallat's original formulation and use instead a formulation more appropriate for spline wavelets [1].

Let $\phi(x) : \mathbb{R} \to \mathbb{R}$ be a real-valued *refinable scaling function*; that is, ϕ is such that there exist constants p_i such that

$$\phi(x) = \sum_i p_i \phi(2x - i).$$

Translates and dilations of $\phi(x)$ are used to define a sequence of linear function spaces; specifically,

$$V^j = \text{Span}(\phi(2^j x - i)_{i \in \mathbb{Z}}).$$

Refinability of $\phi(x)$ implies that the chain of spaces $V^0, V^1, \ldots$ is nested; that is,

$$V^0 \subset V^1 \subset \cdots.$$

Intuitively, the resolution index j governs the amount of detail that can be present in the functions lying in V^j. As j increases it is possible to represent finer and finer detail.

Relative to an inner product $\langle \cdot, \cdot \rangle$, a space $V_\perp^j$ can be defined as the orthogonal complement of V^j in V^{j+1}:

$$V^{j+1} = V^j \oplus V_\perp^j.$$

To simplify the notation, it is common to normalize $\phi(x)$ to have unit norm, and to write the i-th scaling function at resolution j as $\phi_i^j(x) := 2^{j/2}\phi(2^j x - i)$. The normalization factor $2^{j/2}$ ensures that $\phi_i^j(x)$ has unit norm.

A pre-wavelet is a function $\psi(x) \in V_\perp^0$ whose translates span $V_\perp^0$; $\psi(x)$ is a wavelet if in addition the translates of $\psi(x)$ are mutually orthogonal. It is usual to normalize $\psi(x)$ to have unit norm, in which case it is straightforward to prove that the functions $\psi_i^j(x) := 2^{j/2}\psi(2^j x - i)$ form a basis for $V_\perp^j$.

The simplest example of scaling functions and wavelets are piecewise constant functions where $\phi(x)$ is the indicator function of $[0, 1]$, and where $\psi(x)$ is the so-called Haar basis, having a value of 1 on $[0, 1/2]$, a value of -1 on $[1/2, 1]$, and a value of zero elsewhere.

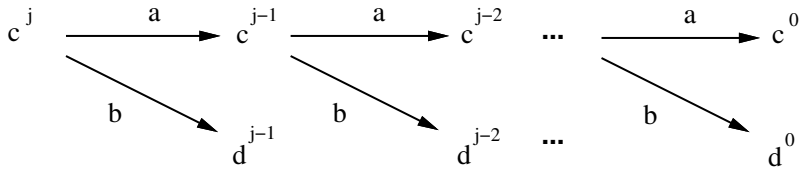

Figure 1: The filter bank process.

2.2 Filter banks

Consider a function

$$f^j(x) = \sum_i c_i^j \phi_i^j(x)$$

in V^j. The function can alternatively be written as a sum of two components, one lying in V^{j-1} and one lying in $V_\perp^j$:

$$f^j(x) = \sum_i c_i^{j-1} \phi_i^{j-1}(x) + \sum_i d_i^{j-1} \psi_i^{j-1}(x).$$

It is not difficult to show [1] that there exist sequences $a = \{..., a_{-1}, a_0, a_1, ...\}$ and $b = \{..., b_{-1}, b_0, b_1\}$ such that

$$c_i^{j-1} = \sum_l a_{l-2i} c_i^j,$$

and

$$d_i^{j-1} = \sum_l b_{l-2i} c_i^j.$$

Conversely, the original sequence c_i^j can be recovered from c_i^{j-1} and d_i^{j-1} using two other sequences p and q according to:

$$c_i^j = \sum_l (p_{i-2l} c_l^{j-1} + q_{i-2l} d_l^{j-1}).$$

The sequences a and b are typically called decomposition or analysis filters, and the sequences p and q are called reconstruction or synthesis filters.

The effect is to split c^j into a low resolution part c^{j-1} and a detail part d^{j-1}. The process can be repeated to split c^{j-1} into c^{j-2} and d^{j-2}, and so on, as indicated in Figure 1. This process, known as a filter bank, represents a decomposition of the original function $f^j(x)$ into a simple "base" function $f^0(x) \in V^0$, together with the wavelet coefficients necessary to recover $f^j(x)$. A filter bank can decompose in linear time an input function of 2^j coefficients into a base shape of scaling function coefficients together with additional information at each increasing level of detail.

For most of the functions encountered in practice, a large percentage of the coefficients resulting from the decomposition are close to zero. Wavelet compression methods therefore approximate the original function $f^j(x)$ by storing only the significant coefficients of the filter bank decomposition. Impressive compression rates have been reported for univariate signals as well as images [4].

2.3 Subdivision curves and surfaces

There are a variety of ways to describe subdivision curves and surfaces. The following presentation provides the background for the later discussion of parametrizing subdivision curves and surfaces.

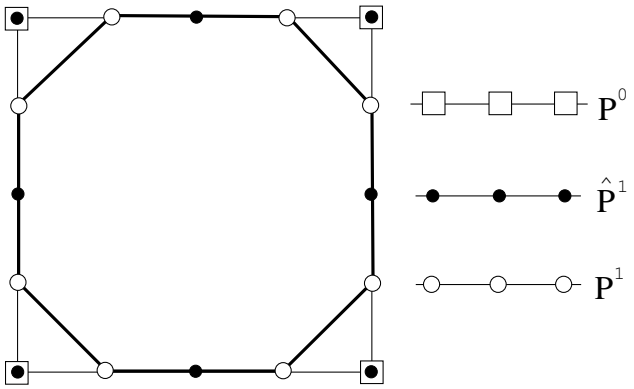

Figure 2: One step of Chaikin's algorithm for quadratic B-splines. The black dots are the $\widehat{P}^1$ split from P^0, which are then averaged with $\{0, 1, 1\}$ to produce the white dots of P^1.

Let us first consider the case of curves. The approach is to iteratively refine an initial control polygon P^0 so that the sequence of increasingly subdivided polygons $P^1, P^2, \ldots$ converges to some limit curve $C = P^\infty$.

It is convenient to think of the refinement step that carries polygon P^{s-1} into polygon P^s as consisting of two substeps: the *splitting* step and the *averaging* step. In the splitting step, each edge of P^{s-1} is split by inserting new vertices at edge midpoints. This results in an intermediate polygon $\widehat{P}^s$ having roughly twice as many vertices as P^{s-1}. The averaging step computes the vertices of P^s as weighted affine combinations of the points in $\widehat{P}^s$. All subdivision schemes share the splitting step – they differ only in the weights used in the averaging step.

For example, if the affine combinations used in the averaging step are such that the i-th control vertex of P^s is computed as the midpoint of the i-th and $(i+1)$-st vertices of $\widehat{P}^s$, then the limit curve can be shown to be a uniform quadratic B-spline. This process, known as Chaikin's algorithm [5], is depicted in Figure 2. We can encode the weights used in the averaging step for Chaikin's algorithm by a picture, known as a *mask*. A mask is a representation of the rule used to redefine the position of a vertex v as a weighted combination of points in the neighborhood of v. Given a point sequence $\widehat{P}$ and a mask $m = \{m_{-n}, \ldots, m_0, \ldots, m_n\}$, the result of applying m to $\widehat{P}$ is to define a new sequence P such that:

$$P_i := \frac{1}{|m|} \sum_{j=-n}^{n} m_j \widehat{P}_{i+j},$$

where $|m|$ is defined to be $\sum_j m_j$, so as to make the combination affine.

As a simple example, consider the averaging rule for quadratic B-splines given in the above paragraph: $P_i^s := \frac{1}{2}(\widehat{P}_i^{s-1} + \widehat{P}_{i+1}^{s-1})$. The mask for this is simply $\{0,1,1\}$. As another example, cubic B-splines are defined by the rule $P_i^s := \frac{1}{4}(\widehat{P}_{i-1}^{s-1} + 2\widehat{P}_i^{s-1} + \widehat{P}_{i+1}^{s-1})$, which translates into the mask $\{1,2,1\}$. There is a rich theory describing the masks that generate uniform B-splines [1], and, more generally, box splines [2]. As a special case, note that piecewise linear curves result if the averaging step is the identity; that is, if none of the vertices are moved.

Subdivision surfaces can be described in a similar fashion. For simplicity, we restrict our attention to subdivision surfaces based on triangular meshes, as described in the introduction. The idea is to again define the surface as the limit of a sequence of triangulated polyhedra (meshes)

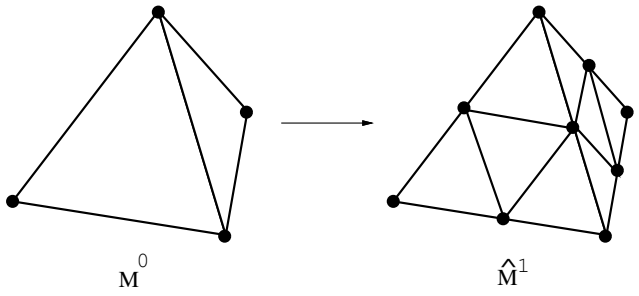

Figure 3: Subdividing a tetrahedral mesh. New vertices are introduced at the midpoints of each edge in the original mesh.

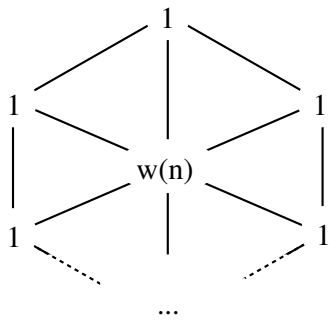

Figure 4: The averaging mask for Loop's scheme.

$M^0, M^1, \ldots$. The refinement step leading from M^{s-1} to M^s is again broken into two substeps, splitting and averaging. In the splitting step, each triangular facet of M^{s-1} is split into four triangular subfacets by introducing vertices at midpoints of edges, creating an auxiliary mesh $\widehat{M}^s$, as shown in Figure 3. The averaging step uses weighted averages to compute the vertex positions of M^s from the vertex positions of $\widehat{M}^s$.

The mesh M^s produced after s subdivision steps includes the vertices of M^0 together with new vertices introduced through subdivision. The vertices of M^s corresponding directly to the original vertices in M^0 retain their original connectivity structure and valence (the number of edges incident to a vertex). The new non-boundary vertices introduced through subdivision, however, are always of valence 6, corresponding to a regular triangular tiling of the surface. As the mesh is further subdivided, the so-called *extraordinary* points (any original vertex of valence other than 6) become increasingly isolated in an otherwise regular tiling of the surface.

The two examples of subdivision surfaces used throughout this paper are piecewise linear surfaces (i.e., triangulated polyhedra) and the subdivision surfaces invented by Loop in his Master's thesis [8]. Just as for curves, piecewise linear surfaces result when the averaging step leaves all vertex positions unchanged.

Loop's surfaces have the averaging mask shown in Figure 4, where n denotes the valence of the

central vertex. By chosing the central weight $w(n)$ in the averaging mask to be

$$w(n) := \frac{n\,\beta(n)}{1-\beta(n)},$$

where

$$\beta(n) := 2\left(\frac{3}{8} + \frac{\cos(\frac{2\pi}{n})}{4}\right)^2 - \frac{1}{4},$$

the limit surfaces can be shown to be tangent plane smooth (using methods such as in [6]). Loop's mask is such that the surface converges to quartic box splines away from extraordinary vertices.

3 Multiresolution analysis via subdivision

As explained in Section 2.1, multiresolution analysis on the real line starts with a single function $\phi(x)$ whose dilates and translates are used to define a sequence of nested linear spaces. To generalize these ideas to domains of arbitrary topological type, one could attempt to make definitions for what it means to dilate and translate a function on an arbitrary topological domain. (For an appreciation of the difficulties one is likely to encounter, try to imagine what it means to translate a function on a two holed torus.) One could then try to find a refinable scaling function and proceed as before to define orthogonal complements, wavelets, and so on. Our approach is to instead reformulate multiresolution analysis using techniques from subdivision surfaces.

As a preview of our approach, we use recursive subdivision to define a collection of functions $\phi_i^j(\mathbf{x})$ that are refinable in the sense that each function with superscript j lies in the span of the functions with superscript $j+1$; the argument $\mathbf{x}$ is a point that ranges over a domain 2-manifold of arbitrary topological type. In one respect, this is a generalization of the approach taken by Daubechies [3], whose locally supported orthogonal scaling functions are also defined through a recursive subdivision procedure. Although in general the $\phi^{j+1}(\mathbf{x})$ are not simple dilates of the $\phi^j(\mathbf{x})$, we can nonetheless use them to define a sequence of nested spaces. Once an inner product is placed on the spaces, we can define orthogonal complements and build basis functions $\psi_i^j(\mathbf{x})$ (i.e., pre-wavelets) for them.

3.1 Nested spaces through subdivision

To show how the subdivision process can be used to define a nested sequence of linear spaces, we define below a parametrization $S(\mathbf{x})$ for the limit surface, where $\mathbf{x}$ is a point that ranges over the faces of the initial mesh M^0. That is, we will treat M^0 as the parameter domain. We will then show that for any $j \geq 0$, and a point $\mathbf{x}$ on some face of M^0, $S(\mathbf{x})$ can be written as

$$S(\mathbf{x}) = \sum_i v_i^j \phi_i^j(\mathbf{x}),$$

where v_i^j denote the vertices of M^j. It will then be relatively easy to show that the basis functions $\phi_i^j(\mathbf{x})$ are refinable, leading to nested spaces.

We define $S(\mathbf{x})$ by means of a 3-step limiting process:

1. $S^0(\mathbf{x}) := \mathbf{x}, \mathbf{x} \in M^0$.

2. Suppose that $S^{s-1}(\mathbf{x})$ lies in triangle $(\hat{v}_a^s, \hat{v}_b^s, \hat{v}_c^s)$ of $\widehat{M}^s$ with barycentric coordinates (α, β, γ). (Recall that $\widehat{M}^s$ is the mesh resulting from applying the splitting step to M^{s-1}.) Then

$$S^s(\mathbf{x}) := \alpha v_a^s + \beta v_b^s + \gamma v_c^s,$$

where (v_a^s, v_b^s, v_c^s) is the triangle of M^s corresponding to $(\hat{v}_a^s, \hat{v}_b^s, \hat{v}_c^s)$ of $\widehat{M}^s$.

3. $S(\mathbf{x}) := \lim_{s\to\infty} S^s(\mathbf{x})$.

Lemma 1 *For all $j \geq 0$ and $s \geq j$ there exist row vectors $\mathbf{\Phi}^{s\leftarrow j}(\mathbf{x})$ such that*

$$S^s(\mathbf{x}) = \mathbf{\Phi}^{s\leftarrow j}(\mathbf{x})\, \mathbf{V}^j,$$

where $\mathbf{V}^j$ denotes the column vector of vertices of M^j.

Proof: The linear combination of Item 2 above can be rewritten in matrix notation as

$$S^s(\mathbf{x}) = \mathbf{b}^s(\mathbf{x})\, \mathbf{V}^s,$$

where $\mathbf{b}^s(\mathbf{x})$ is the barycentric coordinate vector of $\mathbf{x}$ with respect to M^s; that is,

$$\mathbf{b}^s(\mathbf{x}) = (0 \cdots 0\ \alpha\ 0 \cdots 0\ \beta\ 0 \cdots 0\ \gamma\ 0 \cdots 0),$$

where α occurs at index a, β occurs at index b, γ occurs at index c. Since at each refinement step $k = 1, ..., s$, the vertices of M^k can be computed from affine combinations of the vertices of M^{k-1}, there must exist a chain of (non-square) matrices $A^1, ..., A^k$ such that

$$\mathbf{V}^k = A^k A^{k-1} \cdots A^1 \mathbf{V}^0.$$

Thus,

$$S^s(\mathbf{x}) = \mathbf{b}^s(\mathbf{x}) A^s A^{s-1} \cdots A^{j+1} \mathbf{V}^j.$$

The desired result follows immediately by making the definition

$$\mathbf{\Phi}^{s\leftarrow j}(\mathbf{x}) := \mathbf{b}^s(\mathbf{x}) A^s A^{s-1} \cdots A^{j+1}. \ \square$$

As a simple corollary to Lemma 1, we note that

$$\mathbf{\Phi}^{s\leftarrow j}(\mathbf{x}) = \mathbf{\Phi}^{s\leftarrow j+1}(\mathbf{x})\ A^{j+1}. \tag{1}$$

Theorem 1 *For any $j \geq 0$ there exist scaling functions $\phi_i^j(\mathbf{x})$, $\mathbf{x} \in M^0$ such that*

$$S(\mathbf{x}) = \sum_i v_i^j \phi_i^j(\mathbf{x}).$$

Proof: Using Lemma 1 and the definition of $S(\mathbf{x})$:

$$S(\mathbf{x}) = \lim_{s\to\infty} \left(\mathbf{\Phi}^{s\leftarrow j} \mathbf{V}^j \right).$$

By continuity of matrix multiplication, $S(\mathbf{x})$ can be rewritten as

$$S(\mathbf{x}) = \left(\lim_{s \to \infty} \mathbf{\Phi}^{s \leftarrow j}(\mathbf{x}) \right) \mathbf{V}^j.$$

The desired result is obtained by taking

$$\mathbf{\Phi}^j(\mathbf{x}) := \left(\lim_{s \to \infty} \mathbf{\Phi}^{s \leftarrow j}(\mathbf{x}) \right)$$

and then expanding the matrix product $\mathbf{\Phi}^{s \leftarrow j}(\mathbf{x})\mathbf{V}^j$ in summation notation. □

We can now establish the refinability of the scaling functions defined in Theorem 1.

Theorem 2 *The scaling functions $\phi_i^j(\mathbf{x})$ are refinable.*

Proof: Starting with Equation 1 and taking limits as s tends toward infinity, it follows from continuity of inner products that

$$\mathbf{\Phi}^j(\mathbf{x}) = \mathbf{\Phi}^{j+1}(\mathbf{x})\, A^{j+1}. \tag{2}$$

This equation establishes refinability since it states that each of the functions $\phi_i^j(\mathbf{x})$ can be written as a linear combination of the functions $\phi_i^{j+1}(\mathbf{x})$. □

A chain of nested linear spaces $V^j(M^0)$ associated with a mesh M^0 can now be defined as follows:

$$V^j(M^0) := \text{Span}(\phi_i^j(\mathbf{x}) : i \text{ such that } v_i^j \text{ is a vertex of} M^j).$$

Refinability of the scaling functions implies that these spaces are indeed nested; that is,

$$V^0(M^0) \subset V^1(M^0) \subset \cdots$$

3.2 Inner products

We define the inner product of any two functions $f, g \in V^j(M^0)$, $j < \infty$, to be

$$\langle f, g \rangle := \int_{\mathbf{x} \in M^0} f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x},$$

where the area form $d\mathbf{x}$ is taken to be the area form for a triangulation homeomorphic to M^0 consisting of equilateral triangles. Equivalently,

$$\langle f, g \rangle := \sum_{\tau \in \Delta(M^0)} \frac{1}{\text{Area}(\tau)} \int_{\mathbf{x} \in \tau} f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x},$$

where $\Delta(M^0)$ denotes the set of triangles of M^0.

This definition of inner product implies that triangles of different geometric size and shape are weighted equally; that is, the inner product is independent of the geometric positions of the vertices of M^0. This has an important practical consequence: the orthogonal complement spaces are invariant of the geometry of the mesh, meaning that a significant amount of precomputation of inner products and wavelets can be done.

An alternative definition is to weight the inner product by the areas of the triangles of M^0. Although such an approach might lead to more accurate approximations, it has the practical drawback that much less precomputation is possible.

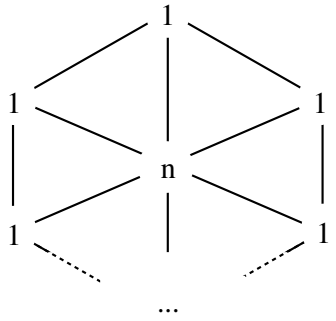

Figure 5: The polyhedral inner product mask.

For any pair of functions f, g, one could estimate the inner product $\langle f, g \rangle$ using numerical cubature. It turns out, however, that it is possible to compute $\langle f, g \rangle$ exactly if f and g are given as expansions in $\phi_i^j(\mathbf{x})$:

$$\begin{aligned} f(\mathbf{x}) &= \sum_i f_i^j \phi_i^j(\mathbf{x}) \\ g(\mathbf{x}) &= \sum_i g_i^j \phi_i^j(\mathbf{x}). \end{aligned}$$

Bi-linearity of the inner product allows $\langle f, g \rangle$ to be written in matrix form as

$$\langle f, g \rangle = \mathbf{g}^T \mathbf{I}^j \mathbf{f},$$

where $\mathbf{f}$ and $\mathbf{g}$ are column matrices consisting of the coefficients of f and g, respectively, and where $\mathbf{I}^j$ is a symmetric bi-linear form whose i, i'-th entry is

$$(\mathbf{I}^j)_{i,i'} = \langle \phi_i^j, \phi_{i'}^j \rangle.$$

We show below that the entries of $\mathbf{I}^j$ can be computed exactly – thus, $\langle f, g \rangle$ can also be computed exactly.

The i-th row of $\mathbf{I}^j$ contains the inner product of ϕ_i^j with each of the other scaling functions $\phi_{i'}^j$. It is convenient to view these entries geometrically by constructing an *inner product mask* around each vertex. The inner product mask for the i-th vertex of M^j assigns to each vertex i' of M^j the number $\langle \phi_i^j, \phi_{i'}^j \rangle$.

In the case of polyhedral subdivision, the scaling functions are piecewise linear functions over M^0, meaning that the entries of $\mathbf{I}^j$ can be computed by explicit integration. Direct calculation leads to the inner product mask around a vertex of valence n shown in Figure 5, where the central weight is simply the valence of the central vertex.

For more general subdivision schemes, such as Loop's, the limit surface has no known closed form, precluding a brute-force explicit integration. However, if the subdivision scheme is local (that is, if the averaging mask of the scheme has local support), it is possible to determine the entries of $\mathbf{I}^j$ by solving a linear system, without resorting to numerical cubature.

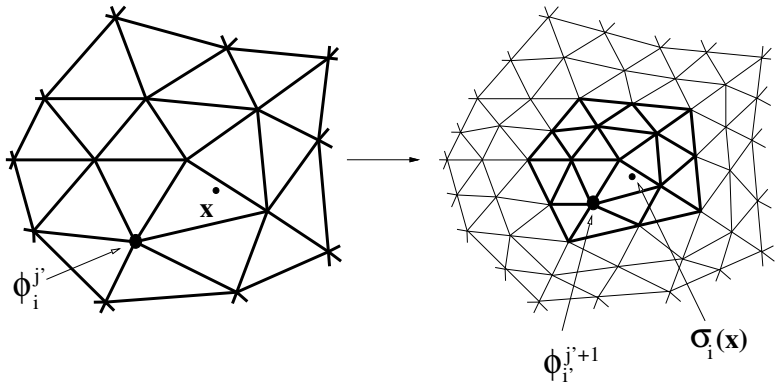

Figure 6: The diagram on the left depicts the triangulation of M^0 produced after j' subdivision steps, that is, after j' recursive midpoint splits; the diagram on the right corresponds to $j'+1$ subdivision steps. $\phi_i^{j'}$ denotes the scaling function for the i-th vertex of $M^{j'}$, and $\phi_{i'}^{j'+1}$ denotes the scaling function for the corresponding i'-th vertex of $M^{j'+1}$. The map $\sigma_i(\mathbf{x})$ is such that the barycentric coordinates of $\mathbf{x}$ and $\sigma_i(\mathbf{x})$ within their respective surrounding triangles are equal.

The key to this linear system is to observe that a recurrence relation exists between $\mathbf{I}^j$ and $\mathbf{I}^{j+1}$. We first note that $\mathbf{I}^j$ can be written as

$$\mathbf{I}^j = \int_{\mathbf{x}\in M^0} (\mathbf{\Phi}^j(\mathbf{x}))^T \; \mathbf{\Phi}^j(\mathbf{x}) \, d\mathbf{x},$$

where the integrand represents a matrix outer product, and where the integral of a matrix of functions is defined to be the matrix of integrals. The refinement property of Equation 2 can now be used to establish the recurrence

$$\begin{aligned} \mathbf{I}^j &= \int_{\mathbf{x}\in M^0} (A^{j+1})^T \; (\mathbf{\Phi}^{j+1}(\mathbf{x}))^T \; \mathbf{\Phi}^{j+1}(\mathbf{x}) \; A^{j+1} \, d\mathbf{x} \\ &= (A^{j+1})^T \; \mathbf{I}^{j+1} \; A^{j+1}. \end{aligned}$$

If the averaging mask is local, the scaling functions will have local support. Thus, after some number of subdivision steps j', the support of each scaling function will contain at most one extraordinary point. Since the scaling functions depend only on the local structure of the mesh, one would expect some relation between the scaling functions in $\mathbf{\Phi}^{j'}(\mathbf{x})$ and $\mathbf{\Phi}^{j'+1}(\mathbf{x})$. More precisely, for each scaling function $\phi_i^{j'}(\mathbf{x})$ there is an i' such that

$$\phi_i^{j'}(\mathbf{x}) = \phi_{i'}^{j'+1}(\sigma_i(\mathbf{x})),$$

where $\sigma_i : M^0 \to M^0$ is a piecewise linear function that maps triangles in the support of $\phi_i^{j'}(\mathbf{x})$ into the corresponding triangles of $\phi_{i'}^{j'+1}(\mathbf{x})$, as depicted in Figure 6. Since triangle areas of M^0 shrink by a factor of four under subdivision, the Jacobian of σ_i is $\frac{1}{4}$. Consequently, each of the entries $(\mathbf{I}^{j'})_{hi}$ in $\mathbf{I}^{j'}$ has one or more corresponding entries $(\mathbf{I}^{j'+1})_{h'i'}$ in $\mathbf{I}^{j'}$, up to a factor of $\frac{1}{4}$. That is, $(\mathbf{I}^{j'})_{hi} = 4(\mathbf{I}^{j'+1})_{h'i'}$

The resulting $m \times m$ matrix equation

$$\mathbf{I}^{j'} = (A^{j'+1})^T \; \mathbf{I}^{j'+1} \; A^{j'+1} \tag{3}$$

represents a homogeneous system of m^2 equations in the m^2 unknown entries of $\mathbf{I}^{j'}$. Due to the symmetry of $\mathbf{I}^{j'}$, the system reduces to $m(m+1)/2$ homogeneous equations in as many unknowns. Once an absolute scale for the homogeneous system is chosen, the system can be uniquely solved.

Finally, after the entries of $\mathbf{I}^{j'}$ have been determined, the remaining inner product matrices $\mathbf{I}^{j'-1}, \mathbf{I}^{j'-2}, ..., \mathbf{I}^0$ can be successively determined via Equation 3.

3.3 Wavelets

Having established nested linear spaces and an inner product, we are now in a position to construct wavelets, that is, a set of functions spanning the orthogonal complements $V_\perp^0(M^0), V_\perp^1(M^0), \cdots$. For notational simplicity we give the construction for a basis of $V_\perp^0(M^0)$.

Our construction consists of two steps. First, we construct a set of functions spanning $V^1(M^0)$ consisting of the scaling functions in $V^0(M^0)$ together with a subset U of the scaling functions in $V^1(M^0)$. Recall from Section 2.3 that new vertices are added to M^0 when constructing M^1 (see Figure 3). We take U to be the set of scaling functions $\phi_i^1(\mathbf{x})$ associated with these new vertices. It is straightforward to prove that this collection does indeed span $V^1(M^0)$. For the example shown in Figure 3, the set U would contain 6 functions, one for each edge of M^0.

The second step is to use a Gram-Schmidt-like process to project each member of U into $V_\perp^0(M^0)$, thereby obtaining a set of functions spanning $V_\perp^0(M^0)$. If $\phi_i^1(\mathbf{x})$ denotes an arbitrary member of U, its projection into $V_\perp^0(M^0)$ may be expressed as

$$\psi_i^0(\mathbf{x}) = \phi_i^1(\mathbf{x}) - \widehat{\phi}_i^1(\mathbf{x}),$$

where $\widehat{\phi}_i^1(\mathbf{x}) \in V^0(M^0)$ is the projection of $\phi_i^1(\mathbf{x})$ into $V^0(M^0)$; that is,

$$\widehat{\phi}_i^1(\mathbf{x}) = \arg \min_{f \in V^0(M^0)} \langle \phi_i^1 - f, \phi_i^1 - f \rangle.$$

In practice, $\widehat{\phi}_i^1(\mathbf{x})$ may be computed by solving a least squares system for the coefficients $\alpha_{i'}$ such that

$$\widehat{\phi}_i^1(\mathbf{x}) = \sum_{i'} \alpha_{i'} \phi_{i'}^0(\mathbf{x}).$$

This least squares system has the form

$$\mathbf{I}^0 \, \alpha = \mathbf{b},$$

where α is the column vector of unknown coefficients, and where $\mathbf{b}$ is a column vector whose k-th entry is $\langle \phi_i^1, \phi_k^0 \rangle$.

The resulting collection of functions ψ_i^0 are pre-wavelets since they span $V_\perp^0(M^0)$, but they are not mutually orthogonal. Figure 7 is a plot of one such pre-wavelet for the polyhedral subdivision scheme. One undesirable feature of these pre-wavelets in practice is that each is globally supported on M^0, implying that filter bank decomposition and reconstruction algorithms require quadratic time.

We currently do not know of a construction leading to locally supported pre-wavelets, nor do we know if such a construction always exists. Our approach for now, which has worked well in practice, is to obtain locally supported functions by relaxing the condition that the ψ_i^0 lie in $V_\perp^0(M^0)$. Instead, we construct them to span a space V_*^0 – that is, some (non-orthogonal) complement of $V^0(M^0)$ in $V^1(M^0)$. We call functions in $V_*^0(M^0)$ *quasi-wavelets*. We show below that it is possible to make $V_*^0(M^0)$ close to $V_\perp^0$, at the expense of increasing the supports of the quasi-wavelets.

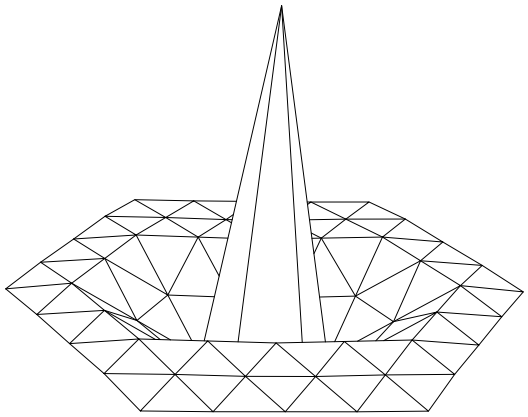

Figure 7: A plot of a polyhedral pre-wavelet centered on a vertex of valence 6.

We construct our quasi-wavelets by first selecting their supports; we choose, for each $\psi_i^0(\mathbf{x})$, which of the $\alpha_{i'}$'s will be non-zero. The resulting $\psi_i^0(\mathbf{x})$ is constructed to be the best locally supported projection of $\phi_i^1(\mathbf{x})$ into $V_\perp^0(M^0)$. By allowing more of the $\alpha_{i'}$s to be non-zero, the supports grow, and $V_*^0(M^0)$ gets closer to $V_\perp^0(M^0)$.

The use of quasi-wavelets instead of globally supported wavelets has two practical consequences. First, since they are locally supported, the filter bank algorithms for quasi-wavelets are linear time. Second, truncation of a wavelet expansion leads to parametric least-squares best approximation; truncation of a quasi-wavelet expansion does not. In practice this does not seem to be much of a disadvantage for use with parametric curves and surfaces since least-squares norms are themselves approximations to more geometric norms such as the Hausdorff or Frechet distance.

4 Example

In this section, we provide figures showing the results of applying our theory to a complicated shape generated by polyhedral subdivision. Details of our implementation are left for a future paper.

The input for this example, shown in Figure 8(d), is a polyhedral mesh consisting of 32,768 triangles created from laser range data provided courtesy of Cyberware, Inc. The triangulation connectivity results from recursively subdividing an octahedron six times. The octahedron therefore serves as the domain mesh M^0, with the input triangulation considered as a parametric function $S(\mathbf{x}), \mathbf{x} \in M^0$ lying in $V^6(M^0)$. More precisely, if the v_i^6 denote the vertices of the input mesh, $S(\mathbf{x})$ can be written as

$$S(\mathbf{x}) = \sum_i v_i^6 \ \phi_i^6(\mathbf{x}), \qquad \mathbf{x} \in M^0,$$

where the scaling functions $\phi_i^6(\mathbf{x})$ are the (piecewise linear) functions defined through the polyhedral subdivision scheme.

The quasi-wavelets $\psi_i^j(\mathbf{x})$ for this example are chosen to be supported on 3-discs. (We define the k-disc around a vertex v of a triangulation to be the set of all triangles whose vertices are reachable from v by following k or fewer edges of the triangulation.) A filter bank process can be

applied in linear time to rewrite $S(\mathbf{x})$ in the form

$$S(\mathbf{x}) = \sum_i v_i^0 \ \phi_i^0(\mathbf{x}) + \sum_{j=0}^{5} \sum_i w_i^j \ \psi_i^j(\mathbf{x}).$$

The first term describes a projection of $S(\mathbf{x})$ into $V^0(M^0)$, which in this case is an approximating octahedron with vertex position given by v_i^0.

The original mesh $S(\mathbf{x})$ can be approximated with simpler meshes by truncating the terms in the expansion having wavelet coefficients w_i^j whose magnitude is below some threshold ϵ. Approximations at various levels of detail can be obtained by varying ϵ. Figure 8 shows a sequence of these approximations for decreasing values of ϵ. Notice that this simple rule causes the mesh to refine in areas of high detail, while leaving large triangles in areas of relatively low detail.

An indication of the usefulness of the compression for high-performance rendering is shown in Figure 9, where each mesh from Figure 8 is Gouraud-shaded (using estimated normals). As a comparison, the original input mesh is shown shaded at the same projected size in Figure 10. Although images (a), (b), and (c) in Figure 9 contain only a fraction of the triangles of the corresponding images in Figure 10, the image quality is not significantly degraded.

5 Summary and future work

In this paper, we have established a theoretical basis for applying multiresolution analysis to surfaces of arbitrary topological type. Subdivision surfaces play an important role in this theory, and appear to be an increasingly attractive primitive for modeling complex surfaces.

There are numerous areas of future work:

- Our current theory applies only to meshes with subdivision connectivity; that is, to meshes resulting from recursively subdividing a simpler mesh. One possible method for dealing with general meshes is to develop a remeshing procedure for approximating an arbitrary mesh by one possessing subdivision connectivity.

- Our surface decomposition retains the original genus of the input surface. When the input is a relatively simple object with many small holes, a more useful approximation would decompose the input into a surface with lower genus.

- We are currently implementing our technique for smooth surfaces described by C^1 continuous subdivision rules. We expect to achieve significant compression rates for many of the smooth surfaces encountered in practice.

- The images in Figure 9 were chosen by simply adding the (vector-valued) quasi-wavelet coefficients of greatest magnitude. We are currently investigating view-dependent heuristics designed to produce improved image quality using even fewer triangles.

6 Acknowledgments

This work was made possible in part by Xerox Corp. and the National Science Foundation under grant no. CCR-8957323.

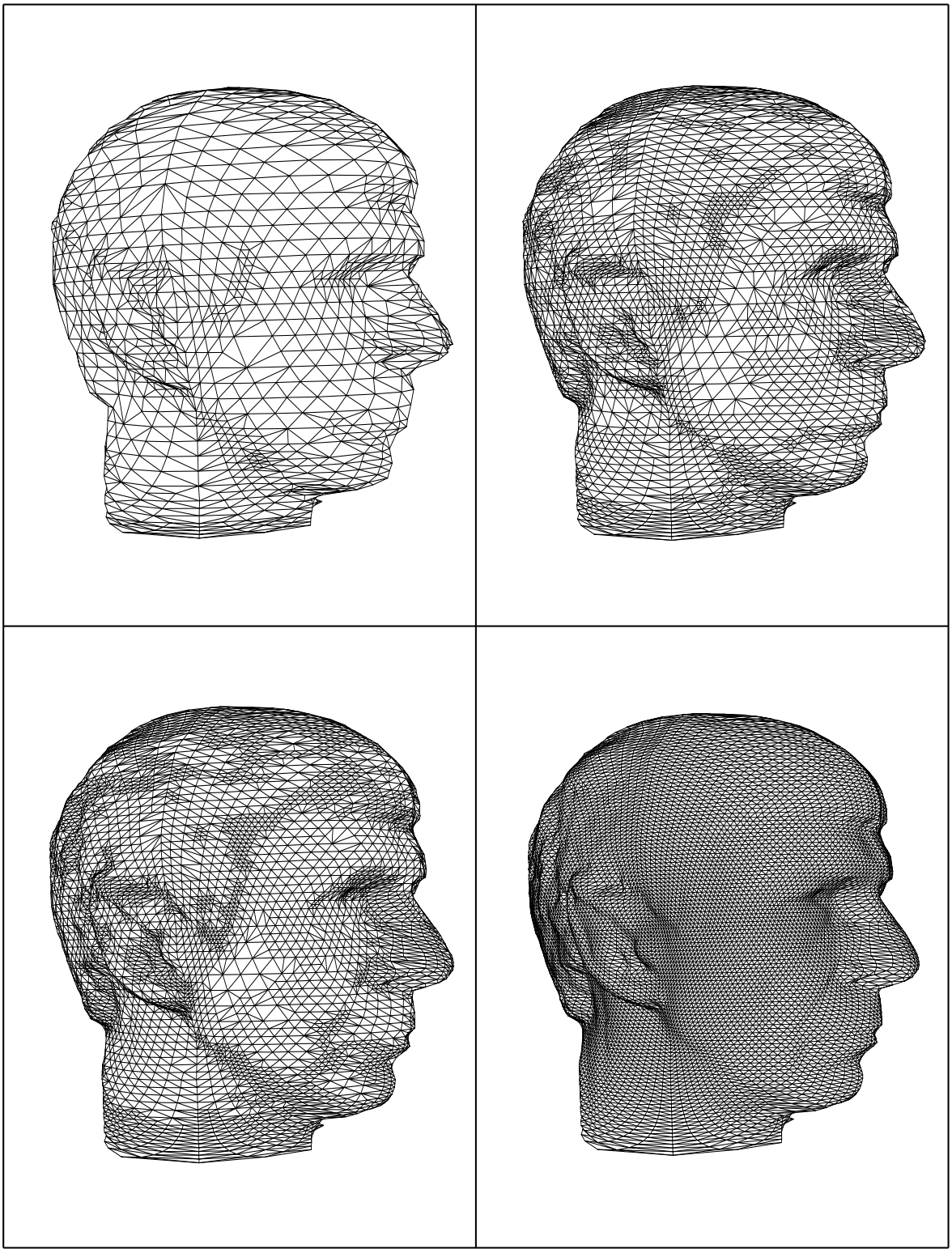

Figure 8: Meshes reconstructed from the input. (a) 4,054 triangles. (b) 12,310 triangles. (c) 16,098 triangles. (d) 32,768 triangles (the original input).

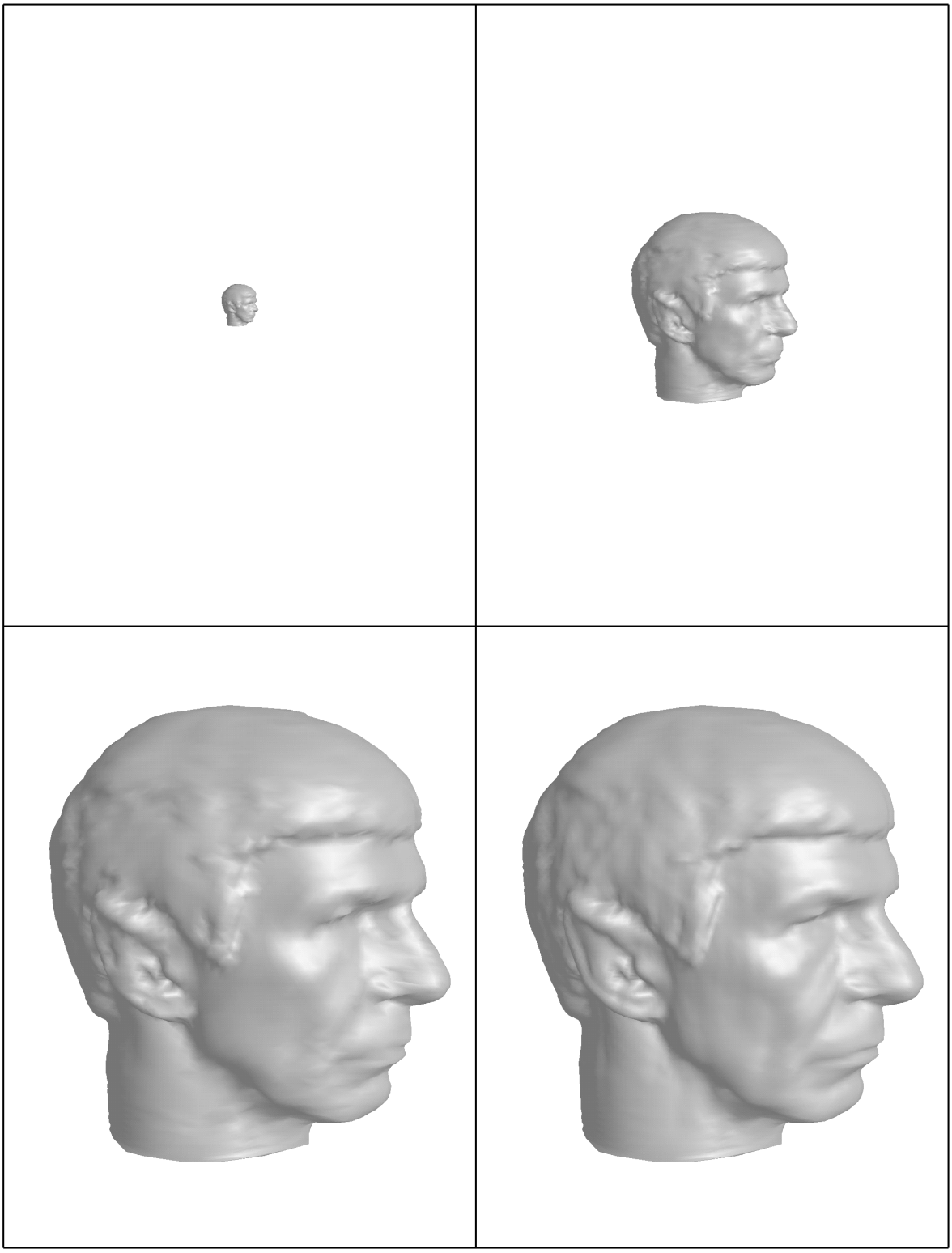

Figure 9: Shaded images of the reconstructed meshes. (a) 4,054 triangles. (b) 12,310 triangles. (c) 16,098 triangles. (d) 32,768 triangles (the original input).

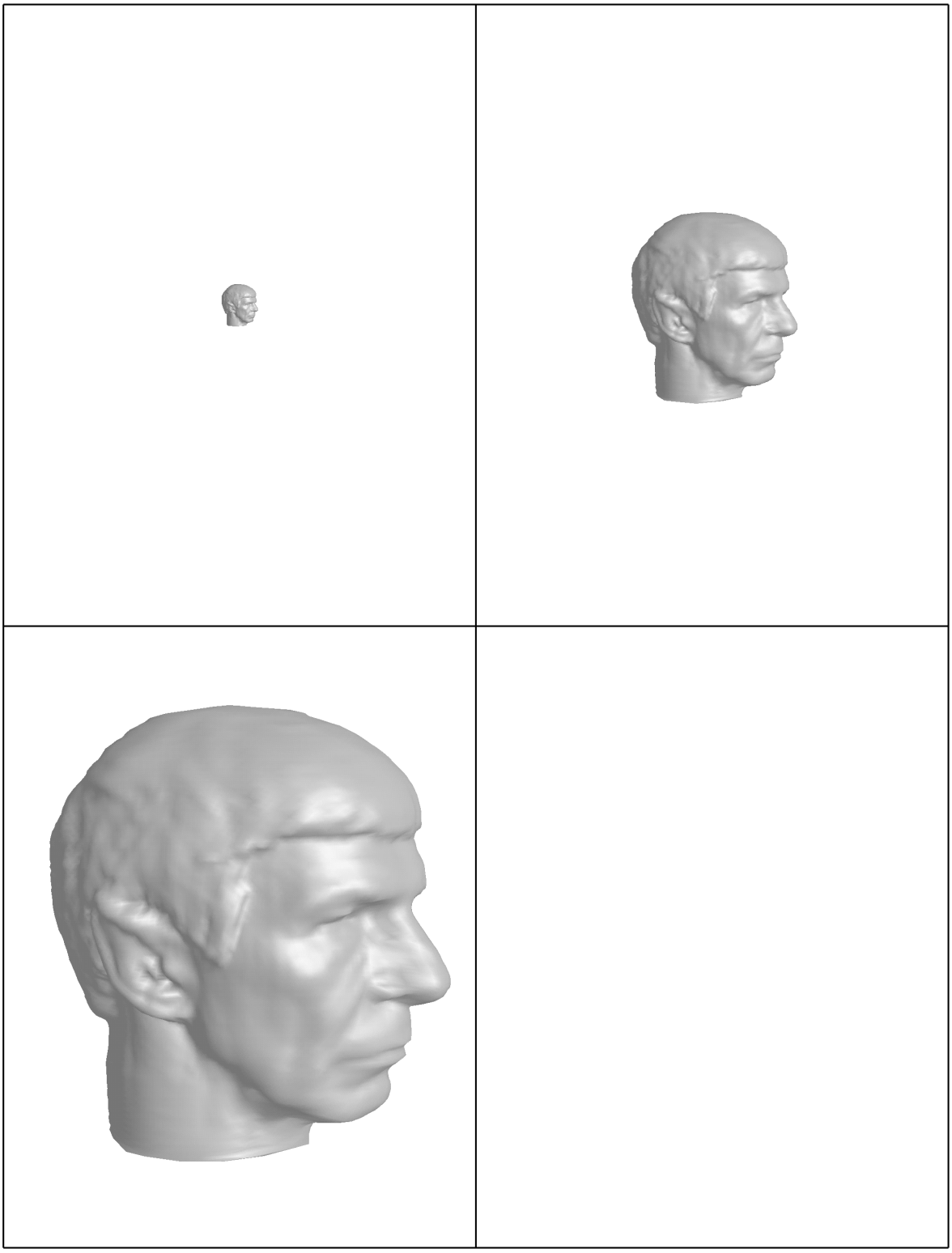

Figure 10: Shaded versions of the complete input mesh, at the same sizes. Each image here is rendered from the complete input of 32,768 triangles.

References

[1] Charles K. Chui. *An Introduction to Wavelets*, volume 1. Academic Press, Inc., Boston, 1992.

[2] M. Dæhlen and T. Lyche. Box splines and approximations. In H. Hagen and D. Roller, editors, *Geometric Modeling: Methods and Applications*, pages 35–93. Springer-Verlag, Berlin, 1991.

[3] Ingrid Daubechies. *Ten lectures on wavelets.* SIAM, Philadelphia, 1992.

[4] R. DeVore, B. Jawerth, and B. Lucier. Image compression through wavelet transform coding. *IEEE Trans. Inform. Theory*, 38(2):719–746, March 1992.

[5] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, third edition, 1993.

[6] Mark Halstead, Michael Kass, and Tony DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. *Computer Graphics*, 27(3):35–44, August 1993.

[7] Pat Hanrahan, David Salzman, and Larry Auperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4):197–206, August 1991.

[8] Charles T. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, Department of Mathematics, University of Utah, August 1987.

[9] Stephane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[10] David Meyers. Multiresolution tiling. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 1994. In preparation.

[11] Edwald Quak and Norman Weyrich. Decomposition and reconstruction algorithms for spline wavelets on a bounded interval. Technical Report 294, Department of Mathematics, Texas A&M University, April 1993.

[12] David Salesin and Adam Finkelstein. Multiresolution B-spline editing. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 1993. In preparation.