The Distance of a Subdivision Surface to its Control Polyhedron

J. Peters and X. Wu

September 15, 2008

Abstract

For standard subdivision algorithms and generic input data, near an extraordinary point the distance from the limit surface to the control polyhedron after m subdivision steps is shown to decay dominated by the mth power of the subsubdominant eigenvalue. Conversely, for Loop subdivision we exhibit generic input data so that the Hausdorff distance at the mth step is greater or equal to the mth power of the subsubdominant eigenvalue.

In practice, it is important to closely predict the number of subdivision steps necessary so that the control polyhedron approximates the surface to within a fixed distance. Based on the above analysis, two such predictions are evaluated. The first is a popular heuristic that analyzes the distance only for control points and not for the facets of the control polyhedron. For a set of test polyhedra this prediction is remarkably close to the distance verified by a posteriori measurement. However, a concrete example shows that the prediction is not safe but can prescribe too few steps. – The second approach is to first locally, per vertex neighborhood, subdivide the input net and then apply tabulated bounds on the eigenfunctions of the subdivision algorithm. This yields always safe predictions that are close for a set of test data.

keywords: subdivision surface, control polyhedron, convergence, distance, characteristic spline, lower bound, box spline

1 Introduction

Alongside splines, recursive subdivision is increasingly used to represent higher-order design surfaces. A key property of subdivision surfaces for applications such as rendering on the computer, is that a so-called *control polyhedron* (Figure 1,*left*) can be used as a proxy for the surface (Figure 1,*right*). Much of the appeal of subdivision surfaces lies in the intuitive refinability of this proxy representation to obtain an ever closer approximation of the limit surface (see Figure 2). It is therefore of interest both to clearly define this representation and to understand its approximation properties. In particular, unless the control polyhedron defines a regular (box-)spline, to date no estimate is published that closely and correctly *predicts* m, the minimal number of subdivision steps necessary to ensure that every point of the subdivision surface is within a given tolerance of a control polyhedron. Predicting m, as opposed to measuring m after each refinement, is practically relevant since it helps preallocating resources, guiding adaptive refinement and optimizing rendering. Evidently, too few subdivision steps result in a visual lack of smoothness, while too many steps are costly due to exponential growth of the number of facets if we refine uniformly – as we must without good local predictions of m to guide adaptive subdivision. For example, estimating the number loosely to be $m = 10$ when the sharp estimate is $m = 5$ means computing and processing millions instead of thousands of facets for each original facet.

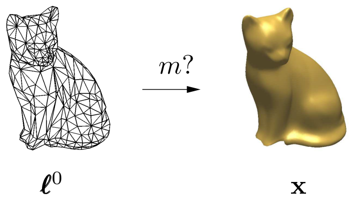

Figure 1: Predicting m, the number of subdivision steps necessary to ensure that the control polyhedron $\boldsymbol{\ell}^m$ is within ϵ of the limit subdivision surface $\mathbf{x}$.

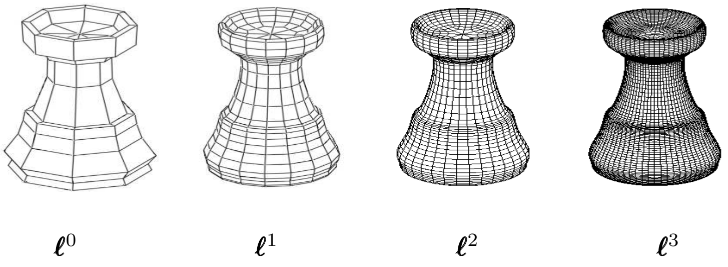

Figure 2: Four steps of Catmull-Clark subdivision (from [SP05]).

The most popular subdivision schemes are Catmull-Clark subdivision [CC78], with a control polyhedron consisting of quadrilateral facets as shown in Figure 2, and Loop subdivision [Loo87], with a control polyhedron consisting of triangles, as illustrated in Figure 1. For these two prototypical schemes and their generalizations, this paper yields the following four insights.

(i) For generic input data, the convergence of the control polyhedron to the limit surface near extraordinary points is dominated by powers of the subsubdominant eigenvalue (Corollaries 2 and 3, page 11). The upper bound estimate also applies when each control point (vertex of the control polyhedron) is replaced by its limit point on the surface and the distance between this interpolant and the surface is measured. — Conversely, for Loop subdivision we exhibit generic input data so that the Hausdorff distance between control polyhedron and surface at the mth step is greater or equal to the mth power of the subsubdominant eigenvalue.

(ii) The easily-computed heuristic prediction of the distance, as the distance between control points and their limit, is in many cases a good guide (Figure 6,*top*, page 16, measurement results labeled *heuristic*), but is unsafe since the maximal distance between the subdivision surface and the control polyhedron is in general not taken on at iterates of the control points (Figure 5).

(iii) Precomputed and tabulated bounds on eigenfunctions can efficiently be combined for predictions that are sharp for specific cases (Equation (15)). However, when applied to the input control polyhedra of typical graphics models (Figure 7), the distance in the first few subdivision steps is overestimated (Figure 6,*top*, measurement results labeled *tab-approx*).

(iv) Local subdivision, without refining data structures, followed by the predictions in (iii) yields safe estimates that, when tested on the polyhedra of Figure 7, are a small and decreasing multiple of the measured distance (Figure 6,*top*, measurements labeled *tab-approx*(1) and *tab-approx*(2)).

Fact (i) has two implications. First, convergence of the control polyhedron to the box-spline under binary subdivision by a factor of $\frac{1}{4}$ in each subdivision step can be viewed as convergence governed by the subsubdominant eigenvalue $\frac{1}{4}$ of the binary subdivision matrix. Second, convergence of the control polyhedron to the surface near points of high valence n can be much slower than $\frac{1}{4}$. For example, for Loop subdivision, if we order the eigenvalues by decreasing size, the subsubdominant eigenvalue $\lambda_3 := (3 + 2\cos\frac{4\pi}{n})/8$ is close to 5/8 for large valence n. And where four steps of subdivision suffice for the regular, box-spline control polyhedron, close to twelve are necessary near high-valence vertices, since $(\frac{5}{8})^{12} \approx (\frac{1}{4})^4$.
Observation (ii) gives a handy heuristic, but not a bound as has been suggested in [WSQ04] and in some manuscripts. Figure 5 shows a concrete example that the simple heuristic is unsafe.
Observations (iii) and (iv) point to a practical compromise between worst-case prediction and practically useful bounds: predictions become both more accurate and more stable after one or two refinement steps. These steps can be computed locally, without actual allocation and refinement of data structures. Followed by the conservative pre-tabulate estimates of (iii), they predict just

one more step than the minimum m for some typical, large test objects (Figure 7).

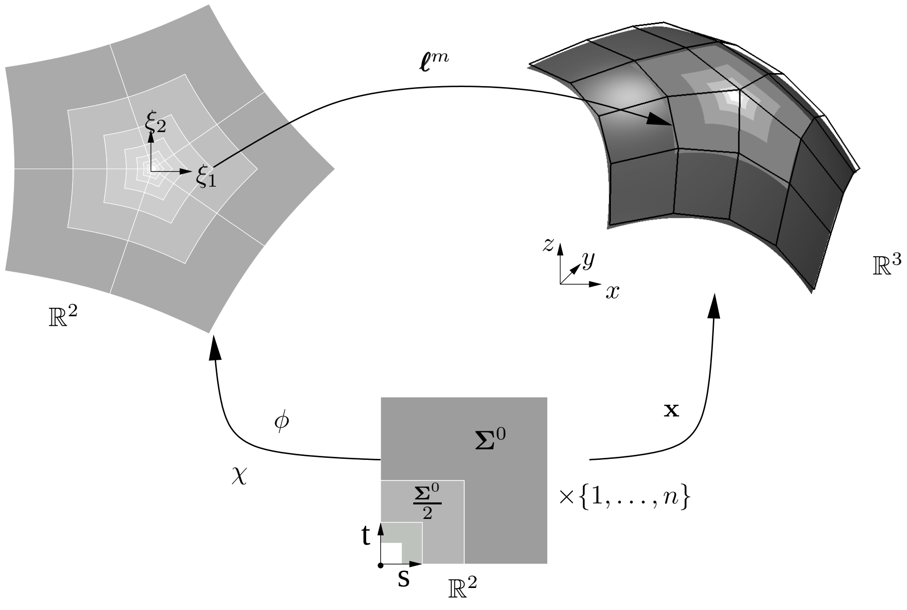

Figure 3: (*bottom*) A piece $\mathbf{\Sigma} = \bigcup_{m=0}^{\infty} \mathbf{\Sigma}^0/2^m$ of the domain $\mathbb{S}_n := \mathbf{\Sigma} \times \{0, 1, \dots, n-1\}$; (*left*) characteristic spline $\chi : \mathbb{S}_n \to \mathbb{D} \subsetneq \mathbb{R}^2$, and an alternative embedding $\phi : \mathbb{S}_n \to \mathbb{D}$; (*top*) control polyhedron $\boldsymbol{\ell}^m : \mathbb{D} \to \mathbb{R}^3$ and (*right*) the spline $\mathbf{x} : \mathbb{S}_n \to \mathbb{R}^3$. (*top*,*right*) The spline surface $\mathbf{x}$ for $n = 5$ is the union of a sequence of nested spline rings $\mathbf{x}^m$.

1.1 Review of Subdivision Basics

Subdivision surfaces are spline surfaces with isolated singularities [PR08]. The word spline is used in a general sense that includes, for example, box-splines. *Singularities* occur because n copies of the natural domain of a spline piece can only be rigidly embedded in $\mathbb{R}^2$ to surround the origin without overlap if n is a specific number of pieces. For box-splines this is the valence of the vertices of the shift-invariant lattice defined by the box-spline convolution directions. For example, the C^2 three-direction box-spline [dHR93] underlying Loop subdivision [Loo87] has exactly $n = 6$ three-sided polynomial patches join. For Catmull-Clark subdivision [CC78] exactly four quadrilateral patches join. This is the regular, tensor-product bicubic spline configuration visible almost everywhere in the meshes of Figure 2. Near the singularities, a subdivision surface is best viewed as a union of *rings* $\mathbf{x}^m$ (see Figure 3):

$$\mathbf{x}^m : \mathbb{S}_n^0 \to \mathbb{R}^d, \qquad \mathbb{S}_n^0 := \mathbf{\Sigma}^0 \times \{1, 2, \dots, n\}$$

and their limit $\mathbf{x}^\infty$, called extraordinary point or central point (cf. Figure 3, *right*). For Catmull-Clark subdivision, $\mathbf{\Sigma}$ is the square $[0,1]^2$ and $\mathbf{\Sigma}^0 := \mathbf{\Sigma} - \mathbf{\Sigma}/2$. For Loop subdivision, $\mathbf{\Sigma}$ is the unit triangle and $\mathbf{\Sigma}^0 := \mathbf{\Sigma} - \mathbf{\Sigma}/2$. For subdivision surfaces in $\mathbb{R}^3$, $d = 3$. Both Catmull-Clark and Loop subdivision are symmetric standard subdivision algorithms.

Standard subdivision algorithms are linear and stationary and defined by a pair (A, G). The first entry, $A \in \mathbb{R}^{\bar{\iota} \times \bar{\iota}}$, is a *subdivision matrix* that maps $\bar{\iota}$ points in $\mathbb{R}^d$ to $\bar{\iota}$ new points, has all rows summing to 1 and values depending on n. The second entry is a vector G of $\bar{\iota}$ *generating rings*. Generating rings will be (at least) C^1 maps $g_i : \mathbb{S}_n^0 \to \mathbb{R}$ that form a partition of unity. For a given vector of control points $\mathbf{Q} \in \mathbb{R}^{\bar{\iota} \times d}$, the sequence of spline rings is then computed by iterated application of the subdivision matrix: $\mathbf{x}^m := GA^m\mathbf{Q}$. Each application corresponds to one *subdivision step*. Now let v_i be an eigenvector of A with corresponding eigenvalue λ_i and let the eigenvalues be sorted so that $|\lambda_l| \geq |\lambda_{l+1}|$. For a standard subdivision algorithm, we have additionally that

$$1 = \lambda_0 > \lambda_1 = \lambda_2 > |\lambda_3|, \qquad A^m v_i = \lambda_i^m v_i, \quad i = 0, \ldots, 2. \tag{1}$$

Of the decomposition $A = VJV^{-1}$ into the matrix V of eigenvectors and the Jordan matrix J, we will assume, that also the other Jordan blocks are singletons so that $A^m v_i = \lambda_i^m v_i, i = 0, \ldots, \bar{\iota}$. Without loss of generality, we may assume that all eigenvectors are effective [PR08]. That is

$$\text{for no } v_i \neq 0 \text{ with } A^m v_i = \lambda_i^m v_i \text{ and } \lambda_i \neq 0 \text{ is } Gv_i = 0.$$

A *spline in subdivision form* is then the union of spline rings and their limit point $\mathbf{x}^\infty$:

$$\mathbf{x} : \mathbb{S}_n \to \mathbb{R}^d, \qquad \mathbb{S}_n := \mathbf{\Sigma} \times \{1, 2, \ldots, n\},$$

with a central singularity at $(0, 0, j)$, for $j \in \{1, \ldots, n\}$. If $d = 3$, the spline is a surface piece. If $d = 2$, we call the spline an *embedding* if it is injective and continuous. Analogous to splines built from B-splines, we can define $\mathbf{x}$ in terms of the vector of *generating splines* B

$$\mathbf{x} = B\mathbf{Q}, \qquad B(\sigma, j) := G(2^m\sigma, j)A^m \text{ for } \sigma \in \mathbf{\Sigma}^0/2^m.$$

So, while G is used in defining a *spline ring* $\mathbf{x}^m$, B defines a whole spline (cap) $\mathbf{x}$ made up of a sequence of spline rings plus a central point. Since A^m only determines the refined control points $A^m\mathbf{Q}$ near the central point, we use a second, non-square matrix M^m of size $\bar{\iota}_m \times \bar{\iota}$ that defines *all* $\bar{\iota}_m$ control points $M^m\mathbf{Q}$ after m subdivision steps.

We can split $\mathbf{Q} = \sum_{i=0}^{\bar{\iota}-1} \mathbf{p}_i v_i$, and correspondingly the spline $\mathbf{x}$ as follows:

$$\mathbf{x} = \sum_{i=0}^{\bar{\iota}-1} \mathbf{p}_i e_i, \qquad \text{where } e_i(\frac{\sigma}{2^m}, j) := \lambda_i^m (Gv_i)(\sigma, j), \text{ for } \sigma \in \mathbf{\Sigma}^0. \tag{2}$$

The weights $\mathbf{p}_i \in \mathbb{R}^d$ are called eigencoefficients, the functions Gv_i eigenrings and the functions e_i eigensplines. If $d = 3$ and any triple of eigencoefficients spans $\mathbb{R}^d$ then we call $\mathbf{Q}$ *generic*. For example, this rules out the case where all control points lie in one plane. Since there are no ineffective eigenvectors, eigensplines corresponding to non-zero eigenvalues are linearly independent [PR08, Lemma 4.25] and

$$e_i(\frac{\sigma}{2^m}, j) = \lambda_i^m e_i(\sigma, j). \tag{3}$$

In particular, since the generating splines form a partition of unity, $e_0 \equiv 1$; and if

$$\chi := (e_1, e_2)$$

is an embedding and normalized, it is called *characteristic spline*. (Normalization means that we need only consider a unique χ in the following and no linear transformations thereof. For example, the characteristic spline of Catmull-Clark-subdivision can be normalized by setting $\chi(1, 1, 0) = (1, 0)$.) The characteristic spline consists of a nested sequence of scaled copies of the characteristic ring.

1.2 Review of Related Work

A general result, that applies to box splines relevant in our context, is [dHR93, Thm 30]: The convergence of the control polyhedron to its spline is proportional to $\left(\frac{1}{2}\right)^{2m}$ in m binary subdivision steps. A few references characterize additionally the constants of proportionality that are of interest when focusing on few refinement steps. Tight estimates are, for example, derived in [PK94, NPL99, Rei00].

For subdivision surfaces, estimates of convergence proportional to the subdominant eigenvalues λ_1 and λ_2 of A or estimates proportional to first control-net differences predict considerably more subdivision steps than necessary. For example, [ZC06] predicts, for Catmull-Clark and $n = 8$, a decrease of the distance with each subdivision step by ca .8125. If this bound were sharp, 12 subdivision steps would be necessary to cut the distance to 10% of the initial. This means generating 4^{12} facets for each original facet, i.e. the estimate does not match observations and is not practically useful. Similar estimates, ever more sophisticated in the details, can be found in [CC06, CY06, CCY06, HW07b, HW07a].

It is possible to obtain estimates proportional to the maximum of 1/4 and the smallest subsubdominant eigenvalue λ_3 instead. For this, it is necessary to make use of the characteristic spline [Rei95] near extraordinary points. The estimate based on $n = 8$ and hence $\lambda_3 = .5$ predicts only four steps or 4^4 facets to cut the distance to 10% of the initial. It is well-known that $\mathbf{x} \circ \chi^{-1}$ is a good way to parameterize a subdivision surface $\mathbf{x}$ near a singularity. However, computing estimates based on χ^{-1}, the inverse of a piecewise polynomial map is cumbersome [BMZ04]. A better approach for characterizing distance, taken in this paper, is to reparameterize the control polyhedron $\boldsymbol{\ell}$ by χ, i.e. to measure the distance between $\mathbf{x}$ and the composition $\boldsymbol{\ell} \circ \chi$ of the control polyhedron and the characteristic map. Chapter 8.1 of [PR08] builds on this approach and the present paper to obtain tight bounds on the distance between subdivision surfaces and their proxy splines near extraordinary points. Mimicking subdivision surfaces, *proxy splines* have generators that form a partition of unity and converge under refinement so that, just like the control polyhedron, a proxy surface approximates its subdivision surface.

The papers [WSQ04, HK04] base their estimates on the hypothesis that the maximal distance between limit surface and control polyhedron is taken on for the iterates of the initial control points. This hypothesis is false. However, Section 5 shows that measuring distance at the control points provides a heuristic that is often good. Despite its seemingly relevant name “Determining a geometric error of a polygon in a subdivision surface”, US patent 7054796 is just a simple heuristic for comparing facet normals and approximate normals at vertices to gauge flatness of the facet from the viewing direction.

There are a number of *a posteriori* estimates both for the regular control polyhedra and for specific subdivision schemes [FMM86, LP01, Kob98, GS01]. To compare results, we combine dense evaluation with tight and safe one-sided *a posteriori* bounds on the distance between the subdivision surface and its control polyhedron, based on the paper [WP04a] and the software [WP04b].

2 Problem statement

Where there is no singularity, the distance between the spline surface and the control polyhedron, defined as interpolant of the control points in $\mathbf{Q}$, is assumed to be known. In particular, for box-splines the distance between the spline and its control polyhedron is well understood in terms of second derivatives [PK94, NPL99, Rei00]: Under subdivision, due to the scaling of the parametrization and hence also of the second derivatives, this distance decays like γ^m for a constant $\gamma < 1$. If several singularities are close together, one or more subdivision steps isolate the singularities. We therefore focus on splines with one isolated singularity at the center and consisting of n patches as depicted in Figure 3.

Definition 2.1 (Control polyhedron) *Given a triangulation (quadrangulation) of the convex hull $\mathbb{D} \subsetneq \mathbb{R}^2$ of the m-times refined control points of χ,*

$$w^m := (M^m v_1, M^m v_2) \in \mathbb{R}^{\bar{\iota}_m \times 2},$$

the mth control polyhedron *for $\mathbf{Q} \in \mathbb{R}^{\bar{\iota} \times d}$,*

$$\boldsymbol{\ell}^m_{[\mathbf{Q}]} : \mathbb{D} \to \mathbb{R}^d,$$

is defined as the interpolant of $(M^m\mathbf{Q})(k)$ at $w^m(k)$ and varying linearly over each triangle (bilinearly over each quadrangle).

We note that M^m, v_1, v_2 and hence w^m and $\boldsymbol{\ell}^m_{[\mathbf{Q}]}$ depend on n and that $\boldsymbol{\ell}^m_{[\mathbf{Q}]}$ maps to $\mathbb{R}$ when $\mathbf{Q} = v_i$, i.e. $d = 1$. The layout of most triangles (quadrangles)

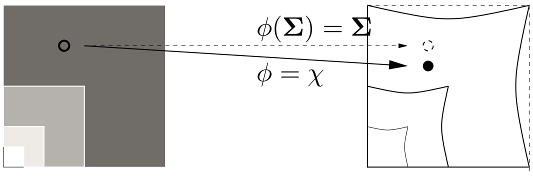

Figure 4: Lemma 1: Only reparametrization by χ allows the characteristic control polyhedron to match the the characteristic spline.

is determined by the knot lines of G. But since the partition of unity of G only guarantees that $\chi(\mathbb{S}_n)$ lies within $\mathbb{D}$, we define $\boldsymbol{\ell}^m_{[\mathbf{Q}]}$ over the full convex hull.

As an example, for Catmull-Clark subdivision, $\boldsymbol{\ell}^m_{[\mathbf{Q}]}$ bilinearly interpolates always four control points. When n is the regular valence, for example $n = 4$ for Catmull-Clark-subdivision, then χ is a rigid embedding and the definition coincides with the standard definition of a tensor-product B-spline control polyhedron. Moreover, $\boldsymbol{\ell}^m_{[\mathbf{Q}]} \circ \chi$ reproduces the first terms of the eigenexpansion (2), as one implication of the following lemma shows (cf. Figure 4).

Lemma 1 (Choice of embedding) *Let $\phi : \mathbb{S}_n \mapsto \mathbb{D}$ be an embedding. Then for each $m \geq 0$*

$$\boldsymbol{\ell}^m_{[v_i]} \circ \phi = e_i, \quad i = 1, 2, \tag{4}$$

if and only if $\phi = \chi$.

Proof By definition of $\boldsymbol{\ell}^m_{[\mathbf{Q}]}$, for every characteristic control point $w^m(k)$,

$$[\boldsymbol{\ell}^m_{[v_1]}, \boldsymbol{\ell}^m_{[v_2]}]\big(w^m(k)\big) = \big(M^m w^0\big)(k) = w^m(k).$$

By (bi-)linear interpolation, $[\boldsymbol{\ell}^m_{[v_1]}, \boldsymbol{\ell}^m_{[v_2]}]$ is therefore the identity on $\mathbb{D}$. This implies the claim. |||

Given the definition of the control polyhedron, we can now define the distance to be minimized.

Definition 2.2 (Distance to the control polyhedron) *Let $\phi : \mathbb{S}_n \to \mathbb{D}$ be an embedding and $\mathbf{x} : \mathbb{S}_n \to \mathbb{R}^d$ a spline with control points $\mathbf{Q}$. The distance between the control polyhedron $\boldsymbol{\ell}^m_{[\mathbf{Q}]}$ and $\mathbf{x}$ is measured as*

$$\min_{\phi} \; \|\mathbf{x} - \boldsymbol{\ell}^m_{[\mathbf{Q}]} \circ \phi\| \tag{5}$$

where $\|\cdot\| := \max_{(\sigma,j)\in\mathbb{S}_n} \|\cdot\|_2$ and $\|\cdot\|_2$ is the Euclidean norm in $\mathbb{R}^d$.

That is, for the neighborhood $\mathbb{S}_n$ of each isolated singularity, $\|\cdot\|$ measures the maximal distance of a point on the spline to a point on its control polyhedron parameterized by the embedding ϕ. For a given step m, we want to minimize this norm by choice of ϕ.

3 Distance and Bounds

To estimate the norm in (5) for the special case $\phi = \chi$ we define

$$\delta(m,n,\mathbf{Q}) : (\sigma,j) \in \mathbb{S}_n \;\mapsto\; (\mathbf{x} - \boldsymbol{\ell}^m_{[\mathbf{Q}]} \circ \chi)(\sigma,j) \quad \in \mathbb{R}^d. \tag{6}$$

Note that $\boldsymbol{\ell}^m_{[\mathbf{Q}]}$ and χ depend on n.

Lemma 2 (Vanishing δ) *For $i = 0,1,2$,*

$$\delta(m,n,v_i) = 0. \tag{7}$$

Proof Constant terms are reproduced since $\boldsymbol{\ell}^m_{[v_0]} \equiv 1 = G\mathbf{1}$ where $\mathbf{1}$ is a vector of ones, the eigenvector v_0 of the leading eigenvalue 1. The remainder of the claim follows from (4). |||

When parameterized by χ, the control polyhedron has a scaling property analogous to that of the eigensplines.

Lemma 3 (Scaling of δ) *For $(\sigma,j) \in \mathbb{S}^0_n$ and $i = 0,\dots,\bar{\imath}-1$,*

$$\delta(m,n,v_i)\big(\frac{\sigma}{2^m},j\big) = \lambda_i^m \delta(0,n,v_i)(\sigma,j). \tag{8}$$

Proof For any characteristic control point $(v_1(k), v_2(k))$,

$$\begin{aligned}\boldsymbol{\ell}^m_{[v_i]}\big(\lambda_1^m v_1(k), \lambda_2^m v_2(k)\big) &=^{(1)} \boldsymbol{\ell}^m_{[v_i]}\big((A^m v_1)(k), (A^m v_2)(k)\big) \\ &=^{Def2.1} A^m v_i(k) =^{(1)} \lambda_i^m v_i(k) =^{Def2.1} \lambda_i^m \boldsymbol{\ell}^0_{[v_i]}\big(v_1(k), v_2(k)\big).\end{aligned} \tag{9}$$

By the convex hull property, for every $(\sigma,j) \in \mathbb{S}^0_n$, there exist nonnegative weights $\alpha(k_l) \in \mathbb{R}$, $l = 1,\dots,\bar{l}$ so that $\chi(\sigma,j) = \sum_{l=1}^{\bar{l}} \alpha(k_l)(v_1(k_l), v_2(k_l))$, i.e. $\chi(\sigma,j)$ lies in some triangle ($\bar{l} = 3$) or quadrangle ($\bar{l} = 4$) with vertices $(v_1(k_l), v_2(k_l))$. The (bi-)linearity of the control polyhedron then implies

$$\boldsymbol{\ell}^m_{[v_i]}(\lambda_1^m \sum_{l=1}^{\bar{l}} \alpha(k_l)(v_1(k_l), v_2(k_l))) = \sum_{l=1}^{\bar{l}} \alpha(k_l) \boldsymbol{\ell}^m_{[v_i]}(\lambda_1^m (v_1(k_l), v_2(k_l))). \tag{10}$$

Together with (9) this yields

$$\begin{aligned}\big(\boldsymbol{\ell}^m_{[v_i]} \circ \chi\big)\big(\frac{\sigma}{2^m},j\big) &=^{(3)} \boldsymbol{\ell}^m_{[v_i]} \circ \lambda_1^m \chi(\sigma,j) \\ &=^{(10)} \sum_{l=1}^{\bar{l}} \alpha(k_l) \boldsymbol{\ell}^m_{[v_i]}(\lambda_1^m v_1(k_l), \lambda_2^m v_2(k_l))) \\ &=^{(9)} \lambda_i^m \boldsymbol{\ell}^0_{[v_i]}(\sum_{l=1}^{\bar{l}} \alpha(k_l)(v_1(k_l), v_2(k_l))) \\ &= \lambda_i^m \big(\boldsymbol{\ell}^0_{[v_i]} \circ \chi\big)(\sigma,j)\end{aligned} \tag{11}$$

so that

$$\begin{aligned} \delta(m,n,v_i)\big(\frac{\sigma}{2^m},j\big) &= (e_i - \boldsymbol{\ell}^m_{[v_i]} \circ \chi)\big(\frac{\sigma}{2^m},j\big) \qquad (12)\\ &=^{(3),(11)} \lambda_i^m (e_i - \boldsymbol{\ell}^0_{[v_i]} \circ \chi)\big(\sigma,j\big) = \lambda_i^m \delta(0,n,v_i)\big(\sigma,j\big) \end{aligned}$$

as claimed. |||

The lemma suggests the following expansion of $\delta(m,n,\mathbf{Q})$.

Theorem 1 (Distance and subsubdominant eigenvalue) *Let (A,G) be a standard subdivision algorithm without generalized eigenvectors. Then for $\sigma \in \mathbf{\Sigma}^0$*

$$\delta(m,n,\mathbf{Q})\big(\frac{\sigma}{2^m},j\big) = \sum_{i=3}^{\bar{\iota}-1} \mathbf{p}_i \lambda_i^m \delta(0,n,v_i)(\sigma,j). \qquad (13)$$

Proof We apply, in turn, the eigenexpansion (2), Lemma 2, and the scaling of Lemma 3 to obtain for $\sigma \in \mathbf{\Sigma}^0$

$$\begin{aligned} \delta(m,n,\mathbf{Q})\big(\frac{\sigma}{2^m},j\big) &=^{(2)} \sum_{i=0}^{\bar{\iota}-1} \mathbf{p}_i \delta(m,n,v_i)\big(\frac{\sigma}{2^m},j\big)\\ &=^{(7)} \sum_{i=3}^{\bar{\iota}-1} \mathbf{p}_i \delta(m,n,v_i)\big(\frac{\sigma}{2^m},j\big)\\ &=^{(8)} \sum_{i=3}^{\bar{\iota}-1} \mathbf{p}_i \lambda_i^m \delta(0,n,v_i)(\sigma,j). \qquad (14) \end{aligned}$$

|||

Since the first two equalities of Equation (14) hold for all σ in $\mathbf{\Sigma}$, we can bound (5) first by $\|\delta(m,n,\mathbf{Q})\|$ and then apply the triangle inequality to obtain an upper bound in terms of eigensplines.

Corollary 1 (Upper bound) *Let (A,G) be a standard subdivision algorithm without generalized eigenvectors. Then*

$$\min_{\phi:\mathbb{S}_n\to\mathbb{D}} \|\mathbf{x} - \boldsymbol{\ell}^m_{[\mathbf{Q}]} \circ \phi\| \le \|\delta(m,n,\mathbf{Q})\| \le \sum_{i=3}^{\bar{\iota}-1} \|\mathbf{p}_i\|_2 \|\delta(m,n,v_i)\|. \qquad (15)$$

The bound of Corollary 1 need not be sharp, because we applied the triangle inequality and because we replace $\delta(m,n,v_i)$ by a maximal absolute value over $\mathbb{S}_n$ and this value is taken on at different parameters (σ,j) for different eigenvectors v_i.

The last equality of (14) can only be used to derive an upper bound near the singularity and does not necessarily determine the convergence rate: for low valences and Loop subdivision, $\lambda_3 < 1/4$ but the convergence is dominated by regular spline subdivision and by powers of $1/4$. Defining $\|\cdot\|_m := \max_{(\sigma,j),\sigma\in\mathbf{\Sigma}^0/2^m} \|\cdot\|_2$, we can use the last equality of Equation (14) to bound near the singularity.

Corollary 2 (Asymptotic Upper bound)

$$\min_{\phi:\mathbb{S}_n\to\mathbb{D}} \|\mathbf{x}-\boldsymbol{\ell}^m_{[\mathbf{Q}]}\circ\phi\|_m \le \sum_{i=3}^{\bar{\iota}-1} \|\mathbf{p}_i\|_2|\lambda_i|^m\|\delta(0,n,v_i)\|_0. \tag{16}$$

Since the eigensplines are linearly independent, $\delta(0,n,v_i)\neq 0$ for $i\ge 3$ and, without loss of generality, we can scale v_i (and inversely scale $\mathbf{p}_i$) so that $\|\delta(0,n,v_i)\|_0=1$. Then, for an important family of input points, we can give the exact distance under characteristic reparameterization.

Corollary 3 (A sharp bound) *Let (A,G) be a standard subdivision algorithm, $\mathbf{0}\in\mathbb{R}^{d-1}$, $\mathbf{Q}:=\sum_{i=0}^{2}\mathbf{p}_iv_i+(\mathbf{0},v_3)$ and v_3 scaled so that $\|\delta(0,n,v_3)\|_0=1$. Then*

$$\|\delta(m,n,\mathbf{Q})\|_m=|\lambda_3|^m.$$

Proof By (14), for $\sigma\in\mathbf{\Sigma}^0$,

$$\delta(m,n,\mathbf{Q})(\frac{\sigma}{2^m},j)=\lambda_3^m\delta(0,n,(\mathbf{0},v_3))(\sigma,j)$$

and hence $\|\delta(m,n,\mathbf{Q})\|_m=|\lambda_3^m|\|\delta(0,n,v_3)\|_0=|\lambda_3|^m$. |||

Since generic data $\mathbf{Q}$ has a component corresponding to an eigenvalue of size $|\lambda_3|$, the asymptotic convergence as $m\to\infty$ is generically proportional to $|\lambda_3|^m$. Indeed, in the example below, λ_3^m is shown to bound the Hausdorff distance from below.

Example 1 (Hausdorff distance of Loop Subdivision) *For Loop subdivision, $n=4$ and $\mathbf{Q}:=(v_1,v_2,v_3)$, λ_3^m bounds from below the Hausdorff distance between the surface and its control polyhedron.*

Proof The control points $\mathbf{Q}$ define the surface (e_1,e_2,e_3). The eigenvalue corresponding to v_3 is $\lambda_3=\frac{1}{4}$ and its Fourier block is 0 indicating elliptic shape. We may assume that v_3 is scaled so that the central control point and its $n=4$ neighbors have coordinates

$$\mathbf{q}_0:=(0,0,1),\quad \mathbf{q}_i:=(\cos(\frac{2\pi}{4}i),\sin(\frac{2\pi}{4}i),-1),\ i=1,2,3,4.$$

All other control points then have a third coordinate less or equal to -1. For Loop subdivision and $n=4$

$$\mathbf{x}^\infty=\frac{1}{2}(\mathbf{q}_0+\frac{1}{n}\sum_{i=1}^{n}\mathbf{q}_i)=(0,0,0).$$

Applying one step of Loop subdivision, the new central control point and its neighbors are at

$$\mathbf{q}_0^1=\frac{5}{8}\mathbf{q}_0+\frac{3}{8}\frac{\sum_{i=1}^{n}\mathbf{q}_i}{4}=\frac{1}{4}(0,0,1)\quad\text{and } \mathbf{q}_i^1:=\frac{1}{4}(\frac{3}{2}\cos(\frac{\pi}{2}i),\frac{3}{2}\sin(\frac{\pi}{2}i),-1).$$

Iterating, we find $\mathbf{q}_0^m = (0, 0, 4^{-m})$ and therefore the distance of the central control point to the limit surface at $(0, 0)$ after m steps is $4^{-m} = \lambda_3^m$. Since e_3 has its maximum at the central limit point, the distance between the surface and the central control point is minimal at $(0, 0)$. The Hausdorff distance from control polyhedron to the limit surface is therefore at least λ_3^m. |||

4 Generalizations

A notationally more involved but analoguous analysis establishes analoguous results when the two subdominant eigenvalues are not equal, when the rings are not differentiable across patch boundaries and, by using bounds on the Jordan decomposition when there are eigenspaces spanned by multiple vectors. Also the analysis of the alternative distance

$$\min_{\tilde{\phi}} \|\mathbf{x} \circ \tilde{\phi} - \boldsymbol{\ell}^m\|_\infty, \quad \text{where } \tilde{\phi} : \mathbb{R}^2 \to \mathbb{S}_n, \quad \text{and } \|\cdot\|_\infty := \max_{\text{range}(\chi)} \|\cdot\|_2, \quad (17)$$

parallels the analysis in Section 3 if we substitute $\tilde{\phi} = \chi^{-1}$.

Dropping linearity or stationarity on the other hand, implies loss of the eigendecomposition (2) used in the analysis. For applications, the functions in G should form a non-negative partition of unity so that the control polyhedron outlines the surface and has the convex hull property.

The distance estimates of Section 3 transfer to the *interpolating control polyhedron* obtained by applying the projection P that maps control points to their limit on the surface under subdivision. The distance at the vertices is now zero, but the distance between the interpolating polyhedron and the surface needs to be estimated. We then redefine $\boldsymbol{\ell}^m$ to map $Pw^m \mapsto PM^m\mathbf{Q}$, and replay the analysis. Approximating (control polyhedron-based) and interpolating (projected polyhedron-based) distances are closely related as evidenced by rows *tab-approx* and *tab-interp* in Figure 6.

One would like to combine Corollary 2 with estimates for box-splines away from the center. For example, one would expect the bound at $\sigma \in \mathbf{\Sigma}^0/2^{\tilde{m}}$ after m steps to be a multiple of $\frac{1}{4^{m-\tilde{m}}}\lambda_3^{\tilde{m}}$. But proving this turns out to be tricky: to prevent subdominant eigenvectors from influencing the error, initial refinement steps need to be estimated using $\phi = \chi$ for the specific valence, while subsequent standard box-spline estimates are based on an rigid embedding that corresponds to the regular valence. Such a switch of parameterizations is addressed in Chapter 8.1 of [PR08] in the setting of *proxy surfaces*.

5 Predictions for few Subdivision Steps

We now discuss a number of measurements to support claims (ii), (iii) and (iv) of the introduction. The estimates are applied to the data sets (control polyhedra) of Figure 7 and Loop's subdivision. For exact measurement, we applied the efficient and tightly sandwiching bounds of [WP04a, WP04b] to

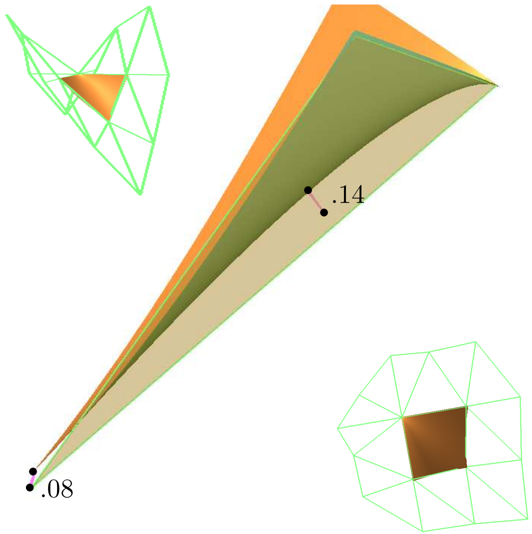

Figure 5: Three views of the relevant parts of the control polyhedron and two patches of the normalized eigenspline e_5 of Loop subdivision for $n = 8$. The central enlargement pinpoints the minimal distance .14 between the patches and the control facets. The maximal distance at a vertex, 0.08, underestimates this Hausdorff distance.

the ten times refined control net (reducing the width between upper and lower bound of the 'sandwich' to at most 10^{-6} times the initial width) and then chose the conservative distance of the farther bound to the control polyhedron. These measurements are labeled *measured* in Table 1 and Figure 6.

The rows marked *heuristic* in Table 1 and Figure 6 record the reduction of the maximal distance after m subdivision steps of a control point to its limit. For Loop subdivision and Catmull-Clark subdivision, the limit is easily computed as the average of the extraordinary node and its neighbors. The results are remarkably close to the row labeled *measured*. While Table 1 shows that the heuristic does underestimate the *measured* distance, the measured distance represents the two-norm of the coordinates and might itself be an overestimate. To prove that the heuristic is not safe requires showing that the heuristic underestimates the Hausdorff distance. Figure 5 shows exactly such an underestimate, for a simple saddle-shaped patch of an eigenspline. The extremum was found by exhaustive comparison using tight conservative, one-sided bounds between points on the surface and points on the facets of the control polyhedron. The extremum is taken on at a nondescript parameter value in the interior, not at the vertices or some vertex after a few binary refinements. Therefore, it is not a matter of applying a few steps of subdivision to make the heuristic safe. A similar example can be constructed for Catmull-Clark subdivision.

The remaining nine types of estimates are based on Equation (14). For valences $n = 3, \ldots, 30$ and refinements $m = 0, \ldots, 5$, we used the bounds of [WP04a] to each eigenfunction and tabulated

$$\delta_{m,n,i} := \|\delta(m, n, v_i)\|.$$

Then we can apply the bound of Corollary 1, (15):

$$\|\delta(m, n, \mathbf{Q})\| \leq \sum_{i=3}^{\bar{\iota}-1} \|\mathbf{p}_i\|_2 \delta_{m,n,i}. \tag{18}$$

This bound is listed as *tab-approx* 0 in Table 1. Applying one local subdivision step followed by the estimate (15) for $m - 1$ yields the estimate listed as *tab-approx* 1. By definition, this row has no entry for $m = 0$. Similarly *tab-approx* 2 uses two local subdivision steps, implemented as a sparse matrix product, and bounds the result to level $m - 2$. By definition, there are no entries for $m = 0$ and $m = 1$.

The bounds listed as *tab-interp* 0, *tab-interp* 1 and *tab-interp* 2 are obtained by the same process except that tabulated distances $\delta(m, n, v_i)$ are replaced by distances between the eigenfunctions and the triangulation formed by mapping the control points of the eigenfunction to their limit position, i.e. by the interpolating triangulation rather than the approximating control polyhedron.

The final set of estimates, listed as *tab-eigen* 0,1,2, combines the standard decay estimate with the last equality of Equation (14) as discussed in the last paragraph of Section 4. It measures

$$\sum_{i=3}^{\bar{\iota}} \|\mathbf{p}_i\|_2 \max\{\lambda_i^m, \frac{1}{4^m}\}\delta_{0,n,i},$$

which turns out to be a weaker and less stable estimate than the upper bound according to Corollary 1.

Cost of bound computations. Since the constants $\delta_{m,n,i}$ are pretabulated, the main runtime cost when estimating m for a particular control polyhedron is the eigendecomposition, i.e. the application of a $\bar{\iota} \times \bar{\iota}$ matrix of precomputed eigenvectors to the control net (see e.g. [Sta98]). We could avoid the eigendecomposition by bounding and tabulating the maximal distance for functions associated with each control point rather than the eigenfunctions. But this would result in a poorer estimate due to a sum including also the subdominant terms, not just those for $i \geq 3$. The $\bar{\iota} \times \bar{\iota}$ matrices for $n = 3, \ldots, 30$ consist of 16002 numbers. We also store another 3276 constants $\delta_{m,n,i}$ for $m = 0, \ldots, 5$. For example, for Loop subdivision and a patch with one node of valence $n = 7$ and subdivision depth $m = 3$, the total runtime cost per point coordinate is 180 adds and multiplications, including the cost of local subdivision.

In our implementation, *a posteriori* estimation costs at least three times as much as the *a priori* estimation, already for two subdivision steps. The advantage of prediction increases with subdivision depth m. For the venus model

(5672 triangles), applying one local subdivision plus *tab-approx* prediction has a *total cost over all facets* of .08s (seconds) plus .27s. By comparison, for the same model and two steps of refinement, subdivision-followed-by-safely-*measured* distance costs .43s (for subdivision) plus .63s (for measuring). Of course in the *a posteriori* estimation all the subdivision work is already done – but that may not be desirable for some applications, e.g. for very large inputs with only a few patches requiring adaptive refinement.

Measurements: Table 1 lists the bounds for Loop subdivision applied to the objects in Figure 7. The units are specific to each model and only the relative reduction of the distances is relevant. From the raw data of Table 1, we extracted the bar-graphs of Figure 6. First, we computed, for each object, the ratio of the estimate to the *measured* distance. Then we selected the median to as represented by the bars in Figure 6, *top*. Figure 6, *bottom*, shows the variance of the ratios from the median. We can make the following observations.
(1) The estimator *tab-eigen* provides the loosest bounds and the bounds with the most fluctuation over different models (The valence-3 spikes of the model `Star` result in high deviation.)
(2) The heuristic has nearly half of the ratios below 1. The estimates are close to *measured*.
(3) The estimator *tab-interp* is consistently slightly smaller than *tab-approx*. The constant factor might be expected since the interpolating triangulation is a linear offset from the control polyhedron.
(4) The variance in the entries of *tab-approx* 2 is low. Except for the model `Star` where two more steps are predicted than required, only one unnecessary step is predicted. This advertises *tab-approx* 2 (and, similarly, *tab-interp* 2) as a practically useful safe distance predictor.

6 Conclusion

It is well-known that near points of high valence, standard subdivision algorithms used in practice exhibit slow contraction of the control polyhedron, dubbed 'polar artifact' [SB02]. Polar artifacts result from large values of λ_1 and λ_2. This paper explains the impact of large values of λ_3. For high valences, the subsubdominant eigenvalue is often closer to $\frac{1}{2}$ than to $\frac{1}{4}$ so that convergence of the control polyhedron to the surface near such singularities (extraordinary points) is much slower than for the rest of the control polyhedron. For example, $\lambda_3 = \frac{1}{2}$ for Catmull-Clark subdivision when $n = 8$. For Loop subdivision $\lambda_3 = \frac{1}{2}$ when $n = 12$.

The well-known quadratic convergence of the control polyhedron to the box-spline under binary subdivision by a factor of $\frac{1}{4}$ in each subdivision step agrees with the convergence according to the subsubdominant eigenvalue $\frac{1}{4}$.

Basing estimates on subsubdominant eigenvalues as opposed to first-order estimates clearly improves the chances to closely predict the minimal number of subdivision steps m so that every point of the limit surface is within a fixed distance of the control polyhedron or its projection.

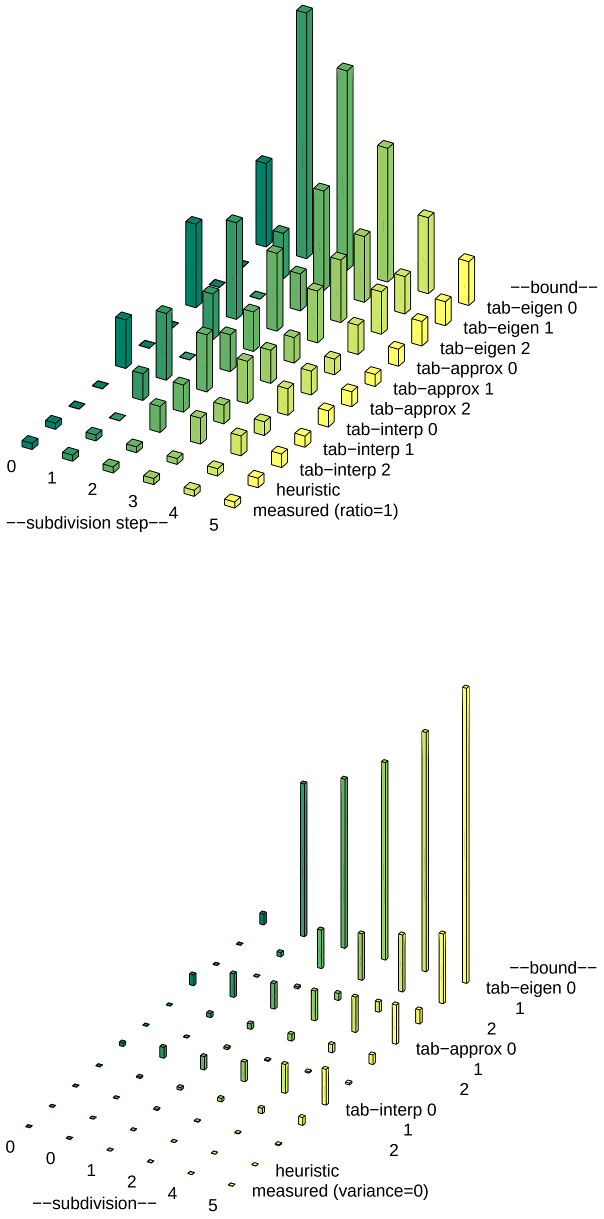

Figure 6: (*top*) Ratio of estimates in Table 1 to *measured*. (*bottom*) Variance of ratios.

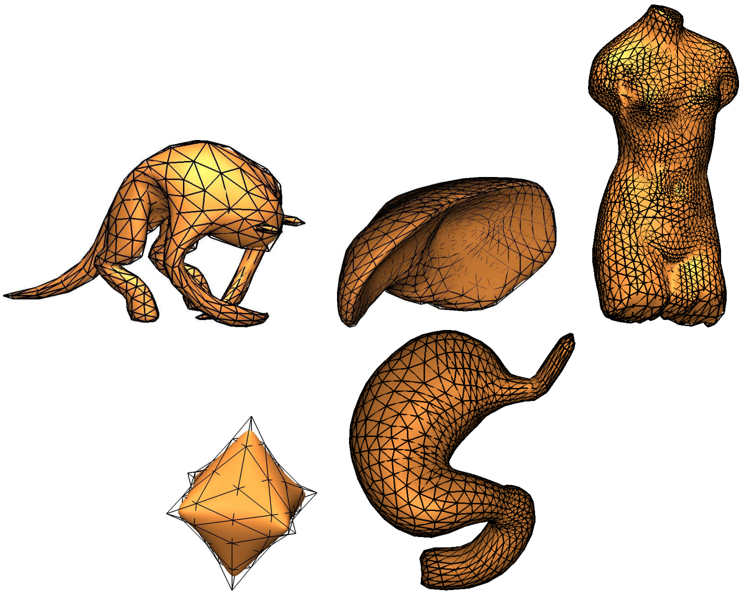

Figure 7: Test objects (number of input triangles): Demon (1096), Liver (1096), Venus (5672), Star (96), Stomach (2136).

Our experiments illustrate that a simple heuristic can yield tight estimates – but an example shows that the heuristic is unsafe. More sophisticated, safe estimates predict more subdivision steps than *a posteriori* measurement proves necessary. This indicates that safe and exact prediction is difficult to achieve. Combining safe prediction with one or two local subdivision steps appears to be the most effective strategy to avoid the cost of subdividing and measuring the entire refined mesh. Together, the experiments and the measurements reveal a scale of options for trading quality (correctness and tightness) for cost: subdivide-followed-by-measured is exact but expensive, the heuristic is cheap but not safe, and the tabulated predictions are safe but not as tight is measuring after subdivision.

Acknowledgement: This work was supported by NSF Grants CCF-0430891 and DMI-0400214. The presentation benefited from high quality reviews and the editor's suggestion to formulate the results in the terms defined in [PR08].

References

[BMZ04] Ioana Boier-Martin and Denis Zorin. Differentiable parameterization of Catmull-Clark subdivision surfaces. In *Proceedings of the Symposium on Geometry Processing*. ACM Press, 2004.

[CC78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10:350–355, 1978.

[CC06] G. Chen and F. Cheng. Matrix based subdivision depth computation for extra-ordinary catmull-clark subdivision surface patches. In *Proceedings of GMP 2006 (LNCS 4077)*, pages 545–552, 2006.

[CCY06] F. Cheng, G. Chen, and J. Yong. Subdivision depth computation for extra-ordinary catmull-clark subdivision surface patches. In *Proceedings of GMP 2006 (LNCS 4077)*, pages 545–552, 2006.

[CY06] F. Cheng and J. Yong. Subdivision depth computation for catmull-clark subdivision surfaces. *Computer Aided Design & Applications*, 3:485–494, 2006.

[dHR93] C. de Boor, K. Höllig, and S. Riemenschneider. *Box splines*, volume 98 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1993.

[FMM86] D. Filip, R. Magedson, and R. Markot. Surface algorithms using bounds on derivatives. *Computer Aided Geometric Design*, 3(4):295–311, 1986.

[GS01] Eitan Grinspun and Peter Schröder. Normal bounds for subdivision-surface interference detection. In Thomas Ertl, Ken Joy, and Amitabh Varshney, editors, *Proc Visualization*, pages 333–340. IEEE, 2001.

[HK04] Wang Huawei and Qin Kaihuai. Estimating subdivision depth of Catmull-Clark surfaces. *J. Computer Sci and Tech*, pages 657–664, 19/5 2004.

[HW07a] Z. Huang and G. Wang. Distance between a catmull-clark subdivision surface and its limit mesh. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 233 – 240, 2007.

[HW07b] Z. Huang and G. Wang. Improved error estimate for extraordinary catmull-clark subdivision surface patches. In *Proceedings of CGI 2007*, 2007.

[Kob98] L. Kobbelt. Tight bounding volumes for subdivision surfaces. In Bob Werner, editor, *Pacific-Graphics'98*, pages 17–26. IEEE, 1998.

[Loo87] Charles T. Loop. Smooth subdivision surfaces based on triangles, 1987. Master's Thesis, Department of Mathematics, University of Utah.

[LP01] D. Lutterkort and J. Peters. Tight linear bounds on the distance between a spline and its B-spline control polygon. *Numerische Mathematik*, 89:735–748, May 2001.

[NPL99] D. Nairn, J. Peters, and D. Lutterkort. Sharp, quantitative bounds on the distance between a polynomial piece and its Bézier control polygon. *Computer-Aided Geometric Design*, 16(7):613–633, Aug 1999.

[PK94] H. Prautzsch and L. Kobbelt. Convergence of subdivision and degree elevation. *Advances in Computational Mathematics*, 2:143–154, 1994.

[PR08] J. Peters and U. Reif. *Subdivision Surfaces*, volume 3 of *Geometry and Computation*. Springer-Verlag, New York, 2008.

[Rei95] U. Reif. A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design*, 12:153–174, 1995.

[Rei00] U. Reif. Best bounds on the approximation of polynomials and splines by their control structure. *Comput. Aided Geom. Design*, 17(6):579–589, 2000.

[SB02] Malcolm Sabin and Loic Barthe. Analysis and tuning of subdivision algorithms. In *Curves and Surfaces*. Vanderbilt Press, 2002.

[SP05] L. Shiue and J. Peters. Mesh refinement based on euler encoding extraordinary points. In *Proceedings of The International Conference on Shape Modeling and Applications*, pages 1–6, 2005.

[Sta98] Jos Stam. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. In Michael Cohen, editor, *SIGGRAPH 98 Proceedings*, pages 395–404. Addison Wesley, 1998.

[WP04a] X. Wu and J. Peters. Interference detection for subdivision surfaces. *Computer Graphics Forum, Eurographics 2004*, 23(3):577–585, 2004.

[WP04b] X. Wu and J. Peters. The SubLiME package (Subdividable Linear Maximum-norm Enclosure), 2004. http://www.cise.ufl.edu/research/SurfLab/SubLiME.

[WSQ04] Huawei Wang, Hanqiu Sun, and Kaihuai Qin. Estimating recursion depth for Loop subdivision. *International Journal of CAD/CAM*, 4(1), 2004.

[ZC06] Xiao-Ming Zeng and X. J. Chen. Computational formula of depth for Catmull-Clark subdivion surfaces. *J. Comp. Appl. Math.*, 195(1):252–262, Oct 2006.

mesh	estimate	$m = 0$	1	2	3	4	5
star	*tab-eigen* 0	9.602865	6.412230	1.603057	0.400764	0.100191	0.025048
	tab-eigen 1		1.189783	0.792618	0.198155	0.049539	0.012385
	tab-eigen 2			0.148528	0.098793	0.024698	0.006175
	tab-approx 0	9.602865	2.520464	0.624280	0.156891	0.038926	0.009289
	tab-approx 1		1.189783	0.312280	0.077330	0.019439	0.004823
	tab-approx 2			0.148528	0.038983	0.009652	0.002427
	tab-interp 0	5.587574	1.735903	0.456955	0.127762	0.035283	0.008853
	tab-interp 1		0.693271	0.215440	0.056655	0.015838	0.004374
	tab-interp 2			0.086626	0.026923	0.007077	0.001978
	heuristic	0.493182	0.082623	0.020656	0.005164	0.001291	0.000323
	measured	0.492493	0.089411	0.022129	0.005563	0.001380	0.000329
demon	*tab-eigen* 0	24.765944	16.506170	4.126542	1.031636	0.257909	0.064477
	tab-eigen 1		3.095206	2.060041	0.515010	0.128753	0.032188
	tab-eigen 2			0.741884	0.342345	0.085586	0.021397
	tab-approx 0	24.765944	6.500835	1.609795	0.404699	0.100407	0.023958
	tab-approx 1		3.095206	0.812431	0.201164	0.050577	0.012698
	tab-approx 2			0.741884	0.191016	0.048463	0.012573
	tab-interp 0	14.433803	4.485724	1.179442	0.329698	0.091071	0.022840
	tab-interp 1		1.805383	0.561142	0.147485	0.041224	0.011652
	tab-interp 2			0.535782	0.151316	0.041203	0.011504
	heuristic	1.416932	0.394898	0.120230	0.047276	0.022095	0.011099
	measured	1.517119	0.404152	0.124367	0.044160	0.018061	0.007068
venus	*tab-eigen* 0	0.593524	0.370544	0.092636	0.023159	0.005790	0.001447
	tab-eigen 1		0.126580	0.061122	0.015281	0.004121	0.001435
	tab-eigen 2			0.061062	0.029721	0.007430	0.001858
	tab-approx 0	0.593524	0.149792	0.037612	0.010438	0.003224	0.001014
	tab-approx 1		0.126580	0.032788	0.008344	0.002606	0.000880
	tab-approx 2			0.061062	0.015708	0.003983	0.001001
	tab-interp 0	0.390317	0.112818	0.031481	0.009628	0.003051	0.000969
	tab-interp 1		0.086038	0.024413	0.007181	0.002444	0.000835
	tab-interp 2			0.043881	0.012317	0.003329	0.000897
	heuristic	0.155856	0.038964	0.009741	0.002727	0.001561	0.000868
	measured	0.159851	0.039848	0.009971	0.002858	0.001006	0.000361
liver	*tab-eigen* 0	1.842082	1.244349	0.311087	0.077772	0.019443	0.004861
	tab-eigen 1		0.227004	0.152919	0.038230	0.009557	0.002389
	tab-eigen 2			0.101263	0.043338	0.010835	0.002709
	tab-approx 0	1.842082	0.483952	0.119934	0.030142	0.007480	0.001785
	tab-approx 1		0.227004	0.059640	0.014776	0.003715	0.001322
	tab-approx 2			0.101263	0.026036	0.006609	0.001661
	tab-interp 0	1.074348	0.334355	0.088078	0.024616	0.006786	0.001702
	tab-interp 1		0.132638	0.041295	0.010863	0.003035	0.001228
	tab-interp 2			0.071502	0.020206	0.005492	0.001488
	heuristic	0.132813	0.030822	0.011009	0.005372	0.002973	0.001761
	measured	0.132811	0.029934	0.008812	0.003498	0.001538	0.000667
stomach	*tab-eigen* 0	1.522140	1.030209	0.257552	0.064388	0.016097	0.004024
	tab-eigen 1		0.189760	0.128229	0.032057	0.008014	0.002249
	tab-eigen 2			0.061201	0.025930	0.006483	0.002249
	tab-approx 0	1.522140	0.399672	0.098984	0.024884	0.006174	0.001784
	tab-approx 1		0.189760	0.049824	0.012338	0.004588	0.001755
	tab-approx 2			0.061201	0.015690	0.004398	0.001736
	tab-interp 0	0.888613	0.276234	0.072625	0.020311	0.005604	0.001628
	tab-interp 1		0.110888	0.034476	0.010681	0.004081	0.001590
	tab-interp 2			0.043091	0.012198	0.003892	0.001569
	heuristic	0.132700	0.033175	0.010174	0.003857	0.001909	0.001003
	measured	0.133740	0.033384	0.012012	0.005087	0.002157	0.000887

Table 1: Distance predictions for m steps of Loop subdivision applied to the objects shown in Figure 7.