4–8 Subdivision

Luiz Velho [a] and Denis Zorin [b]

[a] *Visgraf Laboratory*
IMPA – Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina 110, Rio de Janeiro, RJ, Brazil, 22460-320.
`lvelho@visgraf.impa.br`

[b] *Media Research Laboratory*
Courant Institute of Mathematical Sciences - New York University
715 Broadway, New York, NY 10003
`dzorin@mrl.nyu.edu`

Abstract

In this paper we introduce 4–8 subdivision, a new scheme that generalizes the four-directional box spline of class C^4 to surfaces of arbitrary topological type. The crucial advantage of the proposed scheme is that it uses bisection refinement as an elementary refinement operation, rather than more commonly used face or vertex splits.

In the uniform case, bisection refinement results in doubling, rather than quadrupling of the number of faces in a mesh. Adaptive bisection refinement automatically generates conforming variable-resolution meshes in contrast to face and vertex split methods which require a postprocessing step to make an adaptively refined mesh conforming.

The fact that the size of faces decreases more gradually with refinement allows one to have greater control over the resolution of a refined mesh. It also makes it possible to achieve higher smoothness while using small stencils (the size of the stencils used by our scheme is similar to Loop subdivision).

We show that the subdivision surfaces produced by the 4–8 scheme are C^4 continuous almost everywhere, except at extraordinary vertices where they are is C^1-continuous.

Keywords: subdivision schemes, four-directional grids, Laves tilings, quincunx lattice, binary 4-8 refinement, two-pass smoothing.

1 Introduction

Subdivision surfaces generalize classical spline surfaces. As such, they overcome some limitations of splines and offer several advantages, including the ability to model surfaces of arbitrary topology, as well as the flexibility to combine global smoothness with control of local features, such as creases and corners. They also naturally integrate a continuous surface model with a discrete representation, leading to simple and efficient algorithms.

Practically all previously known subdivision schemes are based on one of the two tilings of the plane: the tiling with regular triangles and the tiling with squares. These tilings are easily refined, and common subdivision schemes are derived from box splines associated with these tilings. For example, three-directional quartic box spline gives raise to the Loop subdivision scheme, and tensor product biquadratic and bicubic splines lead to Doo-Sabin and Catmull-Clark subdivision respectively.

However, regular triangular and quadrilateral tilings are not the only refinable tilings: a complete classification of isohedral tilings, for which all angles at each vertex are equal, was found by Laves (see Section 2.1). This classification includes eleven different types. In this paper we focus on subdivision based on a particular 4–8 tiling, which we demonstrate to have a number of attractive properties. We introduce a new subdivision scheme based on a box spline associated with the 4–8 tiling. There are several advantages specific to this scheme:

- *The basic refinement operation is bisection.* The 4–8 meshes, and their arbitrary topology generalization, 4–k meshes are refined by edge bisection. In contrast to other commonly used refinement operations, such as triangle quadrisection, the result of applying a single bisection to a conforming mesh is a conforming mesh: no cracks can appear. This simplifies adaptive refinement: if a mesh is not refined uniformly, the refined mesh is still guaranteed to be conforming. *Variable resolution meshes* based on 4–8 tilings are discussed in greater detail in Section 2.1.

- *Gradual refinement.* For most common schemes, a single uniform refinement step increases the number of faces or vertices by a factor of 4; a scheme recently introduced by Kobbelt [19] increases the number of faces by a factor of 3. In our case, at each refinement step, the increase is only a factor of 2.

- *Small support and high smoothness.* The masks that we use to implement our scheme have a small support: the support for a vertex mask is even smaller than that of the Catmull-Clark scheme, and the support for the face mask is the same. At the same time, the resulting surface is C^4-continuous on the regular part of the mesh, substantially higher than Catmull-Clark subdivision which is C^2-continuous [1].

- *High symmetry.* The basis function of our scheme in the regular case is invariant with respect to rotations by $\pi/8$; thus it has a large symmetry group compared to basis functions of tensor-product and three-directional box splines.

- *Piecewise-polynomial limit functions.* Like many commonly used schemes, but unlike $\sqrt{3}$ subdivision [19], our scheme has piecewise-polynomial limit functions on the regular meshes, which means it can be evaluated explicitly using techniques proposed in [28].

---

[1] Higher regularity is due to the fact that the mask is applied on every bisection refinement step; two such steps are equivalent to a single face quadrisection step, and the combined mask will be larger than that for the Catmull-Clark scheme.

2

The 4-8 subdivision scheme that we propose has similarities to a number of existing schemes. Uniquely, it combines certain features of triangular and quadrilateral schemes. Like the Loop subdivision scheme [24], it generates a hierarchy of triangular meshes. At the same time, as it operates on triangular meshes partitioned into quadrilateral blocks, it is similar to primal quadrilateral schemes, such as Catmull-Clark [3]. Two steps of our subdivision produce a scheme that can be applied directly to quadrilateral meshes refined by quadrisection, and the support of the masks of such scheme is close to the interpolating scheme scheme described in [17], but smoothness on the regular part is much higher.

As mentioned above, our scheme is also similar to $\sqrt{3}$-subdivision considered in [19]. Our scheme can be viewed as $\sqrt{2}$ subdivision on quadrilateral meshes.

Finally, our scheme is based on a C^4 box spline, a natural higher-degree version of the Zwart-Powell element [36], which was used in [25] and [13] to construct a subdivision scheme for arbitrary meshes.

Our analysis of C^1 continuity is based on [33, 34]; the basic ideas of the analysis were introduced for the first time in [26].

The 4–8 mesh is closely related to quincunx lattices and has also been investigated in image processing [10, 14].

One of the important features of 4–8 meshes, adaptive refinement, was previously considered in the context of rendering of terrain models in [8, 9, 23].

2   4–8 Meshes and Refinement

In this section we review the basic concepts of mesh refinement, introduce regular 4–8 meshes, triangulated quadrilateral meshes, and bisection refinement, For simplicity, in this section we consider only meshes without boundaries; the boundary case is considered in greater detail in Section 4.

2.1   *Regular Refinement and Tilings*

A *refinement rule* is an algorithm that produces a finer mesh from a given mesh. The rule is purely topological: it uses only connectivity information about the mesh and does not use the geometric information (e.g. vertex positions). Typical mesh refinement methods are closely related to regular tilings, that is, tessellations of the plane consisting of identical regular n-gons. A refinement rule applied to the corresponding tiling produces a tiling isomorphic as a graph to the original. In this case we say that a refinement rule leaves a tiling invariant.

There are only three types of regular plane tilings (see e.g. [12]); the tile has to be either a square, an equilateral triangle, or a regular hexagon. Most known refinement schemes are based on square or triangular tilings. Meshes with the same connectivity as a regular tiling are usually referred to as *regular meshes*.

A commonly used refinement rule leaving quadrilateral tiling invariant is *face*

split. This rule operates on arbitrary meshes. A vertex is inserted on each edge, and on each face; the new vertices are connected by edges, so that each face is partitioned into n quads, where n is the number of vertices of the face. After a few refinement steps the refined mesh has the same structure locally as the regular quadrilateral mesh, excluding some vertices inherited from the top level. This property is very important: for a suitable choice of geometric rules for computing new vertex positions, local properties of surfaces generated by iterative refinement (e.g. smoothness) are the same as of surfaces obtained from a regular initial mesh.

2.2 4–8 Meshes

We can obtain new classes of refinement rules if we consider a larger class of tilings, and look for rules that leave these tilings invariant. One possible class to consider is the class of *monohedral tilings* with regular vertices, also known as *Laves tilings*, named after the crystallographer Fritz Laves [12].

In a *monohedral* tiling, every tile is congruent to one fixed tile, called the *prototile*. A vertex v of a tiling is called *regular* if the angle between each consecutive pair of edges that are incident in v is equal to $2\pi/d$, where d is the valence of v. There are eleven tilings that satisfy these two conditions, including three regular tilings.

The prototile of the $[4.8^2]$ Laves tiling [2] is an isosceles right triangle. We refer to it as 4–8 tiling, as it has alternating vertices of valence 4 and 8. This tiling has a rich structure that can be exploited in the context of subdivision with a number of advantages over regular tilings. The basic structure of this tiling is a pair of triangles forming a square block divided along one of its diagonals. We call this structure a *basic block* (Figure 1(a).) We will call the common edge of the two triangles the *interior edge* of the block; all other edges are called *exterior edges*. The $[4.8^2]$ tiling forms a *triangulated quadrangulation*.

We say that a mesh is a *regular 4–8 mesh* if it has the same connectivity as a $[4.8^2]$, or 4–8, Laves tiling. A regular 4–8 mesh has the same block structure as the 4–8 tiling, and its edges can be similarly classified as interior and exterior block edges. Note that each triangle has a single interior edge and two exterior edges. This allows us to use *bisection* as a primitive refinement operation: for each triangle we bisect the unique interior block edge.

Bisection refinement. More precisely, the bisection refinement rule is defined as follows: *Bisection refinement*:

(1) Insert a split vertex on all internal edges of blocks.
(2) Subdivide each face into two sub-faces, by linking the the split vertex on the internal edge to the opposite vertex of the face.

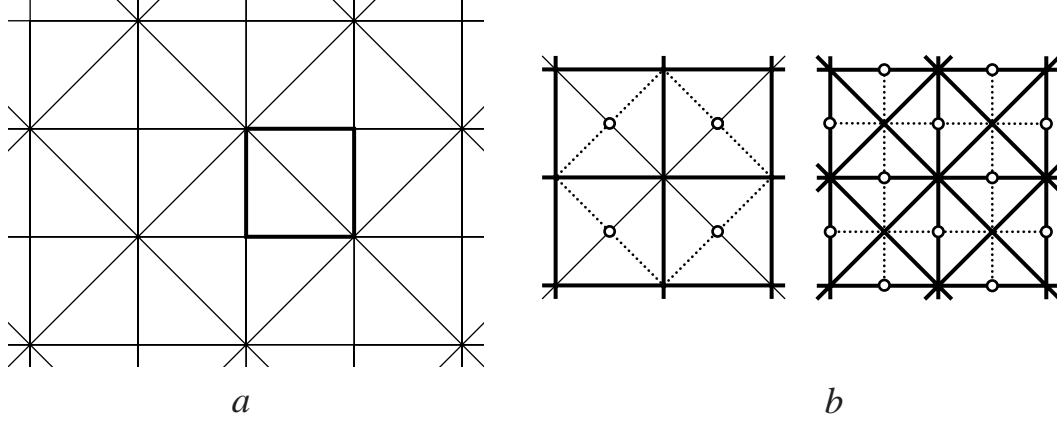[2] An explanation of tiling signatures like $[4.8^2]$ can be found in [12].

Bisection refinement of a regular 4–8 mesh produces a regular 4–8 mesh. On regular 4–8 meshes, bisection refinement is equivalent to quincunx refinement of basic blocks (Section 3.2).

One important property of bisection refinement is that two steps result in a face split of quads formed by pairs of triangles of basic blocks. Thus, bisection refinement can be viewed as a way to decompose the face split into two steps. This is illustrated in Figure 1(b).

Bisection refinement relies on the special topological structure of the mesh, namely, the fact that it is partitioned into the basic blocks. Clearly, it can be applied to an arbitrary mesh partitioned into blocks of two triangles sharing an edge. We call such meshes *triangulated quadrilateral meshes*, or tri-quad mesh for short.

A number of methods can be used to produce tri-quad meshes. If a quadrilateral mesh is given, a tri-quad mesh can be produced by splitting each quad into two triangles. A simple way to convert an arbitrary polygonal mesh into a tri-quad mesh is to use a single step of Catmull-Clark refinement to obtain a quadrilateral mesh, and then split each quad into two triangles. The disadvantage of this method, for arbitrary triangular meshes in particular, is that it may result in six-fold increase in the number of triangles. For triangular meshes, a more complex approach which approximately doubles the number of triangles is described in Section 5.1.

Refinement of tri-quad meshes. Several important observations can be made about the structure and refinement of tri-quad meshes. To state these observations, we introduce notation for vertex types. Note that for any block two vertices are on the diagonal and two are not. We will call a vertex type 1, if for at least one adjacent block it is not on the diagonal. Otherwise, we call it type 2.

- All vertices added by a single refinement step, are type 1 with valence 4.
- A single refinement step converts all vertices of type 1 to type 2 – their valence increases by a factor of no more than two.
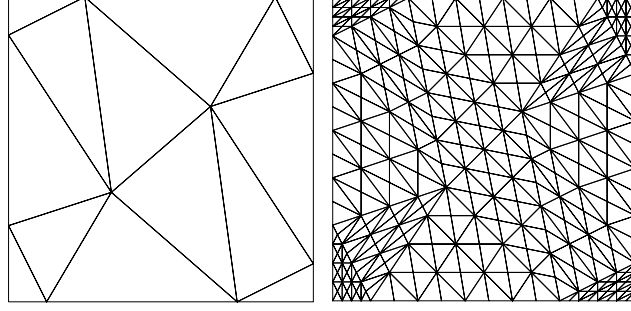- The valence of vertices of type 2 is not changed by refinement.

5

Fig. 2. Bisection refinement of a mesh (4 levels).

After a sufficiently large number of refinement steps, the structure of a neighborhood of a vertex is identical to the structure of a submesh of a 4–8 mesh, excluding the neighborhoods of extraordinary vertices of valence $2n$, $n \neq 4$. The structure of the mesh on such neighborhoods is shown in Figure 2(b). Note that the extraordinary vertices have alternating exterior and interior edges.

Adaptive bisection refinement. Refinement methods such as face or vertex split, cannot be used to refine a mesh adaptively without breaking topological consistency of the mesh, that is, without creating cracks in the mesh. This is particularly clear in the case of face splits, because at every step all edges of a face are subdivided; therefore, to maintain consistency all adjacent faces have to be subdivided to the same level. By induction, the whole mesh would have to be subdivided. To overcome this difficulty, meshes with faces at different refinement levels have to be fixed by a post-process. (see e.g. [15, 32]). This applies to all classical refinement methods. In contrast, bisection refinement generates a hierarchical mesh structure that naturally supports variable resolution.

Suppose for any edge we can evaluate a criterion that tells us whether this edge should be bisected or not. Two cases are possible, as shown in Figure 3: the edge is either interior or exterior. For an interior edge, no other blocks have to be refined to maintain a conforming triangulation. For an exterior edge, two adjacent blocks have to be refined.

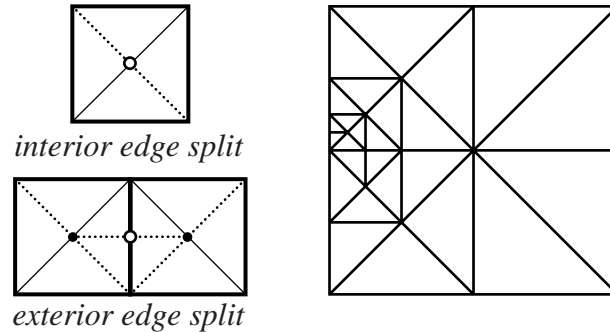*interior edge split*

exterior edge split

Fig. 3. Adaptive refinement of a 4–8 mesh. Left: two cases of edge splits; an interior edge split does not require modification of any other blocks to maintain a conforming mesh. An exterior edge split requires refining a single adjacent block. Right: an example of adaptively refined 4–8 mesh.
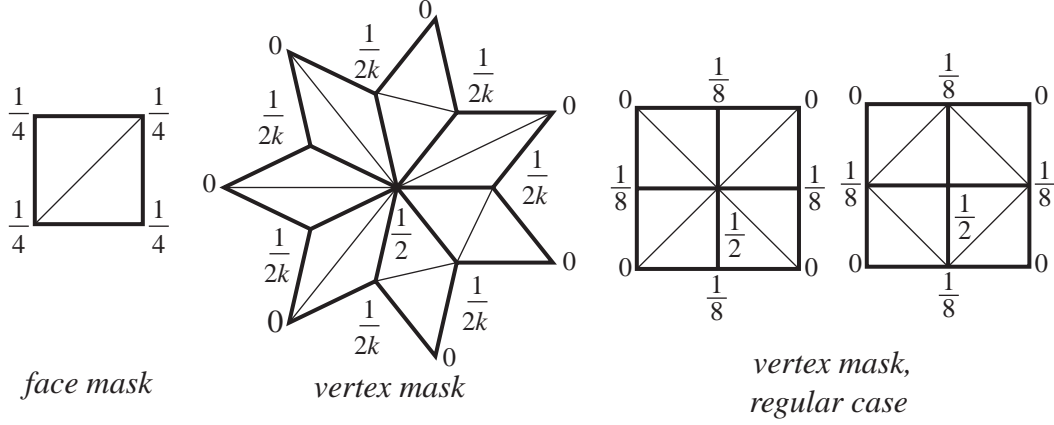
6

Fig. 4. Masks of our subdivision rules; the same face mask is used in all cases; vertex mask coefficients depend on k, the number of incident exterior block edges. In the case shown in the figure, $k = 7$. Note that the same rule applies for any choice of diagonals of the adjacent blocks. For a regular 4–8 mesh only two cases shown on the right are possible.

The above mechanism for refining internal and external edges, makes it possible to build conforming tessellations with different resolution levels. Figure 3 (right) shows an example of constrained resolution propagation.

3   4–8 Subdivision

In this section we present a 4–8 subdivision scheme, based on bisection refinement, and describe its basic properties.

3.1   *Definition of the scheme*

To define a subdivision scheme for a closed tri-quad mesh we need to specify rules for computing positions of the new vertices that we insert when bisecting the basic blocks, and rules to update the positions of the existing vertices. The rules that we propose are very simple (Figure 4):

Face rule: each new vertex inserted as a result of bisection refinement of a basic block is computed as the barycenter of that block;

Vertex rule: the new position of an exisiting vertex v is computed as the average of the old position and barycenter of the vertices sharing an exterior block edge with v.

The vertex rule equally applies to vertices of types 1 and 2. As it was observed in Section 2.2, after one bisection subdivision step all type 2 vertices have even valence and alternating exterior and interior incident block edges; thus, the valence becomes even, and equal to $2k$, where k is the number of basic blocks sharing the vertex. All newly inserted vertices are of type 1, and after one subdivision step turn into type 2 of valence 8.

7

The scheme can be extended to handle meshes with boundary, as discussed in Section 4. Note that the support of the masks is quite small; the face mask is similar to the face mask of the Catmull-Clark scheme, and the vertex mask has even smaller support.

Remarkably, this scheme has a high degree of smoothness:

Proposition 1 *The subdivision scheme defined above has the following properties:*
- *on a regular 4–8 mesh, the limit surface produced by the rules is a C^4 four-directional box spline;*
- *by locality, on a tri-quad the rules produce a C^4-continuous surface away from the extraordinary vertices;*
- *at the extraordinary vertices, the resulting surface is C^1-continuous.*

These properties will be proved in subsequent sections. In the regular case the rules reduce to simple masks shown in Figure 4. The rules of the scheme were obtained as a straightforward generalization of the regular case. The rules for the regular case were obtained as a quincunx subdivision rules for C^4 four-directional box spline as discussed in greater detail in the next section.

Once we establish that the scheme produces C^4 box splines on regular meshes, the analysis of C^1-continuity can be performed using techniques developed in [33]. A detailed outline is presented in Appendix A, where we verify that the scheme is indeed C^1-continuous for all valences.

3.2 C^4 four-directional box spline

Recall that a bivariate box spline is defined by a set of directions $[d_1, \ldots d_n]$. The spline $B_n(x)$, $x \in \mathbb{R}^2$, can be computed using the following recurrence:

$$B_j(x) = \int_0^1 B_{j-1}(x - t d_j) dt$$

with B_0 being the delta function. If the directions d_j are chosen from the set $[1, 0]$, $[0, 1]$, $[1, 1]$, $[1, -1]$ and have equal multiplicities, the spline has the symmetries of the four-directional mesh.

The simplest smooth box spline of this type is the Zwart-Powell function [36], also known as the *ZP element*. It is associated with the set of four directions above, with each direction having multiplicity 1. This spline was used in [25] to define a C^1 dual subdivision scheme on quadrilateral meshes.

A four-directional box spline that exhibits a higher order of smoothness than the ZP element is the function generated by the same set of directions, each taken with multiplicity 2.

We note that this spline is piecewise polynomial of degree 6 ([6], Proposition I.28), and it is C^4 continuous ([6], Proposition I.37). Figure 5 shows a plot of the function, and compares it to other basis functions.

The remarkable property of the C^4 box spline which it shares with ZP spline, is that it is refinable with respect to quincunx refinement, which is equivalent to bisection refinement on regular 4–8 meshes. To define the notion of refinablilty, we need to introduce *dilation matrices* for refinement of regular grids.
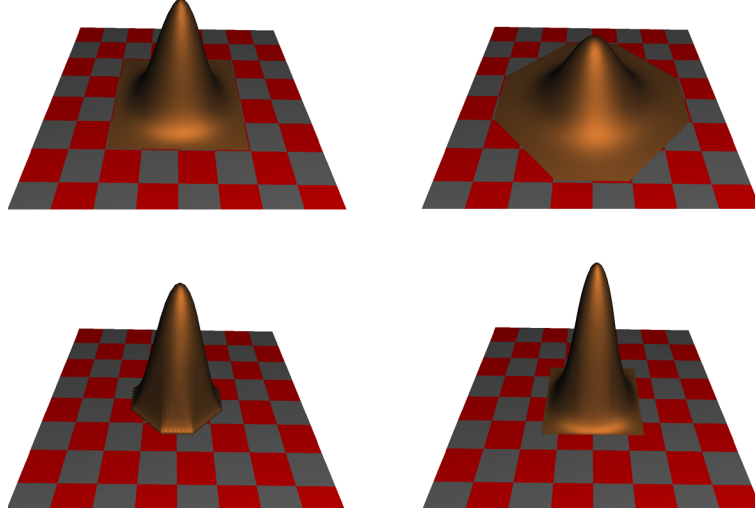
Fig. 5. Top row: bicubic B-spline, C^4 box spline function. Bottom row: Zwart-Powell basis function, biquadratic B-spline. The squares on the horizontal plane are unit size. All plots are stretched 8:1 in vertical direction.

Dilation matrices. Consider a refinement rule on a regular two-dimensional grid, which produces a finer regular grid including the original grid. The rule is completely characterized if we identify the positions of the vertices of the original grid in the refined grid. As the original grid is regular, its vertices form a periodic lattice in the refined grid. This lattice can be characterized by a pair of generating vectors v_1 and v_2: the lattice contains vertices with coordinates $v_1 i + v_2 j$, $i, j \in \mathbb{Z}$. If we use multiindex notation $q = (i, j)$, then the points of the lattice can be represented in the form Mq, where $M = [v_1 v_2]$ is called the *dilation matrix*. The dilation matrix for dyadic refinement, which is the basis of face split refinement, is a diagonal matrix $\mathrm{diag}(2, 2)$. Quincunx refinement has the dilation matrix $M = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

To summarize, *the columns of the dilation matrix are generating vectors of the lattice formed by the the vertices of the coarse mesh in the refined mesh.* For quincunx refinement, this is illustrated in Figure 6.

Scaling relation and subdivision masks. We show that for a particular choice of the refinement mask c_q, $q \in \mathbb{Z}^2$, the C^4 box spline $B(x)$ satisfies the scaling relation

$$B(x) = \sum_{q \in \mathbb{Z}^2} c_q B(Mx - q), \quad x \in \mathbb{R}^2 \tag{1}$$

where M is the dilation matrix of quincunx refinement, and c_q, $q \in \mathbb{Z}^2$, is a simple mask defined below.

It is well known that if a basis function $B(x)$ satisfies (1), then the control points p_q of a surface $\sum_q p_q B(x - q)$ can be refined using the refinement rule
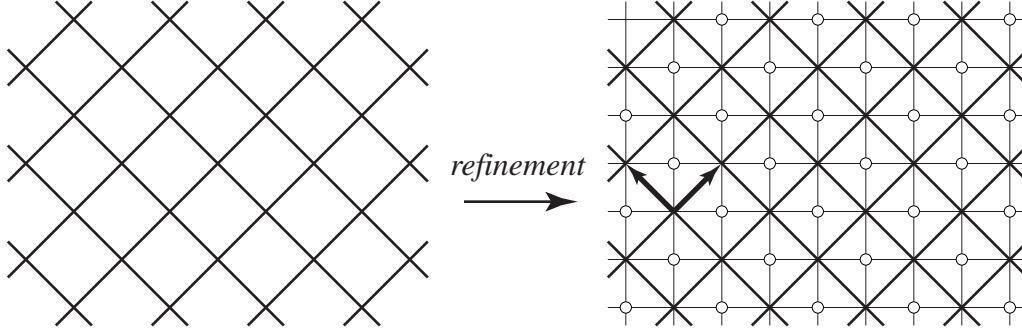
9

Fig. 6. Quincunx refinement. The coarse mesh is shown with thick lines, the refined mesh is shown with thin lines. Inserted vertices are indicated with circles. The column vectors of the dilation matrix generating the coarse mesh are shown.

$$p'_q = \sum_{r \in \mathbb{Z}^2} c_{q-Mr} p_r \tag{2}$$

from which we can deduce separate refinement masks for different vertex types.

To show that (1) is satisfied for the C^4 box four directional spline $B(x)$, and to find the subdivision mask c_q it is convenient to consider the scaling relation in the Fourier domain, where it takes the form

$$\hat{B}(\omega) = \frac{1}{|\det M|} \hat{c}\left((M^{-1})^T \omega\right) B\left((M^{-1})^T \omega\right), \quad \omega \in \mathbb{R}^2 \tag{3}$$

(see, e.g., [21]), and the mask can be computed using $|\det M|\hat{B}(M^T \omega)/\hat{B}(\omega)$.

Following [6], we find the Fourier transform of our box spline, shifted so that its support is centered at 0:

$$\hat{B}(\omega) = \frac{(1 - e^{-i\omega_1})^2 (1 - e^{-i\omega_2})^2 \left(1 - e^{-i(\omega_1+\omega_2)}\right)^2 \left(1 - e^{-i(\omega_1-\omega_2)}\right)^2 e^{3i\omega_1} e^{i\omega_2}}{\omega_1^2 \omega_2^2 (\omega_1 + \omega_2)^2 (\omega_1 - \omega_2)^2}$$

Applying the formula for the Fourier transform of the mask c yields after cancelations

$$\hat{c}(\omega) = \frac{1}{8} \left(1 + e^{-i\omega_1}\right)^2 \left(1 + e^{-i\omega_2}\right)^2 e^{i\omega_1} e^{i\omega_2}$$

This yields the following simple mask:

$$\frac{1}{8} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \tag{4}$$

with entries numbered from -1 to 1 in both directions.

Finally, we have to extract subdivision masks for two types of vertices: newly inserted face centers and vertices of the coarse mesh. Recall that by definition of the dilation matrix the vertices of the coarse mesh have coordinates $(i + j, i - j)$, $1, j \in \mathbb{Z}^2$ in the refined mesh; it is easy to see that the new vertices have coordinates $(i + j, i - j + 1)$. Rewriting (2) for two types we get two rules: $p'_{Mq} = \sum_r c_{M(q-r)} p_r$ (vertex rule), $p'_{Mq+e_1} = \sum_r c_{M(q-r)+e_1} p_r$ (face rule), where $e_1 = [1, 0]$. Substituting the mask (4), we get

$$p'_{i+j\,i-j} = \frac{1}{2} p_{i,j} + \frac{1}{8} (p_{i-1\,j-1} + p_{i-1\,j+1} + p_{i+1\,j-1} + p_{i+1\,j+1})$$

$$p'_{i+j+1\,i-j} = \frac{1}{4} (p_{i\,j} + p_{i+1\,j} + p_{i\,j+1} + p_{i+1\,j+1})$$

These are exactly the rules of our subdivision scheme in the regular case (Figure 4). It is interesting to note that this mask coincides, up to a scale factor, with the mask for midpoint subdivision for dyadic refinement.

We conclude that our scheme produces the C^4 four-directional box spline surfaces on regular 4–8 meshes.

4   Boundaries and Creases

The previous sections presented a 4–8 subdivision scheme that generalizes regular C^4 four-directional box splines for closed surfaces of arbitrary topological type. However, it is often necessary to model surfaces with boundary, which may contain sharp features as well. Thus, it is of practical importance to extend our 4–8 subdivision scheme to support surfaces with smooth boundaries and creases. Furthermore, it is often useful to have surfaces with piecewise smooth boundary. Special rules have to be applied in the neighborhood of *corner* vertices on the boundaries and creases, where the boundary curve is not smooth. Correctly handling of convex and concave corners requires some effort, and can be done for 4–8 subdivision using the techniques described in [1]. Here we present only the rules for smooth boundaries.

Our approach follows the ideas that were previously used to design boundary rules for Catmull-Clark, Loop and other subdivision schemes. We use the observation that each box spline spline has a corresponding univariate spline. In our case, it is easy to see that the corresponding univariate scheme is the degree five B-spline.

Indeed, assume that all control points p_{ij} for the C^4 box spline spline surface have form $(x_i, y_i, 0)$, that is, all points in row i coincide. The resulting limit surface is just a curve in the x, y plane with control points $p_i^{xy} = (x_i, y_i)$, with basis functions computed as $b(u) = \sum_j B(u, v - j)$, where $B(u, v)$ is the basis function of the C^4 box spline. Note that a priori b may depend on u, but in a moment we will see that this is not the case. Taking Fourier transforms, we obtain

$$\hat{b}(\omega_1, \omega_2) = \hat{B}(\omega_1, \omega_2) \sum_j e^{i\omega_2 j} = \delta(\omega_2) \hat{B}(\omega_1, \omega_2) = \delta(\omega_2) \hat{B}(\omega_1, 0)$$

11

As the inverse Fourier transform of $\delta(\omega_2)$ is constant, $\hat{b}$ does not depend on v. Computing $B(\omega, 0)$ using $\lim_{\omega \to 0}(1 - exp(-i\omega))/i\omega = 1$, we obtain Fourier transform of $b(u)$ as a univariate function:

$$\hat{b}(\omega) = \left(\frac{1 - e^{-i\omega}}{i\omega}\right)^6$$

which is precisely the Fourier transform of a degree 5 B-spline. It turns out that it is easy to incorporate refinement rules of a degree 5 spline into bisection refinement, as is shown below. Unfortunately, as it is the case for the Loop and Catmull-Clark schemes [1], we have to modify the rules for the points that we insert near extraordinary boundary vertices to ensure C^1 continuity at extraordinary points on the boundary.

Before we describe the rules, we have to generalize bisection refinement for meshes with boundary.

Bisection refinement on the boundary.   Bisection refinement of closed meshes relies on the fact that after basic blocks are bisected, new blocks can be constructed out of pairs of triangles adjacent to each exterior block edge. In an open tri-quad mesh, the boundary exterior block edges have only a single adjacent triangle. For open meshes we introduce special single-triangle *boundary basic blocks*. The boundary edge of these blocks plays the role of the interior block edge of standard basic blocks. Clearly, we still can apply bisection to the single triangle of the boundary basic block, inserting the split vertex on the boundary edge; on the next refinement level, the boundary edge becomes two exterior block edges (Figure 7).
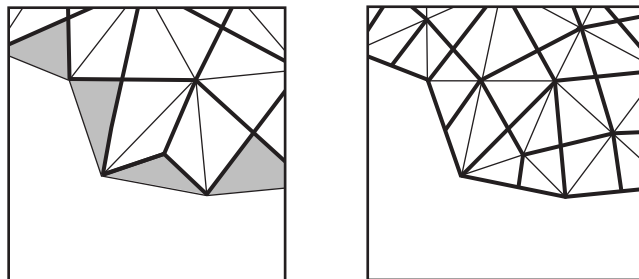

Fig. 7. On two sequential bisection refinement steps either all boundary edges are exterior block edges or none. Boundary basic blocks are shown in gray.

We observe that new vertices on the boundary are inserted only on every other step.

Boundary rules.   Our boundary subdivision rules are based on the Lane-Riesenfeld algorithm [20]. A degree n B-spline can be computed by recursively applying the following two-step algorithm: replicate the control points; apply midpoint averaging n times. To adapt the algorithm to our refinement procedure, we combine the first three averaging steps with point replication. This is equivalent

to applying a cubic B-spline subdivision rule on the step when new vertices are inserted. The masks are shown in Figure 8. The remaining two averagings are performed on the next step, when no new vertices are inserted on the boundary. The combined two averaging steps amount to convolution with the mask $1/4, 1/2, 1/4$.
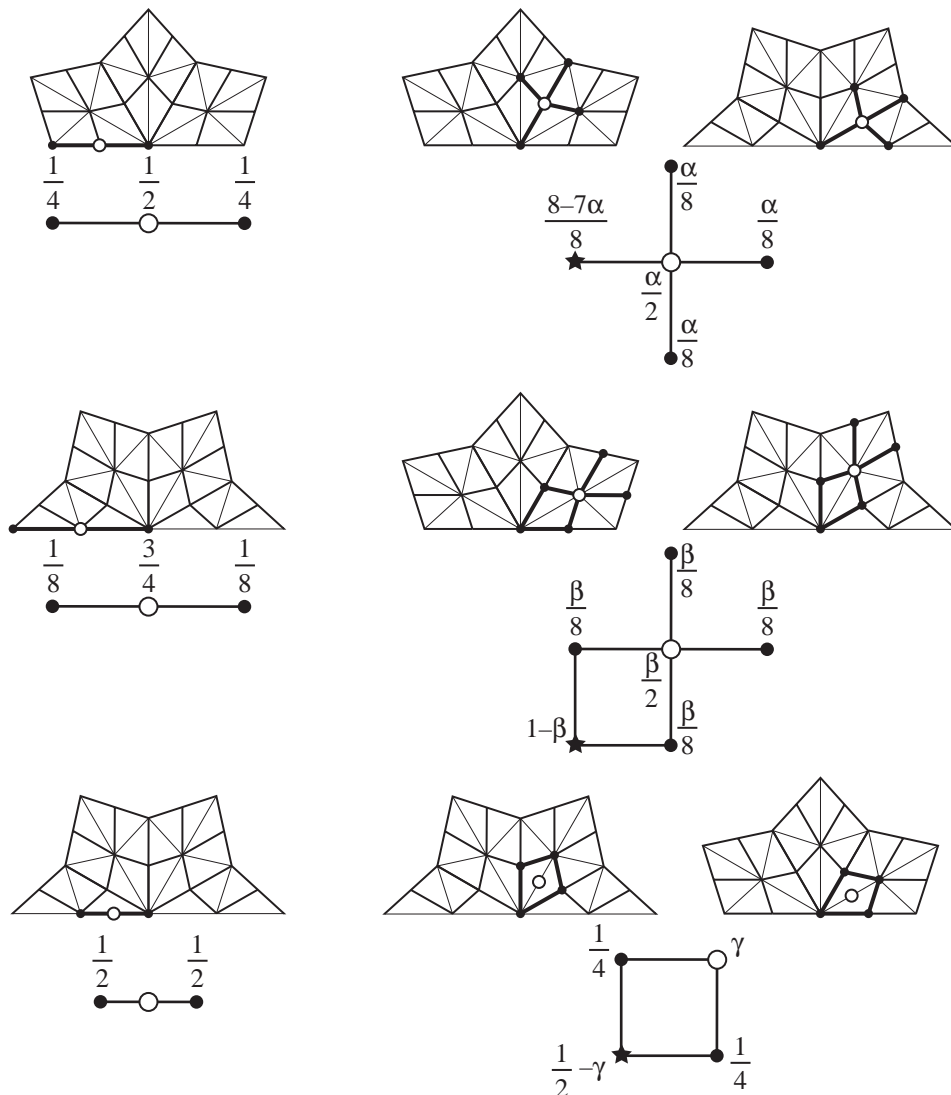

Fig. 8. Boundary subdivision rules. Modified subdivision rules; the constants are given by (5).

This observations lead to the following rules:

- If for a boundary vertex the incident boundary edges are exterior block edges, apply the smoothing mask. If these edges are edges of single-triangle boundary blocks, apply cubic B-spline vertex mask.
- To insert a new vertex on the boundary, use midpoint subdivision.

One difficulty with these rules is that they lead to surfaces which are not C^1-continuous near extraordinary vertices on the boundary, similar to the case of Loop and Catmull-Clark subdivision [1]. We use a similar approach to eliminate this problem, modifying the rules for the interior vertices adjacent to the extraordinary
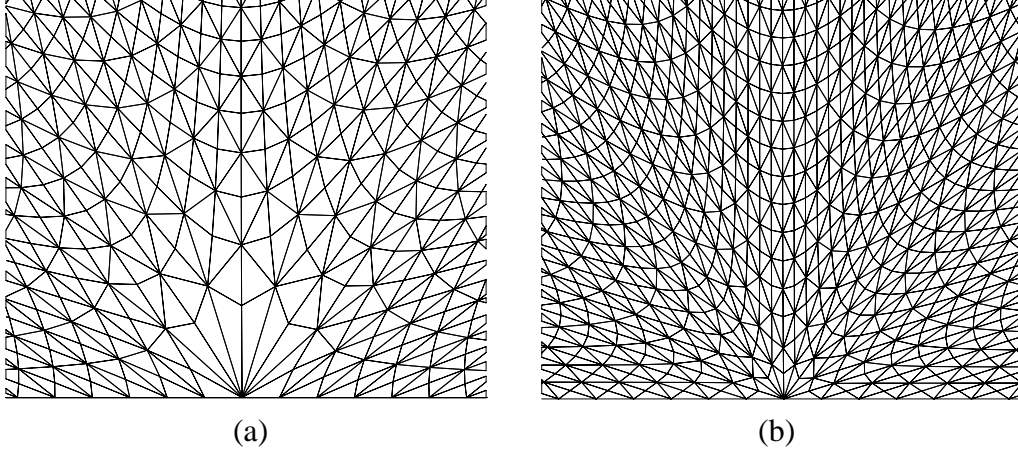
Fig. 9. Left: subdivided mesh with boundary after 10 iterations for unmodified rules. Right: Subdivided mesh with boundary after 10 iterations for modified rules. Note that the convergence is uniform along different directions.

vertex on the boundary. Here we present an informal explanation of the problem, and propose the rules that are likely to produce C^1-continuous surfaces; this claim is supported by empirical evidence, and we leave precise analysis along the lines described in [35].

Rules for extraordinary boundary vertices. First we examine the problem with using a combination of the degree 5 B-spline rules together with the standard rules in the interior. Our argument is based on the standard subdivision matrix considerations; basic concepts are discussed in [31, 34]. We emphasize that this is an informal argument, which is used as a motivation for the choice of coefficients; we leave formal analysis as future work.

Consider the minimal invariant neighborhood of an extraordinary point on the boundary, that is, a neighborhood $N^j(v)$ of the vertex v such that all control points in a similar neighborhood $N^{j+1}(v)$ on a finer level $j + 1$ can be computed using only the points in $N^j(v)$. The vector of control points p^{j+1} on $N^{j+1}(v)$ is given by the Sp^j, where p^j is the vector of control points on level j and S is the subdivision matrix of minimal size (smaller than the one that we need for the analysis of the characteristic map in Appendix A).

A typical appearance of a mesh obtained by using unmodified rules after a few subdivision steps can be seen in Figure 9(a).

This behavior is characteristic for schemes which are not C^1 on the boundary; the consequence of the fact that the control points in 1-neighborhood of the extraordinary vertex on the boundary converge to a common limit faster than the points in the interior, formally corresponds to the fact that two subdominant eigenvectors of the subdivision matrix defining the tangent plane have zero values on the boundary. This means that the tangent plane, if the surface were C^1-continuous, could be chosen independently from the position of the control points on the boundary, by manipulating the points in the interior. However, the boundary curve by construction is independent from the interior. Its tangent, which should be in the tangent
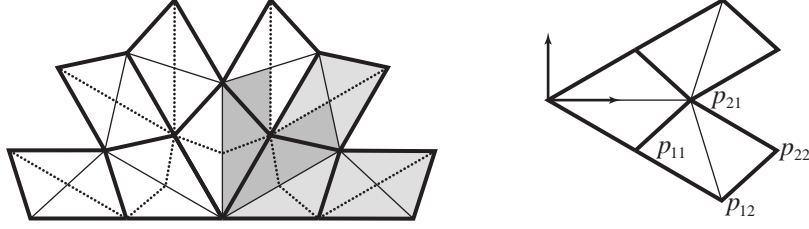
14

Fig. 10. The self-similar configuration used to derive the modified rules for vertices adjacent to the boundary. Right: the complete configuration; it can be obtained by rotating a single sector shown in light gray. A single sector of the image of the configuration after a subdivision step is shown in dark gray. Left: the notation for vertices and the coordinate system are shown.

plane of the surface depends only on the boundary control points. This means that the surface cannot be C^1-continuous.

The general way to improve the situation is to change the coefficients in such a way that the neighborhood of the extraordinary vertex shrinks uniformly; as on the boundary the 1-neighborhood is scaled by the factor approaching $1/2$ as we subdivide twice, the interior part should also shrink by this factor. Our idea is to modify the rules in such a way that for a boundary vertex of valence $2k + 1$ there is a self-similar configuration which scales by the factor of $1/2$ in all directions when subdivision is applied twice, or, equivalently, by a factor $1/\sqrt{2}$ when it is applied once A single subdivision step also introduces a rotation, which is eliminated after two steps.

Furthermore, we force this configuration to be one half of a configuration invariant with respect to rotations by π/k and reflections about the coordinate axes. We would like to limit our modifications to the rules for vertices that are connected to the extraordinary vertex by an edge, and keep the size of the support for the rules. These conditions can be used to find the modified coefficients.

Figure 10 shows the configuration that we are using. It should be noted that it is defined uniquely by our requirements; indeed, it follows from rotation invariance that the configuration consists of identical sectors, with each sector symmetric. As any such configuration is defined up to a scale factor, we fix $\|p_{11}\| = 1$. It follows from the condition on scaling by $1/2$ after two steps and the subdivision rules for the boundary, that $\|p_{12}\| = 2$. As we require scaling by factor of $1/\sqrt{2}$ after one subdivision step, we get $\|p_{12}\| = \sqrt{2}$. The leaves only the choice of the position of the point p_{22}. Its position is computed using the fact that away from the extraordinary vertex we do not modify the rules, so the position of the p'_{22} in the refined mesh is computed using the standard face rule. This yields $p_{22} = [\sqrt{2} + c, -s]$, with $c = \cos(\pi/k)$, $s = \sin(\pi/k)$ in the coordinate system shown in Figure 10.

Once the configuration is fixed, we introduce parameters into the subdivision rules, and find their values from the condition that the configuration above is scaled by a factor of $1/\sqrt{2}$ after subdivision. We parameterize the rules as shown in Figure 8 and find the following values for the coefficients:

15

$$\beta = \frac{4\sqrt{2}}{2c + 3\sqrt{2}}, \quad \alpha = \frac{4}{3 + \sqrt{2}c}, \quad \gamma = \frac{1}{2} - \frac{\sqrt{2}c}{4}. \tag{5}$$

One can show formally that this choice of rules guarantees that $1/2$ is the subdominant eigenvalue, and that the configuration of Figure 10 can be interpreted as a complex eigenvector for the eigenvalue $1/2$, which gives rise to two real subdominant eigenvectors (the real and imaginary part of the complex one). When extended to a larger neighborhood, these eigenvectors form the control net for the characteristic map [26, 35], which needs to be examined to verify C^1-continuity.

5 Implementing 4–8 Subdivision

In this section we consider in greater detail two important aspects of implementation of 4–8 subdivision: conversion of arbitrary triangular meshes to tri-quad meshes, and implementation of adaptive subdivision.

5.1 Triangular Mesh Preprocessing

As we have observed we could use one step of Catmull-Clark subdivision to produce a quadrilateral mesh, and then add diagonals to obtain a tri-quad mesh from an arbitrary mesh. However, one can achieve the desired result without increasing the number of faces by a factor of six. In addition our experience was that the surfaces obtained using our preprocessing have a more pleasing appearance. We propose a preprocessing procedure with two passes (Figure 11), which increases the number of faces by a factor only slightly more than two:

Tri-quad preprocess:
 (1) Find an independent set of basic blocks, remove interior edges of the blocks; the result is an intermediate mesh with triangular and quadrilateral faces.
 (2) Perform barycenter refinement on the intermediate mesh and mark the resulting basic blocks.

To implement pass 1, we select basic blocks based on edge length. This heuristic guarantees that we obtain triangles with good aspect ratio, and, for a planar mesh, convex quadrilateral blocks. The selection process ensures that we obtain an independent set of basic blocks. Algorithm 1 describes the implementation.

In general, it is not possible to cover the whole mesh with basic blocks. There will be a few isolated triangular faces remaining, and the intermediate mesh, obtained after removal of interior edges, will have two types of faces.

The second pass inserts barycenters into the intermediate mesh, splitting each quadrilateral into four triangles and each triangle into three triangles. (Figure 11).

It is easy to show that the result of preprocessing is a tri-quad mesh. In the processed mesh exactly one vertex of every triangle is a barycenter of a face of the

16

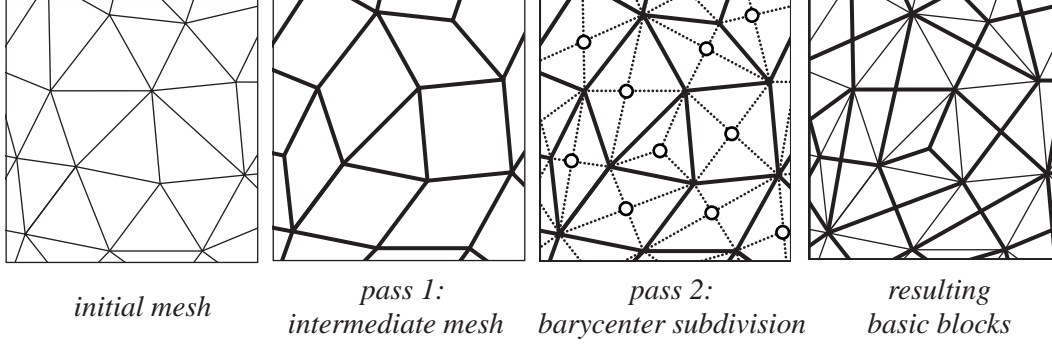| *initial mesh* | *pass 1:*
intermediate mesh | *pass 2:*
barycenter subdivision | *resulting*
basic blocks |

Fig. 11. Preprocessing of a triangular mesh. The result of the third step is a mesh partitioned into basic blocks and single-triangle boundary blocks.

Algorithm 1 : find_blocks

   store interior edges in priority queue Q
   while $Q \neq \emptyset$ **do**
     get e from Q
     mark basic block corresponding to e
     remove from Q edges sharing a face with e

---

intermediate mesh, and exactly one edge is an edge of the intermediate mesh. Thus, with each edge of the intermediate mesh we can associate a unique pair of triangles of the processed mesh sharing this edge, and any triangle of the processed mesh is associated with an edge of the intermediate mesh. These pairs of triangles form the basic blocks.

We note that in practice it is unnecessary to remove the interior edges on the first step: we can leave them in, and instead of reconnecting barycenters of the quadrilaterals with their corners, simply perform bisection refinement on the corresponding basic blocks.

As a result of preprocessing, valences of vertices increase at most from n to $2n$. This upper bound occurs mostly in the case of lower valence vertices. Because of geometric reasons, valences greater than 8 tend to change very little. The net effect is an equalization of vertex valence over the mesh. This is, in part, a consequence of the longest edge criteria used in pass (1) for selecting basic blocks [27].

In practice, for interior vertices with valence n, we have observed the following behavior:

- $n = 3$ — new valence 6;
- $4 \leq n \leq 8$ — new valence 8;
- $n > 8$, odd — new valence $n + 1$;
- $n > 8$, even — new valence n.

In the mesh obtained by our preprocessing algorithm, the type 1 vertices have valence 3 or 4, but after a single bisection refinement step, the processed mesh no longer has any type 1 vertices with valence 3.

For meshes with boundary our preprocessing produces a mesh with single-triangle boundary basic blocks, of the type shown in Figure 7 on the left.

17

Similar techniques for generating quadrilateral meshes from triangulations have been discussed in [2, 18].

5.2 *Adaptive 4–8 Subdivision*

As we have seen in Section 2.2, the 4–8 mesh structure is well suited for adaptive refinement. In order to exploit this capability in the context of subdivision, we need to devise an scheme for applying our rules to nonuniformly refined meshes. For such a mesh, it is possible that we need to compute a control point $p^{l+1}(v)$ for a vertex v at level $l + 1$, but the vertices in the stencil of the subdivision mask have control points evaluated only at levels less than l. This means that we have to evaluate the control points for these vertices up to level l before we can compute $p^{l+1}(v)$.

Our approach is to evaluate the required control points lazily. Note that a new vertex v can only be added to the mesh as a result of bisection of an internal edge of a basic block. The control point for v is computed by the face mask, which in turn uses the control points of the four neighbors v^S, v^N, v^E, v^W. Figure 12 shows the notation that we use for vertices of a basic block, and the face that is required to exist for the refinement to proceed.
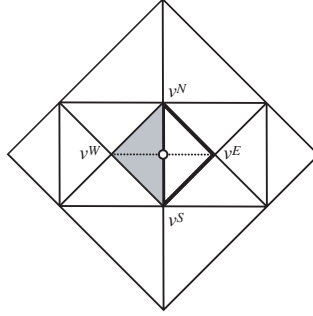

Fig. 12. To compute the control point for the newly inserted vertex denoted with a circle, the gray face would have to be created if it did not exist.

The initial control point $p^{l+1}(v)$, of a vertex inserted at level $l + 1$, is computed by the face mask, while subsequent values $p^j(v)$, $j = l + 2, \ldots$, are computed by the vertex mask.

To apply the face mask to a basic block f the following preconditions must hold: the two faces of the basic block should be at the same level l of refinement, and the control points of the four vertices at level l must be available.

Recall that a basic block has four vertices, two type 1 vertices (opposite to the diagonal) and two type 2 vertices (i.e. endpoints of the diagonal). Because of the properties of the 4–8 mesh structure, type 1 vertices are first generation vertices, inserted at level l. As these vertices exist, they are guaranteed to have control points evaluated at level j. So, we just need to make sure that the control points of type 2 have been correctly updated using the vertex mask.

The implementation of the adaptive face rule is shown in Algorithm 2. Note that the level of the adjacent face $f.mate.level$ in the algorithm can be either l or $l - 1$

(only faces which are on the same level or differ by a single level can share an edge).

Algorithm 2 :   adapt_face (f)

---
   if f has been refined **then**
     return
   $l \leftarrow f.level$
   if $f.mate.level < l$ **then**
     adapt_face ($f.mate$)
   adapt_vertex (v^S, l)
   adapt_vertex (v^N, l)
   refine_quadblock (f, $f.mate$)
   $v \leftarrow$ split vertex of internal edge
   $p^{l+1}(v) = \dfrac{1}{4} \sum p^l(v^S + v^N + v^E + v^W)$

To apply the vertex mask at level l all the faces sharing this vertex should be refined to level $l - 1$ or higher. The implementation of the adaptive vertex rule is shown in Algorithm 3. We define $N_1^{l-1}(v)$ to be the collection of vertices appearing on level $l - 1$, which share an exterior block edge with v on that level. These are exactly the vertices we need to apply the subdivision rule.

Algorithm 3 :   adapt_vertex (f, l)

---
   if $p^l(v)$ has not been computed **then**
     for f containing v **do**
       while $f.level \neq l$ **do**
         adapt_face (f)
   $p^l(v) = \dfrac{1}{2} p^{l-1}(v) + \dfrac{1}{2n} \displaystyle\sum_{v_j \in N_1^{l-1}(v)} p^{l-1}(v_j)$

---

6   Examples and Comparisons

In this section we present some examples of applying 4–8 subdivision for surface modeling. Surfaces produced by our scheme are constructed and compared with surfaces generated by other subdivision schemes.

The examples in Figures 13 and Figure 14 show a closed surface and a surface with boundary generated by 4–8 subdivision, together with their control meshes. The appearance of the surfaces is quite similar to the appearance of the surfaces obtained using such schemes as Loop and Catmull-Clark.

Figure 15 shows multiple surfaces with boundary joined with C^0 continuity. The model shown in Figure 15(a) consists of separate meshes, for the eyes, eyebrows, and face. Figure 15(b) shows a detail view of a region near the eye, and Figure 15(c) shows the same region with several parts removed.

19

The examples in Figures 16 demonstrate adaptive 4–8 subdivision. Figure 16(a) shows the inital mesh and Figure 16(b) shows the smoothed polygonal mesh after applying 4 subdivision steps uniformly to the mesh.

In Figure 16(c) the adaptation uses a simple geometric criterion: if the distance between a position of a new control point computed by subdivision and the midpoint of the corresponding edge is less than a threshold, the edge is not bisected.

In Figure 16(d) the adaptation is based on a spatial threshold function (the characteristic function of the half-space $x > 0$). The mesh has finest resolution on one side of the plane $x = 0$, and coarsest resolution on the other side of the plane. Note the fast transition between two regions.

The last examples (Figure 17 and Figure 18) show a comparison between 4–8 subdivision and other schemes. We use four meshes with different complexity: a cube, an extruded pentagon, a rook model and a mask model. The figures show the control polyhedron of each mesh and the surfaces generated by Loop [24], Catmull-Clark [3], Midedge [25], Doo-Sabin [7], and 4–8 subdivision. One can observe that 4–8 subdivision results in more smoothing and more shrinking than all other schemes on the mask model and the rook model. Overall, the surface appearance is close to Catmull-Clark. The only case where the difference in appearance is substantial is the cube model. The reason for this is unclear at this time; it is likely to be related to the differences in the handling of vertices of valence 3 for Catmull-Clark scheme and exterior block edge valence 3 for 4–8 subdivision.

7 Conclusions

We have presented 4–8 subdivision, a new scheme using bisection refinement that extends the four-directional box splines of class C^4 to surfaces of arbitrary topological type. We have proved that closed limit surfaces are C^4 continuous almost everywhere, except at extraordinary vertices where they are C^1-continuous (Appendix A).

Bisection refinement generates a hierarchical mesh structure that supports adaptive refinement, while keeping the meshes conforming. In [30], the hierarchical structure for tri-quad [3] meshes is discussed in greater detail, and its applications are reviewed.

Applications of bisection refinement are not limited to the presented scheme. One can implement other schemes extending two-directional and four-directional box splines. For example, [29] describes the implementation of Doo-Sabin and Catmull-Clark schemes. We have also implemented the Midedge scheme based on ZP element using tri-quad meshes.

While the 4–8 subdivision scheme has a number of nice properties it also has several drawbacks compared to schemes using face splits and lower degree splines. For example, the support of the masks for tangents and limit positions is rather large. While direct evaluation along the lines proposed in [28] is possible, more

---

[3] In [30] and [29] tri-quad meshes are referred to as 4–k.

subdivision steps are required before it can be performed, and the evaluation itself is computationally more expensive. Finally, the translates of the C^4 four-directional box spline are not linearly independent ([5], Proposition II.57). This might lead to problems with fitting a 4–8 subdivision surface to data. While in some cases high degree of smoothness is an advantage, quite predictably it also results in greater shrinkage of a surface and greater attenuation of high-frequency detail.

(a) Control polyhedron   (b) Suface
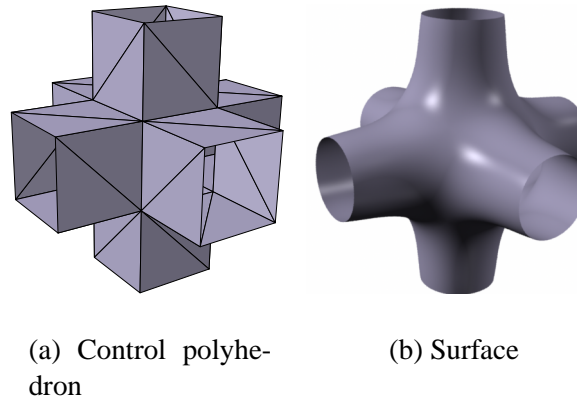
Fig. 13. A surface generated using 4–8 subdivision.

(a) Control polyhe-   (b) Surface
dron

Fig. 14. Surface with boundary.

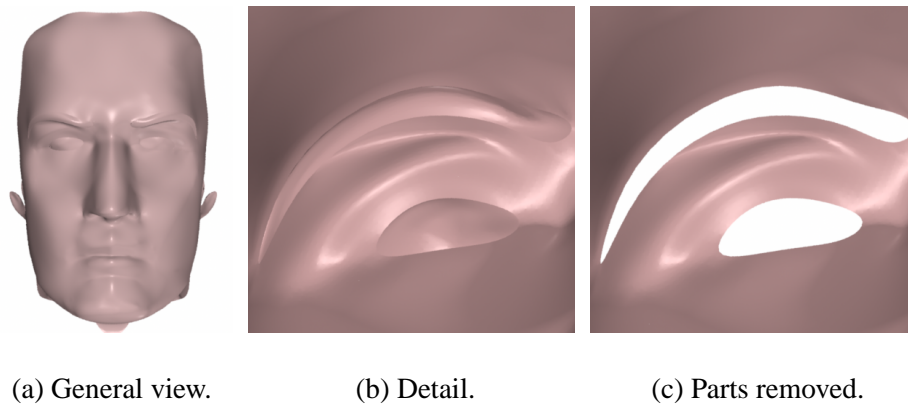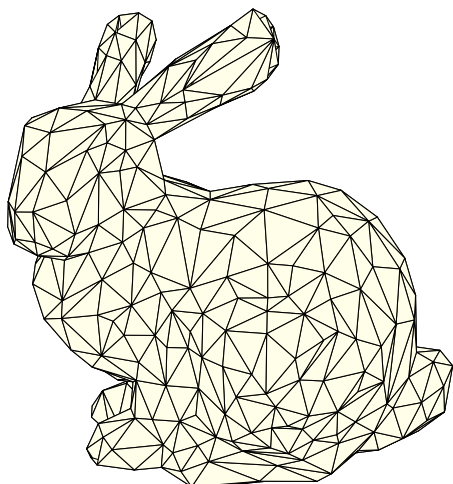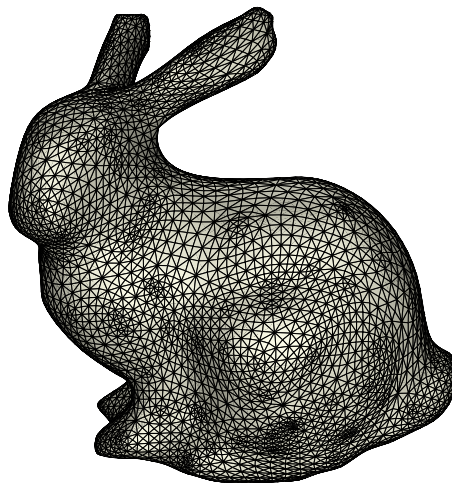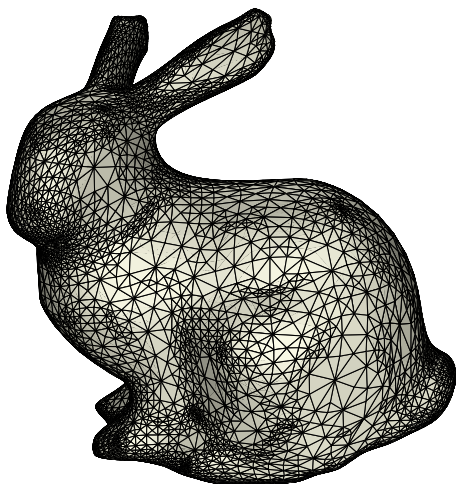(a) General view.   (b) Detail.   (c) Parts removed.

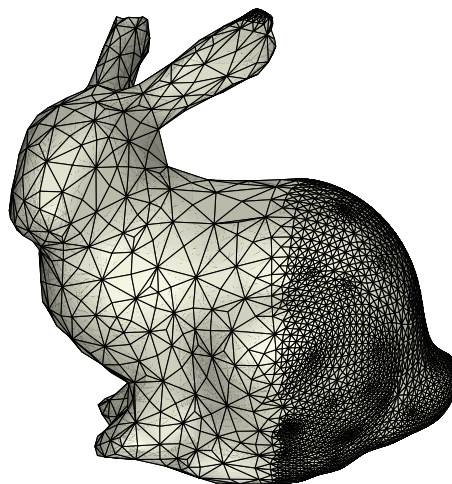Fig. 15. Multiple surfaces joining with C^0 continuity.

(a) Control mesh.

(b) Uniform subdivision.

(c) Adaptation to surface geometry.

(d) Adaption to spatial threshold.

Fig. 16. Adaptive refinement. The model is courtesy of Stanford Computer Graphics Lab.

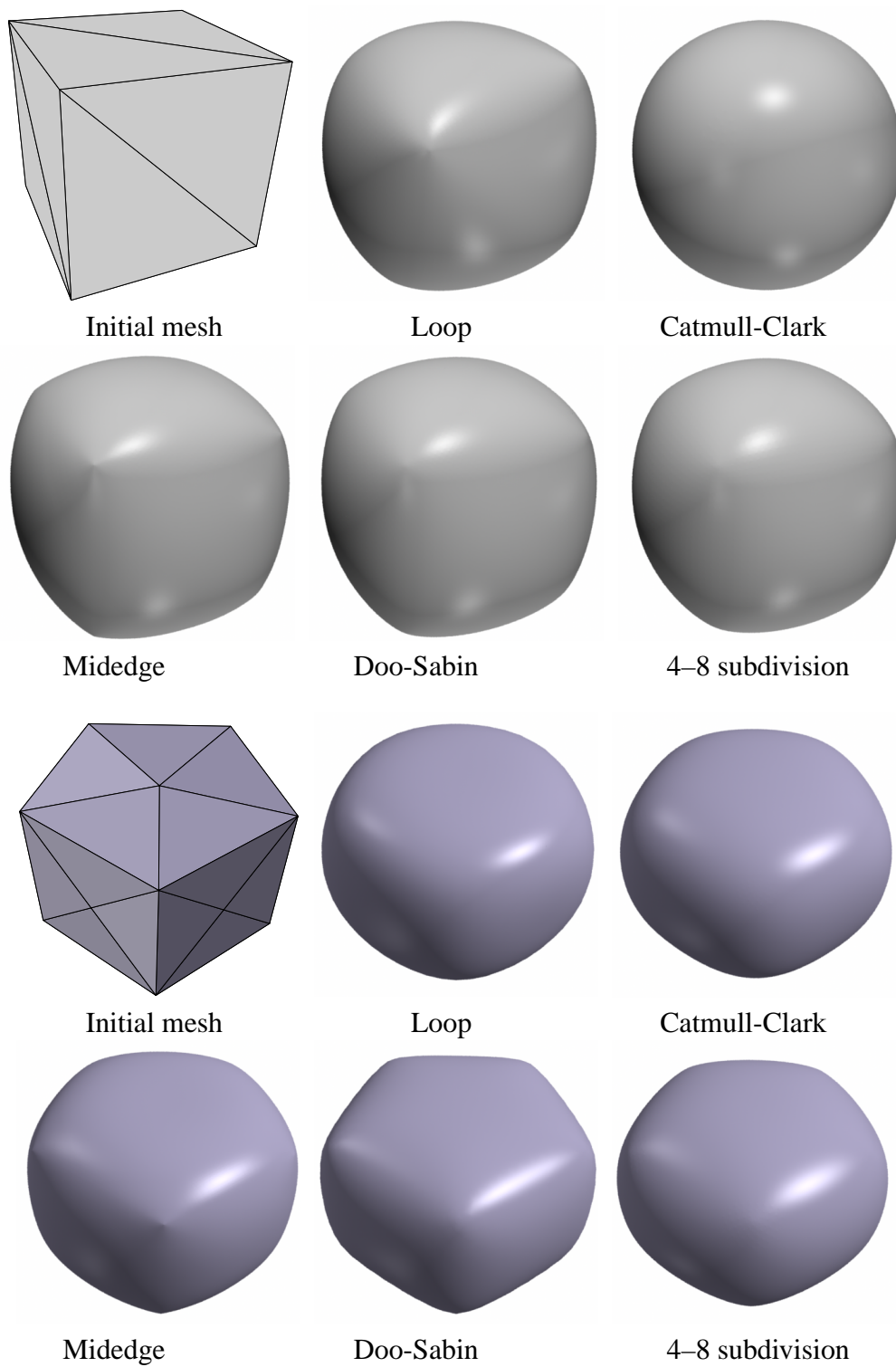|		
Initial mesh	Loop	Catmull-Clark
Midedge	Doo-Sabin	4–8 subdivision
Initial mesh	Loop	Catmull-Clark
Midedge	Doo-Sabin	4–8 subdivision

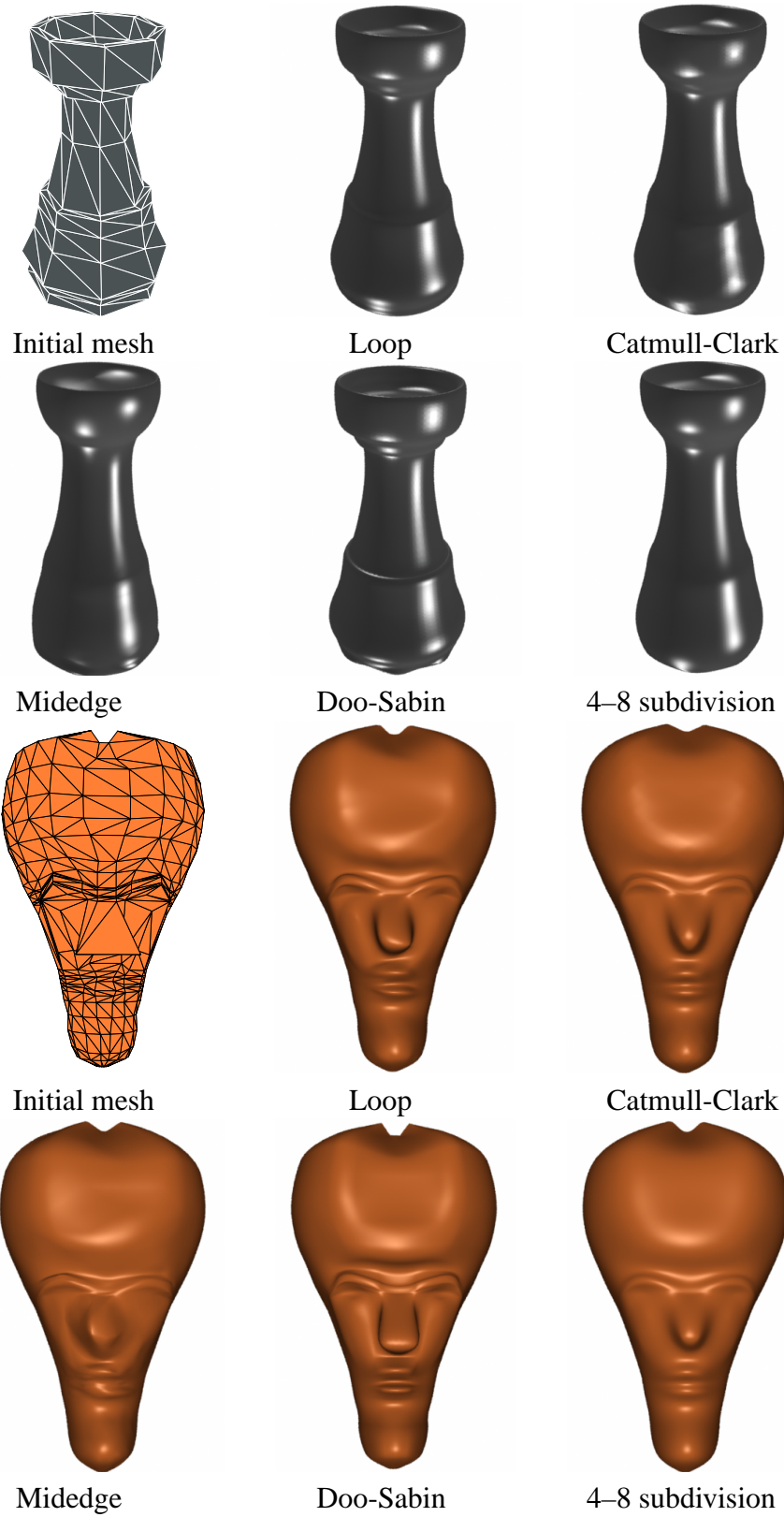Fig. 17. Comparison between surfaces generated by different subdivision schemes.

Fig. 18. Comparison between surfaces generated by different subdivision schemes. The head model is courtesy of of Khrysaundt Koenig, Caltech.

A Smoothness Analysis

Our analysis of the 4–8 subdivision scheme is based on the techniques presented in [33]. For the purposes of the C^1-continuity analysis we can regard the scheme as being defined on a quadrilateral mesh. Indeed, after sufficient number of refinement steps, locally near an extraordinary vertex (that is, a vertex of valence $2k$, $k \neq 4$), the tri-quad mesh is a quadrilateral mesh with diagonals added (Figure 2). Removing the diagonals results in a k-regular quadrilateral mesh. Thus, the extraordinary vertices of valence k in the original mesh become vertices of valence k in the quadrilateral mesh. In this section, valence of a vertex is the valence in the quadrilateral mesh, not in the original tri-quad mesh. While the facts that are proved in [33] assume subdivision schemes defined on triangular meshes, the extension to the case of quadrilateral schemes is straightforward, and requires only minor changes in notation.

To establish C^1-continuity of a subdivision scheme for valence k, it sufficient to verify that the *characteristic map* ([26]) for this valence is regular and injective.

A.1 Eigenstructure of the Subdivision Matrix

After applying the standard DFT approach, the subdivision matrix for the scheme (see [33] for definitions) is converted into the block-diagonal form with $k - 1$ of the following 12 by 12 blocks B_m, $m = 1 \ldots k - 1$ on the diagonal:

$$
B\left(\frac{2\pi m}{k}\right) = \frac{1}{64}
\begin{pmatrix}
16+8c & 6\bar\omega+6 & 2 & 0 & 0 & 0 & 0\,0\,0\,0\,0\,0 \\
14+\bar\omega+14\omega+\omega^2 & c+14 & 1+\omega & 1 & 0 & 1 & 0\,0\,0\,0\,0\,0 \\
24+2\bar\omega+2\omega & 8\bar\omega+8 & 8 & 2 & 0 & 2\bar\omega & 0\,0\,0\,0\,0\,0 \\
18+6\omega & 2\bar\omega+18 & 6 & 6 & 0 & 2 & 0\,0\,0\,0\,0\,0 \\
8+8\omega & 24 & 2+2\omega & 8 & 2 & 8 & 0\,0\,0\,0\,0\,0 \\
6+18\omega & 18+2\omega & 6\omega & 2 & 0 & 6 & 0\,0\,0\,0\,0\,0 \\
18 & 6\bar\omega+6 & 18 & 6 & 0 & 6\bar\omega & 2\,0\,0\,0\,0\,0 \\
\omega+14 & \bar\omega+14 & 14 & 14 & 1 & \bar\omega+1 & 1\,1\,0\,0\,0\,0 \\
6+2\omega & 18 & 6 & 18 & 6 & 6 & 0\,2\,0\,0\,0\,0 \\
1+\omega & 14 & 1+\omega & 14 & 14 & 14 & 0\,1\,1\,0\,1\,1 \\
2+6\omega & 18 & 6\omega & 6 & 6 & 18 & 0\,0\,0\,0\,0\,2 \\
1+14\omega & \omega+14 & 14\omega & 1+\omega & 1 & 14 & \omega\,0\,0\,0\,0\,1 \\
\end{pmatrix}
$$

where $\omega = \exp(2\pi i m/k)$, $c = \cos(2\pi m/k)$. In addition, there is a single 13×13 block $B(0)$:

$$B(0) = \begin{pmatrix}
24 & 32 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
24 & 32 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
18 & 32 & 12 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
14 & 30 & 16 & 2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & 28 & 16 & 8 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 24 & 20 & 6 & 6 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 16 & 24 & 4 & 8 & 2 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 24 & 20 & 6 & 2 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 18 & 12 & 18 & 6 & 0 & 6 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 15 & 15 & 14 & 14 & 1 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 8 & 18 & 6 & 18 & 6 & 6 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 2 & 14 & 2 & 14 & 14 & 14 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 8 & 18 & 6 & 6 & 6 & 18 & 0 & 0 & 0 & 0 & 0 & 2 \\
1 & 15 & 15 & 14 & 2 & 1 & 14 & 1 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

The eigenvalues of the block $B(0)$ can be found explicitly, and are all in the range $0 \dots 1/4$, excluding the eigenvalue 1.

The characteristic polynomial of degree 12 of each block $B\left(\frac{2\pi m}{k}\right)$ always factorizes into linear factors and a single polynomial of degree 5:

$$\lambda^5 + \left(-\frac{25}{32} - \frac{5}{32}c\right)\lambda^4 + \left(\frac{17}{512}c + \frac{21}{128} - \frac{1}{512}c^2\right)\lambda^3 +$$
$$\left(-\frac{25}{2048} - \frac{13}{8192}c + \frac{7}{8192}c^2\right)\lambda^2 + \left(-\frac{1}{32768}c^2 + \frac{1}{4096} - \frac{1}{32768}c\right)\lambda$$
$$+ \frac{1}{2097152}c^3 + \frac{5}{2097152}c^2 + \frac{1}{524288}c$$

In general, the roots cannot be computed explicitly. For fixed m and k, we can easily find the roots numerically, with guaranteed lower and upper bounds on the roots. Numerically computed roots of this polynomial are plotted as functions of c in Figure A.1.

Analysis of the eigenvalues. The analysis is similar to the analysis of the eigenvalues of the subdivision matrix for Kobbelt's quadrilateral scheme presented in [33]. From the plot it is clear that the largest eigenvalue increases as a function of c; therefore, it appears that the largest eigenvalue of the subdivision matrix for any valence corresponds to $m = 1$. Moreover, our calculations show that the largest
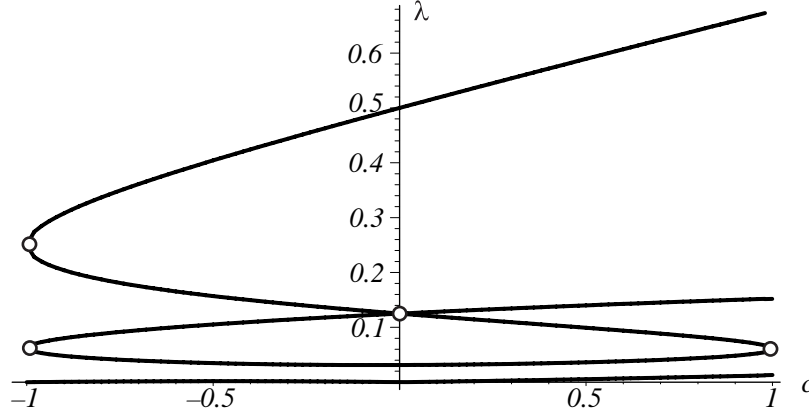
Fig. A.1. The magnitudes of the eigenvalues of the subdivision matrix as functions of $c = \cos 2\pi/k$. Only 5 eigenvalues which depend on c are shown. Double eigenvalues are indicated with circles.

eigenvalue is always real. Using interval methods, we prove the following proposition:

Proposition 1 *For any valence k, and any $m = 1 \ldots k-1$ the largest eigenvalue is real and unique, and for any block $B(2m\pi/k)$, $m \neq k-1, 1$ the largest eigenvalue is less than the largest eigenvalue of the blocks $B(2\pi/k)$ and $B(2\pi(k-1)/k)$. The unique largest eigenvalue is the only eigenvalue in the interval $[0.5, 1]$, for $k > 4$.*

The detailed proof with all calculations can be found in a Maple worksheet available from the authors. Here we present an outline of the proof. The proof is performed in several steps:

(1) We show that for $c < 0$, all roots of the characteristic polynomial $P(c, \lambda)$ are less than 0.51.
(2) We show that for any $c \in [0 \ldots 1]$, there is a unique real root μ in the interval $[0.47+0.2c, 0.51+0.2c]$, and the function $\mu(c)$ is C^1-continuous and increases.
(3) We "deflate" the characteristic polynomial (that is, divide by the monomial $\lambda - \mu$), and verify that all roots of the deflated polynomial are inside the circle of radius 0.5 for $c \in [0, 1]$.

We use the Marden-Jury test [16] to show that the roots of a polynomial are inside a circle of radius 0.51 and 0.5 in the complex plane on steps 1 and 3 respectively. This test requires only a simple algebraic calculation on the coefficients of the polynomial, and can be performed easily for symbolic and interval coefficients. See [33] for details of application of this test.

Proposition 1 allows us to compute the subdominant eigenvalue with arbitrary precision for any k, and establishes that it always corresponds to the blocks with $m = 1$ and $m = k - 1$.

Eigenvectors. We compute the complex eigenvectors of the matrix as a function of c and λ, solving the linear system $B\left(\frac{2\pi m}{k}\right) - \lambda I = 0$. Two real eigenvectors are obtained as the real and imaginary part of the complex eigenvector. To obtain the subdominant eigenvectors, which define the control mesh for the characteristic

28

map, we evaluate the eigenvectors obtained as above for $m = 1$ and λ equal to the subdominant eigenvalue.

The resulting control mesh for the characteristic maps of the scheme for several valences is shown in Figure A.2.
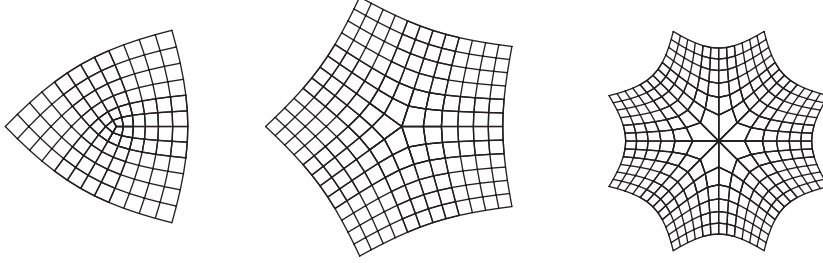

Fig. A.2. Control meshes for the characteristic map of the 4–8 subdivision scheme for extraordinary valences 3, 5 and 8; the diagonal edges in the two-triangle clusters for the 4-k mesh are not shown.

A.2 Analysis of the Characteristic Map

Instead of proving that the characteristic map is injective, it is sufficient to verify that it maps only zero to zero, and has index 1. More precisely, the following theorem was proved in [33]:

Theorem 2 *Suppose a characteristic map Φ satisfies the following conditions:*
 (1) the preimage $\Phi^{-1}(0)$ contains only one element, 0;
 (2) the characteristic map has a Jacobian of constant sign everywhere on $\mathbf{R}^2$ except zero.
Then the extension of the characteristic map is a surjection and a covering away from 0. In particular, if the winding number with respect to the origin of the image $\Phi(\gamma)$ of a simple curve is 1, the characteristic map is injective and the scheme is C^1-continuous.

For the scheme that we present, the characteristic map can be computed locally in closed form. Instead we use general tools that evaluate a sufficiently close approximation of the Jacobian of the characteristic map to verify regularity. Similarly, rather than computing the image of a curve enclosing zero under the characteristic map in explicit form, we use a sufficiently close piecewise linear approximation.

The characteristic map Φ satisfies the scaling relation $\Phi(y/2) = \mu\Phi(y)$, where μ is the subdominant eigenvalue; to establish its regularity, it is sufficient to examine its behavior on a ring around an extraordinary vertex, such that under the iterative application of the scaling transformation $t \to t/2$, the copies of the ring cover the k-gon U_k. We choose the ring in such a way that the rules that are applied to refine the control mesh of the ring are the standard box spline rules. For our scheme, the control mesh of a minimal ring with this property consists of 6 layers of vertices. Further, due to the rotational symmetry of the characteristic map, only one segment needs to be examined.

29

Let L^∞ be the limit function of subdivision, L^m be its approximation after m steps, and let p^m be the vector of control points of a subdivision surface. Then the following estimate holds [4, 33]:

$$\left\| L^\infty - L^m \right\|_\infty \le \frac{c}{1-\gamma} D\left(p^m\right) \tag{A.1}$$

where D is the contraction function, which we choose to be $\left\| \nabla p \right\|_\infty$, with ∇ being the vector of finite directional differences. The constants γ and c can be computed from the coefficients of subdivision.

For a subdivision scheme S, one can always find a *matrix* difference subdivision scheme S' (that is, a scheme whose coefficients are matrices) acting on the vectors of differences at each vertex, such that $\nabla S p = S' \nabla p$. If the scheme is C^1 on regular grids, then an estimate similar to (A.1) holds for the difference scheme, with different constants γ^D and c^D. These estimates can be used to compute approximations to the characteristic map and its derivatives with guaranteed error bounds; if the lower estimate for the Jacobian of the characteristic map obtained in this way is positive (or upper estimate is negative), the map is guaranteed to be regular. For our scheme, the convergence constants for the scheme are $c = 9/8$, $\gamma = 1/2$. The convergence constants for the difference scheme are $c^D = 21/16$, $\gamma^D = 1/2$.

Figure A.3 shows the dependence of the upper and lower estimates of computed Jacobians on the valence for valences up to 444. As the control meshes depend on the eigenvalues, which are known only approximately, all calculations are performed using interval arithmetic. In this way, if the guaranteed intervals for the eigenvalue μ are known, we also know the guaranteed intervals for the upper and lower bounds of the Jacobian.

Behavior at infinity.   Our proof of C^1-continuity for high valences (in the case of 4–8 subdivision scheme, greater than 444) is based on the following observation. One of the subdominant eigenvectors depends only on $c = \cos(2\pi/k)$; the other has the form $\sin(2\pi/k)w(c)$, where w depends only on c. Clearly, the sign of the Jacobian of the resulting map does not depend on a scale factor; therefore, we can rescale the control mesh of the characteristic map by $\sin(2\pi/k)$ in the vertical direction. The rescaled control mesh for a single segment of the ring approaches a nondegenerate limit configuration (Figure A.4) as $k \to \infty$. We are using interval arithmetic to examine approximations to characteristic maps. Suppose the control mesh was computed using the interval $[1 - \epsilon, 1]$ for c. If we verify that for this control mesh with interval control points the Jacobian has constant sign, we have verified this fact simultaneously for all control meshes for which $c \in [1 - \epsilon, 1]$.

To complete the analysis of the scheme we need to describe the behavior of $\mu(c)$ at infinity. Specifically, to use our algorithm for verification of smoothness for all valences, for an interval $c = [1 - \epsilon, 1]$ we need to estimate the corresponding interval value $\mu(c)$, so that the eigenvectors can be computed. As $\mu(c)$ changes slowly, linear approximation is sufficient for our purposes. The upper bound for
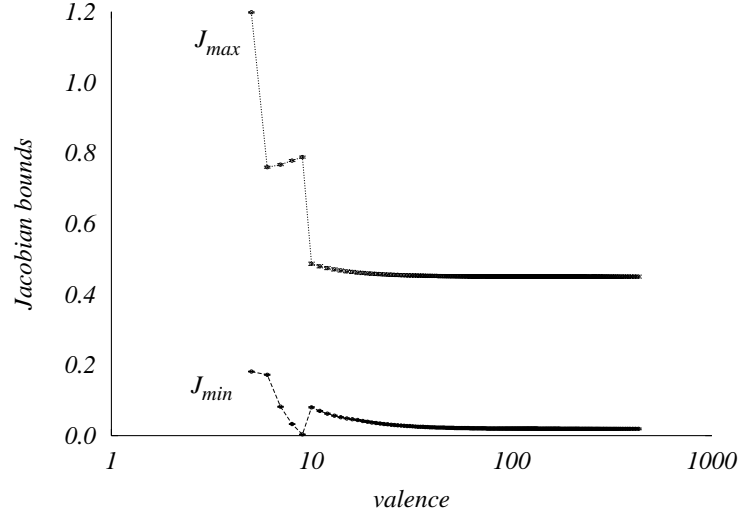
Fig. A.3. The upper and lower bounds for the Jacobian of the characteristic maps as functions of the valence for the 4–8 subdivision scheme. The error bars indicate the size of the interval (these intervals are quite small and are not clearly visible). The interval size for c was chosen to be 1×10^{-4}; the maximal examined valence was 444; as the difference between control meshes for large valences was smaller than the size of the intervals for the control points, a total of 229 valences had to be examined.

the derivative μ_c' at $c = 1$ can be computed easily if we regard the characteristic polynomial as a function $F(\mu, c)$ of two variables μ and c, and estimate $\mu'(c)$ using the ratio of the components of the gradient of this function. The upper bound for $\mu'(c)$ in the region of interest is approximately 0.45.
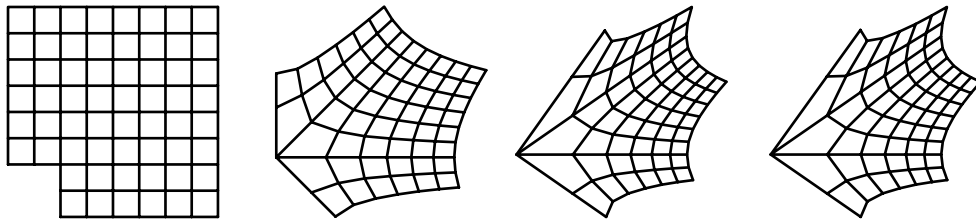

Fig. A.4. As $k \to \infty$, $c = \cos(2\pi/k) \to 1$, and rescaled control meshes for a segment of the characteristic map converge to a limit; rescaled segments for valences 4, 8, 32 and the limit configuration are shown.

We conclude that *the 4–8 subdivision scheme is C^1-continuous for all valences*.

31

References

[1] H. Biermann, A. Levin, and D. Zorin. Piecewise smooth subdivision surfaces with normal control. In *Proceedings of SIGGRAPH 2000*, 2000.

[2] H. Borouchaki, P. Frey, and P. George. Unstructured triangular-quadrilateral mesh generation, 1996.

[3] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput. Aided Design*, 10:350–365, 1978.

[4] A. S. Cavaretta, W. Dahmen, and C. A. Micchelli. Stationary subdivision. *Memoirs of the AMS*, 12(453), 1991.

[5] C. de Boor, D. Hollig, and S. Riemenschneider. *Box Splines*. Springer-Verlag, New York, NY, 1994.

[6] C. de Boor, K. Höllig, and S. Riemenschneider. *Box splines*. Springer-Verlag, New York, 1993.

[7] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Comput. Aided Design*, 10:356–360, 1978.

[8] M. Duchaineau, M. Wolinsky, D. Sigeti, M. Miller, C. Aldrich, and M. Mineev-Weinstein. Roaming terrain: Real-time optimally adapting meshes. *IEEE Visualization '97*, pages 81–88, November 1997.

[9] W. Evans, D. Kirkpatrick, and G. Townsend. Right triangular irregular networks. Technical Report 97-0, University of Arizona, 1997.

[10] J. C. Feauveau. *Wavelets and their Applications*, chapter Wavelets for the Quincunx Pyramid. Jones and Bartlet, 1992.

[11] Larry Gritz and James K. Hahn. BMRT: A global illumination implementation of the renderman standard. *Journal of Graphics Tools*, 1(3):29–47, 1996.

[12] B. Grünbaum and G. Shephard. *Tilings and Patterns*. W. H. Freeman, 1987.

[13] Ayman Habib and Joe Warren. Edge and vertex insertion for a class of C^1 subdivision surfaces. *CAGD*, 16(4):223–247, 1999. Previously available as a TR, Rice University, August 1997.

[14] D. J. Hebert. Cyclic interlaced quadtree algorithms for quincunx multiresolution. *Journal of Algorithms*, 27:97–128, 1998.

[15] Brian Von Herzen and Alan H. Barr. Accurate triangulations of deformed, intersecting surfaces. *Computer Graphics (Proceedings of SIGGRAPH 87)*, 21(4):103–110, July 1987. Held in Anaheim, California.

[16] E. I. Jury. *Theory and Applications of the z-Transform Method*. Wiley, New York, 1964.

[17] L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Computer Graphics Forum*, 15(3):409–420, 1996.

[18] L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *Computer Graphics Forum (Proc. EUROGRAPHICS '96), 15(3)*, pages 409–420, 1996. Eurographics '96 issue.

[19] L. Kobbelt. $\sqrt{3}$ subdivision. In *SIGGRAPH 2000 Conference Proceedings*, 2000.

[20] J. Lane and R. Riesenfeld. A theoretical development for the computer generation and display of piecewise polynomial surfaces. *IEEE Trans. Pattern Analysis Machine Intell.*, 2(1):35–46, 1980.

[21] W. Lawton, S. L. Lee, and Zuowei Shen. Stability and orthonormality of multivariate refinable functions. *SIAM J. Math. Anal.*, 28(4):999–1014, 1997.

[22] S. Levy, T. Munzner, and M. Phillips. Geomview. Geometry Center, University of Minnesota, 1991.

[23] P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hughes, N. Faust, and G. Turner. Real-Time, continuous level of detail rendering of height fields. In *SIGGRAPH 96 Conference Proceedings*, pages 109–118, 1996.

[24] C. Loop. Smooth subdivision for surfaces based on triangles. Master's thesis, University of Utah, 1987.

[25] J. Peters and U. Reif. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics*, 16(4):420–431, October 1997.

[26] U. Reif. A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design*, 12(2):153–174, 1995.

[27] M. C. Rivara. Mesh refinement processes based on the generalized bisection of simplices. *SIAM J. Numer. Anal.*, 21(3):604–613, 1984.

[28] J. Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *SIGGRAPH 98 Conference Proceedings*, pages 395–404, 1998.

[29] Luiz Velho. Using semi-regular 4-8 meshes for subdivision surfaces. *Journal of Graphics Tools*, 2001.

[30] Luiz Velho and Jonas Gomes. Hierarchical 4–k meshes: *Concepts and Applications*. *Computer Graphics Forum*, 2000.

[31] D. Zorin and P. Schröder, editors. *Subdivision for modeling and animation*. Course notes of Siggraph 2000. ACM SIGGRAPH, July 2000.

[32] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *SIGGRAPH 97 Conference Proceedings*, pages 259–268, 1997.

[33] Denis Zorin. A method for analysis of C^1-continuity of subdivision surfaces. *SIAM Journal of Numerical Analysis*, 37(4), 2000.

[34] Denis Zorin. Smoothness of subdivision on irregular meshes. *Constructive Approximation*, 16(3), 2000.

[35] Denis Zorin, Tom Duchamp, and H. Biermann. Smoothness of subdivision surfaces on the boundary. Technical report, New York University, Dept. of Computer Scinece, 2000. In preparation.

[36] P. B. Zwart. Multivariate splines with nondegenerate partitions. *SIAM J. Numer. Anal.*, 10(4):665–673, 1973.