Semi-Regular 4–8 Refinement and Box Spline Surfaces

LUIZ VELHO

IMPA–Instituto de Matemática Pura e Aplicada,
lvelho@visgraf.impa.br

Abstract. In this paper we introduce a new mesh refinement method for subdivision surfaces. It generates a semi-regular 4-direction hierarchical structure from control meshes representing 2D manifolds of arbitrary topology. The main advantage of this structure is that it allows the extraction of conforming variable-resolution meshes based on spatially varying adaptation functions. We also present a smoothing method designed to work in conjunction with our semi-regular refinement. It produces generalized four direction Box spline surfaces of class C^1. Together the refinement and smoothing operators result in a subdivision scheme that is very effective in multiresolution applications.

Keywords: subdivision schemes, four-directional grids, Laves tilings, quincunx lattice, refinement, smoothing.

1 Introduction

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

For the above reasons, subdivision surfaces received considerable attention from the research community in recent years. As a consequence, the field developed rapidly and reached a mature stage. Now, that the theoretical foundations have been established, the main emphasis has shifted towards the application area.

In that respect, perhaps the most important feature of subdivision surfaces to practical applications is its intrinsic multiresolution structure. This property is essential to the *scalability* of large modeling and visualization systems. Multiresolution also makes it possible the use of *adaptive* solutions, resulting in more accurate and efficient algorithms.

Adaptivity can only be fully exploited if the multiresolution structure supports variable level of detail across the domain of interest. We call this type of structure a *variable resolution mesh.*

Current subdivision schemes are based on uniform refinement methods that cannot produce a variable resolution mesh. This fact restricts the adaptation capabilities of subdivision surfaces and, in order, to overcome the problem, ad-hoc solutions are often employed.

In this paper, we introduce a new refinement method that generates a variable resolution mesh structure for subdivision surfaces. We also present a smoothing method designed to take advantage of the characteristics of the hierarchical mesh structure. These refinement and smoothing methods are combined to define a subdivision scheme that is very effective in multiresolution applications.

2 Refinement and Mesh Hierarchies

Mesh refinement methods are usually based on regular tilings. That is, tessellations of the plane composed by regular n-gons.

The basic idea is to start with an initial uniform tessellation, and then, apply repeatedly some refinement rule, such that, at every step, a finer tessellation, similar under scaling to the original, is produced.

There are only three types of plane tilings formed by tiles that are congruent to a single regular polygon [4]. They correspond to uniform tessellations generated, respectively by squares, equilateral triangles and regular hexagons. The most common ones are the triangle and quadrilateral tessellations. Figure 1 shows these tilings.

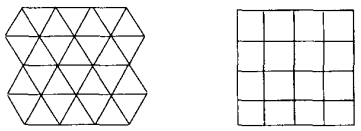


Figure 1: Uniform triangle and quadrilateral tilings.

Note that the above tilings have the desired property: it is possible to subdivide the tiles obtaining a new tiling made by similar elements of smaller size.

A planar mesh that is homeomorphic to a regular tiling is called a *regular mesh.* Since there is a 1–1 mapping between vertices, edges and faces of a regular mesh and those of the corresponding tiling, it follows that refinement methods also apply to regular meshes.

0-7695-0878-2/00 $10.00 © 2000 IEEE

The refinement procedure introduces new vertices, edges and faces to create a finer mesh from a coarse one. The local neighborhood structure of the mesh is invariant by regular refinement. Therefore, the valence of new vertices is the same as the valence of vertices of the initial mesh.

There are two main strategies for refinement: in a *primal refinement*, the old mesh is a subset of the new mesh. New faces are constructed through subdivision of old ones, in which edges are divided and reconnected. In a *dual refinement*, new faces are constructed by inserting new vertices in the the interior of old faces and connecting these vertices. The old mesh is discarded. The refinement rule is conveniently described by a diagram, called *refinement template*.

The principles of regular planar refinement can be generalized for tilings of 2D manifolds. The generalization consists simply in applying the refinement method to an initial mesh that tessellates the manifold.

The main difference is that, in general, it is not possible to tessellate a manifold of arbitrary topology using a regular mesh. In such cases, the initial mesh will contain vertices of arbitrary valence, and it is called an irregular mesh. Vertices with the same valence as in a regular mesh are known as *regular vertices*, while vertices with other valences are known as *extraordinary vertices*.

Since the refinement process, as we have seen, creates only ordinary vertices, in a refined mesh, the only extraordinary vertices are the ones inherited from the initial mesh. As the mesh is refined, these vertices will be isolated, surrounded only by vertices with regular valence. For this reason, such meshes are called *semi-regular*.

A refinement method generates a sequence of meshes with increasing resolution. This sequence is called a *multiresolution mesh* and can be represented by a hierarchical data structure, such as a tree. For primal refinement, an n-ary tree of faces is the natural representation, since every face is decomposed into n subfaces. For dual refinement, the best choice is an n-ary tree of vertices.

We remark that the multiresolution structure generated by refining regular triangle and quadrilateral meshes do not allow the transition between faces of different resolutions while maintaining the topological consistency of the mesh. This is particularly clear in the case of primal refinement, because at every step all edges are subdivided. Therefore, the subdivision of one face has to be propagated to all other faces at the same level of resolution.

In order to overcome this deficiency, meshes with faces at different resolution have to be modified by a post-process that fixes topological inconsistencies. The usual solution is to enforce a restricted hierarchical structure, such that, adjacent faces differ, at most, by one level. When there is a difference in resolution across an edge, the face at the lowest level is subdivided to re-establish consistency. Figure 2 illustrates this procedure for a quadrilateral mesh. Note that, faces of a different type (i.e. triangles) had to be introduced.

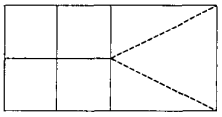


Figure 2: Post-process to fix topological inconsistencies.

3 Adaptive Tilings

The analysis in the previous section motivates us to look for a hierarchical mesh structure that supports variable resolution and can be generated by refinement. This quest leads us to a larger class of tilings: the monohedral tilings with regular vertices, also known as *Laves tilings*, named after the crystallographer Fritz Laves, who studied them [4].

In a *monohedral* tiling, every tile is congruent to one fixed tile, called *prototile*. This means that all faces in the tessellation have the same shape and size.

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. We classify these tilings by listing the degree of the vertices of their prototile in cyclic order. Thus, they are named using the following notation $[d_1, \ldots, d_k]$, where d_i is the valence of vertex v_i. We also use superscripts to indicate repetition of symbols.

The Laves tilings have the property that we want – they are refinable and more general than regular tilings. As it was expected, regular triangle, quadrilateral and hexagonal tilings also belong to this class. They are, respectively, the Laves tilings of type $[6^3]$, $[4^4]$ and $[3^6]$.

Only certain types of Laves tilings can be used to generate hierarchical meshes that support non-uniform variable resolution. These are the Laves tilings of type $[4.8^2]$ and [4.6.12] They are both composed of right triangles: in the first, the prototile is an isosceles triangle; while in the second, it is a 30-60 triangle. Figure 3 shows these two tilings.

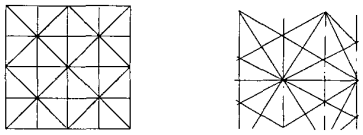


Figure 3: Laves tilings $[4.8^2]$ and [4.6.12]

We have chosen the $[4.8^2]$ instead of the [4.6.12] tiling, for two main reasons: its vertices have lower valence; and, as we shall see, it allows a better transition between different resolution levels.

It is worth noting that the $[4.8^2]$ tiling forms a *triangulated quadrangulation*. Thus, it gives, at the same time, the advantages of triangular and quadrilateral tessellations.

The basic structure in a $[4.8^2]$ tiling is a pair of triangles forming a square block divided along one of its diagonals. We call this structure a *basic block*. Note that, splitting the internal edge of one basic block causes the subdivision of its two triangles, without affecting any of the neighbors. This procedure is illustrated in Figure 4 (left). Consider now, the refined block formed by four triangles. Splitting one of its external edges, causes the subdivision of just one triangle. Since the refinement was on the boundary of the block, it does not affect the other three triangles. But, it affects the neighbor block sharing that edge, and forces the subdivision of one of its triangles. This is shown in Figure 4 (right).

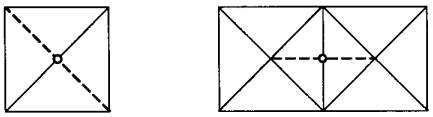


Figure 4: Refinement of a basic square block.

The above mechanism for refining internal–external edges of a basic block, makes it possible to build tessellations with different resolution levels. This is the key property of $[4.8^2]$ tilings, related to variable level of detail.

It is important to realize that refining the boundary of a block depends on the refinement of its interior. This dependency propagates across the tessellation, ensuring a gradual transition between resolution levels. In fact, it results in a restricted quad-tree structure. Figure 5 shows an example of constrained resolution propagation.

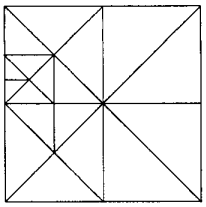


Figure 5: Transition between resolution levels.

In summary, the $[4.8^2]$ tiling has, not only the adaptation property that we were looking for. This makes it the perfect choice as the underlying structure for a multiresolution mesh. Indeed, this structure is known by the GIS community as *hierarchy of right triangles* [9]. It has been used very successfully for rendering terrain models by Lindstrom et al. [7], Kirkpatrick et al.[3], and Duchaineau el al. [2].

4 Regular 4–8 Meshes

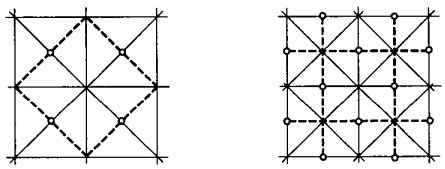
A *regular 4–8 mesh* is a cell complex that is homeomorphic to a $[4.8^2]$ Laves tiling. By definition, every face has one vertex of valence 4 and two vertices of valence 8. The 4–8 mesh has edges of two types: 8-8 edges, linking two vertices of valence 8; and 4–8 edges, linking one vertex of valence 4 to one vertex of valence 8.

Now, we would like to devise a refinement procedure based just on topological information. Observe that, edges of type 8–8 occur only as the diagonal edges of basic blocks. This follows directly from the regularity condition.

Using the observation above, we specify a binary subdivision procedure:

1. Split all edges of type 8–8, by inserting a split vertex.
2. Subdivide all faces into two subfaces, by linking the degree 4 vertex to the split vertex of the opposite edge.

Note that, in order to produce a self-similar mesh, binary subdivision has to be applied twice [1]. For this reason, the *regular 4–8 refinement* is defined as a double step of binary subdivision. This is illustrated in Figure 6.



(a) step 1 (b) step 2

Figure 6: Two binary subdivision steps of regular 4–8 refinement.

Note also that, because the 4–8 mesh is a triangulated quadrangulation, the double subdivision step performs primal refinement on triangular faces, as well as, a cyclic primal and dual refinement on quadrilateral blocks (see Figure 7).

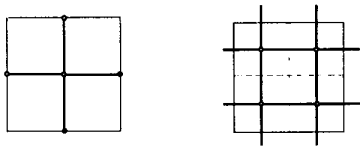


Figure 7: Primal and dual refinement of quadrilateral blocks.

[1] Applying just one subdivision step results in a mesh that when rotated by 45 degrees is self-similar to the original.

It is remarkable that two steps of 4–8 refinement are equivalent to Catmull-Clark subdivision of quadrilateral blocks at level j, and, at the same time, to Doo-Sabin subdivision of quadrilateral blocks at level $j+1$. This fact will be exploited in Section 6 for the design of a smoothing operator.

The regular 4–8 refinement procedure relies on the special topological structure of the mesh. In order to make it widely applicable, particularly for the representation of 2D manifolds, it is necessary to extend it to accept arbitrary initial meshes. This will be discussed in the next section.

5 Semi-Regular 4–8 Refinement

The generalization of regular 4–8 refinement exploits the fact that subdivision operates on quadrilateral blocks. Thus, our strategy is to take a triangulation as input, and, in a pre-processing step, construct a block structure that can be handled by regular refinement. We call the complete procedure, *semi-regular 4–8 refinement*, and the mesh produced under its action, *semi-regular 4–8 mesh*.

The main problem consists in transforming an arbitrary initial mesh into a triangulated quadrangulation, making just few modifications to the mesh.

Our solution is based on the fact that applying one step of Catmull-Clark subdivision to an arbitrary tessellation, produces a mesh containing only four-sided faces [6]. Figure 8 shows an example.

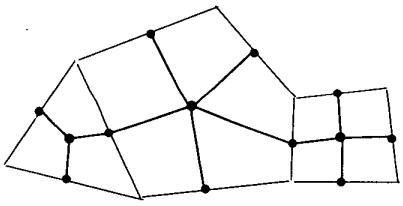


Figure 8: Example of Catmull-Clark subdivision first step.

Since this result is valid, in particular, for generic triangulations, we could just use Catmull-Clark subdivision as our pre-processing step. Nonetheless, we pursue a different strategy that minimizes changes to the original mesh, produces well shaped faces, and avoids increasing vertex valences.

First, we select a set of two-face clusters, Q, that covers most of the mesh. At the same time, we identify the remaining triangle faces, $T = \overline{Q}$. Then, we apply one step of Catmull-Clark subdivision to $T \cup Q$, but treating the two-face clusters in Q, as quadrilateral faces.

The complete pre-processing method is summarized below:

1. Find an independent set of two-face clusters, and identify the remaining isolated triangle faces;
2. Perform a hybrid binary subdivision step, and mark the resulting quadrilateral blocks.
3. Perform one step of regular 4–8 binary subdivision.

We actually use the machinery developed in the previous section for regular 4–8 refinement to perform Catmull-Clark subdivision (note that, together (2) and (3) are equivalent to one step of Catmull-Clark).

To implement pass (1) we select clusters based on edge length. This heuristic guarantees that we obtain convex quadrilateral blocks in case of planar meshes. The pseudo-code is shown in Algorithm 1

Algorithm 1 : find_clusters ($K = \{V, E, F\}$)

```
sort_edges (E)
store e ∈ E in priority queue Q
while Q ≠ ∅ do
  get e from Q
  if e not marked then
    mark_cluster (e)
```

The routine sort_edges, sorts edges by decreasing length and radially around each vertex. The routine mark_cluster(e) marks an edge e and the edges sharing a face with e. Marking ensures that we obtain an independent set of two-face clusters.

In general, it is not possible to cover the whole mesh with these two-face clusters. There will be a few isolated triangular faces remaining.

The purpose of the hybrid subdivision step in pass (2) is to make these isolated triangular faces compatible with the two-face clusters, such that, after subdivision we get a quadrangulation that covers the whole mesh.

The hybrid subdivision procedure applies distinct refinement rules to two-face clusters and isolated triangles. Two-face clusters are subdivided in the regular way through binary subdivision. Isolated triangles are subdivided using barycentric subdivision: three new faces are created by linking the barycenter to each old vertex of the triangle. Figure 9 illustrates hybrid refinement.

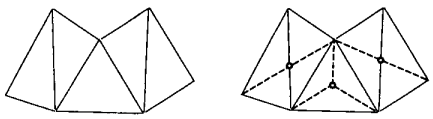


Figure 9: Hybrid refinement.

After the pre-processing, two-face clusters subdivide into four basic blocks, and isolated triangles subdivide into three basic blocks. The union of these blocks covers the mesh, providing the desired triangulated quadrangulation structure (see Figure 10).

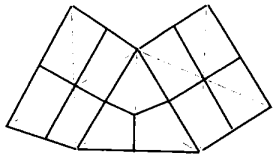


Figure 10: Quadrangulated triangulation.

As a result of the pre-process step, the valence of every interior vertex will be even, while the valence of boundary vertices will be odd. Valences may 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 (1) for selecting two-face clusters [10].

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.

The behavior for boundary vertices is similar.

6 The Four Directional Grid and the ZP Element

4–8 meshes are closely related to the four directional grid, well known in the theory of Box splines [1]

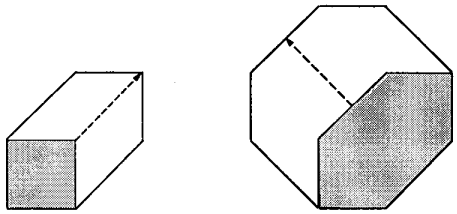
A *Box spline* is generated by convolutions of the characteristic function of the unit partition, along a prescribed set of directions. They are smooth piecewise polynomial functions with compact support. They are refinable, and their translates form a basis. Box spline basis are usually specified by a set of direction vectors.

Box spline functions can be used to create surfaces that are defined parametrically by a function $g : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$. In this setting, a box spline surface is specified by control points $c_{uv} \in \mathbb{R}^3$ that are associated with grid points $(u, v) \in \mathbb{Z}^2$ of the domain U.

The simplest smooth Box spline over a four directional grid is the Zwart-Powell function [13], (also known as the ZP element). It is associated wit the set of vectors $D = \{e_1, e_2, e_1+e_2, e_1-e_2\}$, where $e_1 = (1, 0)$ and $e_2 = (0, 1)$. Or, in matrix form

$$D = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix}$$

The construction of the ZP element is as follows: we start with the characteristic function of the unit square, given by $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Then we perform convolution: first in the direction $(1, 1)$, and next in the direction $(1, -1)$. This construction is shown in figure 11.



(a) (b)

Figure 11: Construction of the Zwart-Powell element

The functions are piecewise quadratic, with C^1 continuity across grid lines. Figure 12 shows a plot of the Zwart-Powell function, as well as, its support on the underlying four directional grid.

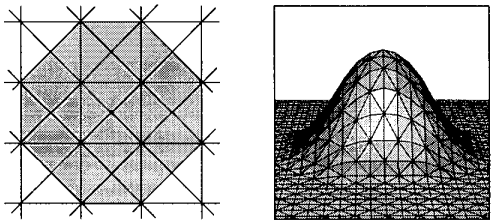


Figure 12: Zwart-Powell function

Using refinement we express the Zwart-Powell function on a coarse grid as a linear combination of, scaled and translated, functions on a finer grid. This two-scale relation can be computed from the *generating function*, $S(z_1, z_2)$, associated with the induced subdivision scheme (See [12]). [2]

The generating function for subdividing the ZP element is

$$S(z_1, z_2) = \frac{1}{4}(1 + z_1)(1 + z_2)(1 + z_1 z_2)(1 + z_1/z_2) \tag{1}$$

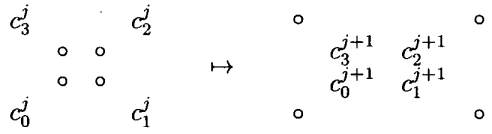
Expanding this equation, we obtain the coefficients of the two-scale relation. They are the coefficients of the monomials of $S(z_1, z_2)$, where the weight at grid point (u, v) is the coefficient of $z_1^u z_2^v$.

[2] A generating function is a Laurent polynomial specifying the transfer function of a discrete convolution operator [11]

The coefficients of the subdivision formula, non-normalized by the factor $\frac{1}{4}$, are shown below:

$$\begin{array}{cccc} & 1 & 1 & \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ & 1 & 1 & \end{array}$$

As the grid is refined, values at grid points of the finer grid are calculated as linear combinations of values at grid points of the coarse grid. The ZP Box spline uses dual refinement, and subdivision employs the following update scheme for values c_k at the vertices of each quadrilateral block



where the control point c_0^{j+1} at level $j+1$ is computed by

$$c_0^{j+1} = \frac{1}{2}c_0^j + \frac{1}{4}c_1^j + \frac{1}{4}c_3^j$$

and similarly, for the other control points.

The above equation defines a smoothing rule that is conveniently represented by a mask (or stencil) indicating the computation. Figure 13 shows the mask for the control point c_0 (the other masks, for c_1, c_2 and c_3, are versions of this mask rotated by 90, 180 and 270 degrees, respectively).

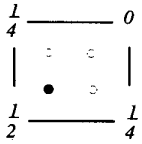


Figure 13: Mask for c_0 using dual quad-mesh refinement of the ZP subdivision scheme.

The fact that, as we have seen in Section 5, regular 4–8 refinement consists of two binary subdivision steps, allows us to factor the smoothing operator into simpler rules. We employ the mask shown in Figure 14, that performs the averaging of two values, and is the simplest convolution filter.

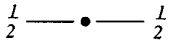


Figure 14: Averaging mask.

The factored subdivision scheme is as follows: At level j, the grid is refined in the horizontal and vertical directions. The values at new grid points are the average of their two neighbors. At level $j+1$, the grid is refined in the two diagonal directions. The values at new grid points are the averages of the two opposite vertices. See Figure 15.

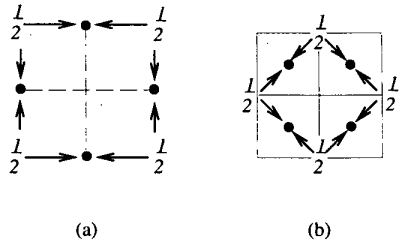


Figure 15: Hybrid refinement.

Note that, convolution in a particular direction corresponds to one factor of the generating function $S(z_1, z_2)$ in equation (1).

Due to the dual nature of refinement, values of old grid points are not used after two refinement steps. For this reason, at every double step, old points are updated to be the average of the surrounding four new grid points (e.g. they become the centroid of the newly created quadrilateral blocks). See Figure 16.

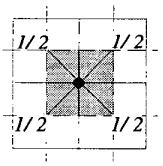


Figure 16: Updating the center of quad-blocks.

The generalization to irregular meshes is immediate since the factorization of the subdivision scheme also applies to semi-regular 4–8 refinement. It is worth noting that, in the case of four direction Box splines, it is not even necessary to design special rules for extraordinary faces.

We remark that, the above principles were employed by Peters and Reif to construct Zwart-Powell Box spline surfaces using quadrilateral meshes [8]. They called the scheme, midedge subdivision. Habib and Warren also adopted a similar scheme in [5].

Continuity analysis showed that the limit surface is of class C^1. Unfortunately, the convergence can be uneven for non-quadrilateral faces. It is too fast for 3-sided faces and too slow for n-sided faces, with $n > 4$. (See Figure 18(b)). Special rules can minimize this problem [8].

7 Examples and Comparisons

We now present some results of applying the methods developed in the previous sections for surface modeling. We combine semi-regular 4–8 refinement with four direction Box spline smoothing, to create a subdivision scheme that generalizes the C^1 four directional Box spline.

Examples of surfaces produced by such scheme are constructed. The adaptation capabilities of the multiresolution 4–8 mesh representation is also demonstrated.

The first example, in Figure 17, illustrates the integration of refinement and smoothing. Figure 17(a) shows an irregular base mesh containing two extraordinary vertices of valence 6 in its interior. Figure 17(b) depicts the mesh after applying two times the semi-regular 4–8 refinement operator only. Figures 17(c) shows the effect of using the complete C^1 subdivision scheme — both refinement and smoothing were applied.

The second example, in Figure 18, is a compact surface of genus 0. Figure 18(a) shows the control polyhedron, obtained by extruding a regular pentagon. Figures 18(b) shows an approximation of the C^1 surface, after two levels of subdivision.

The third example, in Figure 19, is the "Stanford Bunny", generated from sampling a real object. The control polyhedron, shown in Figure 19(a), was created from a dense mesh using simplification. Figure 19(b), shows the smoothed polygonal mesh after applying 1 step of the C^1 subdivision scheme.

The last example, in Figures 20 and 20, briefly demonstrates the power and flexibility of the 4–8 mesh structure as a multiresolution representation. In this example, spatially-variable threshold functions are used to extract topologically consistent approximations of the Bunny, based on various criteria. The adaptation function specifies the desired resolution at each point of the mesh.

In Figure 20, adopted criteria was point location. The extracted tessellation, shown in Figure 20(a) was constrained to exhibit highest resolution at one point on the surface. A detail of the transition region is shown in Figure 20(b). Note that the mesh quickly adapts to the specified constraint.

In Figure 20, the threshold function varies along the horizontal direction in two different ways. Figure 20(c) shows a tessellation in which the resolution varies from left to right according to a linear ramp. Figure 20(d) shows a tessellation that was based on a step function. The mesh has finest resolution on one side of a plane, and coarsest resolution on the other side of the plane. Note that the multiresolution 4–8 mesh structure was able to support both the sharp transition in Figure 20(d), as well as, the gradual transition in Figure 20(c).

Acknowledgements

The figures in section 7 were generated with Geomview. We wish to thank Michael Garland and Paul Heckbert for making Qslim publicly available. We acknowledge the Stanford Computer Graphics Laboratory and Viewpoint for providing the original data for the examples.

The authors are partially supported by research grants from the Brazilian Council for Scientific and Technological Development (CNPq) and Rio de Janeiro Research Foundation (FAPERJ).

References

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

[2] 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.

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

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

[5] A. Habbib and J. Warren. Edge and vertex insertion for a class of C^1 subdivision surfaces. preprint, 1997.

[6] P. L. King. On local combinatorial Pontjagrin numbers – i. *Topology*, 16:99–105, 1977.

[7] 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.

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

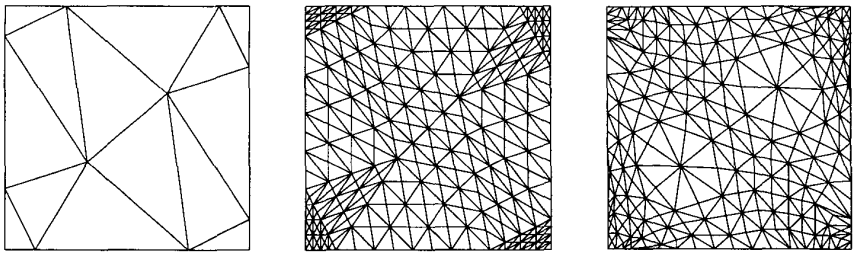
[9] E. Puppo and R. Scopigno. Simplification, LOD and multiresolution – principles and applications, 1997. Eurographics'97 Tutorial Notes.

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

[11] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA, 1996.

[12] J. Warren. Subdivision methods for geometric design. Unpublished monograph, 1995.

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



(a) base mesh (b) refinement — 2 levels (c) C^1 smoothing

Figure 17: Semi-Regular 4–8 Refinement and four directional Box spline smoothing

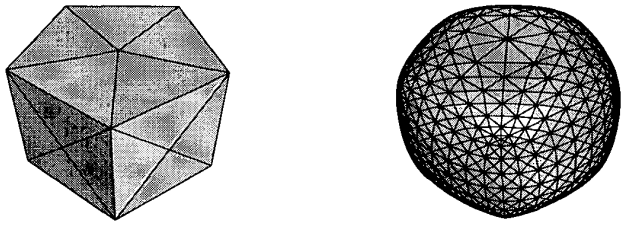


Figure 18: Extruded Pentagon

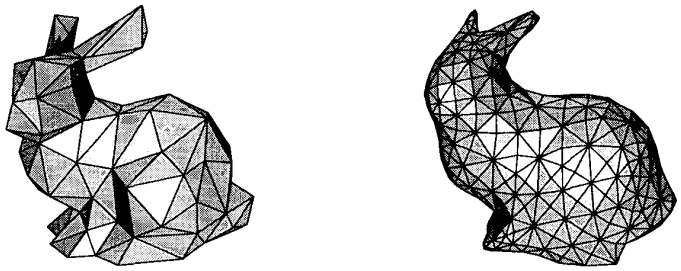
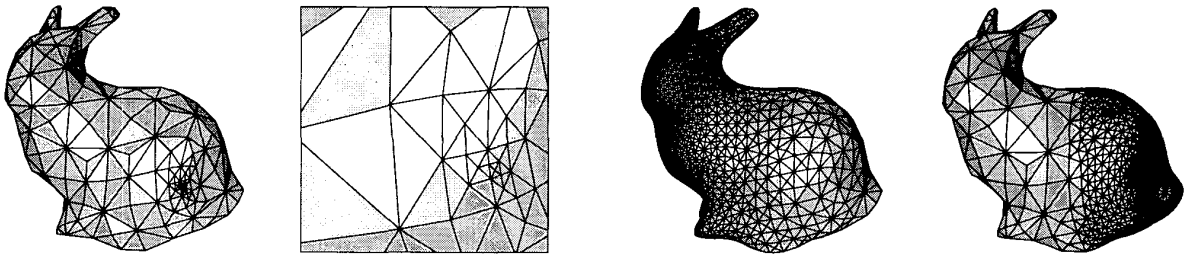


Figure 19: Stanford Bunny



(a) adapted mesh (b) transition region (c) linear ramp (d) step function

Figure 20: Variable resolution tesselations based on: point location (a-b); threshold functions (c-d).