Generalizing the C^4 Four-directional Box Spline to Surfaces of Arbitrary Topology

Luiz Velho

Abstract. In this paper we introduce a new scheme that generalizes the four-directional box spline of class C^4 to surfaces of arbitrary topological type. The scheme is composed of semi-regular binary refinement together with separable two-pass smoothing by repeated convolution.

§1. The $[4.8^2]$ Laves Tiling and Bisection Refinement

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. They correspond to uniform tessellations generated by squares, equilateral triangles, and regular hexagons. The most common ones are the triangle and quadrilateral tessellations. 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 larger class of refinable tilings are the monohedral tilings with regular vertices, also known as Laves tilings, named after the crystallographer Fritz Laves, who studied them [2].

In a monohedral tiling, every tile is congruent to one fixed tile, called the 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 notation $[d_1, d_2 \ldots, d_k]$, where d_i is

Mathematical Methods in CAGD: Oslo 2000
Tom Lyche and Larry L. Schumaker (eds.), pp. 1–10.

ISBN 0-8265-xxxx-x.

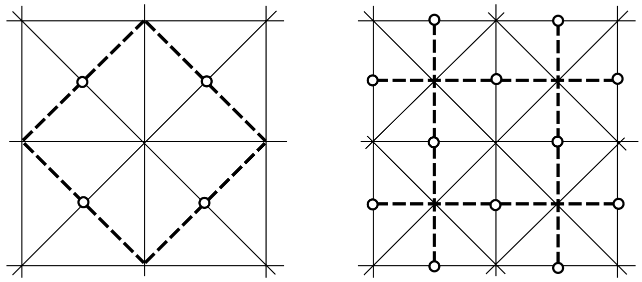

Fig. 1. Two bisection refinement steps of the 4–8 the refinement procedure.

the valence of vertex v_i (we also use superscripts to indicate repetition of symbols). As 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 $[3^6]$. Naturally, it is possible to extend all refinement concepts to the Laves tilings.

The $[4.8^2]$ Laves tiling has a rich structure that can be exploited in the context of subdivision with many advantages over the traditional $[6^3]$ and $[4^4]$ tilings. Among other things, refinement is based on bisections, and uniform as well as non-uniform refinement are both supported.

The $[4.8^2]$ tiling is composed of isosceles right triangles. The basic structure is a pair of triangles forming a square block divided along one of its diagonals. We call this structure a basic block.

A regular 4–8 mesh is a cell complex that has the same connectivity as 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.

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 this observation, we specify a 4–8 bisection refinement procedure.

The procedure is as follows: First, we bisect all edges of type 8–8, by inserting a split vertex. Then, we subdivide all faces into two sub-faces, by linking the vertex of degree 4 to the split vertex of the opposite edge.

Note that, in order to produce a self-similar mesh, 4–8 bisection refinement has to be applied twice (applying just one subdivision step results in a mesh that when rotated by 45 degrees is self-similar to the original). For this reason, the regular 4–8 refinement is defined as a double step of 4–8 bisection refinement This is illustrated in Figure 1.

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 the refinement procedure to accept arbitrary initial control meshes.

The generalization of regular 4–8 refinement to non-regular meshes exploits the fact that subdivision operates on quadrilateral blocks. Thus, our strategy is to take a triangulation as input, and, in a pre-processing phase,

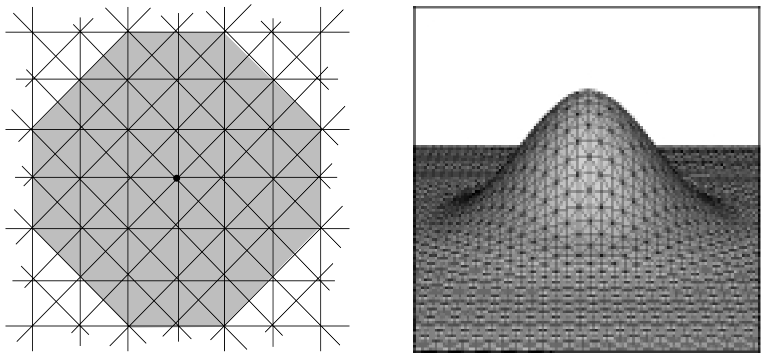

Fig. 2. C^4 Box spline function.

construct a block structure that can be handled subsequently by regular 4–8 refinement. This pre-processing transforms an arbitrary initial mesh into a triangulated quadrangulation [3]. We call the complete procedure a semi-regular 4–8 refinement, and the mesh produced under its action semi-regular 4–8 mesh. Thus, a semi-regular 4–8 mesh is a triangle mesh in which original vertices may have arbitrary valence and all other vertices have either valence 8 or valence 4. First, when a vertex is generated at refinement step i, its valence is 4. Then, at the subsequent refinement step $i+1$, its valence becomes 8.

§2. C⁴ Box Spline Subdivision

4–8 meshes are closely related to the four-directional grid, well known in the theory of 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 $g: U \subset \mathbf{R}^2 \to \mathbf{R}^3$. In this setting, a box spline surface is specified by control points $c_{uv} \in \mathbf{R}^3$ that are associated with grid points $(u,v) \in \mathbf{Z}^2$ of the domain U.

The simplest smooth box spline over a four-directional grid is the Zwart-Powell function [7], also known as the ZP element. It is associated with the set of vectors $D = \{e_1, e_2, e_1+e_2, e_1-e_2\}$, where $e_1 = (1,0)$ and $e_2 = (0,1)$.

A four-directional box spline that exhibits a higher order of smoothness than the ZP element is the function generated by the set of direction vectors

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

This function is a piecewise polynomial of degree 6 and it is C^4 continuous. Figure 2 shows a plot of the function, as well as its support on the four-directional grid.

Using refinement, we express the function on a coarse grid as a linear combination of scaled and translated functions on a finer grid. This two-scale

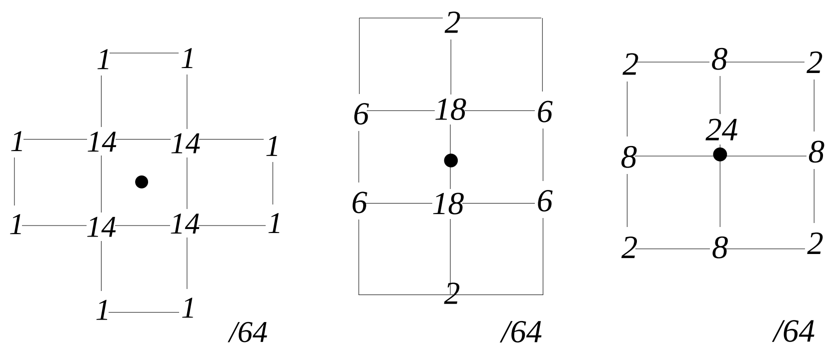

Fig. 3. Face, edge and vertex masks for C^4 box spline.

relation can be computed from the generating function $S(z_1, z_2)$ associated with the induced subdivision scheme (See [6]). The generating function of the C^4 four-directional box spline is given by

$$S(z_1, z_2) = \frac{1}{64}(1 + z_1)^2(1 + z_2)^2(1 + z_1 z_2)^2(1 + z_1/z_2)^2.$$

Expanding it, we compute 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 $z_1^u z_2^v$. These coefficients are shown below (without normalization by the factor $\frac{1}{64}$).

		1	2	1		
	2	6	8	6	2	
1	6	14	18	14	6	1
2	8	18	24	18	8	2
1	6	14	18	14	6	1
	2	6	8	6	2	
		1	2	1		

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. From the subdivision formula we obtain the smoothing masks for face, edge and vertex points, shown in Figure 3. The masks extend over a 2-neighborhood, as expected.

The large support of the masks makes the implementation of a subdivision scheme based on the C^4 box spline difficult. The separability property of semi-regular 4–8 refinement, allows us to factorize such a high order scheme into manageable pieces.

We decompose the C^4 smoothing operator into two masks, shown in Figure 4. It is easy to see that the smoothing filter corresponds to averaging in the horizontal, vertical, and diagonal directions. When applied successively in the two binary steps comprising the 4–8 refinement, the result is a discrete convolution twice in each of the four directions.

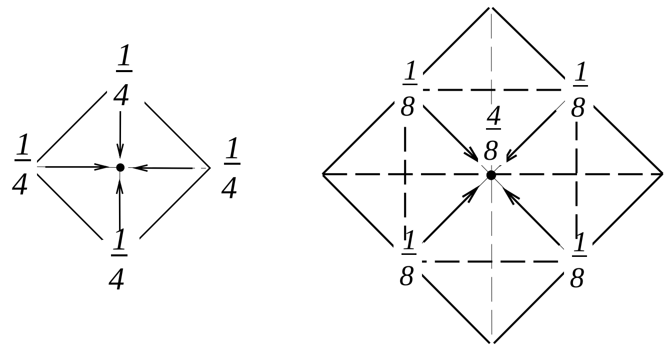

Fig. 4. Factorized face and vertex masks for C^4 Box spline.

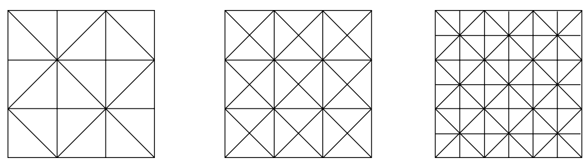

Fig. 5. Three consecutive subdivision levels, $j-1$, j and $j+1$.

First, we use the face mask shown in Figure 3 (left) to compute the values at new vertices $v' \in V'$ (recall that, by construction, their 1-neighborhood $N_1(v')$ consists of exactly 4 vertices). The filter function is an average of the values at the neighbors. Next, we update the values of old ordinary vertices as follows: for any vertex v with valence 8 *after subdivision*, (i.e. old ordinary vertices), update it by weighted average of its neighboring valence-8 vertices, $\{v' \in N_1(v); deg(v') = 8\}$, (always 4 vertices) and v itself. This vertex mask is shown in Figure 4 (right).

The two factorized face and vertex masks in Figure 4 combine to produce the original face, edge, and vertex masks of the C^4 box spline in Figure 3.

We now describe in more detail the factorization of the C^4 subdivision scheme. We will analyze three consecutive levels of subdivision, $j+1$, j and $j-1$. We want to compute the values of the vertices of the mesh, $v_i^{j+1} \in V$ at level $j+1$. The face, vertex, and edge masks in Figure 3 compute new values based on previous values of the vertices of the mesh at level $j-1$. The factorization of these masks will employ a combination of the face and vertex masks in Figure 3 at levels $j+1$ and j, using values of the vertices at levels j and $j-1$, respectively. We remark that the factorized masks appear rotated by 45 degrees at two consecutive levels. In the following figures, the face and vertex masks will be indicated, respectively, by dashed arrows on a gray background and by solid arrows. Figure 5 shows a piece of the mesh at these three consecutive levels of subdivision, which we will use in the analysis. For clarity, in the figures we will only show the grid lines of level j.

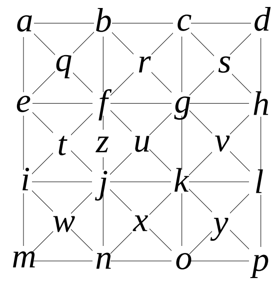

Fig. 6. Labeling scheme for the vertices of the mesh.

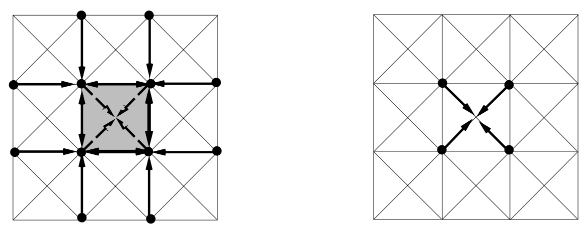

Fig. 7. Decomposition of the center mask.

Figure 6 shows the labeling scheme that we will use to identify the vertices of the mesh.

The C^4 face mask is decomposed into an application of face and vertex masks at level j, followed by an application of vertex mask at level $j+1$. Figure 7 shows the flow of computations used for updating vertex u.

At level $j+1$ the new value of vertex u is computed using the vertex mask

$$u^{j+1} = \frac{4}{8}u^j + \frac{1}{8}\left(f^j + g^j + k^j + j^j\right)$$

The values of u^j, f^j, g^j, k^j, and j^j are computed at level j using the face mask for computing u^j

$$u^j = \frac{1}{4}\left(f^{j-1} + g^{j-1} + k^{j-1} + j^{j-1}\right)$$

and the vertex mask for computing the values of f^j, g^j, k^j, and j^j. So, we have

$$f^j = \frac{4}{8}f^{j-1} + \frac{1}{8}\left(b^{j-1} + g^{j-1} + j^{j-1} + e^{j-1}\right)$$

and similarly for vertices g^j, k^j, and j^j. Substituting the values of vertices u^j, f^j, g^j, k^j, and j^j into the equation for u^{j+1}, we obtain the C^4 face mask, which gives the value of u^{j+1} in terms of the values of the vertices of the mesh ate level $j-1$.

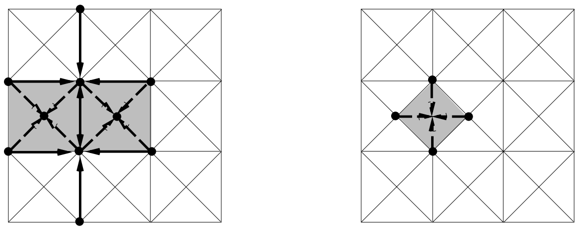

Fig. 8. Decomposition of the edge mask.

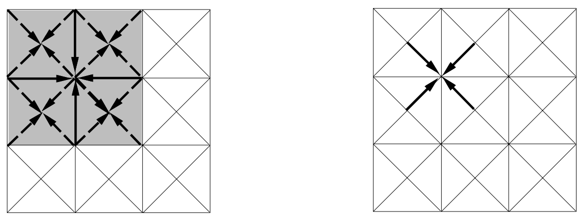

Fig. 9. Decomposition of the vertex mask.

The C^4 edge mask is decomposed into an application of face and vertex masks at level j, followed by an application of face mask at level $j+1$. Figure 8 shows the flow of computations used for updating vertex z.

At level $j+1$ the new value of vertex z is computed using the face mask

$$z^{j+1} = \frac{1}{4}\left(f^j + u^j + j^j + t^j\right)$$

The values of f^j, u^j, j^j, and t^j are computed at level j using the vertex mask for computing f^j and j^j, and the face mask for computing u^j and t^j. The computation for j^j is similar to the computation of f^j, while the computation for t^j is similar to the computation of u^j. Substituting the values of f^j, u^j, j^j, and t^j into equation for z^{j+1}, we obtain the C^4 edge mask.

The C^4 vertex mask is decomposed into an application of face and vertex masks at level j, followed by an application of vertex mask at level $j+1$. Figure 9 shows the flow of computations used for updating vertex f.

At level $j+1$ the new value of vertex f is computed using the vertex mask

$$f^{j+1} = \frac{4}{8}f^j + \frac{1}{8}\left(r^j + u^j + t^j + q^j\right)$$

The values of f^j, r^j ,u^j, t^j, and q^j are computed at level j using the vertex mask for computing f^j and the face mask for computing r^j ,u^j, t^j, and q^j. The computation for r^j, t^j, and q^j is similar to the computation of u^j. Again, substituting the values of f^j, r^j ,u^j, t^j, and q^j, into the equation for f^{j+1}, we obtain the C^4 vertex mask.

The factorization described above applies only for regular vertices of valence 4 and 8. In order to extend the subdivision scheme to arbitrary meshes, we have to devise masks for extraordinary vertices of valence $2n$ (since vertices of a semi-regular 4-8 mesh have even valence). It turns out that is only necessary to generalize the factorized vertex mask. This mask will combine with

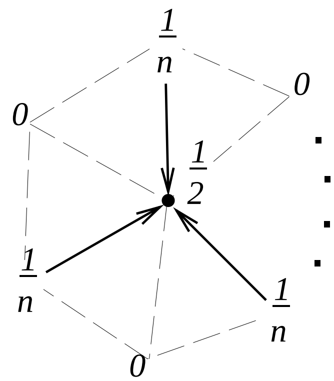

Fig. 10. Generalization of factorized vertex masks for valence $2n$.

the face mask to produce new C^4 face, edge and vertex masks for arbitrary meshes.

Figure 10 shows the generalization of the factorized vertex mask. Note that extraordinary vertices have n new neighbors with valence 4 and n old neighbors with valence greater that 4. The mask is a weighted average of the extraordinary vertex and its old neighbors.

The smoothing operator is computed by a simple procedure that first applies the face mask to new vertices of valence 4, $v'_i \in V^{l+1} \setminus V^l$,

$$p^{l+1}(v'_i) = \frac{1}{4} \sum_{v_j \in N_1(v'_i)} p^l(v_j),$$

and then applies the vertex mask to old vertices of arbitrary valence, $v_i \in V^l$,

$$p^{l+1}(v_i) = \frac{1}{2} p^l(v_i) + \frac{1}{2n} \sum_{v_j \in N_1(v_i) \cap V^l} p^l(v_j).$$

Smoothness of the subdivision scheme follows from the regular C^4 four-directional box spline. The limit surface is C^4 continuous almost everywhere, except at isolated extraordinary vertices. In [3] we show that the surface is C^1 continuous at extraordinary vertices.

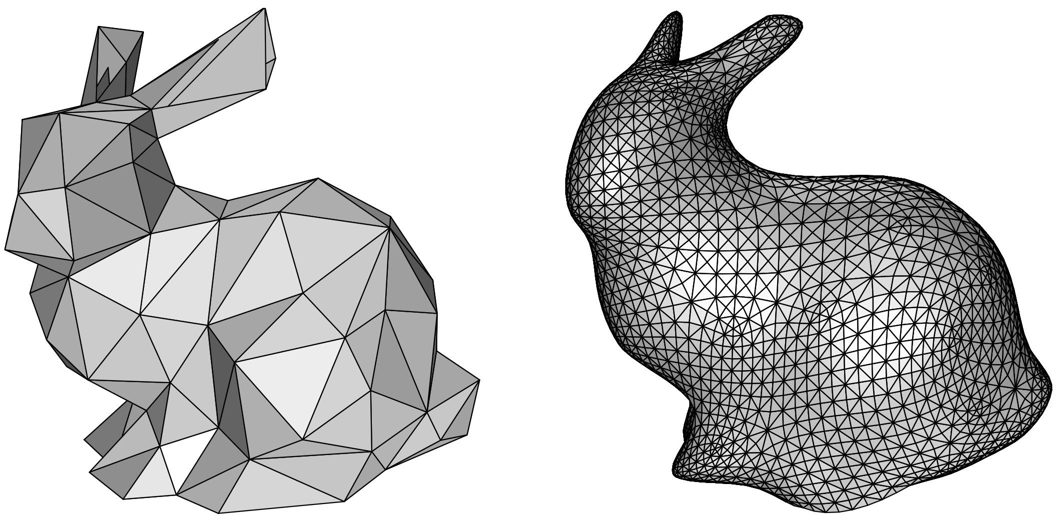

Fig. 11. Stanford Bunny.

Figure 11 shows an example of a C^4 continuous box spline surface generated from a mesh of control points with arbitrary connectivity.

§3. Conclusions

We have presented a new subdivision scheme based on the $[4.8^2]$ Laves tiling that extends the four-directional box splines of class C^4 to surfaces of arbitrary topological type. Our scheme is composed of a semi-regular 4–8 refinement operator a separable two-pass smoothing operator. The characteristics of these two operators make 4–8 subdivision a powerful tool for CAGD.

Semi-regular 4–8 refinement employs only bisections, and generates a hierarchical mesh structure that supports adaptive multiresolution. Separable two-pass smoothing allows a simple and efficient implementation of large masks through a decomposition of the subdivision rules.

In addition to the C^4 box spline, the 4–8 subdivision framework makes possible the construction of other schemes based on two-directional and four-directional grids.

We have described in [4] the implementation of Doo-Sabin and Catmull-Clark surfaces using the 4–8 subdivision framework. We have also implemented the Midedge scheme, that is based on the C^1 four-directional box spline.

An important aspect of 4–8 subdivision is its support for adaptive multiresolution. In [5], we discuss the hierarchical structure for 4–k meshes and give an overview of its applications.

References

1. C. de Boor, D. Hollig, and S. Riemenschneider, *Box Splines*, Springer-Verlag, New York, NY, 1994.
2. Grünbaum and G. Shephard, *Tilings and Patterns*, W. H. Freeman, 1987.
3. L. Velho, D. Zorin, 4–8 subdivision, Comput. Aided Geom. Design, 2001.
4. L. Velho, Using semi-regular 4–8 meshes for subdivision surfaces Journal of Graphics Tools, 2001.
5. L. Velho, J. Gomes, Variable resolution 4-K meshes: concepts and applications, Computer Graphics Forum, 2000.
6. J. Warren, Subdivision methods for geometric design, Unpublished monograph, 1995.
7. P. B. Zwart, Multivariate splines with nondegenerate partitions, SIAM J. Numer. Anal., 1973, **10:4**, 665–673.

Luiz Velho
IMPA – Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina 110, Rio de Janeiro, RJ, Brazil, 22460-320
`lvelho@visgraf.impa.br`