

Quasi 4–8 Subdivision Surfaces

LUIZ VELHO JONAS GOMES

IMPA–Instituto de Matemática Pura e Aplicada,
{lvelho|jonas}@visgraf.impa.br

Abstract. This paper presents a new scheme for subdivision surfaces that is based on four-directional meshes. It combines geometry-sensitive refinement with convolution smoothing. The scheme has a simple, efficient implementation and generates well-shaped meshes which converge to smooth surfaces.

keywords: subdivision schemes, four-directional meshes, quincunx lattice, refinement, smoothing.

1 Introduction

In recent years subdivision surfaces have become one of the most important mathematical tools for shape modeling. Subdivision surfaces provide a natural generalization of spline surfaces with several advantages: they can handle control meshes of arbitrary topology; they guarantee global surface smoothness while making possible the control of local features; they effectively bridge the gap between continuous models and discrete representations; and, lastly, they are associated with efficient algorithms that are simple to implement. Another important point is that subdivision is intimately related with multiresolution and wavelets. This is the key to scalability and adaptation, two fundamental properties for dealing with complex shapes in a modeling system.

The power of a subdivision surface model lies in the quality of the underlying limit surface, as well as, in the applicability of the associated computational scheme. The various types of subdivision surfaces offer different compromises between these two aspects. In this paper, we present a new type of subdivision surface that achieves a good balance of features. It is another addition to the existing repertoire of shape design tools.

1.1 Basic Concepts

A *subdivision surface* is defined as the limit surface when a *subdivision scheme* is applied to a *control polyhedron*. Repeated subdivision refines the polyhedral shape, which converges to a piecewise continuous smooth surface.

A subdivision scheme is composed of two basic operations:

- *refinement*: changes the topology of the polygonal mesh by introducing new vertices, edges and faces. As a consequence, mesh density increases.
- *smoothing*: changes the geometry of the polygonal surface by modifying the coordinate values of its vertices. The end effect is an improvement of local shape regularity.

In the 1-dimensional setting (i.e. curves), these two operations are relatively simple to define, but in the 2-dimensional setting (i.e. surfaces), their definition is a lot more complex. This is because, while in 1D a discrete local neighborhood is always an interval, in 2D discrete neighborhoods depend on the topological structure of the underlying polygonal mesh.

Before going into more details, let's give some definitions:

The *first order neighborhood* $N_1(v)$, (or 1-neighborhood) of a vertex v in a mesh, consists of the set of vertices v_i that share a face with v . Higher order neighborhoods (or k -neighborhoods) are obtained from $N_1(v)$, by adding all 1-neighborhoods of the neighbors $v_i \in N_1(v)$, and so on.

In a triangular mesh, the 1-neighborhood of a vertex v consists of the vertices v_i linked to v by an edge. The set of edges incident to a vertex v is called *star* of v . In a triangular mesh, there is a correspondence between the 1-neighborhood and the star of a vertex.

The *valence* (or *degree*) of a vertex is the number of edges in its star (which in a triangular mesh is the number of 1-neighbors).

Subdivision schemes can be classified according to the mesh structure, as well as, the type of refinement and smoothing operators employed.

The mesh structure can be *regular* or *irregular*. A regular mesh is isomorphic to some uniform tiling of the plane by regular polygons. The possible cases are: equilateral triangles; squares; and hexagons. In a regular mesh, all vertices have the same valence (6 for triangular meshes and 4 for rectangular meshes). A irregular mesh is an affine cell complex. A vertex whose valence adheres to the corresponding regular case is called *ordinary*, otherwise it is called *extraordinary*. Note that regular meshes cannot represent shapes of arbitrary topology.

The refinement operator can apply a *primal* or *dual* decomposition method. In the primal decomposition a face is subdivided by splitting its edges and reconnecting them. The old mesh is contained in the new one. In the dual de-

composition new vertices are created inside the faces and connected across the edges to construct a dual mesh. The old mesh is discarded. The connectivity graph for mesh refinement can be described by a *subdivision template*.

The smoothing operator computes the new position of a vertex as a function of its neighbors. This function is a low-pass filter, and is usually linear (but can be non-linear). Also, it can be *uniform* or *non-uniform*¹. In the uniform setting, a fixed rule for each type of neighborhood is applied throughout the mesh. The filter coefficients relative to different neighborhood types are usually indicated by a *stencil* (or *mask*). In a non-uniform setting, the rule minimizes some measure of the curvature subject to additional constraints. The function is determined by an energy functional defined over the entire mesh. Thus, this method is called *variational*.

Depending on the constraints imposed on the smoothing operator, the subdivision scheme can be *approximating* or *interpolating*.

An important issue concerning subdivision surfaces is the analysis of convergence of a given subdivision scheme. This is paramount to characterize the smoothness of the limit surface.

For schemes on regular meshes, is often possible to obtain an expression of the continuous surface in closed form.

For schemes on irregular meshes using fixed templates and stencils, it is possible to employ the notion of stationary subdivision. A subdivision scheme is called *stationary* [3] when the refinement operator does not change the neighborhood structure of a vertex and the smoothing operator is a linear function uniform over the mesh.

Under stationary subdivision, for every vertex v , it is possible to find a matrix S , called *subdivision matrix*, that maps the coordinate values $p_j^i = p^i(v_j)$ of a neighborhood $N_k^i(v)$ of v at level i into an isomorphic neighborhood $N_k^{i+1}(v)$ at level $i + 1$, i.e., $p^{i+1} = Sp^i$, $i > 0$.

The action of the subdivision scheme is encoded in the subdivision matrices, and the eigenvalues of S reveal the surface regularity at v . In short, if the eigenvectors of S form a basis, and its eigenvalues satisfy $\lambda_0 = 1 > \lambda_1 \geq \lambda_2 > |\lambda_m|$, with $m > 2$, then the limit surface is of class C^1 , provided that the *characteristic map* induced by S is regular and injective [17].

For irregular meshes using variable templates or stencils (e.g. non-stationary subdivision), convergence analysis is still an open question. The characterization of such schemes is difficult because it strongly depends on the topology of the mesh at every level of refinement.

¹The term *uniform* here means that the smoothing function depends only on the local neighborhood structure, but not on the coordinate values of a vertex.

1.2 Previous Work

Subdivision surfaces started as a generalization of uniform splines. The very idea of a subdivision scheme draws upon *knot insertion* techniques [13], and have its roots on the “De Boor” algorithm [1]. Therefore, tensor product b-splines and box splines can be viewed as a special case of subdivision surfaces on regular meshes [4].

Nonetheless, the beginning of the field is identified with the development of the first subdivision surfaces for irregular meshes. Catmull and Clark [2], and Doo and Sabin [5], generalized bicubic and biquadratic B-splines, respectively, to rectangular meshes with irregularities. Later, Loop [15], created a generalization of quartic 3-direction Box splines to arbitrary triangular meshes. Hoppe, De Rose, et al. [9], proposed an extension of the Loop scheme for piecewise C^1 surfaces. Peters and Reif [16] introduced schemes that generalize quadratic 4-direction Box splines on irregular rectangular meshes. Taubin [19] proposed a non-shrinking filtering method for arbitrary triangular meshes.

All the schemes mentioned so far result in surfaces that approximate a control net. Dyn, Gregory and Levin [6], designed the “Butterfly” scheme for interpolating C^1 surfaces based on triangular meshes. Kobbelt [10] described a C^1 interpolating scheme for quadrilateral meshes with arbitrary topology.

A rigorous study of the convergence of subdivision schemes based on the characteristic map was proposed by Reif [17].

The stationary subdivision methods cited above still have a number of shortcomings. Variational subdivision schemes attempt to overcome such limitations. This issue have been addressed by Kobbelt [11] and by Warren [20].

1.3 Contribution

We remark that, although the refinement and smoothing operators of a subdivision scheme have been considered separately in our discussion, there is a strong interdependency between these two operators. This is due to the fact that the refinement operator constructs the neighborhood structure where the smoothing operator is applied. For regular meshes, this interdependency is not so relevant because the neighborhood structure is the same everywhere, as in the 1-dimensional setting. On the other hand, for irregular meshes this interdependency must be taken into account in the design of a subdivision scheme, so that convergence to a smooth surface is guaranteed.

Convolution schemes that generalize Box splines employ carefully designed smoothing rules which depend on the local neighborhood structure of a vertex. Variational schemes on irregular meshes employ a smoothing energy functional which depends on the global neighborhood structure and its geometry.

It is interesting to note that, in all existing schemes, most of the effort was concentrated on the design of the smoothing operator. In fact, almost all of them adopt the same standard refinement operators. One characteristic of these refinement operators is that they do not depend on the geometry of the net.

In this paper we describe a new subdivision scheme that uses a geometry-sensitive refinement operator in conjunction with a linear smoothing operator. Our scheme induces a quasi-stationary subdivision. It has a simple implementation and generates smooth surfaces that approximate the initial control mesh.

2 Background

Before going into the details of our subdivision scheme and its operators, we provide some background on the mesh structure adopted for surface representation.

2.1 4–8 Meshes

A *4–8 mesh* is a triangular mesh which has only vertices of valence 4 and 8. More formally, a 4–8 mesh is a 2D simplicial complex $K = (V, E, F)$, where V , E and F are respectively the sets of vertices, edges and faces of K . Moreover, V is divided into two classes $V = V_4 \cup V_8$, where $V_4 = \{v; \deg(v) = 4\}$, and $V_8 = \{v; \deg(v) = 8\}$.

A *regular 4–8 mesh* is a homogeneous simplicial complex in which the 1-neighborhood of every internal vertex of valence 4 has only neighbors of valence 8, and the 1-neighborhood of every internal vertex of valence 8 consists of a ring of vertices with alternating valences 4 and 8. (See Figure 1)

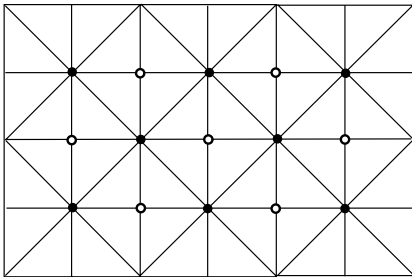


Figure 1: Regular 4–8 mesh: \circ - valence 4; \bullet - valence 8.

Note that in a regular 4–8 mesh, every internal face is formed by linking two vertices of valence 8 with one of vertex valence 4.

2.2 Refinement of 4–8 Meshes

A *refinement operator* for a regular 4–8 mesh $K = (V, E, F)$ is defined by the following procedure:

1. Split all edges $e = (v, w) \in E$ by inserting a *split vertex* $s_{vw} \in V'$, and connecting it to the endpoints $v, w \in V$ of e . That is, $e \mapsto \{e_v, e_w\}$, where $e_v = (v, s_{vw})$, $e_w = (s_{vw}, w)$, and $e_v, e_w \in E'$.
 2. Subdivide all faces $f \in F$ into four new faces by linking the vertex of degree 4, $u \in V_4$, to the split point s_{vw} of the opposite edge, and linking s_{vw} to the split points s_{uw} and s_{uv} of the remaining edges. That is, $f \mapsto \{f_w, f_{uw}, f_{uv}, f_v\}$, where $f = (u, v, w)$, $f_w = (w, s_{wu}, s_{vw})$, $f_{uw} = (u, s_{vw}, s_{wu})$, $f_{uv} = (u, s_{uv}, s_{vw})$, $f_v = (v, s_{vw}, s_{uv})$, and $f_w, f_{uw}, f_{uv}, f_v \in F'$.
- (The regularity property ensures that is possible to choose $u \in V_4$, and $v, w \in V_8$.)
3. Update the complex: $K' \mapsto K$, where $K' = (V \cup V', E', F')$.

The subdivision template corresponding to this procedure is illustrated in Figure 2.

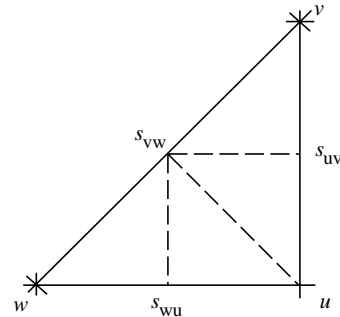


Figure 2: Subdivision template for 4–8 mesh.

It is easy to see that the above refinement operator preserves the 4–8 mesh regularity. After one refinement step, new vertices of valence 4 are inserted, old vertices of valence 4 change to valence 8, and old vertices of valence 8 remain unchanged. Therefore, the 1-neighborhood structure of valence 8 vertices is invariant under this operator.

An important property of this refinement operator is that it can be decomposed into two interleaved refinement sequences. This will be useful for the development of our quasi 4–8 refinement algorithm. The operator decomposition is as follows: the quaternary subdivision performed in one refinement step is replaced by nested binary subdivisions performed in two subsequent steps.

The interleaved refinement procedure is very similar to the normal one. The difference is that step 1 splits only edges connecting valence 8 vertices, and step 2 subdivides faces in two, accordingly. The regularity of the mesh guarantees that just one edge bisects in each face. Figure 3 compares the normal and interleaved refinement procedures.

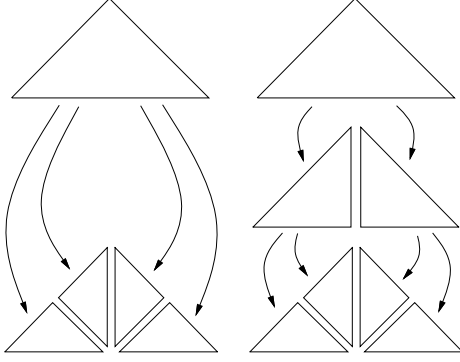


Figure 3: Normal and interleaved 4–8 refinement.

2.3 Planar 4–8 Meshes

The *quincunx* lattice is the set of points $Q = \{Mx; x \in \mathbb{Z} \times \mathbb{Z}\}$, where

$$M = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

is the *quincunx matrix*.

A *uniform 4–8 mesh* is a planar embedding of a 4–8 mesh in which the vertices of valence 4 and 8 belong, respectively, to two interleaved quincunx lattices. That is, $v \in V_8 \Rightarrow p(v) \in Q$, and $v \in V_4 \Rightarrow p(v) \in Q_1 = \{(1, 0) + Mx; x \in \mathbb{Z} \times \mathbb{Z}\}$.

Note that a uniform 4–8 mesh is regular, the main difference is that a regular 4–8 mesh is a topological complex, while the uniform 4–8 mesh has also a geometric realization on the plane.

Uniform 4–8 meshes are also known as four-direction meshes. These meshes are closely related with the 4 direction Box splines [21], that are generated from the set of four direction vectors $\{e_1, e_2, e_1 + e_2, e_1 - e_2\}$, where $e_1 = (1, 0)$ and $e_2 = (0, 1)$.

Since a uniform 4–8 mesh is regular, it can be refined using the same procedures described in subsection 2.2. On the other hand, the fact that a uniform mesh is embedded in \mathbb{R}^2 , makes it possible to exploit a geometric component in the design of the refinement operator. Note that the topological criteria for edge bisection can be replaced by a geometric criteria. A uniform 4–8 mesh has edges of length 1 (horizontal and vertical), and $\sqrt{2}$ (diagonals). Thus, in the case of uniform 4–8 meshes, it is easy to verify that an interleaved refinement procedure which splits the *longest* edges would produce the same results as the one using the topological criteria (i.e. edges with vertices of valence 8). This observation will be crucial in the design of our quasi 4–8 refinement operator.

The geometry based refinement of four direction meshes gives a very powerful multiresolution structure which has been exploited in many application areas. It has

been used for rendering terrain models with variable level of detail by Lindstron et al. [14] and Kirkpatrick et al.[7]. The sequence of refined meshes is known in the GIS community as *hierarchy of right triangles*.

3 The Scheme

In this section we describe the basic operation of our subdivision scheme and sketch an analysis of its convergence properties.

3.1 Refinement

As we have seen 4–8 meshes possess very nice properties but, unfortunately they cannot represent surfaces of arbitrary topology. This motivates us to look for a generalization of 4–8 meshes.

A quasi 4–8 mesh is a simplicial complex which has mostly vertices of valence 4 and 8, except for isolated vertices with some other valence. These vertices are *extraordinary*.

Obviously, the above characterization is only applicable to dense meshes. In fact, the real interest is to find a method to generate dense meshes with these properties from coarse meshes. This is our first goal.

A *quasi 4–8 refinement operator* is a transformation on simplicial complexes embedded in \mathbb{R}^n , such that its iterated application to an initial arbitrary (coarse) mesh will produce a quasi 4–8 mesh.

We now present an algorithm that implements this operator.

Algorithm 1: quasi_4–8_refinement(K)

```

sort_edges( $E$ )
store  $e \in E$  in priority queue  $Q$ 
while  $Q \neq \emptyset$  do
  get  $e$  from  $Q$ 
  if  $e$  not marked then
    split( $e$ )
    mark_cluster( $e$ )
  for all  $f \in F$  do
    subdiv( $f$ )

```

The routine `sort_edges`, sorts edges by decreasing length and radially around each vertex. The routine `mark_cluster` of edge e , marks e and the edges sharing a face with e . This is illustrated in Figure 4. The routine `subdiv`, performs a binary decomposition of a face by linking the split point of its longest edge to the opposite vertex. (See Figure 4.)

Note that, the longest side bisection gives the best aspect ratio of the triangles in a binary subdivision [18]. Also, edge cluster marking ensures that, at most, one edge in each faces splits.

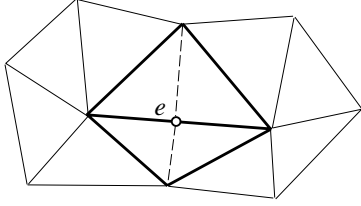


Figure 4: Cluster marking and subdivision.

The above remarks indicate that algorithm 1 produces a simplicial complex with compatible geometry. Now, it remains to be shown that the algorithm defines a quasi 4–8 refinement operator. Because the refinement procedure is geometry sensitive, the analysis also depends on the action of the smoothing operator. Consequently, we will postpone the discussion to the end of this section.

3.2 Smoothing

After the application of the refinement operator to the complex K , the set of vertices V of K , can be naturally divided into two classes: newly inserted vertices, $v' \in V'$, which we will call *new vertices*, and previously existing vertices, $v \in V$, which we will call *old vertices*.

The smoothing operator is a convolution filter. It uses a different smoothing rule for each class of vertices.

The stencil for new vertices is depicted in Figure 5. Recall that the 1-neighborhood of internal new vertices, by construction, consists of exactly 4 vertices. The filter function is an average of the coordinate values of these 1-neighbors.

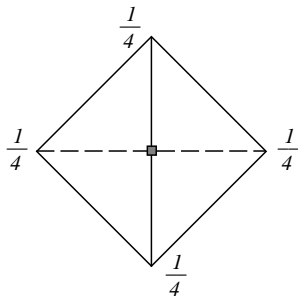


Figure 5: Filter mask for new vertex.

The stencil for ordinary old vertices of valence 8 is depicted in Figure 6. Observe that the filter kernel extends beyond the first neighbors of the vertex. Actually, the end effect is like using the 1-neighborhood structure of a quadrilateral mesh. Note also that this smoothing filter produces less shrinking than a gaussian filter, because more weight is put on the central vertex [19].

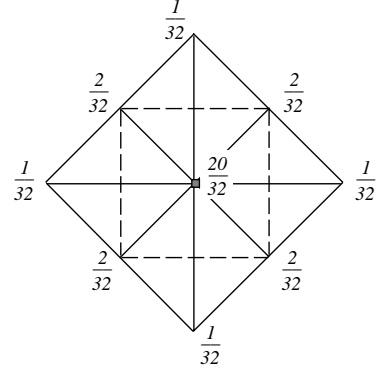


Figure 6: Filter mask for regular old vertex of valence 8.

The stencil for an ordinary old vertex of valence n is a direct generalization of the above filter, and it is depicted in Figure 7.

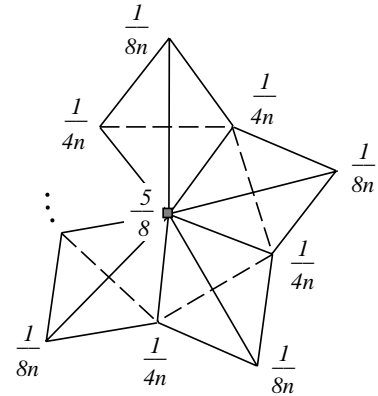


Figure 7: Filter mask for old vertex of valence n .

The smoothing operator can be implemented very efficiently using a cascade convolution technique.

First, filtering is applied to new vertices:

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

Then, filtering is applied to old vertices using new vertices:

$$\text{for } v_i \in V \text{ do} \\ p^{l+1}(v_i) = \frac{1}{2} p^l(v_i) + \frac{1}{2n} \sum_{v'_j \in N_1(v_i) \cap V'} p^{l+1}(v'_j)$$

where n is the number of new vertices in $N_1(v_i)$.

This scheduling makes it possible to execute all the computations “in-place”. Figure 8 illustrates the process.

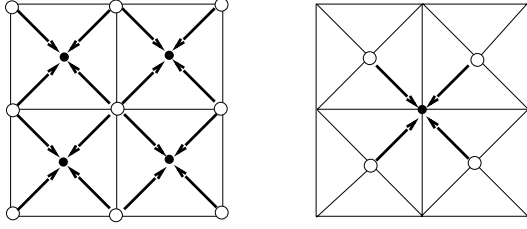


Figure 8: Computation flow of cascade convolution.

3.3 Quasi-Stationary Subdivision

The combination of refinement and smoothing described in this section leads to a subdivision scheme in which these two aspects are well integrated. The refinement operator is sensitive to geometry and produces a neighborhood structure suitable for filtering. The smoothing operator relaxes the mesh making it better for subdivision. We conjecture that this subdivision scheme is *quasi-stationary*. We also have strong indications that the scheme generates limit surfaces of class C^1 .

In order to prove quasi-stationarity and C^1 continuity there are, essentially, three conditions to verify: First, that algorithm 1 is indeed a refinement operator; Second, that the refinement is quasi-stationary; and finally, its convergence properties.

We will not give a complete proof in this paper. Instead, we will elaborate on some of the arguments that could lead to a formal demonstration.

The main condition imposed on a refinement operator is that its action increases the mesh density everywhere. This means that, for primal subdivision, all edges of the mesh at any given level must split in a small number of iterations. It is not difficult to verify that algorithm 1 satisfies such condition, since it splits edges based on their length.

Quasi-stationarity is more delicate to verify. We have to show that the refinement operator obey the following restrictions:

1. newly inserted vertices have valence 4.
2. old vertices of valence 4 change to valence 8.
3. old vertices of valence 8 do not change.
4. vertices of any other valence change little.

Restrictions 1 to 3 guarantee that vertices created during refinement are ordinary. Restriction 4 is admittedly vague, but it captures the idea that extraordinary vertices of the input mesh have a predictable neighborhood structure.

In order to continue the discussion, we will concentrate on planar subdivision. This is justified since, under smoothing, the geometry of the refined net becomes locally flat.

Consider a regular n -sided polygon $B \in \mathbb{R}^2$, inscribed in the unit circle. Triangulate B by connecting its vertices $b_i, i = 1, \dots, n$, to the center. Let's analyze the behavior of the subdivision algorithm on this triangulation.

When $n < 6$, the length of the boundary edges is greater than the length of the internal edges. Thus, the algorithm splits all boundary edges and the valence n of the central vertex becomes $2n$.

When $n > 6$, the length of the internal edges is greater than the length of the boundary edges. Thus, it is possible to pick a sequence of internal edges such that $\frac{n}{2}$ alternating edges split. If n is even, these will be the only vertices that split and the valence of the central vertex remains the same. (See Figure 12(a).) If n is odd, one boundary edge also splits and the valence of the central vertex increases by 1. (See Figure 12(c).)

After the first subdivision step, $n \geq 6$, and is even. Therefore, we only need to investigate this case in the rest of the analysis. But, since subdivision is based on geometry, we also have to take into account the effect of smoothing. We will focus on two adjacent triangles, (o, b_i^0, b_{i-1}^0) and (o, b_{i+1}^0, b_i^0) , whose common edge (o, b_i^0) splits. (See Figure 9.) The first subdivision step inserts vertex b_i^1 , which after smoothing will be the average of its four neighbors, $b_i^1 = \frac{1}{4}(o + b_{i-1}^0 + b_i^0 + b_{i+1}^0)$. In the second subdivision step, edges (o, b_{i-1}^0) and (o, b_{i+1}^0) are longer and will split.

As the process continues, by the very nature of the geometric criteria, subdivision occurs radially around the central vertex o . The vertices b_i^j introduced at subdivision steps, $j = 0, \dots, \infty$, approach o in geometric progression, as shown in Figure 9.

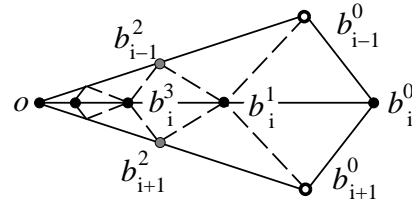


Figure 9: Convergence of a vertex under 4–8 subdivision.

The arguments just presented suggest that our subdivision scheme is quasi 4–8 stationary. We are currently studying the convergence properties of the method and hope to show that it produces C^1 surfaces.

4 Examples

In this section, we give some examples of surfaces generated with our subdivision scheme.

4.1 N-Regular Neighborhoods

The following examples illustrate more concretely the discussion in the previous section.

Figure 10 shows the subdivision of regular planar polygons with 3 to 6 sides. This is the first case in subsection 3.3. Note that for $n = 2, 3,$ and $4,$ the central vertex has valence $2n,$ as predicted by our analysis. Note also that, except for boundary vertices, all other vertices are ordinary and structured according to a regular 4–8 pattern. For the regular hexagon, which is the limit case, a small perturbation was added to the initial vertices of the polygon in Figure 10(d). As a consequence, the valence of the central vertex remained unchanged. Observe that the algorithm constructed the same subdivision as in the triangle of Figure 10(a).

Figure 11 reveals how the algorithm behaves in the case of a non-regular initial triangulation. The input mesh, shown in Figure 11(a) is the 6-regular triangulation of Figure 10(d), but warped such that the horizontal internal edges have a 1:2 length ratio. Note that, because the subdivision is based on geometry, the final triangulation adapts nicely to the polygonal domain. Since one of the horizontal edges is longer than the rest, it has split in the first subdivision step. The end result is that the final mesh in Figure 11(b), gradually transitions from a 4-regular structure (on the left hand side) to a 5-regular structure (on the right hand side).

Figure 12 exemplifies the subdivision of n -regular planar polygons for which $n > 6.$ It contrasts the behavior of the algorithm when the valence n is even and odd. Figure 12(a) shows the mesh for a 12-sided polygon after the first subdivision step. Note the radial decomposition structure around the central vertex. The final mesh, shown in Figure 12(b), has a quasi 4–8 symmetric structure. Figure 12(c) shows the mesh for a 9-sided polygon after the first subdivision. The decomposition cannot be completely radial, because n is odd. The final mesh, shown in Figure 12(d), has a quasi 4–8 structure, but exhibits a slight asymmetry.

Observe that in both cases, the triangulations of Figures 12(b) and (d) are somewhat distorted near the boundaries. They contain a few extraordinary vertices in that region, as well. This is, in part, due to the fact that boundary vertices are not allowed to move.

4.2 Simple Objects

The next examples show subdivision surfaces created from simple control meshes.

The first example is a closed surface of genus 0. The control mesh, shown in Figure 13(a), is a triangulated cube. The polygonal mesh after four subdivision steps is shown in Figure 13(b). Note that the only extraordinary vertices are the 8 corners of the original cube. Note also that there is very little distortion near these extraordinary vertices.

The second example is a genus 1 surface. The control mesh, shown in Figure 14(a), is a coarse approximation of a torus. The polygonal mesh after four subdivision steps is shown in Figure 14(b). Note that the mesh is 4–8 regular (i.e. it does not have extraordinary vertices). Note also that the resulting surface gives a good piecewise linear approximation of the torus.

4.3 Feature Control

The next example demonstrates the result of incorporating control of shape features in the subdivision scheme. It also shows the treatment of surfaces with boundary.

Boundary vertices are constrained to move along the surface normal direction.

As in other schemes we employ a tagged mesh for controlling features. Currently, we have implemented only the simplest kind of control mechanism, in which a tagged vertex or edge is not affected by smoothing. We plan to further investigate this issue, and experiment with different types of context sensitive smoothing filters.

The example in Figure 15 is a surface with boundary. The control mesh, shown in Figure 15(a), is a triangulated box shape with two open sides. The surface shown in Figure 15(b) was constructed by tagging all boundary edges and vertices. Note the smooth blend between the two open extremities. To produce the surface shown in Figure 15(c), boundary edges were untagged and one internal longitudinal edge was tagged. Note that, now, the boundary curves are smooth, except at the vertex shared with the tagged edge.

4.4 Complex Objects

The next examples deal with more complex shapes obtained from sampling real objects.

The first example is the well known “Stanford Bunny”. The control mesh, shown in Figure 16(a), was created with the mesh simplification software *Qslim*, developed by Garland and Heckbert [8]. Figure 16(b) shows the smoothed polygonal mesh after a few steps of our subdivision scheme. It is interesting to compare this mesh with the one in Figure 1 of [12], that was generated by smoothing the original dense mesh.

The final example is a cow model used in [8]. The control mesh shown in Figure 17(a) was also created by simplification. Figure 17(b) shows the resulting subdivision surface. Note that the shape is very smooth and, yet the main features are preserved.

5 Conclusions

In this paper we presented a new scheme for subdivision surfaces. It is based on quasi 4–8 regular meshes, and combines a geometry sensitive refinement operator with a convolution smoothing operator.

Overall, the scheme integrates effectively the two main operations and exploits the adaptivity of the mesh structure. The implementation is simple and efficient. It generates well-shaped meshes which converge to smooth surfaces. There is strong evidence that the resulting limit surfaces are piecewise C^1 continuous.

We have demonstrated the capabilities of the scheme through examples of different surfaces: with and without boundary; of arbitrary topological type; as well as, from shapes with variable level of complexity.

This scheme is an addition to the repertoire of modeling tools for subdivision surfaces. In the range of existing techniques, it lies halfway between stationary and variational subdivision methods.

The main novel ideas proposed in this paper are the introduction of the concept of quasi-stationary subdivision and mesh topology refinement based on geometric criteria.

References

- [1] W. Boehm. *Surfaces in Computer Aided Geometric Design*, chapter The de Boor algorithm for triangular splines, pages 109–120. North-Holland, 1982.
- [2] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput. Aided Design*, 10:350–365, 1978.
- [3] A. S. Cavaretta, W. Dahmen, and C. A. Micchelli. Stationary subdivision. *Memoirs of the AMS*, 12(453), 1991.
- [4] E. Cohen, T. Lyche, and R. Riesenfeld. Discrete box splines and refinement algorithms. *Computer Aided Geometric Design*, 1(2):131–148, 1984.
- [5] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Comput. Aided Design*, 10:356–360, 1978.
- [6] N. Dyn, D. Levin, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160–169, April 1990.
- [7] W. Evans, D. Kirkpatrick, and G. Townsend. Right triangular irregular networks. Technical Report 97-0, University of Arizona, 1997.
- [8] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH 97 Conference Proceedings*, pages 209–216, 1997.
- [9] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Proceedings of SIGGRAPH '94*, pages 295–302, 1994.
- [10] L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Computer Graphics Forum*, 15(3):409–420, 1996.
- [11] L. Kobbelt. A variational approach to subdivision. *Computer Aided Geometric Design*, 13(8):743–761, 1996.
- [12] L. Kobbelt, S. Campagna, J. Vorsatz, and H. P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH 98 Conference Proceedings*, pages 105–114, 1998.
- [13] 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.
- [14] 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.
- [15] C. Loop. Smooth subdivision for surfaces based on triangles. Master's thesis, University of Utah, 1987.
- [16] J. Peters and U. Reif. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics*, 16(4):420–431, October 1997.
- [17] U. Reif. A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design*, 12(2):153–174, 1995.
- [18] M. C. Rivara. Mesh refinement processes based on the generalized bisection of simplices. *SIAM J. Numer. Anal.*, 21(3):604–613, 1984.
- [19] G. Taubin. A signal processing approach to fair surface design. In *SIGGRAPH 95 Conference Proceedings*, pages 351–358, 1995.
- [20] J. Warren. Subdivision schemes for variational splines. preprint, 1997.
- [21] P. B. Zwart. Multivariate splines with nondegenerate partitions. *SIAM J. Numer. Anal.*, 10(4):665–673, 1973.

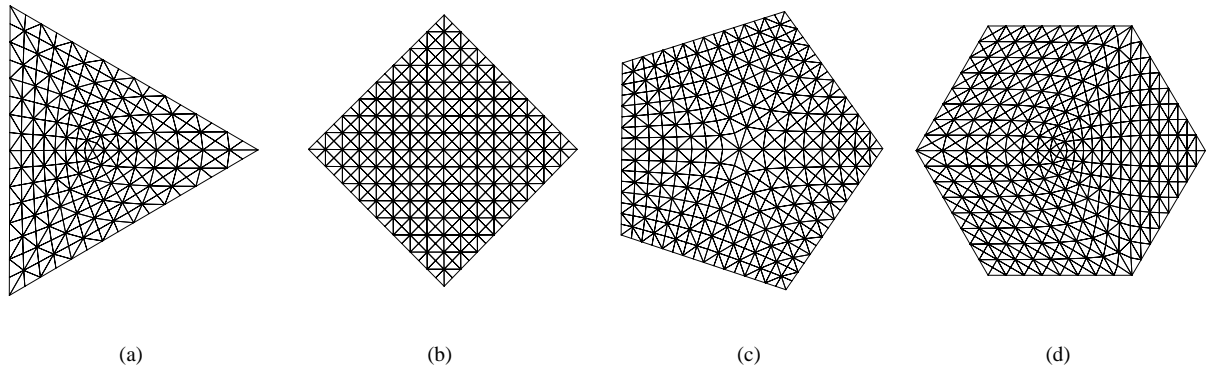


Figure 10: Subdivision of n -regular planar polygons.

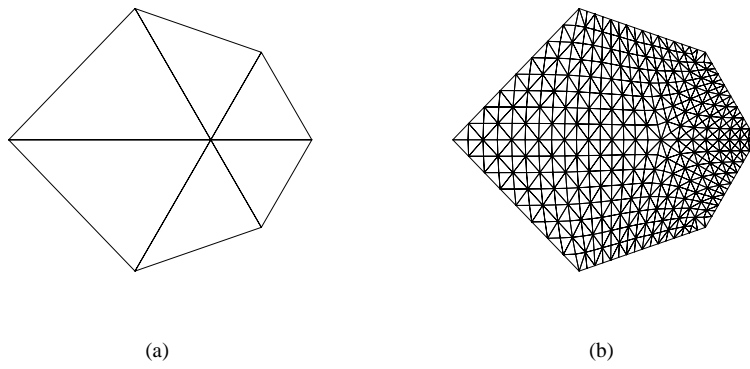


Figure 11: Subdivision of a warped hexagon.

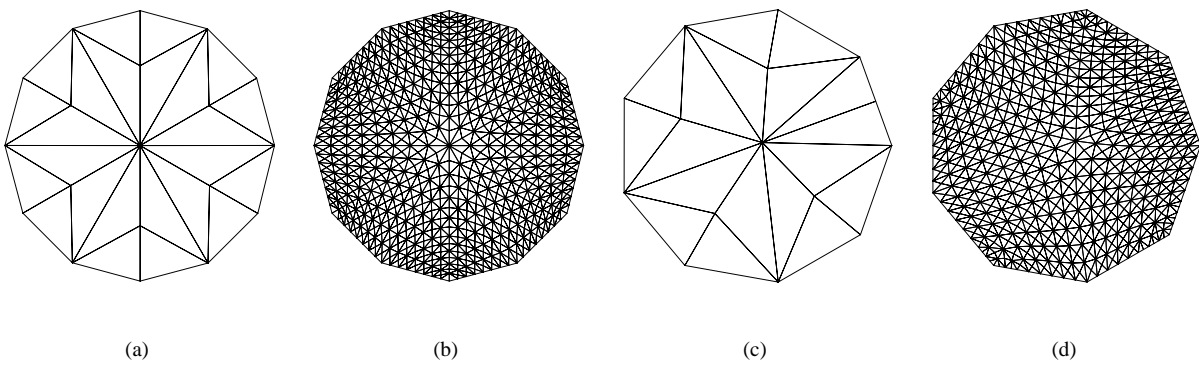


Figure 12: Subdivision of regular polygons with even (top row) and odd (bottom row) number of sides.

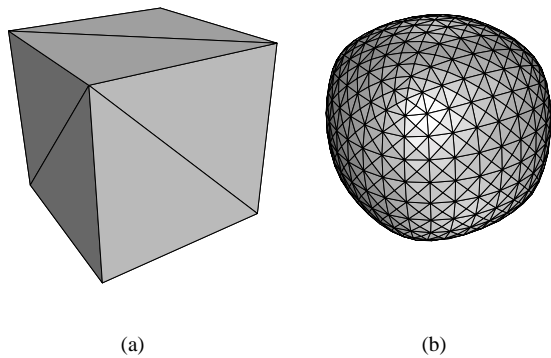


Figure 13: Surface generated from cube after 4 steps.

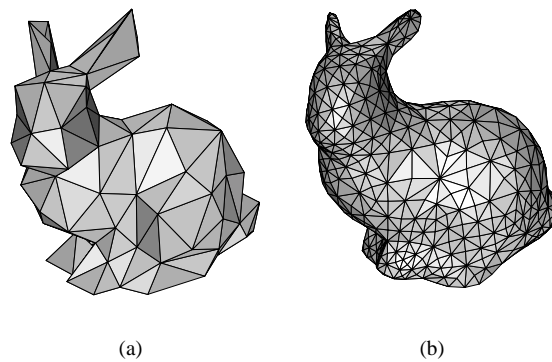


Figure 16: Bunny: (a) control net; (b) smooth net.

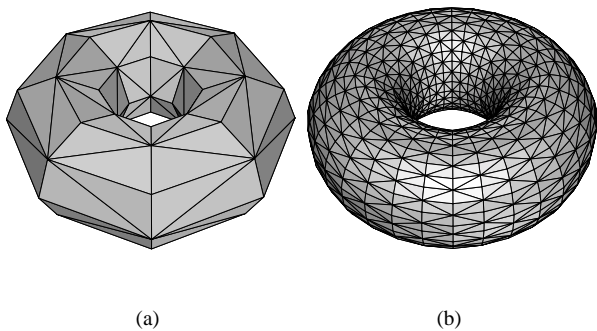


Figure 14: Surface generated after 4 steps from torus.

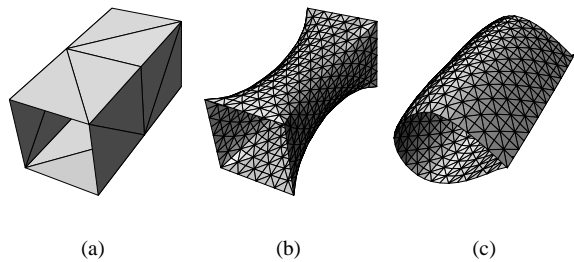
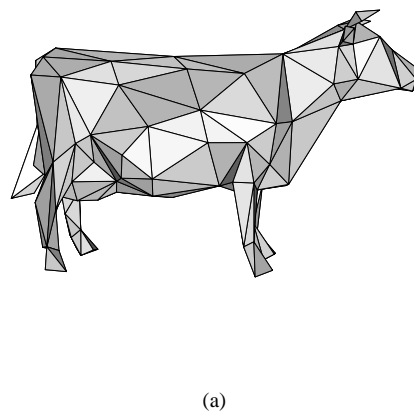


Figure 15: Control of features and boundary treatment

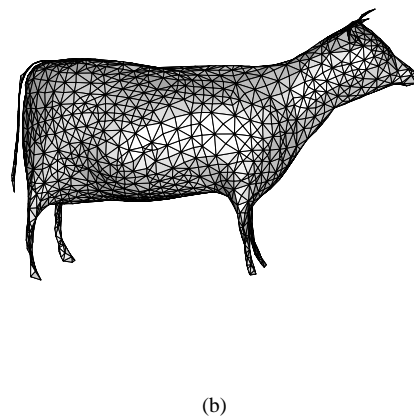


Figure 17: Cow: (a) control net; (b) smooth net.