EUROGRAPHICS 2000 / M. Gross and F.R.A. Hopgood
(Guest Editors)

Volume 19 (2000), Number 3

Interpolatory $\sqrt{3}$-Subdivision

U. Labsik G. Greiner

Computer Graphics Group
University of Erlangen-Nuremberg
Am Weichselgarten 9, 91058 Erlangen, Germany

Abstract

We present a new interpolatory subdivision scheme for triangle meshes. Instead of splitting each edge and performing a 1-to-4 split for every triangle we compute a new vertex for every triangle and retriangulate the old and the new vertices. Using this refinement operator the number of triangles only triples in each step. New vertices are computed with a Butterfly like scheme. In order to obtain overall smooth surfaces special rules are necessary in the neighborhood of extraordinary vertices. The scheme is suitable for adaptive refinement by using an easy forward strategy. No temporary triangles are produced here which allows simpler data structures and makes the scheme easy to implement.

1. Introduction

Subdivision allows to generate smooth surfaces from a given control mesh $\mathcal{M}_0$. With this coarse mesh a sequence of refined meshes $\mathcal{M}_1, \ldots, \mathcal{M}_n$ can be computed. In the limit this sequence of meshes converges to a continuous smooth surface. Each refinement step can be divided into two different aspects. First, a topological operation is performed. Therefore new vertices are added to the mesh and the triangles are split. Then the geometry of the mesh is changed by a smoothing operation. In order to converge to a smooth limit surface M_∞ the subdivision scheme has to satisfy certain necessary and sufficient conditions [1, 12, 11].

In a subdivision mesh $\mathcal{M}_{i+1}$ two different kinds of vertices can be distinguished. These are *even* vertices corresponding to the vertices of the mesh $\mathcal{M}_i$ and *odd* vertices which are newly inserted. Subdivision schemes can be classified in *approximating* [3, 4, 10] and *interpolating* [6, 15, 8] schemes. In an approximating scheme the positions of the even vertices of $\mathcal{M}_{i+1}$ are local averages of the vertices of $\mathcal{M}_i$. In general, approximating schemes produce smoother surfaces. But for many applications it is mandatory that the position of the vertices in the control mesh is not changed. Therefore interpolating schemes have to be used.

The most popular splitting operation for triangular meshes is a 1-to-4 split. A new vertex for every edge of the original mesh is computed and the vertices are triangulated so that one triangle of the mesh is split into four triangles of the refined mesh. This splitting operation is for example used in the Butterfly scheme [6] and in the Loop scheme [10]. In [9, 7] a new splitting operator was introduced. Here the refinement is done by a combination of vertex insertion and edge flipping.

In this paper we present a new subdivision scheme which uses this vertex insertion and edge flipping operator for the refinement of the mesh. This operation has the advantage that only three new triangles are generated out of one triangle in the coarser mesh. Therefore more different refinement levels can be computed within a prescribed mesh complexity.

Since the subdivision scheme is interpolatory, only the positions of new vertices have to be computed. As for other subdivision schemes we have to distinguish between topologically regular settings where all vertices of a triangles have *valence* 6 (this is the number of adjacent edges of a vertex) and topologically irregular settings. In this case *extraordinary vertices* (i.e. vertices with valence $\neq 6$). In the regular case for the computation of a new vertex a 12-neighborhood is needed. With the rule presented in this paper smooth surfaces over a regular triangular mesh can be generated. To avoid unwanted artifacts like creases and cusps in topologically irregular settings we modify this rule for vertices not having valence 6.

The paper is organized as follows. In Section 2 we give a short overview of well known interpolatory subdivision schemes, especially the Butterfly scheme. In Section 3 the splitting operator for uniform and adaptive subdivision is explained and in Section 4 the rules for the computation of new vertices are presented. The refinement of boundaries is shown in Section 5. In Section 6 we show examples produced with our new subdivision scheme. Finally, in Section 7 we give a conclusion.

2. Interpolatory Subdivision Schemes

The most famous interpolatory subdivision scheme for triangular meshes is the Butterfly scheme [6]. This scheme is a generalization of the 4 point subdivision scheme for curves [5]. The Butterfly scheme leads to smooth surfaces over topologically regular triangular meshes where all vertices have valence 6. In the Butterfly scheme a 8 point stencil is used to compute a new vertex, arranged in a configuration which this scheme is named after (cf. Figure 1).

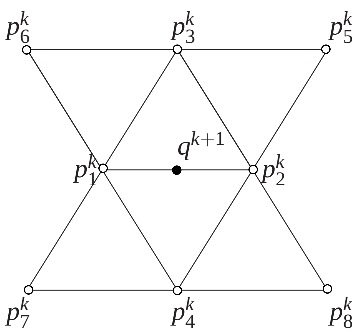

Figure 1: *Stencil of the Butterfly scheme.*

The position for a new edge point q^{k+1} is

$$q^{k+1} = \frac{1}{2}(p_1^k + p_2^k) + 2w(p_3^k + p_4^k) - w(p_5^k + p_6^k + p_7^k + p_8^k), \quad (1)$$

with a tension parameter w, normally set to $\frac{1}{16}$. Instead of a three dimensional mesh we now think of a scalar valued function over a three directional grid. When the function values are constant along one of these directions then the Butterfly scheme reduces to the 4 point scheme

$$q^{k+1} = (\frac{1}{2} + w)(p_i^k + p_{i+1}^k) - w(p_{i-1}^k + p_{i+2}^k) \quad (2)$$

along the other two directions. In the next section we will use this idea to construct an interpolatory rule for the $\sqrt{3}$-subdivision scheme.

The Butterfly scheme has got the advantage that it provides local rules to compute new vertex positions. No global set of equations has to be solved. But it exhibits unwanted artifacts in the neighborhood of extraordinary vertices with valences not equal to 6. Zorin et al. [16] have developed modified rules for these cases. With the modified Butterfly scheme overall C^1 continuous surfaces can be computed. A generalization of the 4 point scheme for subdividing quadrilateral meshes with arbitrary topology was proposed by Kobbelt [8].

3. $\sqrt{3}$-Subdivision

In this section we present the splitting operator used for our subdivision scheme. This operator was introduced in [7] for an approximatory subdivision scheme. We also show how this splitting operator can be used for adaptive subdivision.

3.1. Uniform Refinement

As described in the introduction the subdivision scheme presented in this paper does not use the normal 1-to-4 splitting operator where the mesh is refined by inserting one new vertex per edge. In our scheme in the middle of every triangle t_j of a mesh $\mathcal{M}_k$ a new vertex q_j is computed. This vertex is connected to the old vertices of the triangle. To achieve a regular mesh structure, all old edges of the mesh are now flipped. This means that an edge between two triangles t_{j_1} and t_{j_2} is removed and an edge between the new vertices p_{j_1} and p_{j_2} is inserted. This process is illustrated in Figure 2.

After two steps of the scheme every triangle is divided into 9 and every edge is split into 3. This property leads to the name $\sqrt{3}$-*Subdivision*. All vertices inserted into the mesh have valence 6 and the valence of old vertices is not changed. Therefore the number of extraordinary vertices is constant during the refinement process. The special kind of mesh topology obtained by applying the refinement operator uniformly is called *subdivision connectivity*.

Using this splitting operator the number of triangles in the mesh grows only by factor 3 in every refinement step instead of factor 4 with the normal 1-to-4 split. This slower growth of the mesh size allows the computation of more refinement levels until a prescribed mesh complexity is reached.

3.2. Adaptive Refinement

Uniform subdivision triples the number of triangles in the mesh. This leads to an exponential growth of the mesh with the number of refinement steps. To avoid this problem the technique of *adaptive refinement* is used. Here only triangles are split which do not satisfy a prescribed *flatness criterion*. This means that in areas of high local curvature more refinement levels are computed than in rather flat areas of the mesh.

When using the normal 1-to-4 split of triangles problems occur where different refinement levels of the mesh meet. Holes can be produced when one edge is split into a triangle but not in the neighboring one. This has to be fixed with the so-called *red-green triangulation* [2, 14, 13].

Green splits bisect a given triangle and are only temporary. If a green split triangle has to be further refined the split will be removed and a normal red split will be applied to the original triangle. This makes the data structures of the triangle mesh more complicated.

The problem of holes does not occur with the $\sqrt{3}$-subdivision scheme because no edges are split. A technique

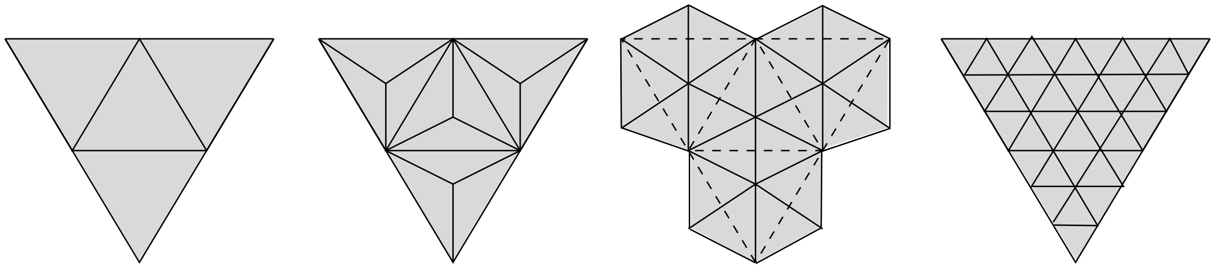

Figure 2: *Splitting of the mesh. One new vertex per triangle is computed, this vertex is connected with the vertices of the triangle and the edges between old vertices are flipped. After two refinement steps every triangle is split into 9 new triangles.*

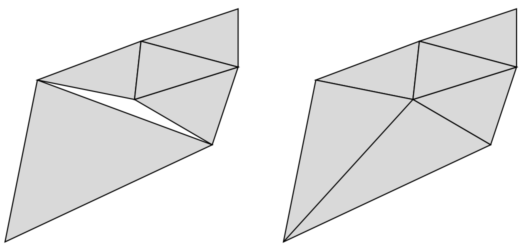

Figure 3: *Holes in the mesh has to be fixed by using a red-green triangulation.*

for the adaptive refinement with the vertex insertion and edge flipping operation used to refine the mesh was introduced in [9,7]. This technique can also be applied to the interpolating version of the $\sqrt{3}$-subdivision scheme.

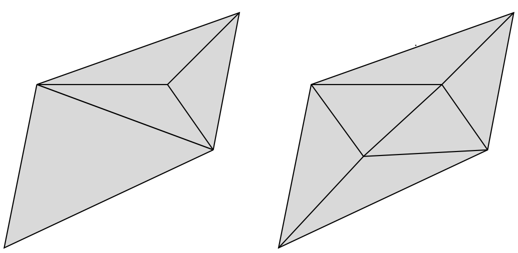

Figure 4: *Forward adaptive refinement.*

A triangle is subdivided by inserting a vertex into a triangle and connecting this new vertex to the old vertices of the triangle. If a neighboring triangle is already refined the edge between the triangles is removed and a new edge between the face points of both triangles is inserted, i.e. the old edge will be flipped . This process is illustrated in Figure 4. A triangle can only be further refined if it has been generated by an edge flipping operation.

To compute a new face point all stencil vertices have to be collected. If some of them are missing they have to be produced recursively by refining triangles in the neighborhood.

By this approach a balancing of the mesh is achieved. In fact, it is not possible that triangles with a level difference greater than one are neighbors.

4. Stationary subdivision rules

In this section we introduce the smoothing rules of the new interpolatory $\sqrt{3}$-subdivision scheme for the computation of new vertices. Here we have to distinguish between topologically regular and irregular settings.

4.1. Regular meshes

In [9,7] this splitting operator was used for an approximatory subdivision scheme. Two rules were used, one for the computation of a new vertex as the average of the vertex positions of a triangle and a second for the new position of the already existing vertices.

In our scheme we want to develop rules for an interpolatory subdivision scheme. Hence, we only need one rule for the computation of a new vertex. The stencil of the subdivision scheme has to be symmetric and should be as small as possible. We suggest the stencil shown in Figure 5 defining a 12-neighborhood around the triangle.

Because of the symmetry of the shown stencil the rule must have the form

$$\begin{aligned} q^{k+1} &= a(p_1^k + p_2^k + p_3^k) + b(p_4^k + p_5^k + p_6^k) \\ &\quad + c(p_7^k + p_8^k + p_9^k + p_{10}^k + p_{11}^k + p_{12}^k). \end{aligned} \tag{3}$$

We think again of a scalar valued function over a three directional grid with constant function values along one of the directions. In contrast to the Butterfly scheme we do not insert the midpoint between two of the grid lines. The new vertex divides the distance between two lines with a ratio of 2 to 1. Therefore the weights along the other directions cannot be reduced to the normal 4 point scheme.

For the construction of the weights in the one dimensional case we have to solve the interpolation problem for four given points $f(0), f(1), f(2), f(3)$. These four points

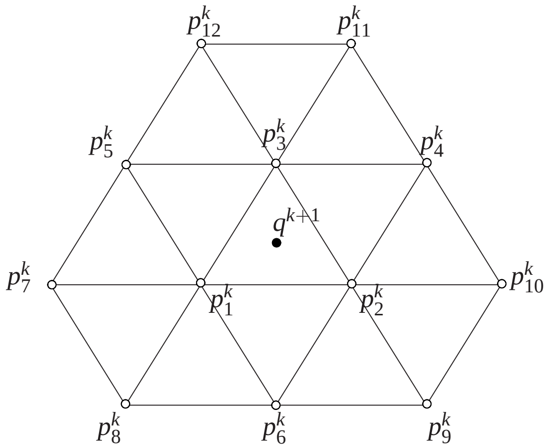

Figure 5: *Stencil for the computation of a new vertex.*

uniquely define a cubic interpolation polynomial $p_3(x)$. By evaluating this polynomial at the position $x = \frac{5}{3}$ the weights for a subdivision scheme can be computed. We get

$$p(\frac{5}{3}) = -\frac{4}{81}f(0) + \frac{10}{27}f(1) + \frac{20}{27}f(2) - \frac{5}{81}f(3). \quad (4)$$

We want our subdivision scheme to be a generalization of this equation. Hence, we compute the weights $,a,b,c$ so that

$$-\frac{4}{81} = 2c \quad (5)$$

$$\frac{10}{27} = a + 2b \quad (6)$$

$$\frac{20}{27} = 2a + 2c \quad (7)$$

$$-\frac{5}{81} = b + 2c. \quad (8)$$

These equations have the solution $a = \frac{32}{81}$, $b = -\frac{1}{81}$ and $c = -\frac{2}{81}$ and consequently the scheme is

$$\begin{aligned} q^{k+1} = \frac{32}{81}(p_1^k + p_2^k + p_3^k) - \frac{1}{81}(p_4^k + p_5^k + p_6^k) \\ -\frac{2}{81}(p_7^k + p_8^k + p_9^k + p_{10}^k + p_{11}^k + p_{12}^k). \end{aligned} \quad (9)$$

Another way of finding these weights is to compute a bicubic interpolation polynomial for the 12 points of the stencil in a regular setting. Evaluating this polynom at the midpoint of the central triangle leads to the same formula.

It is obvious that this scheme leads to a C^0 continuous surface because the sum of the weights $3a+3b+6c$ is 1. We do not want to give a mathematical rigorous proof of the C^1 continuity of the scheme for a regular mesh here.

The analysis of the *subdivision matrix* we will present in the following provides a necessary condition for the convergence of the subdivision scheme to a C^1 continuous surface. The subdivision matrix **S** describes how a certain region of the mesh $\mathcal{M}_i$ is mapped to a "scaled" region in the mesh $\mathcal{M}_{i+1}$. In our case the smallest region which is mapped to a corresponding region in a finer level is a 37-neighborhood of a given vertex. The coefficients of **S** are determined by the weights from equation 9.

For the applied refinement operator it is not possible to directly use this 37×37 matrix. This is because of a 30° rotation being performed when splitting the triangles. Therefore we subdivide the mesh twice leading to a 60° rotation which is corrected by resorting the vertices. This is done by multiplying the matrix with a permutation matrix **R**. So we have to analyze the matrix

$$\tilde{\mathbf{S}} = \mathbf{RSS} \quad (10)$$

The behavior of the subdivision scheme is specified by the leading eigenvalues of the matrix $\tilde{\mathbf{S}}$. From [12] it is known that $\tilde{\mathbf{S}}$ can only be a convergent subdivision scheme leading to a C^1 continuous surface if the leading eigenvalues are

$$\lambda_0 = 1, \lambda_1 = \lambda_2 = \frac{1}{3} \quad (11)$$

and $|\lambda_i| < \frac{1}{3}$ for $i = 3, \ldots 36$. This condition is satisfied for the presented scheme.

4.2. Modifications for extraordinary vertices

The subdivision rule presented above leads to a smooth surface for topologically regular settings. But it can easily be shown that this rule is not sufficient to achieve smoothness also for the case of extraordinary vertices. The same problem also occurs for the Butterfly scheme. So special rules have to be applied for the subdivision near extraordinary vertices [15]. This will be explained in the following.

For the computation of special rules for irregular settings we consider the mesh not as vertices in the $\mathbb{R}^3$ but as a planar region with height values for every vertex. First we study the case for the computation of a new vertex in a triangle where one vertex has valence $k \neq 6$ and the others have valence 6. This configuration is illustrated in Figure 6.

We set up the subdivision rules by examining the necessary conditions for a convergent subdivision scheme as found in [12]. In order to keep the subdivision rules as simple as possible we only use the vertices in the 1-neighborhood of p^k for the computation of the new vertex q^{k+1}. So a new vertex is computed as

$$q^{k+1} = \alpha p^k + \sum_{i=0}^{n-1} \alpha_i p_i^k \quad (12)$$

For the given configuration we can build the local subdivision matrix S. If the valence of the vertex p^k is n, this is a

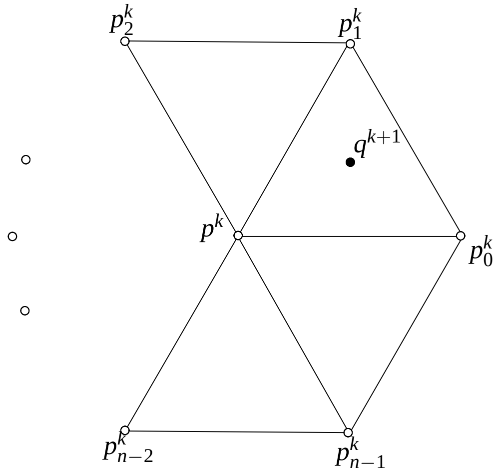

Figure 6: *One vertex in the triangle has valence $k \neq 6$, the others have valence 6.*

$(n+1)\times(n+1)$ matrix. For $n=5$ the matrix S is given by

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha & \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \\ \alpha & \alpha_5 & \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \alpha & \alpha_4 & \alpha_5 & \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \\ \alpha & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_0 & \alpha_1 & \alpha_2 \\ \alpha & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_0 & \alpha_1 \\ \alpha & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_1 \end{pmatrix} \tag{13}$$

As in the regular case we cannot directly use this matrix for the analysis of the subdivision scheme but have to take a matrix $\tilde{S} = RSS$. Again, this matrix represents two refinement steps of our scheme. With the help of a sophisticated eigen analysis of $\tilde{S}$ we find that for a vertex p with valence $n \geq 5$ the weights for such a double step are

$$\tilde{\alpha} = \frac{8}{9} \tag{14}$$

$$\tilde{\alpha}_i^n = \frac{\frac{1}{9} + \frac{2}{3}\cos(\frac{2\pi i}{n}) + \frac{2}{9}\cos(\frac{4\pi i}{n})}{n}, \tag{15}$$

for $n=3$ we have to use $\tilde{\alpha} = \frac{8}{9}, \tilde{\alpha}_0^3 = \frac{7}{27}, \tilde{\alpha}_1^3 = \tilde{\alpha}_2^3 = \frac{-2}{27}$ and for $n=4$ we take $\tilde{\alpha} = \frac{8}{9}, \tilde{\alpha}_0^4 = \frac{7}{36}, \tilde{\alpha}_1^4 = \tilde{\alpha}_3^4 = \frac{1}{27}, \tilde{\alpha}_2^n = \frac{-5}{36}$.

With these weights the leading eigenvalues of $\tilde{S}$ for $n \geq 5$ are

$$\lambda_0 = 1, \lambda_1 = \lambda_2 = \frac{1}{3}, \lambda_3 = \lambda_4 = \lambda_5 = \frac{1}{9}. \tag{16}$$

For the subdivision algorithm we now need to find the weights of the matrix S, which is a suitable square root of S^2. The square root is determined by an eigenvector analysis of S^2. For the implementation of the subdivision scheme we have precomputed the weight coefficients to make the algorithm faster.

After one subdivision step there are only triangles with at least two regular vertices and at most one irregular vertex with valence $n \neq 6$. In this case the face point in such a triangle is computed with the modified rule. If all vertices of the triangle have valence 6 the normal regular rule is applied. Only in the coarsest level of the mesh triangles with two or more irregular vertices may exist. In this case we compute the face point of the triangle for every irregular vertex and take the average as the resulting new vertex.

5. Boundaries

When meshes with boundaries are subdivided two problems occur. First, there has to be a special subdivision rule to compute a smooth boundary curve. Second, for the computation of new vertices in triangles which have at least one vertex on the boundary of the mesh not all stencil points for the subdivision rule exist.

5.1. Subdividing the boundary polygon

With the normal 1-to-4 split edges on the boundary curve have to be split in two parts. For the butterfly scheme the interpolatory 4-point scheme is used. In our case a boundary edge is only divided in every second refinement step and split into three parts (cf. Fig. 7). Hence, two vertices have to be inserted. Therefore we cannot use the normal 4-point scheme and have to find a modification.

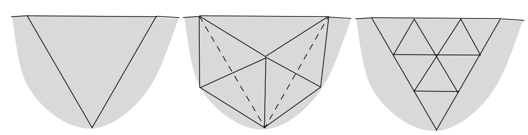

Figure 7: *Subdividing boundary triangles.*

For the computation of new boundary vertices no inner vertices of the mesh should be used. The advantage is that two different meshes with a common boundary polygon can be divided separately and the subdivided meshes still share a common boundary curve. This enables the generation of cusps and creases.

For the splitting of a boundary edge into three parts we can use equation (4) resulting in a modified 4 point scheme where two vertices are inserted in one step. The subdivision rules are

$$p_{3i-1}^{k+1} = -\frac{4}{81}p_{i-2}^k + \frac{10}{27}p_{i-1}^k + \frac{20}{27}p_i^k - \frac{5}{81}p_{i+1}^k \tag{17}$$

$$p_{3i}^{k+1} = p_i^k \tag{18}$$

$$p_{3i+1}^{k+1} = -\frac{5}{81}p_{i-1}^k + \frac{20}{27}p_i^k + \frac{10}{27}p_{i+1}^k - \frac{4}{81}p_{i+2}^k \tag{19}$$

and the configuration of the stencil points can be found in Fig. 8.

These subdivision rules lead to a C^1 continuous boundary

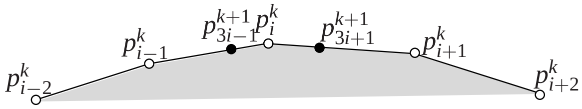

Figure 8: *Stencil points for boundary subdivision.*

curve. This can be shown by analyzing the eigenvalues of the 7×7 subdivision matrix for the invariant neighborhood of the scheme. The eigenvalues are

$$\lambda_0 = 1, \quad \lambda_1 = \frac{1}{3}, \quad \lambda_2 = \frac{1}{9} > \lambda_3 > \cdots > \lambda_6. \qquad (20)$$

5.2. Triangles near the boundary

To compute the face point of a triangle all stencil points have to be collected. For triangles near the boundary some of them may not exist. This happens when one or more vertices of the triangle are on the boundary of the mesh. In this case virtual points are computed which lie outside of the mesh.

The easiest way to do this is to reflect vertices across the boundary of the mesh. We compute the 'virtual' points by only using the vertices of the triangle which shall be splitted. A typical configuration is shown in Figure 9.

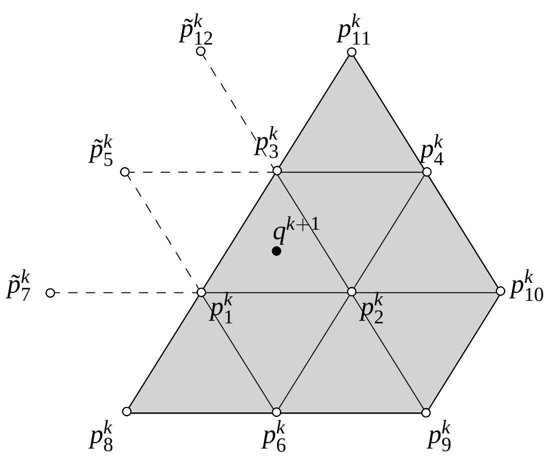

Figure 9: *Reflecting vertices across the boundary of the mesh.*

In the given example the virtual points $\tilde{p}_5^k$ and $\tilde{p}_7^k$ are computed by $\tilde{p}_5^k = p_1^k + p_3^k - p_2^k$ and $\tilde{p}_7^k = p_1^k + p_1^k - p_2^k$.

With the help of the virtual points the normal regular subdivision rule can be used if the inner vertices of the triangle have valence 6. This will be the case for all triangles after the first refinement of the mesh. If the inner vertex of the triangle has a valence not equal to 6 the rule for extraordinary vertices is used. This rule does not need any virtual points.

6. Examples

We have applied our subdivision scheme to a number of examples. The first one is a simple diamond model with a vertex with valence 12 on the top (cf. Fig. 10). It can be seen that both the modified butterfly scheme and the new interpolatory $\sqrt{3}$-subdivision scheme are not able to completely avoid the creases on the surface. These artifacts are a consequence of the interpolation with with smoothing rules which are not adapted to the local geometry.

In the next example we applied the $\sqrt{3}$-subdivision scheme adaptively for the refinement of a cat model. We used a curvature depend adaptive refinement strategy based on the dihedral angle between two adjacent triangles. The results are shown in Figure 11. The left picture shows the original data set, in the middle an angle of 20° is allowed and in the right the angle is reduced to 10°. In regions of high curvature more refinement levels are computed than in rather flat areas. The original dataset consists of 728 triangles, the refined meshes have 9358 and 34228 triangles. In comparison with the modified butterfly scheme this is a reduction by about 30%, where the refined meshes with the same flatness criteria have 13364 and 49908 triangles.

In a third example in Figure 12* we show the subdivision of a mannequin head model (courtesy University of Washington). The original data set consists of 1355 triangles and was subdivided four times with our scheme. As in [15] we obtained the coarse mesh by mapping each vertex onto the boundary surface being produced by the Loop scheme. The figure clearly shows that the mesh converges to a smooth surface.

7. Conclusions

We have presented an effective interpolating subdivision scheme for triangle meshes. It produces smooth surfaces in the regular setting where all vertices have valence 6 by using a local stencil with 12 points. For the case that in a triangle not all vertices are regular we have computed special rules with minimal support leading to overall smooth surfaces.

With the special refinement operator we use, only three new triangles are computed out of a coarse triangle. Therefore we reduce the growth of the mesh size as compared to the normal 1-to-4 split used by nearly every other subdivision scheme for triangular meshes and more refinement levels can be computed within a prescribed mesh complexity.

When subdividing the mesh adaptively the refinement operator has got the advantage that no red-green triangulation is necessary where different refinement levels of the mesh meet. This allows the use of simpler data structures for storing the mesh which makes the implementation of the scheme easier.

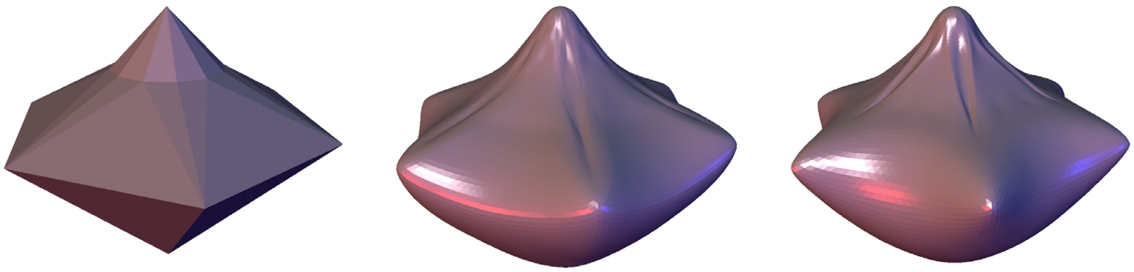

Figure 10: *Subdividing a simple diamond model with a valence 12 vertex (left), subdivided with the modified Butterfly scheme (middle) and the interpolatory* $\sqrt{3}$*-subdivision scheme.*

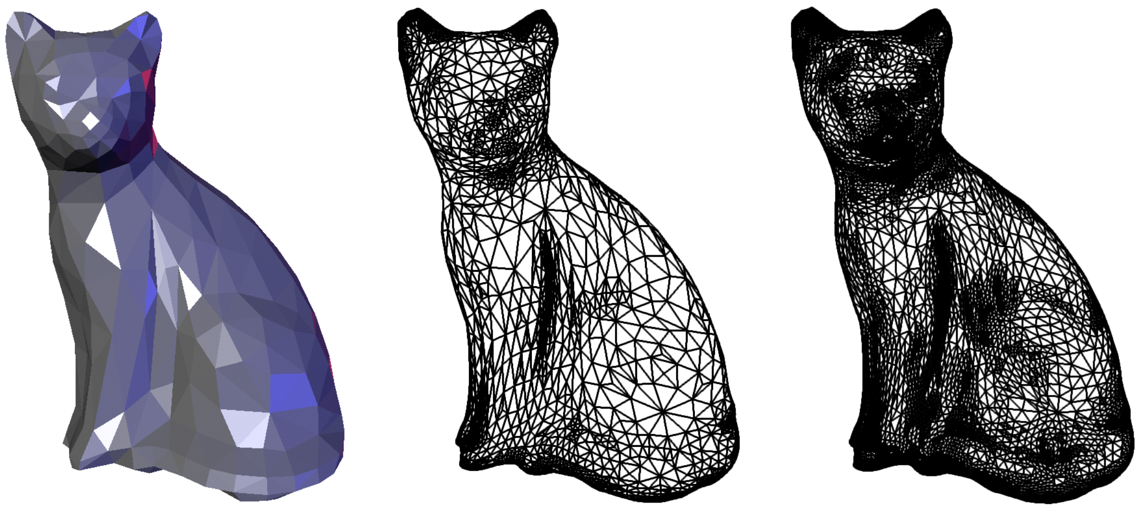

Figure 11: *Adaptive subdivision of the cat model. From left to right: original data set and adaptive refined meshes allowing a dihedral angle of* 20° *and of* 10°.

Acknowledgments

We are grateful to Leif Kobbelt for directing our attention to $\sqrt{3}$-subdivision schemes and for several helpful hints.

References

1. A. Ball, and D. Storry: *Conditions for Tangent Plane Continuity over Recursive Generated B-Spline Surfaces*, pp. 83–102. ACM Trans. Graph 7., 1988. 1
2. R. E. Bank, A. H. Sherman, and A. Weiser: Refinement Algorithms and Data Structures for Regular Local Mesh Refinement. In *Scientific Computing*, R. Stepleman (ed.), pp. 3–17. IMACS/North Holland, Amsterdam, 1983. 2
3. E. Catmull, and J. Clark: *Recursively generated B-spline surfaces on arbitrary topological meshes*, pp. 350–355. CAD 10, 1978. 1
4. D. Doo, and M. Sabin: Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design 10*, pp. 356–360, 1978. 1
5. N. Dyn, D. Levin, and J. Gregory: *A 4-point interpolatory subdivision scheme for curve design*, pp. 257–268. CAGD 4, 1987. 2
6. N. Dyn, D. Levin, and J. Gregory: *A Butterfly Subdidvision Scheme for Surface Interpolation with Tension Control*, pp. 160–169. ACM Trans. Graph. 9, 1990. 1, 2
7. L. Kobbelt, U. Labsik, and H.-P. Seidel: $\sqrt{3}$-subdivision and forward adaptive refinement. In *Proceedings of Korea-Israel Bi-National Conference on Geometrical Modeling and Computer Graphics in the World Wide Web Era*, Korea, September 1999. 1, 2, 3
8. L. Kobbelt: Interpolatory subdivision on open

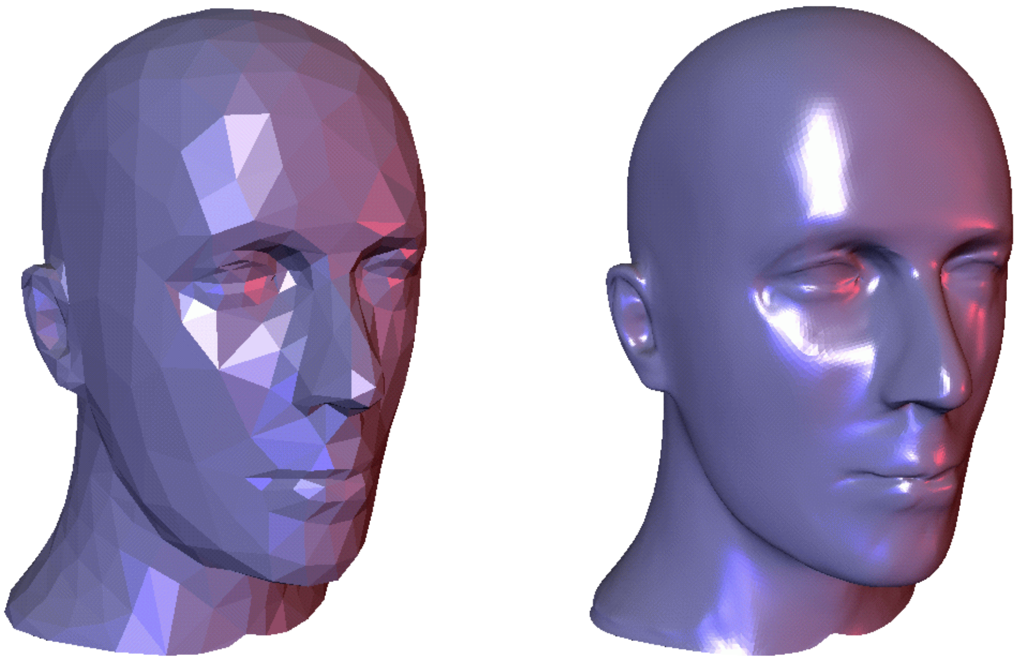

Figure 12: *Using interpolatory $\sqrt{3}$-subdivision for refining the mannequin head.*

quadrilateral nets with arbitrary topology. *Computer Graphics Forum 15(3)*, pp. C409–C420, C485, Sept. 1996. 1, 2

9. U. LABSIK: Restrukturierung und progressive Übertragung polygonaler Netze. Master's thesis, University of Erlangen-Nuremberg, 1999. 1, 3
10. C. LOOP: Smooth sudivsion surfaces based on triangles. Master's thesis, Utah University, 1987. 1
11. H. PRAUTZSCH: Smoothness of subdivision surfaces at extraordinary points. *Adv. Comp. Math. 14*, pp. 377–390, 1998. 1
12. U. REIF: *A unified approach to subdivision algorithms near extraordinary vertices*, pp. 153–174. CAGD 12, 1995. 1, 4
13. M. VASILESCU, AND D. TERZOPOULUS: Adaptive meshes and shells: Irregular triangulation, discontinuities, and hierarchical subdivision. In *Proceedings of Computer Vision and Pattern Recognition conference*, pp. 829–832, 1992. 2
14. R. VERFÜRTH: *A review of a posteriori error estimation and adaptive mesh refinement techniques*. Wiley-Teubner, 1996. 2
15. D. ZORIN, P. SCHRÖDER, AND W. SWELDENS: Interpolating subdivision for meshes with arbitrary topology. In *ACM Comp. Graph. (SIGGRAPH '96 Proc.)*, pp. 189–192, 1996. 1, 4, 6
16. D. ZORIN, P. SCHRÖDER, AND W. SWELDENS: Interpolating subdivision for meshes with arbitrary topology. Tech. Rep. CS-TR-96-06, Caltech, Department of Computer Science, 1996. 2