Vol. 14, No. 2: 61–76

Removing Polar Rendering Artifacts in Subdivision Surfaces

Ursula H. Augsdörfer and Neil A. Dodgson
University of Cambridge

Malcolm A. Sabin
Numerical Geometry Ltd.

Abstract. A polar artifact occurs in subdivision surfaces around high valency vertices. It manifests as large polygons in an otherwise finely subdivided mesh. It is particularly noticeable in subdivision schemes that have been tuned to improve the appearance and behaviors of the limit surface. Using the *bounded curvature Catmull-Clark scheme* as an example, we describe three practical methods by which this rendering artifact can be removed, thereby allowing us to benefit from the improved character of such tuned schemes.

1. Problem Statement

A subdivision surface is obtained by iteratively refining a coarse polygon mesh referred to as a control mesh. Most control meshes include extraordinary vertices, at which either more or fewer than the regular number of edges meet, referred to as valency. A number of subdivision schemes are available with Catmull-Clark [Catmull and Clark 78] and the Loop algorithm [Loop 87] being the most commonly used schemes for quadrilateral and triangular meshes, respectively. A typical subdivision process is linear and can be expressed as a matrix multiplication with existing vertices to obtain a new, denser set of vertices. The refinement matrix is referred to as the subdivision matrix. The local subdivision matrix varies, depending on the valency and the subdivision algorithm used.

© A K Peters, Ltd.

1086-7651/09 $0.50 per page

Eigenanalysis of the subdivision matrix provides us with information about the subdivision surface in the limit. In regular regions, all good binary subdivisions schemes have a subdivision matrix that has a subdominant eigenvalue $\lambda = 1/2$, which implies that edges in these regions reduce by half with every subdivision step. At extraordinary points with larger than regular valency, tuning of the scheme may require a λ that is very different than $1/2$. Such tuning should ideally be done with respect to the mathematical properties of the limit surface, not in relation to any polygonal approximation. Indeed, certain desirable properties, particularly bounded curvature, are only possible with λ greater than $1/2$ [Augsdörfer et al. 06]. The practical outcome of having λ greater than $1/2$ is that the polygons around an extraordinary point do not shrink in size as quickly as those in regular regions of the mesh, and that the extraordinary vertex and its neighbors take longer to reach their limit position [Sabin and Barthe 03].

A typical application will want to subdivide until every polygon is roughly the size of a pixel on screen, but no further. For an initial mesh of 100 polygons displayed on a 1000×1000-pixel screen, roughly seven subdivision steps are needed. If there is an extraordinary vertex around which $\lambda = 0.9$ then, after seven subdivision steps, while the majority of polygons will have edge lengths on the order of a pixel, the edges radiating from the extraordinary vertex will be of the order of $(0.9/0.5)^7 \approx 60$ pixels long. As a result the majority of the surface will appear smooth while the region around the extraordinary vertex appears faceted. An example can be seen in Figure 1.

Due to the polar artifact, there is a belief that subdivision schemes require the subdominant eigenvalue λ to be the same around extraordinary vertices as in the regular regions of the mesh [Barthe and Kobbelt 04, Zulti et al. 06, Ni and Nasri 06]. By constraining the tuning of subdivision schemes to solutions that fulfill this condition, we may prevent ourselves from finding the optimal limit surface [Zorin and Schröder 00, pp. 95–97].

We show that the polar artifact is purely a rendering artifact caused by the particular polygonal approximation and that it does not reflect the quality of the underlying limit surface, which itself is smooth. Simply rendering the polyhedron after k subdivision steps (Figure 1(d)) does not provide a sufficiently good approximation to the limit surface for reasonable values of k. Smoothing the subdivided surface and using Phong shading may improve the appearance (see Figure 1(e)), but this does not solve the problem as can be clearly seen when applying certain textures or shaders to the smoothed surface (see Figure 1(f)).

We need to make clear that we *are* interested in values of λ that are closer to 1.0 than to 0.5, for high valency extraordinary vertices. Augsdörfer's tuned Catmull-Clark scheme [Augsdörfer et al. 06] and Cashman's NURBS with extraordinary points [Cashman et al. 09] both use high λ values to get high quality limit surfaces around extraordinary vertices. For the tuned Catmull-

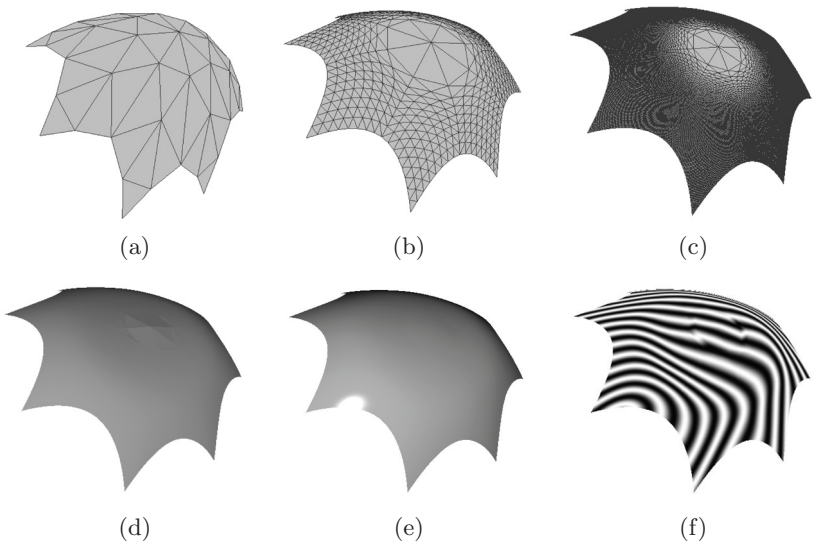


Figure 1. Bounded curvature Catmull-Clark subdivision: (a) The control polygon with a vertex of valency 8 at its center has edges of approximately the same length everywhere. (b) The control polygon has been subdivided three times using a tuned algorithm. The polar artifact appears: long edges arise around extraordinary vertices with a higher than regular valency. (c) The control polygon has been subdivided seven times. While in the regular region the edges are the size of a pixel, edges emanating from the extraordinary vertex are about 60 pixels long. (d) The polar artifact is clearly visible if the subdivided mesh is rendered naïvely. (e) Smoothing the surfaces in a post-process and using Phong shading on the smooth surface reduces the effect of the polar artifact. (f) Using reflection lines or textures on the surface still show up the artifact, even if applied to the smoothed surface: another solution is needed (see Section 2).

Clark scheme, $\lambda = 0.89$ around a vertex of valency 8 [Augsdörfer et al. 06]. This makes it impractical to continue subdividing until the polygons around the extraordinary vertex are small enough. For $\lambda = 0.9$, this constraint would require 50 subdivision steps in the example given above, by which point there would be around 10^{36} polygons, most with edges of length 10^{-15} pixel (compared with the 10^6 polygons and the unit edge length generated by seven steps). This is clearly untenable. Even if we make the assumption that we can tolerate longer edges, say 1/9 of the length of the longest edge after seven steps, we still need $\lceil \log_\lambda 1/9 \rceil$ steps. When $\lambda = 0.9$, 21 further steps are required. Therefore, approximations need to be made that are computationally efficient and produce a close approximation to the true limit surface.

In the next section we discuss ways to improve the appearance around high valency extraordinary vertices. Figures 8 and 9 (b)–(e) show four better

approximations to the limit surface, of these (c) and (e) are visually acceptable. We discuss solutions (b) and (d) also, because solution (e) builds on both of them.

2. Solutions

Although triangles are the only surface primitives supported by common graphics hardware, the solutions presented here are applicable for any mesh type. We demonstrate the solutions using a tuned Catmull-Clark subdivision scheme, which works on quadrilateral meshes. Quadrilaterals are rendered as a pair of triangles after splitting them along either diagonal. The scheme has been tuned to achieve bounded curvature at and around extraordinary vertices. Unbounded curvature, a problem present in surfaces subdivided using the original Catmull-Clark scheme [Peters and Reif 04], should not be confused with the effect caused by the polar artifact.

2.1. Pushing Vertices to the Limit Surface

If, after a number of subdivision steps, the polygons in the regular region of the subdivided mesh are sufficiently small to be considered a good approximation to the limit surface, then the vertices of those polygons are sufficiently close to the limit surface to be considered approximately on it. This follows from the convex hull property of the box splines on which the standard subdivision methods are based. However, the extraordinary vertex itself is not in this category, and the vertices in the one-ring around it may not be in this category.

The limit points on a subdivision surface can be obtained from the row eigenvector corresponding to the dominant eigenvalue [Halstead et al. 93]. We can form stencils from these to determine the limit points at regular and extraordinary vertices. The row eigenvector differs for each subdivision scheme used and for each valency. For example, the stencil derived from the dominant row eigenvector for the Catmull-Clark scheme for a regular vertex is given by

$$\begin{bmatrix} \gamma & \beta & \gamma \\ \beta & \alpha & \beta \\ \gamma & \beta & \gamma \end{bmatrix}, \tag{1}$$

where $\alpha = 16$, $\beta = 4$, and $\gamma = 1$. The stencil is normalized such that its entries sum to unity by dividing it by 36. For the bounded-curvature variant of the Catmull-Clark scheme [Augsdörfer et al. 06], we apply stencil (1) with the above coefficients in regular regions, but, e.g., for valency $n = 8$ the stencil

has the entries $\alpha = 0.3742$, and eight entries for $\beta = 0.0610$ and $\gamma = 0.0172$ to cover all vertices around the extraordinary vertex. To push the extraordinary vertex to the limit surface, we apply this stencil. Table 1 gives α, β, and γ values for valencies up to 10 for the tuned Catmull-Clark method. To determine the limit positions for vertices in the one-ring, we simply apply one extra subdivision step, which allows us to determine their limit position using the standard regular limit stencil (1).

If we push all vertices at and around the extraordinary vertex onto the limit surface by convolving the mesh with the limit stencils, we get some improvement in the rendered result. But the artifact is still clearly visible due to the long edges emanating from the extraordinary vertex (see Figure 2). Phong shading across the large polygons, although useful, is not sufficient because it preserves the coarse polygonal silhouette. What is required, instead, is an appropriate polygonization that allows for a smooth variation of surface

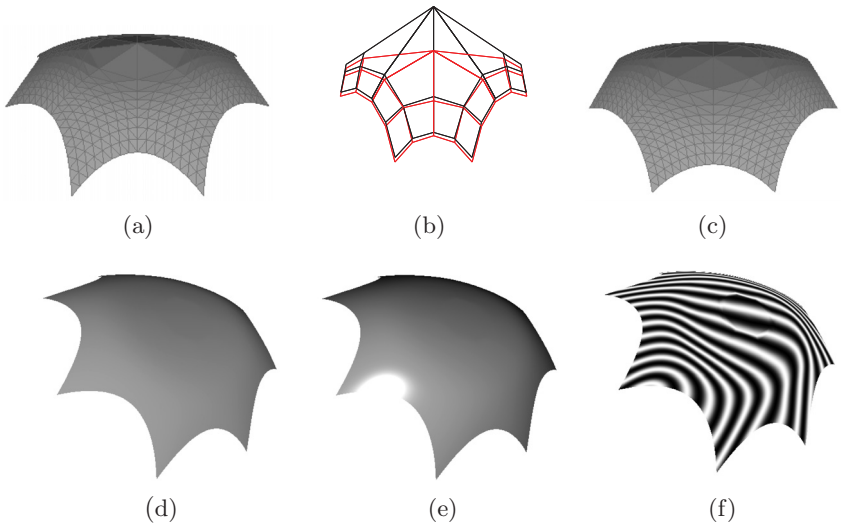


Figure 2. Pushing vertices to the limit surface. (a) The naïvely rendered polygon after three subdivision steps clearly shows the polar artifact, because the extraordinary vertex approaches its limit position slower then the regular vertices. (b) By pushing the points after subdivision (black mesh) onto the limit surfaces (red mesh), the polar artifact effect improved, but it is still clearly visible after rendering naïvely (c) because of the long edges emanating from the extraordinary vertex. (d) The polygon is smoothed in a postprocess before rendering seven subdivision steps. The long edges cause a visible ring around the extraordinary vertex, despite vertices being pushed onto the limit surface. (e) Phong shading improves but does not solve the problem. (f) Using reflection highlights the polar artifact effect.

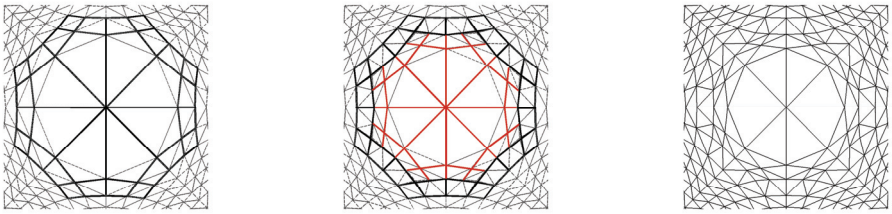
n	α	β	γ
4	0.4444	0.1111	0.0278
5	0.4228	0.0915	0.0239
6	0.4020	0.0785	0.0212
7	0.3855	0.0687	0.0190
8	0.3742	0.0610	0.0172
9	0.3661	0.0548	0.0156
10	0.3602	0.0497	0.0143

Table 1. The normalized entries in the limit stencils for valencies n up to 10.

normal as we move across the one-ring. However, we will need to use the idea of pushing to the limit surface in our solutions in Sections 2.4 and 2.5.

2.2. Adaptive Subdivision

Adaptive subdivision has been widely implemented (e.g., by [Müller and Jaeschke 98]). The issues in the implementation are how to maintain a good mesh with no cracks and how to identify when to switch between levels. In the present situation, we switch between levels in rings around the extraordinary vertex and change from lower to higher levels of refinement as we get closer to the extraordinary vertex (see Figure 4). This means that the join between levels of refinement, shown in detail in Figure 3, always occurs in the same



(a) (b) (c)

Figure 3. (a) The one-ring around the extraordinary vertex of the quadrilateral mesh (black lines) is shown together with the mesh triangulated for rendering (grey dashed lines). (b) The one-ring around the extraordinary vertex before adaptive subdivision (black lines) is shown together with the one-ring after one step of subdivision has been applied (red lines). The new triangulation is also shown (grey dashed lines). (c) The triangulation after one step of subdivision has been applied. More steps of adaptive subdivision are still necessary to obtain a more densely triangulation around the extraordinary vertex.

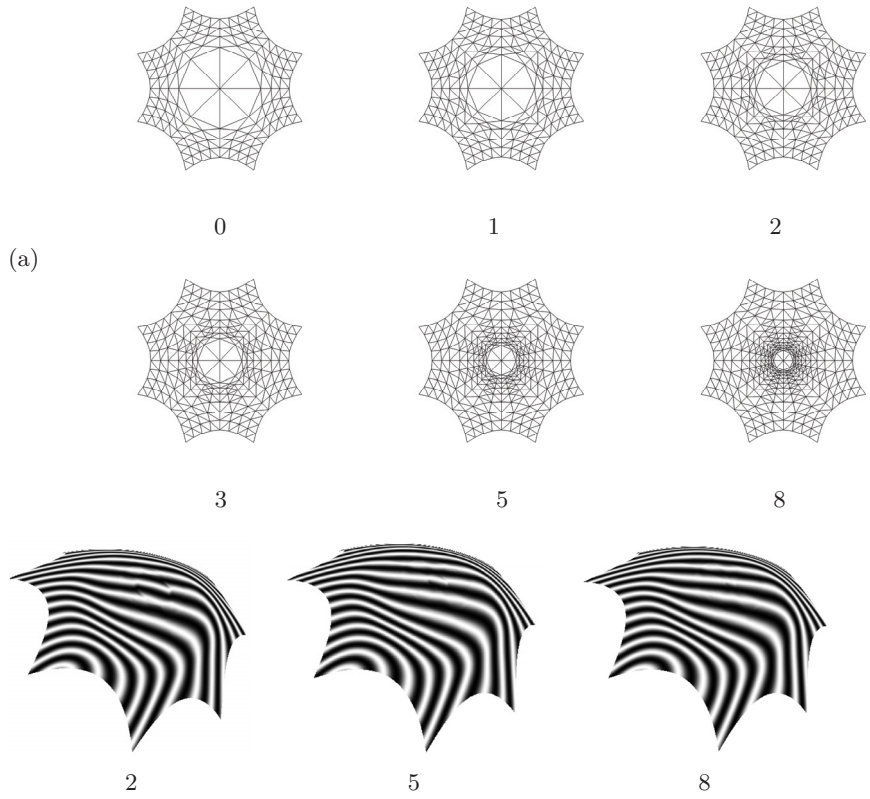


Figure 4. Adaptive subdivision. (a) The large polygons around an extraordinary vertex of valency $n = 8$ are reduced using adaptive subdivision. The level of subdivision within the one-ring around the extraordinary vertex increases from 0 to 7. (b) Reflection lines on the smoothed surface show the improved appearance around the extraordinary vertex with increasing number of adaptive subdivision steps (after two, five, and eight additional subdivision steps within the one-ring).

way. These facts make it relatively easy to implement adaptive subdivision to solve the rendering artifact problem compared to implementing general adaptive subdivision.

As an example, assume that all edges in the base mesh are of roughly the same length and that k subdivision steps are sufficient in the regular regions of the mesh. The spoke edges (those emanating from the extraordinary vertex to the one-ring) will be too long. The ratio of edge lengths between polygons in the two-ring and polygons in the one-ring is approximately $\sigma = (1 - \lambda)/\lambda$. Adaptive subdivision thus needs to ensure that the maximum edge length is σ times the spoke-edge length; for $\lambda = 0.9$, $\sigma = 1/9$.

Figure 4(a) shows an example of adaptive subdivision until spoke edges are equal to or smaller than σ. Using reflection lines (Figure 4(b)) shows how progressive adaptive subdivision improves the rendered surface.

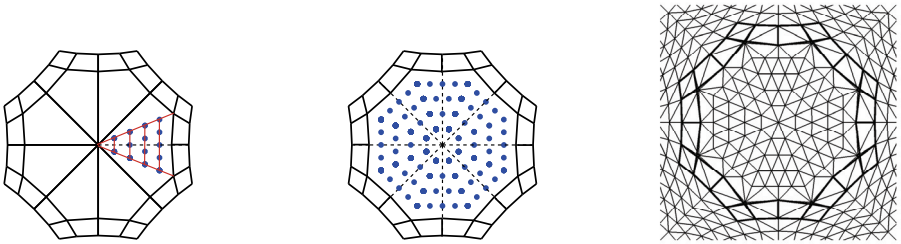
Adaptive subdivision produces a rendered solution, which is a visually acceptable approximation to the limit surface, and thus solves the polar artifact problem. It has the practical drawback that some of the polygons near the extraordinary vertex are extremely small.

2.3. Exact Evaluation

Stam's exact evaluation [Stam 98] is another method that is obviously relevant. If we polygonize the region around the extraordinary vertex in a uniform fashion, then we can evaluate the limit surface at every vertex of that polygonization.

The uniform polygonization in the parameter space can be generated by placing vertices at steps of length σ within the quadrilaterals around the extraordinary vertex, as shown in Figures 5(a) and (b), and discarding existing long spokes from the extraordinary vertex. Re-meshing the region around the extraordinary vertex using the newly inserted points (see Figure 5(c)) leads to a uniform limit mesh around the extraordinary vertex with edges no longer than σ.

The issues here are that we need to handle Stam's evaluation to high depths for large λ (at least 21 steps for the $\lambda = 0.9$ example above), that we need



(a) (b) (c)

Figure 5. (a) Points at equidistant intervals have been inserted replacing one spoke emanating from the extraordinary vertex. (b) All points added at equidistant intervals within the one ring around the extraordinary vertex are shown in blue. The long spoke edges emanating from the extraordinary vertex (dotted lines) are discarded. (c) The uniform triangulation around the extraordinary vertex using the newly inserted points.

to precalculate and store all of the necessary matrices, and that we should ideally store a uniform mesh for each valency of extraordinary vertex. This is time-consuming to implement.

For a sufficiently fine polygonization, this produces a surface that is indistinguishable from the limit surface and indistinguishable from the surface created using adaptive subdivision described in the previous section. It improves on adaptive subdivision by not generating a large number of tiny polygons.

However, we would prefer a solution that is easier to implement and faster to evaluate than either of these methods.

2.4. Bézier Curve Approximation

An early idea was to approximate the edges emanating from the extraordinary vertex by quadratic Bézier curves. The end points are the extraordinary vertex P_0 and the vertex in the one-ring P_2, both pushed onto the limit surface. To determine the third point P_1 of the quadratic Bézier curve, we define tangent planes for the extraordinary vertex and the vertex in the one-ring.

To determine the tangent plane, the normal vector N at the extraordinary vertex and all vertices in the one-ring have to be computed. This can be done by taking cross products of all edges meeting at the vertex, adding them together, and then dividing them by the valency of that vertex. Because we only need the tangent planes corresponding to the extraordinary vertex and the vertices in the one-ring, we only need to define $n+1$ normals, where n is the valency of the extraordinary vertex.

The tangent plane of a vertex P_x is spanned by the vectors v and w, where vector $v = (N_{P_0} \times N_{P_2})$ is orthogonal to the normal vectors at the extraordinary vertex N_{P_0} and at the one-ring vertex N_{P_2} and given by their cross product; vector $w = (v \times N)$ is orthogonal to v and the normal vector N at the vertex where the tangent plane is determined.

The line of intersection q of the tangent plane at the extraordinary vertex and the tangent plane at a point in the one-ring can then be determined according to

$$q = P_0 + w\frac{(N_{P_2}.(P_2 - P_0))}{(N_{P_2}.w)}, \tag{2}$$

where N_{P_2} is the normal at a vertex in the one-ring. We defined the third control point P_1 to be the point on the intersection line of the two tangent planes that lies closest to the straight line joining the end points P_0 and P_2. To find this point, we employ

$$P_1 = q + v\frac{((q - P_0).(P_2 - P_0))}{\frac{(v.v)}{v.(P_2 - P_0)}((P_2 - P_0).(P_2 - P_0)) - (v.(P_2 - P_0))}. \tag{3}$$

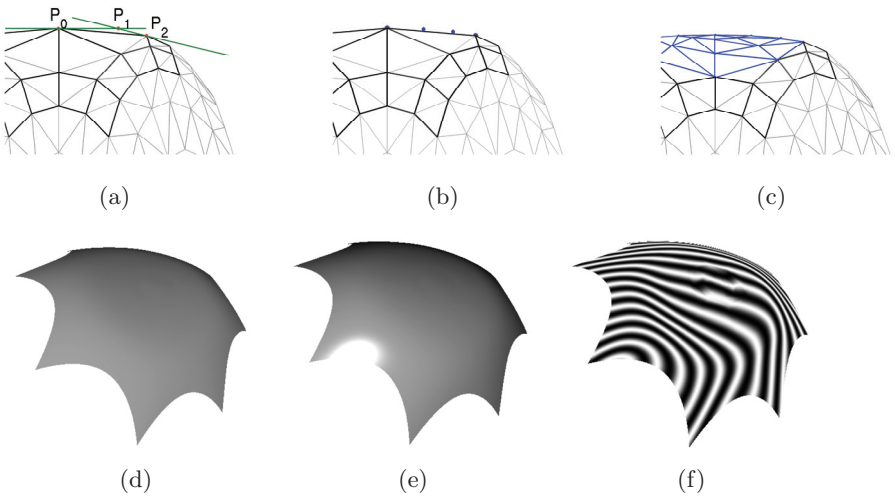


Figure 6. Bézier curve approximation. (a) The intersection of the tangent planes on the limit position of the extraordinary vertex and on a limit vertex on the one-ring around the extraordinary vertex gives the third point necessary to define a quadratic Bézier curve instead of the straight line connecting the two points after subdivision. (b) We define equidistant points on the quadratic Beźier curve with a spacing not exceeding the distance between the one- and two-ring. (c) After defining the additional points on each of the edges around the extraordinary vertex, the points are connected with straight lines to form smaller triangles around the extraordinary vertex. (d) When rendering the surfaces naïvely, the long edges are still visible. (e) The smoothed and Phong-shaded surface. (f) Reflection lines used on the smoothed surface.

We now have the three points necessary to define a quadratic Bézier curve instead of a straight line between the extraordinary vertex and the one-ring.

We replace each of the straight edges emanating from the extraordinary vertex with a quadratic Bézier curve and determine points at intervals of length σ along each curve as shown in Figure 6(b). We connect these newly defined points to form a number of triangles around the extraordinary vertex as shown in Figure 6(c).

However, for all but very high valency, the triangles created between the spokes are still too large, causing rendering artifacts, clearly visible in Figures 6(d)-(f). We build on this idea to produce a workable solution using Bézier triangles.

2.5. Bézier Triangle Approximation

Bézier triangle approximation produces an adequate approximation to the limit surface without generating tiny polygons and without the need to

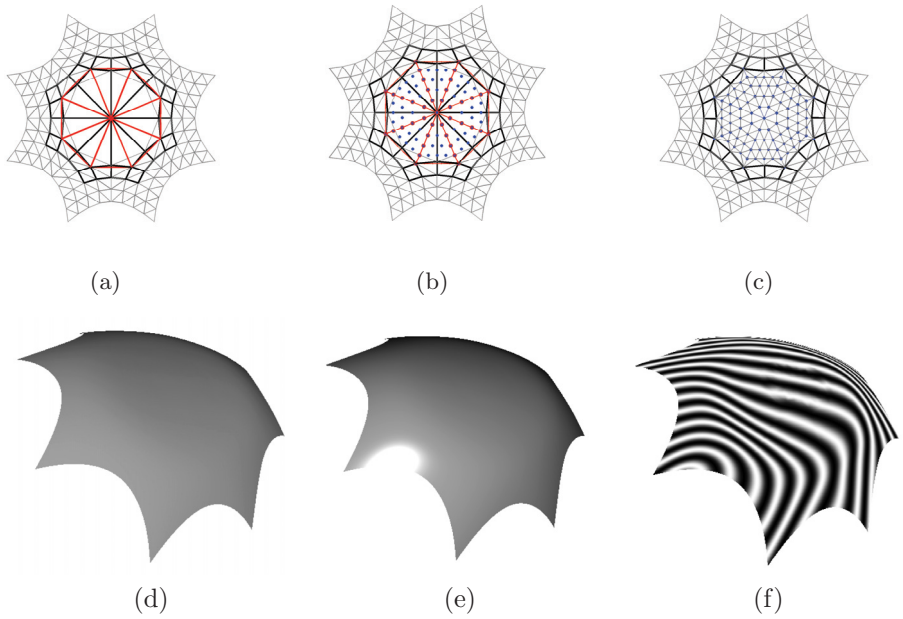


Figure 7. Bezier triangle approximation (Section 2.5): (a) The equidistant points along the three Bézier curves which form the edges of the Bézier triangle are shown in red. They are defined in the same way as described in Figure 6. (b) Equidistant points are defined on the Bézier triangle. All points used for re-meshing are shown in blue. (c) The newly re-meshed two-ring region around the extraordinary vertex. (d) The naïvely rendered surface after re-meshing. (e) The smoothed and Phong shaded surface. (f) Reflection lines used on the rendered on the smoothed surface.

perform Stam's exact evaluation. We approximate the limit surface around the extraordinary vertex by Hermite interpolation, using a set of Bézier triangles, and we sample from this approximation.

The three corners of the triangle are the extraordinary vertex and two adjacent vertices in the two-ring, all pushed onto the limit surface as explained in Section 2.1. The three other control points of the quadratic Bézier triangles are determined analogously to the Bézier curves as explained in Section 2.4. We discard the limit points in the one-ring and use vertices in the two-ring so that the Bézier triangles cover the entirety of the quadrilaterals surrounding the extraordinary vertex, as shown in Figure 7(a). This way we do not encounter T-junctions when re-meshing around the extraordinary vertex. We use the same uniform polygonization as in Stam's exact evaluation (see Section 2.3), at lower computational cost, with the result shown in Figure 7(d)-(f). This only has a guarantee of C^0 continuity across the spoke edges, but the

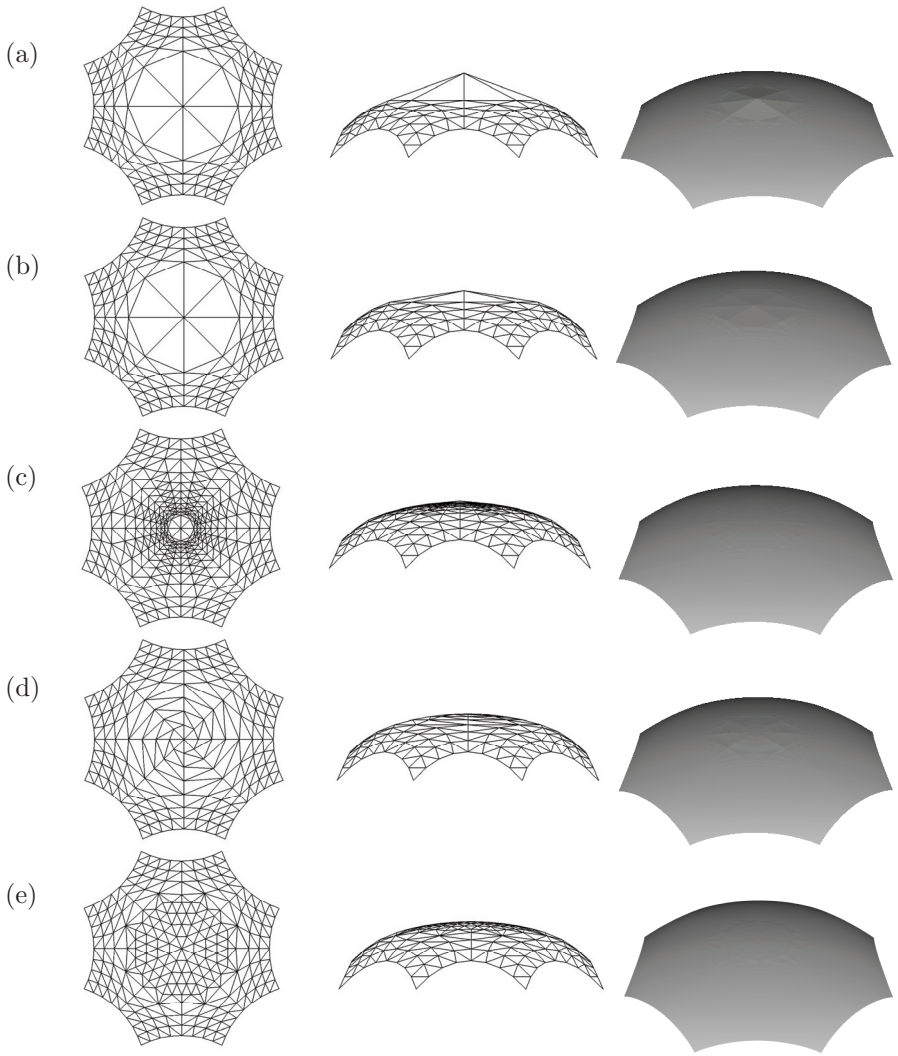


Figure 8. Bounded curvature Catmull-Clark [Augsdörfer et al. 06] applied to an elliptic shape with an extraordinary vertex of valency 8 at its center. In each case, the polygon edges are shown on the left and the center, respectively, and the naïvely rendered surface is shown on the right. (a) Basic subdivision, where the polar artifact is clearly visible and the solutions (b), vertices pushed to the limit surface. (c) Adaptive subdivision. (d) Bézier curves on spoke edges. (e) Bézier patches between spoke edges.

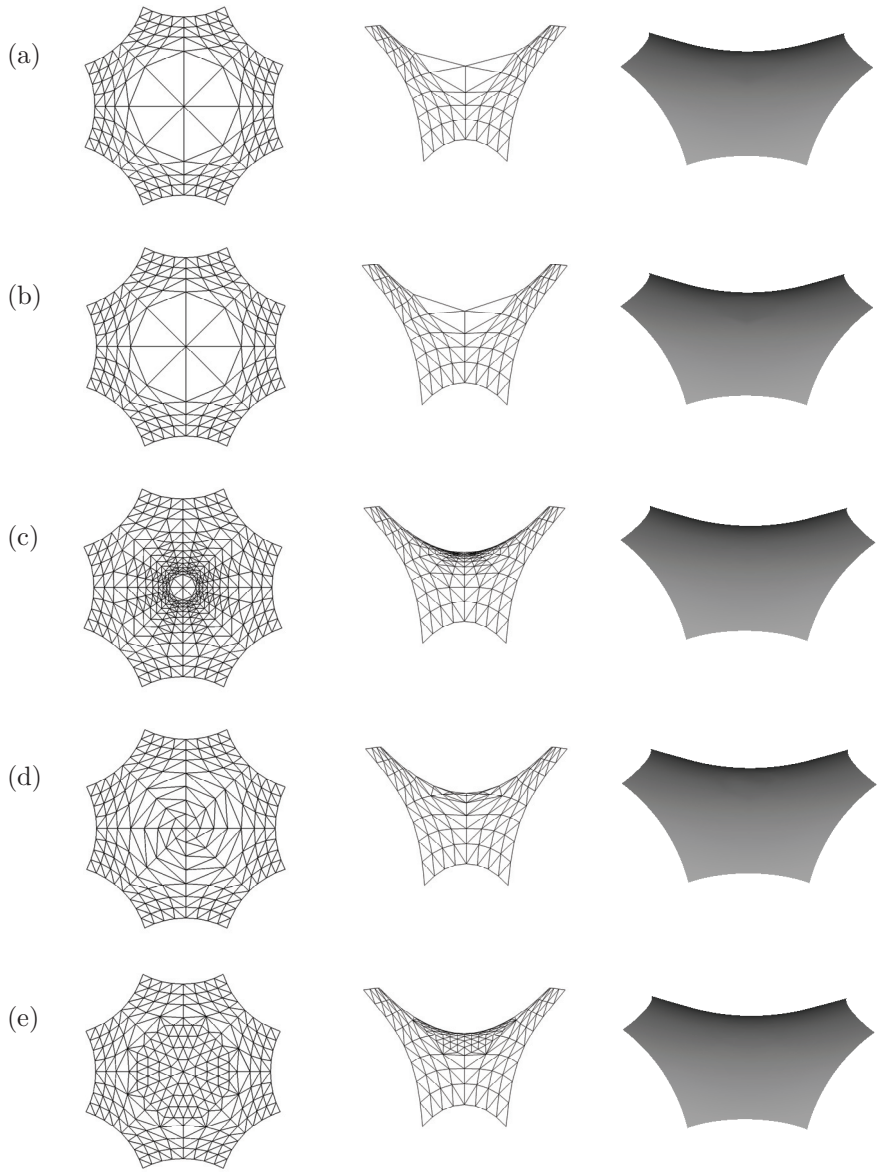


Figure 9. Bounded curvature Catmull-Clark [Augsdörfer et al. 06] applied to a hyperbolic shape with an extraordinary vertex of valency 8 at its center. In each case, the polygon edges are shown on the left and the center, respectively, and the naïvely rendered surface is shown on the right; (a)-(e) are as in Figure 8.

angles between the tangent planes of adjacent Bézier triangles are significantly less than between the facets that are actually being rendered. Figures 8 and 9(e) show that despite its simplicity, this approximation provides good results.

3. Discussion

Large values of λ are required to get highest quality limit surfaces with optimal mathematical properties around extraordinary vertices of high valency. The polar rendering artifacts, which occur around extraordinary vertices for large λ, are artifacts caused by subdividing too few times to get a good polygonal approximation to the limit surface. They can be removed by providing a better approximation to the limit surface than is given by subdividing a small number of times. Any of the solutions shown in Figures 8 and 9 improve the polar artifact. However, adaptive subdivision, Stam's exact evaluation, and Bézier triangle evaluation all provide a good rendering of the limit surface and achieve visual smoothness.

The comparative efficiency of the three algorithms can be measured using the number of memory accesses required, the number of floating-point operations performed, and the number of triangles that need to be drawn. The Bézier triangle method is the most efficient on all three measures (see below); therefore, it will be competitive regardless of the location of the bottleneck in any particular rendering pipeline.

For the Bézier triangle approach and adaptive subdivision, the cost in terms of number of calculations is $O(n)$, where n is the valency of the vertex, while Stam's exact evaluation is of complexity $O(n^2)$. For example, to fill the region around a valency $n = 8$-vertex for a Catmull-Clark subdivision surface with $\lambda = 0.9$, one requires approximately 1900 multiplications in the Bézier triangle approach, 2100 multiplications for adaptive subdivision, but 4750 multiplications when using exact evaluation.

The cost in terms of rendering is $O(n)$ for the Bézier triangle approach or Stam's exact evaluation, while the numerous little triangles created using adaptive subdivision lead to a complexity of $O(n\frac{\lambda}{1-\lambda})$. In the example above, 176 triangles are required using exact evaluation or Bézier triangles, but 512 triangles need to be rendered when the adaptive subdivision approach is used.

The cost in terms of memory look-up is $O(n)$ for Bézier triangles or Stam's exact evaluation, but $O(n\frac{\lambda}{1-\lambda})$ for adaptive subdivision. In the example above, we require 24 memory look-ups to fit Bézier triangles, 96 memory look-ups for Stam's exact evaluation, and 720 memory look-ups for adaptive subdivision.

Bézier triangle approximation, although less accurate, is the most efficient rendering of the three, as it does not produce the tiny triangles of adaptive subdivision, is considerably easier to implement than exact evaluation, and is faster.

References

[Augsdörfer et al. 06] Ursula H. Augsdörfer, Neil A. Dodgson, and Malcolm A. Sabin. "Tuning Subdivision by Minimising Gaussian Curvature Variation Near Extraordinary Vertices." *Computer Graphics Forum* 25:3 (2006), 263–272.

[Barthe and Kobbelt 04] Loïc Barthe and Leif Kobbelt. "Subdivision Scheme Tuning Around Extraordinary Vertices." *Comp. Aided Geom. Des.* 21 (2004), 561–583.

[Cashman et al. 09] Thomas J. Cashman, Ursula H. Augsdörfer, Neil A. Dodgson, and Malcolm A. Sabin. "NURBS with Extraordinary Points: High-degree, Non-uniform, Rational Subdivision Schemes." *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2009)* 28:3 (2009), Article 46.

[Catmull and Clark 78] Ed Catmull and Jim Clark. "Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes." *Computer Aided Design* 10:6 (1978), 183–188.

[Halstead et al. 93] M. Halstead, M. Kass, and T. DeRose. "Efficient, Fair Interpolation using Catmull-Clark Surfaces." In *Proceedings of SIGGRAPH 93, Computer Graphics Proceedings, Annual Conference Series*, edited by James T. Kajiya, pp. 35–44. New York: ACM Press, 1993

[Loop 87] Charles Loop. "Smooth Subdivision Surfaces Based on Triangles." Master's thesis, Department of Mathematics, University of Utah, 1987.

[Müller and Jaeschke 98] H. Müller and R. Jaeschke. "Adaptive Subdivision Curves and Surfaces." Technical Report No. 676, Universität Dortmund, 1998.

[Ni and Nasri 06] T. Ni and A. H. Nasri. "Tuned Ternary Quad Subdivision." In *Geometric Modelling and Processing*, Lecture Notes in Comp. Sci. 4077, pp. 441–450. Berlin-Heidleberg: Springer-Verlag, 2006.

[Peters and Reif 04] J. Peters and U. Reif. "Shape Characterization of Subdivision Surfaces: Basic Principles." *Comput. Aided Geom. Des.* 21:6 (2004), 585–599.

[Sabin and Barthe 03] Malcolm Sabin and Loïc Barthe. "Artifacts in Recursive Subdivision Surfaces." In *Curve and Surface Fitting: Saint-Malo 2002*, pp. 353–362. Brentwood, TN: Nashboro Press, 2003.

[Stam 98] Jos Stam. "Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values." In *Proceedings of SIGGRAPH 98, Computer Graphics Proceedings, Annual Conference Series*, edited by Michael Cohen, pp. 395–404. Reading, MA: Addison Wesley, 1998.

[Zorin and Schröder 00] D. Zorin and P. Schröder, editors. "Subdivision for Modelling and Animation." *SIGGRAPH 2000 Course Notes*. New York: ACM Press, 2000.

[Zulti et al. 06] Avi Zulti, Adi Levin, David Levin, and Mina Teicher. "C^2 Subdivision over Triangulations with One Extraordinary Point." *Comp. Aided Geom. Des.* 23 (2006), 157–178.

Web Information:

http://jgt.akpeters.com/papers/AugsdoerferEtAl09/

A video explaining how Bézier triangles are fitted is available online at http://www.cl.cam.ac.uk/∼uha20/PolarArtifactMovie.wmv

Ursula H. Augsdörfer, The Computer Laboratory, University of Cambridge, 15 J.J. Thomson Ave., Cambridge CB3 0FD (uha20@cl.cam.ac.uk)

Neil A. Dodgson, The Computer Laboratory, University of Cambridge, 15 J.J. Thomson Ave., Cambridge CB3 0FD (nad10@cl.cam.ac.uk)

Malcolm A. Sabin, Numerical Geometry Ltd., 19 John Amner Close, Ely, Cambridge CB6 1DT (malcolm@geometry.demon.co.uk)

Received May 20, 2009; accepted in revised form October 7, 2009.