Subdivision shading

Marc Alexa Tamy Boubekeur
TU Berlin

Abstract

The idea of Phong Shading is applied to subdivision surfaces: normals are associated with vertices and the same construction is used for both locations and normals. This creates vertex positions *and* normals. The vertex normals are smoother than the normals of the subdivision surface and using vertex normals for shading attenuates the well known visual artifacts of many subdivision schemes. We demonstrate how to apply subdivision to normals and how blend and combine different normals for achieving a variety of effects.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Surface representations; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Shading

Keywords: Shading, subdivision, visual quality, irregular vertices

1 Introduction

Meshes are the dominant representation of computer generated objects for visual effects. They are simple and versatile, but suffer from discontinuous derivatives and the resulting visual artifacts. Subdivision surfaces define smooth surfaces over a given base mesh and are popular because their recursive construction allows generating discrete approximations to the smooth surface at arbitrary resolutions. Generally, faces are subdivided by inserting new vertices and then the locations of old and new are updated. The update rules often are derived from analogies to spline surfaces. The basic schemes cover triangle and quad meshes and interpolate the vertex positions or approximate them providing a surface with better quality[Catmull and Clark 1978; Doo and Sabin 1978; Loop 1987; Dyn et al. 1990].

For many of the popular subdivision schemes, the surface is C^2 continuous and well-behaved except for irregular vertices (i.e. vertices with degree other than 4 in a quad mesh or 6 in a triangle mesh) where continuity is only C^1. Unfortunately, this is not only a theoretical defect and even if C^2 or a higher degree of parametric smoothness is enforced, surfaces do exhibit shape defects leading to visual artifacts at irregular vertices [Reif 1995; Zorin 1998; Prautzsch 1998; Peters and Reif 2004; Karčiauskas et al. 2004]. There are several approaches to improve the situation at irregular vertices (see, e.g., [Levin 2006; Karčiauskas and Peters 2007]), but they usually come at the expense of a more complicated scheme, or a fundamental change in the modeling paradigm, which make them less attractive in some applications.

A standard trick in computer graphics, where the actual shape is only necessary to generate the visual result, is using normals for shading computations that are not orthogonal to the surface. Phong

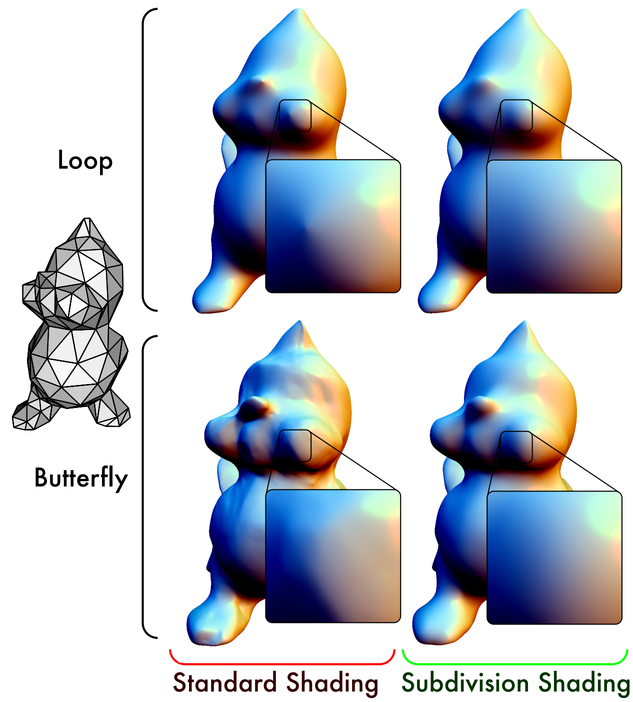

Figure 1: *The coarse* TWEETY *mesh is subdivided using Loop and modified Butterfly subdivision.* Subdivision Shading *uses subdivision rules for computing vertex normals instead of using surface normals from the limit surface. It results in visually smoother shading, especially around irregular vertices.*

Shading [Phong 1975] associates normals with vertices and linearly interpolates them across faces. Note that the same geometric construction is used for positions and normals inside the triangles spanned by vertex positions and vertex normals. In this way, the idea can be extended to quadratic [Boubekeur and Schlick 2007] interpolation, however, the degree of normal interpolation could also be higher [van Overveld and Wyvill 1997; Loop and Schaefer 2008] or lower [Vlachos et al. 2001; Boubekeur and Alexa 2008] than the degree of geometric interpolation.

Subdivision surfaces have a smooth geometry so that silhouettes will look very good. The remaining visual problem appears in the shading around irregular vertices, where the normals are only C^0 (for standard schemes). Our idea for hiding this problem is a generalization of Phong Shading: vertex positions and vertex normals are treated in the same way, i.e. the interpolation process applied to vertex locations is also applied to vertex normals. For a subdivision surface this means we associate normals with vertices of the base mesh, and then compute vertex normals for all vertices by applying the subdivision rules to the normal vectors.

In the following we will briefly recall the basics of subdivision and then explain how to apply the schemes to vertex normals rather than positions. In particular, we explain how to interpolate normals and how to blend the normals of subdivision surface where they are expected to be well behaved with the normals derived from subdivision around irregular vertices. The results show a clear improvement of the shading (see Fig. 1). We also discuss some applications together with emerging open problems.

ACM Reference Format
Alexa, M., Boubekeur, T. 2008. Subdivision shading. *ACM Trans. Graph. 27*, 5, Article 142 (December 2008), 4 pages. DOI = 10.1145/1409060.1409095 http://doi.acm.org/10.1145/1409060.1409095.

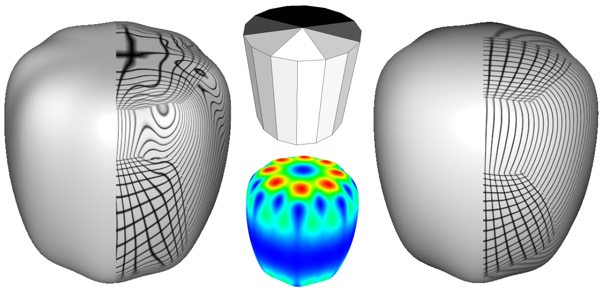

Figure 2: *Catmull-Clark subdivision applied to a cylinder. The left image is rendered with the normals of the subdivided mesh, the right image shows* subdivided *normals computed using Catmull-Clark rules. The color plot in the middle illustrates the difference of the vertex normals*

2 Background

Subdivision surfaces are constructed from a base mesh $\mathscr{M}^0 = (\mathsf{K}^0, \mathbf{V}^0)$ whose vertex positions $\mathbf{V}^0 = \{\mathbf{v}_1^0, \mathbf{v}_2^0, \ldots\}$ and combinatorial structure is given. A subdivision scheme is defined as a set of rules for inserting new vertices (i.e. updating K) and then modifying the positions of the old and new vertices (i.e. updatig $\mathbf{V}$). Applying the rules to $\mathscr{M}_0$ yields $\mathscr{M}_1, \mathscr{M}_2$, and so forth. For many sets of rules it has been proved that the sequence of meshes $\mathscr{M}_i$ converges to a smooth limit surface.

The modified positions for old and new vertices are described in terms of affine weights for the old positions, arranged in a *mask* describing which vertices receive how much weight. Note that each local combinatorial structure requires its own mask. While the inserted vertices always require a mask to define their position in space, the update of existing vertices is only done in some schemes. A usual assumption is that most vertices are *regular*, i.e. have degree 6 in a triangle mesh or degree 4 in a quad mesh. Irregular vertices are assumed to be isolated, i.e. surrounded only by regular vertices. This allows limiting the number of masks necessary for implementing the procedure. Specific masks for most common subdivision schemes and more details on their variants can be found in the recent book by Peters and Reif [2008]. In most applications subdivision schemes are considered practical if they **(i)** generate a *smooth limit surface*, i.e. the limit surface is C^1 everywhere, preferably C^2, ideally with bounded curvature, **(ii)** use the same mask in each step, i.e. they are *stationary* (the i-th subdivision pass is independent of i), **(iii)** with the number of entries in the mask being small so that the evaluation is fast.

The position of a point on the limit surface can be computed from its parametric position for several schemes [Stam 1998] – this can be applied to any coordinate in a linear space, for example texture coordinates [DeRose et al. 1998] and tangents, the latter offering normals for shading the surfaces in practice.

3 Subdividing normals for shading

The main idea of our approach is equipping each vertex with a normal and then applying the subdivision mask to geometry *and* normals (used for shading) in each step of subdivision.

Formally, if $\mathbf{N}^0 = \{\mathbf{n}_1^0, \mathbf{n}_2^0, \ldots\}$ are the vertex normals of the base mesh, vertex normals for refined meshes are computed by computing weighted combinations of these normals, with the weights specified by the mask of the subdivision scheme. The normals $\mathbf{N}^0$ could be supplied by the user, as are the vertex positions, or they could be computed from the vertex positions in the base mesh, for example the angle-weighted average of incident faces normals.

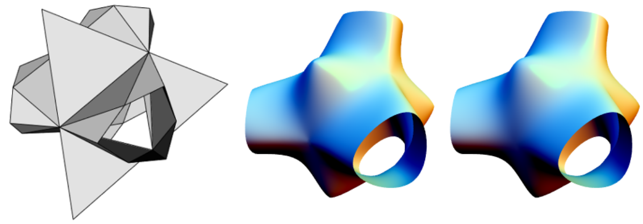

Figure 3: *Subdividing vertex normals also works on boundaries.*

If the subdivision scheme provides a C^1 geometry, we would like that it provides a similarly smooth limit when applied to the vertex normals. However, normals are entities on $\mathbb{S}^2$ rather than $\mathbb{R}^3$, and for the continuity property to carry over we need to define linear combination in an appropriate way on the sphere. The problem of defining weighted combinations of points on the sphere has been analyzed by Buss & Fillmore [2001] and they essentially define the weighted combination of normals $\mathbf{n}_1, \mathbf{n}_2, \ldots$ with weights $w_1, w_2, \ldots$ as

$$\min_{\mathbf{n}\in\mathbb{S}^2} \sum_i w_i \, \left(\mathrm{dist}_{\mathbb{S}^2}(\mathbf{n}, \mathbf{n}_i)\right)^2 \tag{1}$$

where the distance on $\mathbb{S}^2$ is measured geodesically. The following description of weighted combinations on the sphere illustrates this definition: the normal $\mathbf{n}$ has the property that the linear combination of normals mapped to the exponential map at $\mathbf{n}$ reproduces $\mathbf{n}$. Linear interpolation in exponential maps around intermediate solutions converges to this stationary point. This concept leads to the following algorithm:

1. As a starting normal we use the linear combination of the normal vectors and then normalize: $\mathbf{n}^{k=0} = \sum_i w_i \mathbf{n}_i / \|\sum_i w_i \mathbf{n}_i\|$
2. Map all input normals orthogonally to a plane orthogonal to $\mathbf{n}^k$, scale them to have distance to origin corresponding to the angle to $\mathbf{n}^k$ (*exponential map*):

$$\tilde{\mathbf{n}}_i = \frac{\angle(\mathbf{n}^k, \mathbf{n}_i)}{\left\|\mathbf{n}_i - \left(\mathbf{n}_i^{\mathsf{T}}\mathbf{n}^k\right)\mathbf{n}^k\right\|} \left(\mathbf{n}_i - \left(\mathbf{n}_i^{\mathsf{T}}\mathbf{n}^k\right)\mathbf{n}^k\right) \tag{2}$$

3. Perform linear combination in the map, i.e. $\tilde{\mathbf{n}}^{k+1} = \sum_i w_i \tilde{\mathbf{n}}_i$.
4. Map the result back onto the sphere, by rotating $\mathbf{n}^k$ around the axis $\mathbf{n}^k \times \tilde{\mathbf{n}}^{k+1}$ with an angle corresponding to $\|\tilde{\mathbf{n}}^{k+1}\|$ to define $\mathbf{n}^{k+1}$.
5. If $(\mathbf{n}^k)^{\mathsf{T}}\mathbf{n}^{k+1} > \varepsilon$ go back to step 2.

Note that accuracy can be traded for speed by performing only few iterations. We have found that performing many iterations has usually little effect on the visual quality; in particular, simple linear combination and normalization usually works well. This is consistent with the concept of proximity [Wallner and Dyn 2005], which suggests that the linear combination and subsequent normalization of normal vectors also defines a smooth subdivision scheme (albeit a proof for this concept is missing).

4 Controlling the normals

In the following we discuss several ways to control the normals. We consider first how and at what level of subdivision to generate base normals and, second, linearly blending subdivided and surface normals for achieving certain effects.

Figure 4: *The subdivision of normals can be started from the base mesh or after few levels of geometric subdivision. The images show the results starting with the normals of the base mesh, the normals of the mesh subdivided one time and two times. The right image is shaded with normals computed at the fully subdivided mesh.*

4.1 Generating base normals

As subdivision defines a smooth surface, it is typically not necessary to supply normals with the base mesh (an exception is if normal control is wanted [Biermann et al. 2000]). We have found that computing the vertex normals from the base mesh works well in practice. Using different techniques for computing the vertex normals results in only slightly different visual results.

As mentioned before, for the subdivision process to be well defined for normals, normals in the support of the subdivision mask need to be in a common hemisphere. If this is not the case for a very coarse base mesh, this problem can be corrected by subdividing the mesh before computing vertex normals from the face normals. Moreover, the idea of starting the generation of vertex normals by subdivision only after the second or third level of subdivision can be used to get results that appear less smoothed as compared to the surface normals. Fig. 4 compares several levels.

4.2 Blending vertex normals and surface normals

In a number of applications, one may want to correct the shading only where the shape exhibits artifacts (i.e., irregular vertices). Subdivision Shading enables an elegant solution to this problem, avoiding an explicit tracking of where to use which normal. At initialization, we equip each vertex i with a *blend weight* $b_i \in [0,1]$. This weight will be interpolated to vertices inserted by subdivision using linear interpolation (or potentially subdivision). The normal in each vertex necessary for rendering is computed by interpolating between the subdivided vertex normal and the surface normal in the vertex using b_i as the weight.

The typical scenario is to assign $b_i = 1$ to irregular vertices and $b_i = 0$ to regular vertices. Then the local correction of vertex normals using the subdivided normals is only applied to irregular vertices and the vertices inserted in their neighborhood. Fig. 5 shows examples of the resulting shaded images and the blending function.

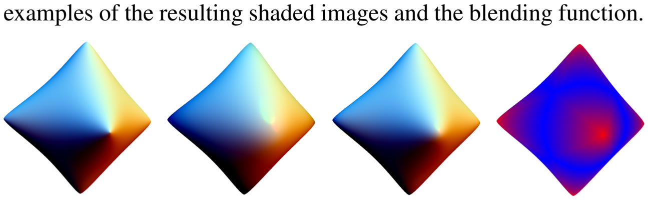

Figure 5: *Blending surface normals in the vertices with subdivided vertex normals. The images show from left to right: standard subdivision surface, subdivision shading, subdivided vertex normals blended with normals of the subdivision surface, and the color coded blend weights, where red corresponds to subdivided normals and blue corresponds to surface normals.*

As mentioned by Cignoni et al. [2005] and Rusinkiewicz et al. [2006], amplifying local contrast through normal manipulation can greatly improve visual features. Note that subdivided vertex normals are smoother than the normals of the subdivision surface. Instead of blending between the surface normals in vertices and the subdivided vertex normals with positive weights, we can use negative weights to exaggerate the lacking smoothness of the surface normals (see Fig. 6). This effect can be also controlled by the number of geometric subdivisions without performing subdivision on the normals.

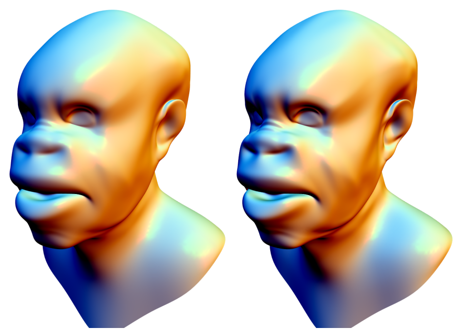

Figure 6: *Extrapolating from the smooth subdivided vertex normals over the surface normals (shown left) results in a normal field that enhances features in the subdivision surface. These normals can be used for exaggerated shading, as demonstrated in the right image*

5 Discussion

While our approach is admittedly very simple, it offers a striking balance between simplicity, speed, and visual quality. Its simplicity allows every developer a quick implementation and evaluation, even within a real-time context [Shiue et al. 2005]. Computation times are similar to the generation of normals of the subdivision surface, making sure production environments can keep their adopted work flows. It works for any standard scheme (see Fig. 2 for Catmull-Clark, and all other figures for subdivision on triangle meshes), surfaces with boundary (see Fig. 3), and under motions of light sources or the objects (please see the accompanying video). The visual quality in the vicinity of irregular vertices is always clearly better than using surface normals.

In particular, interpolating schemes such as Butterfly could become a very attractive alternative to other constructions of smooth interpolating surfaces aiming mostly at the visual quality such as curved PN triangles [Vlachos et al. 2001] (see Fig. 7 for a visual comparison): especially in games it is important to reuse the mesh defining a character as the control mesh for a smooth geometry. Curved PN triangles as interpolating subdivision preserve the original dimensions of the shapes. While Butterfly is usually considered to generate shapes that are seriously lacking the necessary visual fidelity, Subdivision Shading is attenuating the defects enough to make it attractive again (see Figures 1, 4).

In the spirit of Phong Shading, one could stop the subdivision of the geometry earlier than the subdivision of normals. This generates a *procedural* normal texture for each triangle, which can be useful in applications that are sensitive to the polygon count, or for realtime *screen-space* sampling which might perform normal subdivision in the fragment shader. In the extreme case, geometry would not be subdivided at all (similar to Phong Shading). The larger stencil of subdivision as compared to linear interpolation can be useful for tuning the appearance of highlights, as demonstrated in Fig. 8.

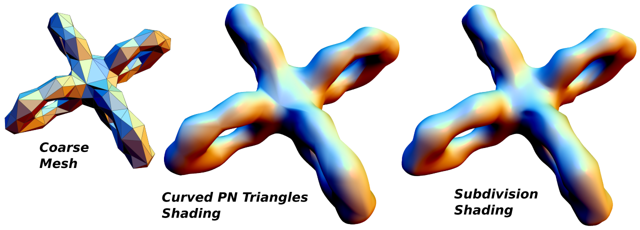

Figure 7: *Comparing modified Butterfly subdivision with Subdivision Shading (right) to the quadratic normal fields of curved PN triangle (middle). Note that the visual smoothness of Subdivision Shading applied to Butterfly subdivision is similar or better.*

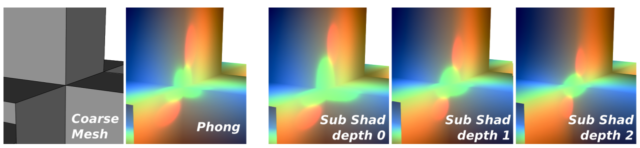

Figure 8: *Phong Shading compared to Subdivision Shading: when subdividing only the normal field and keeping the original coarse geometry, subdivision shading offers a variety of different light spread effects over the surface.*

We have found that computing normals after one level of standard subdivision usually yields the best results. In fact, the only potential problem we have encountered with Subdivision Shading so far is backface normals influencing the front facing part of the surface (this is a known problem of Phong Shading as well [Olano and Yoo 1993]). Computing base normals after a few steps of subdivision corrects this problem.

Interesting avenues for research are devising special subdivision rules for the normals as well as geometry-aware blending of vertex and surface normals. Furthermore, it might be possible to modify positions based on the normals. This raises the question of the integrability of the so-defined normals, and the potential surface they define.

References

BIERMANN, H., LEVIN, A., AND ZORIN, D. 2000. Piecewise smooth subdivision surfaces with normal control. In *Proceedings of ACM SIGGRAPH 2000*, 113–120.

BOUBEKEUR, T., AND ALEXA, M. 2008. Phong tessellation. In *ACM Trans. Graph (Proc. SIGGRAPH Asia)*.

BOUBEKEUR, T., AND SCHLICK, C. 2007. QAS: Real-time quadratic approximation of subdivision surfaces. In *Proceedings of Pacific Graphics 2007*, 453–456.

BUSS, S. R., AND FILLMORE, J. P. 2001. Spherical averages and applications to spherical splines and interpolation. *ACM Trans. Graph. 20*, 2 (Apr.), 95–126.

CATMULL, E., AND CLARK, J. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10*, 6 (Sept.), 350–355.

CIGNONI, P., SCOPIGNO, R., AND TARINI, M. 2005. A simple normal enhancement technique for interactive non-photorealistic renderings. *C&G 29*, 1 (feb), 125–133.

DEROSE, T., KASS, M., AND TRUONG, T. 1998. Subdivision surfaces in character animation. In *ACM SIGGRAPH*.

DOO, D., AND SABIN, M. 1978. Behaviour of recursive division surfaces near extraordinary points. *Computer-aided Design 10*, 6, 356–360.

DYN, N., LEVIN, D., AND GREGORY, J. A. 1990. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph. 9*, 2, 160–169.

KARČIAUSKAS, K., AND PETERS, J. 2007. Bicubic polar subdivision. *ACM Transactions on Graphics 26*, 4 (Oct.), 14:1—-14:6.

KARČIAUSKAS, K., PETERS, J., AND REIF, U. 2004. Shape characterization of subdivision surfaces–case studies. *CAGD 21*, 6, 601–614.

LEVIN, A. 2006. Modified subdivision surfaces with continuous curvature. *ACM Transactions on Graphics 25*, 3 (July), 1035–1040.

LOOP, C., AND SCHAEFER, S. 2008. Approximating catmull-clark subdivision surfaces with bicubic patches. *ACM Trans. Graph. 27*, 1, 1–11.

LOOP, C. 1987. *Smooth subdivisions surfaces based on triangles*. Master's thesis, Department of Mathematica, University of Utah.

OLANO, T. M., AND YOO, T. S. 1993. Precision normals (beyond Phong). Tech. rep., University of North Carolina.

PETERS, J., AND REIF, U. 2004. Shape characterization of subdivision surfaces–basic principles. *Computer Aided Geometric Design 21*, 6, 585–599.

PETERS, J., AND REIF, U. 2008. *Subdivision surfaces*. Springer.

PHONG, B. T. 1975. Illumination for computer generated pictures. *Commun. ACM 18*, 6, 311–317.

PRAUTZSCH, H. 1998. Analysis of C^k-subdivision surfaces at extraordinary points. Tech. Rep. iratr-1998-4, University of Karlsruhe, Germany, Computer Science.

REIF, U. 1995. A unified approach to subdivision algorithms near extraordinary vertices. *Comput. Aided Geom. Des. 12*, 2, 153–174.

RUSINKIEWICZ, S., BURNS, M., AND DECARLO, D. 2006. Exaggerated shading for depicting shape and detail. *ACM Trans. Graph. 25*, 3, 1199–1205.

SHIUE, L.-J., JONES, I., AND PETERS, J. 2005. A realtime gpu subdivision kernel. *ACM Transactions on Graphics 24*, 3 (Aug.), 1010–1015.

STAM, J. 1998. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Proc. ACM SIGGRAPH 98*, 395–404.

VAN OVERVELD, C. W. A. M., AND WYVILL, B. 1997. Phong normal interpolation revisited. *ACM Trans. Graph. 16*, 4, 397–419.

VLACHOS, A., PETERS, J., BOYD, C., AND MITCHELL, J. L. 2001. Curved PN triangles. In *2001 ACM Symposium on Interactive 3D Graphics*, 159–166.

WALLNER, J., AND DYN, N. 2005. Convergence and c1 analysis of subdivision schemes on manifolds by proximity. *Comput. Aided Geom. Des. 22*, 7, 593–622.

ZORIN, D. 1998. Smoothness of stationary subdivision on irregular meshes. Technical Report CSL-TR-98-752, Stanford University, Computer Systems Laboratory, Jan.