

Postprocessing of Compressed 3D Graphic Data

Ka Man Cheang, Wenlong Dong, Jiankun Li, and C.-C. Jay Kuo

*Department of Electrical Engineering-Systems, Integrated Media Systems Center,
University of Southern California, Los Angeles, California*

E-mail: kcheang@sipi.usc.edu, wdong@sipi.usc.edu, jiankunl@sipi.usc.edu, cckuo@sipi.usc.edu

Received September 17, 1998; accepted July 6, 1999

This work presents a postprocessing technique applied to a 3D graphic model of a lower resolution to obtain a visually more pleasant representation. Our method is an improved version of the modified butterfly subdivision scheme developed by Zorin and his colleagues. Our main contribution is to exploit the flatness information of local areas of a 3D graphic model for adaptive refinement in relatively coarser areas. We can avoid unnecessary subdivision in regions which are smoother. Consequently, the new algorithm reduces the computational complexity and also saves the storage space for many models. Our algorithm can also remove overshoots around sharp edges and retain sharp features. © 2000 Academic Press

Key Words: 3-D meshes; postprocessing; adaptive subdivision; modified butterfly subdivision.

1. INTRODUCTION

3D graphic models represented by polygonal meshes have become increasingly popular due to the fast development of the Internet and the standardization of VRML (Virtual Reality Modeling Language). However, the complexity of graphic models has increased much faster than the throughput of today's graphics hardware and computer networks. In the past several years, several algorithms for mesh simplification and compression have been applied successfully to level-of-detail (LOD) generation and progressive transmission [7, 12, 18, 19]. A quite different approach was adopted in [6, 11], where classical multiresolution analysis and subdivision techniques were extended to the adaptive parameterization of mesh surfaces.

Due to the loss of information in these simplification techniques, a graphic model may become very coarse and have an inferior appearance in some areas, especially in the case of high compression ratios. Quality degradation occurs because the compressed model usually contains a fewer vertices and polygons than the original one and because geometry data, such as vertex positions and normal vectors, are often represented with less numerical precision than their raw values. A solution to this problem is to perform postprocessing on the compressed model to enhance its visual quality. By postprocessing, we mean the way to use surface subdivision techniques on the simplified model to obtain a finer and smoother

mesh. We see at least two applications of mesh postprocessing, i.e., mesh browsing over the Internet and LOD rendering. When meshes are transmitted through the Internet, visual quality is greatly degraded because of the high compression ratio required. By applying postprocessing, meshes can have more pleasant visual quality at relatively coarser areas. For LOD rendering, we can use relatively coarser mesh when the object is far from the viewer. If the object is closer, coarser areas can be further subdivided for a better resolution.

There are two major subdivision methods to achieve C^1 -continuous limit surfaces: one is the butterfly scheme pioneered by Dyn *et al.* [3], which is an interpolation scheme and the other is Loop's scheme [13] which is an approximating scheme for a given model. The butterfly scheme of Dyn *et al.* only considered the subdivision over triangular meshes in which every vertex is regular (i.e., with six neighbors). Zorin *et al.* [21, 22] proposed a modified butterfly subdivision (MBS) scheme on an arbitrary topology with extraordinary points. MBS constructs a smooth surface that encloses the original mesh and retains old vertices. Thus, it is an interpolating scheme. In contrast, Loop's scheme obtains a shrunk model with the old mesh as an envelope. Both schemes are very fast, since only linear interpolation of local vertices is used. The advantage of MBS over Loop's scheme is that it allows adaptive subdivision in local areas as presented in this paper, and it may also allow dynamic subdivision which shows view-dependent features similar as discussed in [8].

Based on MBS, we propose a new postprocessing algorithm for 3D meshes by using a selective subdivision algorithm in this research. Our main contribution is the development of a new adaptive subdivision method by using the local flatness information to perform selective refinement. To be more specific, by defining the flatness of two neighboring faces, we specify coarser areas in the 3D mesh for local subdivision. Furthermore, the extraordinary behavior at boundaries of subdivided areas is carefully analyzed, and irregular cases in MBS are improved to lead to a robust algorithm. We have tested our algorithm on a variety of models and observed very good results for most models. With our new algorithm, a large number of relatively smoother areas are not subdivided. This technique is able to reduce the time for mesh updating and rendering and the storage space while preserving a similar visual effect.

2. REVIEW OF UNIFORM SUBDIVISION

The butterfly scheme [3] is a local interpolating scheme. It is local in the sense that it requires only a small neighborhood (8-point stencil with the shape of a butterfly) to calculate values of new vertices and generates a smooth surface over triangular meshes with regular vertices (i.e., vertices with six neighbors). However, it exhibits undesirable artifacts for irregular meshes and the surface smoothness can be lost. Zorin *et al.* [21] developed a MBS scheme that retains the simplicity of the butterfly scheme and provides a smoother surface for meshes of arbitrary topology by using the discrete Fourier transform analysis. The scheme extends the 8-point butterfly neighborhood to a 10-point stencil for regular points as shown in Fig. 1 and finds a new formula for extraordinary vertices. Due to the nature of the construction, all new vertices are of valence 6 and the limiting surface is thus of C^1 -continuity.

The MBS scheme can be specified under four cases:

1. *Edge connects two regular points.* The neighborhood is a 10-point stencil as shown in Fig. 1a. The weights for the new vertex are given by $a = 1/2 - w$, $b = 1/8 + 2w$, $c = -1/16$, and $d = w$.

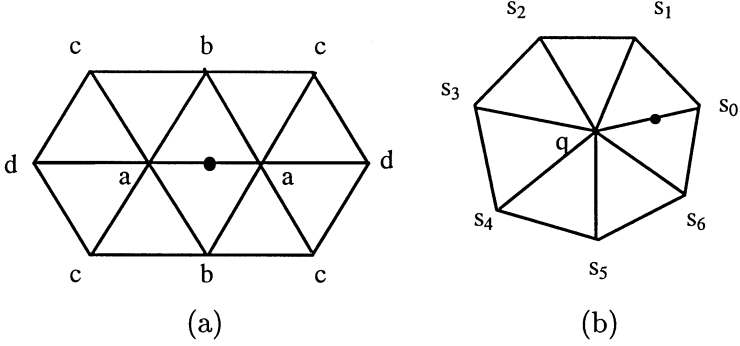


FIG. 1. (a) The 10-point stencil and (b) the 1-neighborhood stencil for a 7-vertex, where the black dot indicates the spot of a new vertex obtained by splitting the corresponding edge.

2. *Edge connects a K -vertex ($K \neq 6$) and a regular vertex.* We show the neighborhood of the K -vertex with $K = 7$ in Fig. 1b, which consists of vertices with distance 1. Weights for these 1-neighbor vertices are chosen as follows. For $K = 3$, $s_0 = 5/12$, $s_1 = s_2 = -1/12$. For $K = 4$, $s_0 = 3/8$, $s_1 = s_3 = 0$, $s_2 = -1/8$. For $K \geq 5$, $s_j = (1/4 + \cos(2\pi j/K)) + 1/2 \cos(4\pi j/K)/K$. In all cases, the weight for the central K -vertex is $q = 1 - \sum s_j = 0.75$.

3. *Edge connects two extraordinary vertices.* The position of the new vertex is the average of the values computed using the above formula for each endpoint.

4. *Boundary edge.* 1-dimensional 4-point scheme is used: $s_{-1} = -1/16$, $s_0 = 9/16$, $s_1 = 9/16$, $s_2 = -1/16$.

Before describing the proposed selective subdivision process, let us address the problem of exponential growth in the computational complexity and the storage requirement due to uniform subdivision. For uniform subdivision, each triangle is divided into four new faces at each subdivision level. The number of vertices also increases by approximately four times. Even with a small model, we will have difficulty in performing several levels of subdivision. In the next section, we propose the use of local neighborhood flatness information to achieve selective subdivision to solve this problem.

3. SELECTIVE SUBDIVISION

3.1. Subdivision Based on Flatness

Our basic idea is simple. Edges that are relatively flat are skipped so that no subdivision is performed on flat patches. Only triangles in a visually rough surface are subdivided. Thus, by defining a meaningful decision rule on surface flatness, we are able to skip a large amount of flat triangles without performing subdivision on them and concentrate on subdividing a small portion of the surface to give a more pleasant visual effect.

The flatness information of an edge is calculated by using the unit normal vectors of its two adjacent triangles. Suppose that the dihedral angle between two adjacent faces is θ and the two corresponding unit normal vectors are \mathbf{n}_1 and \mathbf{n}_2 . Then, the *flatness* of the joint edge can be defined as

$$\tau = -\cos \theta = \mathbf{n}_1 \cdot \mathbf{n}_2,$$

providing that \mathbf{n}_1 and \mathbf{n}_2 point to the same side of the connected patch. The initial direction

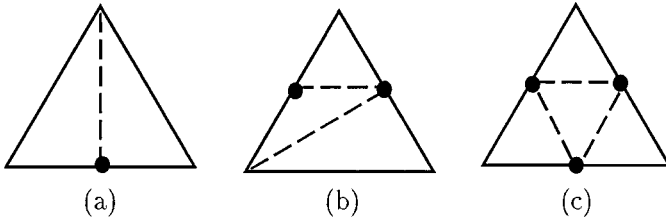


FIG. 2. Triangles with one, two, or three target edges are split into two, three, or four new patches.

of the normal of each face is defined as the outside of the surface by the corresponding order of vertices in VRML files. Thus, we have to be careful about the directions of new faces after subdivision.

We have $\tau \in [-1, 1]$. It is clear that the part at the edge of the patch is smoother when τ is large while the edge is in a rough area if τ is small. $\tau = 0$ implies that the two faces are perpendicular to each other. Thus, given a threshold t , we can classify an edge as *flat* if

$$t < \tau < 1.$$

Otherwise, it is classified as a *target* edge for subdivision. A target edge can be subdivided by MBS and a new vertex is obtained along the edge.

When a triangle has target edges, it can have one, two, or three target edges. After appropriately connecting new and old vertices, a triangle is replaced by a new patch of two, three, or four faces. When a triangle has just one target edge, we can simply connect the new inserted point with the old vertex residing in the opposite of the edge. When a triangle has two target edges, we connect the two new vertices and link one of the new vertices with the opposite old vertex. If a triangle has three target edges, the old triangle can be simply split into four. These situations are shown in Fig. 2.

As observed in experiments, very good results are obtained by choosing threshold t from 0.75 to 0.95. The choice depends on topological structures of different models. Generally, for simple models such as those under the first one or two levels of subdivision, we choose smaller t and then gradually increase t for finer subdivisions. By limiting the threshold t to 0.75 or above, we allow the dihedral angle between two triangles to be classified as flat by having an incident angle of 139° or above.

4. FURTHER MODIFICATIONS

4.1. Problem of Successive Edge Subdivision

Directly applying the above scheme several times to a mesh may result in successive subdivision on the same edge of the original triangle. As illustrated in Fig. 2a, since new vertices are connected with the opposite old vertex in the case of a triangle with only one target edge, the smoothness may not be recovered by subdivision. Instead, there may be a loss in efficiency by introducing more faces and vertices. One such example is shown in Fig. 3b. To solve this problem, we introduce a new way to mark target edges. After edges are marked as targets by using the flatness initially, we further mark the 1-neighboring edges of each previously marked edge for subdivision. This method can avoid the case of subdividing an old edge again and again. It also prevents a vertex from connecting to too many other vertices. This change greatly improves the appearance of the subdivided mesh even if we

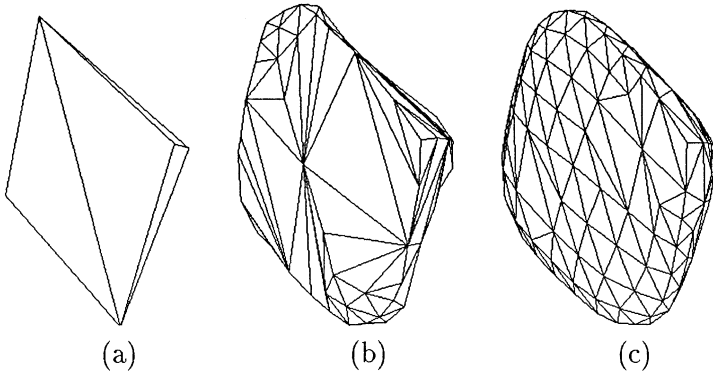


FIG. 3. (a) The original coarse mesh, (b) the subdivided mesh obtained by using the scheme described in Section 3, and (c) the subdivided mesh obtained by using the scheme described in Section 4.1. Three levels of subdivision operations were performed for these two cases with thresholds $t = 0.8, 0.85,$ and $0.9,$ respectively.

select a much smaller threshold t for the flatness. The result of the modified scheme is shown in Fig. 3c. When the mesh is relatively large (of several hundreds of vertices and faces) and most areas are smooth, this method can still retain the effect of saving vertices and faces.

4.2. Problem of Overshoot

We performed the above algorithm on many models such as Bunny and Dinosaur. The visual effect of final results looks almost the same as those obtained by using the uniform subdivision (see results in Section 5). Zorin *et al.* also got very good results in [21] and [22]. However, when we work on some simple models with very sharp edges, MBS attempts to construct a quadratic surface to interpolate the original mesh and, consequently, the refined mesh suffers from the overshoot problem as shown in Fig. 4. This phenomenon is extremely undesirable when the original mesh is greatly simplified. A practical example of the overshoot problem is given in Fig. 5b. Kobbelt *et al.* [10] addressed the same problem for the 1-dimensional case and used the variational subdivision scheme to handle the overshoot problem.

The overshoot phenomenon results from the discontinuity of the surface around sharp edges, which is the same as the 1-D case. As shown in Fig. 6, when we split edge $AC,$ the

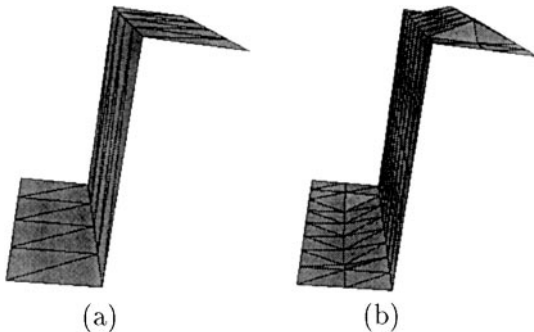


FIG. 4. The overshoot phenomenon associated with MBS: (a) the original mesh and (b) the subdivided mesh with MBS.

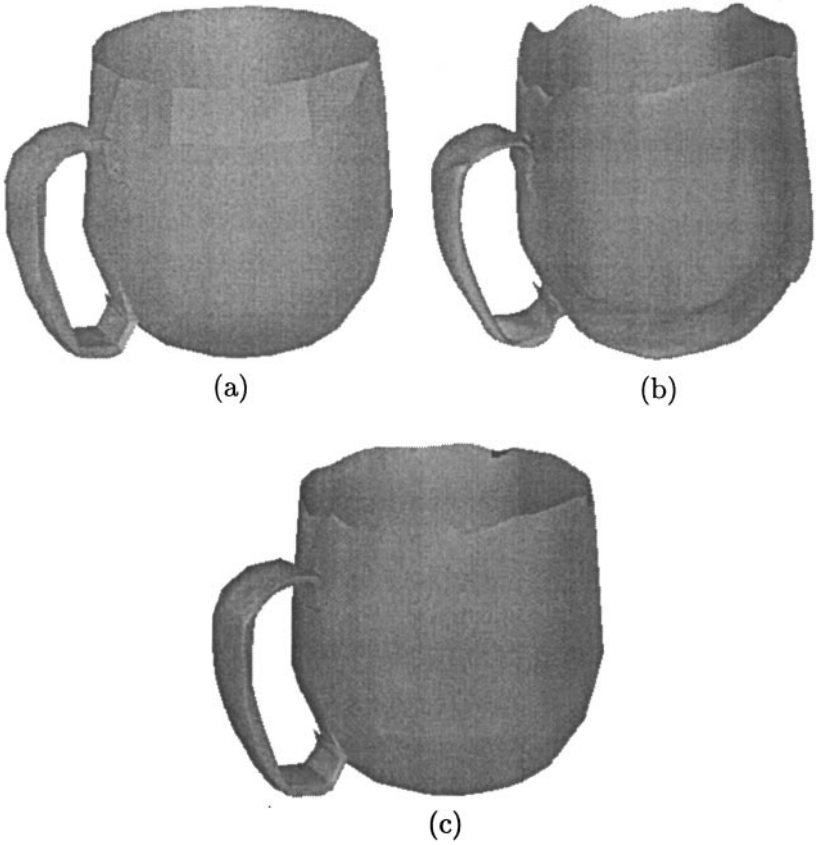


FIG. 5. The overshoot problem for the Mug model: (a) the simplified mesh, (b) subdivision with MBS, and (c) postprocessing by using the tension factor.

1-neighboring vertices of A are used to interpolate this new vertex. Suppose that the two adjacent triangular faces $\triangle ACB$ and $\triangle ADC$ of AC have unit normals \mathbf{n}_1 and \mathbf{n}_2 , respectively. Let \mathbf{n}_e be the unit vector in the direction of $\mathbf{n}_1 + \mathbf{n}_2$. Then, the plane determined by the edge AC and parallel to \mathbf{n}_e equally divides the dihedral angle of the two triangular faces. When we project the 1-neighboring vertices of A onto this plane, the reason of the overshoot phenomenon becomes clear. For example, let us consider vertex E , which is the 1-neighboring

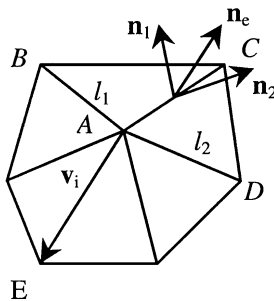


FIG. 6. Splitting AC by using the tension factor.

vertex of A . When the projection of E is lower than the horizontal line AC , it pushes the new vertex obtained by subdividing AC higher to smooth the corner. On the other hand, if E is higher than AC , it pushes the new vertex lower. Thus, in order to reduce the overshoot phenomenon, we should reduce the interpolating coefficients of those vertices with larger projections in the direction of \mathbf{n}_e . This can be achieved by introducing *tension factors* for 1-neighboring vertices of a K-vertex. Let $\mathbf{v}_i = \vec{AE}$, l_1 and l_2 be the lengths of edges AB and AD . For a given threshold a ($a = 0.2$ was used in the experiment), we can define the tension factor of E with respect to A via

$$t_E = \begin{cases} 1 & \text{if } b_E \leq a, \\ C\alpha^{-b_E}, & \text{otherwise,} \end{cases} \quad (1)$$

where C is a constant less than 1, α is a number greater than 1, and

$$b_E = \frac{2|\mathbf{n}_e \cdot \mathbf{v}_i|}{l_1 + l_2}.$$

It is easy to see that the larger the projection of \mathbf{v}_i onto \mathbf{n}_e is, the larger b_E and the smaller t_E . The tension factor t_E decreases in an exponential speed as b_E increases. Thus, if b_E is larger than threshold a , we classify vertex E to one that contributes to the overshoot problem. Then, coefficient s_i (with $i \neq 0$, i.e., not including the vertex C) at E with respect to A as defined in MBS subdivision is reduced to $t_E s_i$.

In our implementation, $C = 0.8$ and $\alpha = 5$ were adopted. In order to satisfy the smooth requirements, we renormalize all weights of the K-vertex A so that the sum of all weights is still equal to 1. Then when α increases, the new vertex approaches the midpoint of the edge AC . The modified algorithm is usually able to remove the overshoots around sharp edges. The result for the Mug model with the modified algorithm is shown in Fig. 5c.

4.3. Sharp Feature Preservation

In order to get more realistic visual effects, it is desirable to preserve sharp features for graphic models. Hoppe *et al.* [6] modified Loop's subdivision rules locally to model sharp features such as creases and corners specified by users. We choose a different way to do this by using the tension factor defined in Section 4.2. This is automatically done by specifying different parameters for the tension factor. When a decreases, we can get more and more sharp corners with b_E greater than a . When α increases, weights of steeper neighbors have less absolute values so that the new vertex approaches the midpoint of the corresponding edge. From the results of the model Mug given in Fig. 5, we see that sharp edges are retained with the proposed algorithm while relatively coarse regions are also smoothed.

5. EXPERIMENTAL RESULTS

We performed our postprocessing algorithm on many graphic models. Our experiment was performed on a Pentium-II 450 MHz with 128 MB memory. The total processing time of one level of subdivision including the rendering time was about 1 to 5 s for models of about 10,000 vertices. The time for computing the tension factor is negligible in comparison with the rendering time.

TABLE 1
Results of Different Subdivision Schemes

Model	Bunny			Dinosaur			Mug		
	Vertex	Face	Level	Vertex	Face	Level	Vertex	Face	Level
Original	34834	69451	—	2832	5647	—	1725	3450	—
Simplified	268	500	—	409	801	—	172	344	—
Uniform MBS	4078	8000	2	25715	51238	3	11008	22016	3
Our scheme	2613	5141	2	11679	23229	3	3447	6894	3
	$t = 0.75, 0.85$			$t = 0.75, 0.85, 0.90$			$t = 0.75, 0.85, 0.90$		

Table 1 summarizes the resulting number of vertices and faces, the corresponding thresholds t , and the levels of subdivision (if any) for three models, i.e., Bunny, Dinosaur, and Mug. We see from this table that there are about 40 to 80% of faces and vertices can be saved with our new method in comparison with uniform subdivision without loss of visual quality. By gradually increasing thresholds for different levels of subdivision, we see more coarser patches are smoothed. We got very good results when t ranges from 0.75 to 0.95. Our method tends to get more savings for simplified models of about 1000 to 5000 vertices with some local rough areas.

Figures 5 and 7 show the results of Mug and Cannon, which are all subdivided three times with thresholds $t = 0.75, 0.85$, and 0.90 , respectively. From Fig. 5a, we see that the Mug model is very simple with only 172 vertices and 344 faces. After subdivision with MBS, the top boundary of the Mug is rugged because of overshoots at the first two levels as shown in Fig. 5b. The result obtained by using the tension factor is given in Fig. 5c, whose top boundary is much smoother. In Fig. 5, we illustrate the sharp feature preserving capability by using our new algorithm for the Cannon model given in Fig. 7a. The MBS scheme smoothes out all sharp edges with severe overshoots in the first level of subdivision for this model as shown in Fig. 7b. Figure 7c gives the result of using the tension factor to control the overshoots around sharp corners and to preserve sharp features. It is clear that the result is visually more similar to the original one.

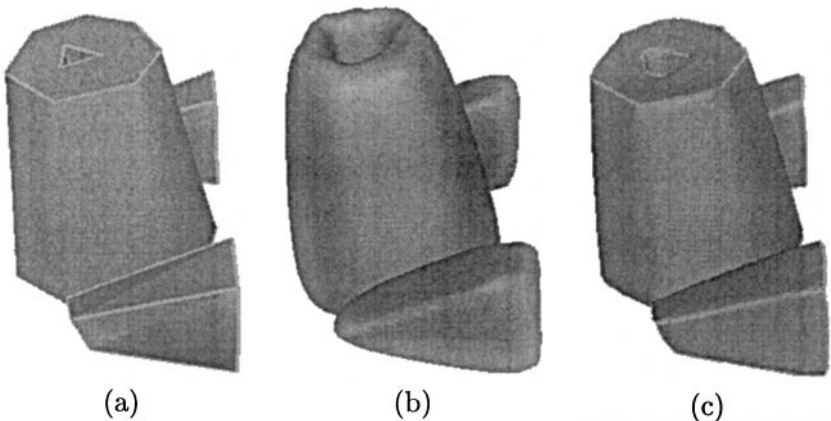


FIG. 7. The edge-preserving subdivision for the Cannon model: (a) sharp edges shown on the simplified model, (b) subdivision with MBS, and (c) postprocessing by using the tension factor.

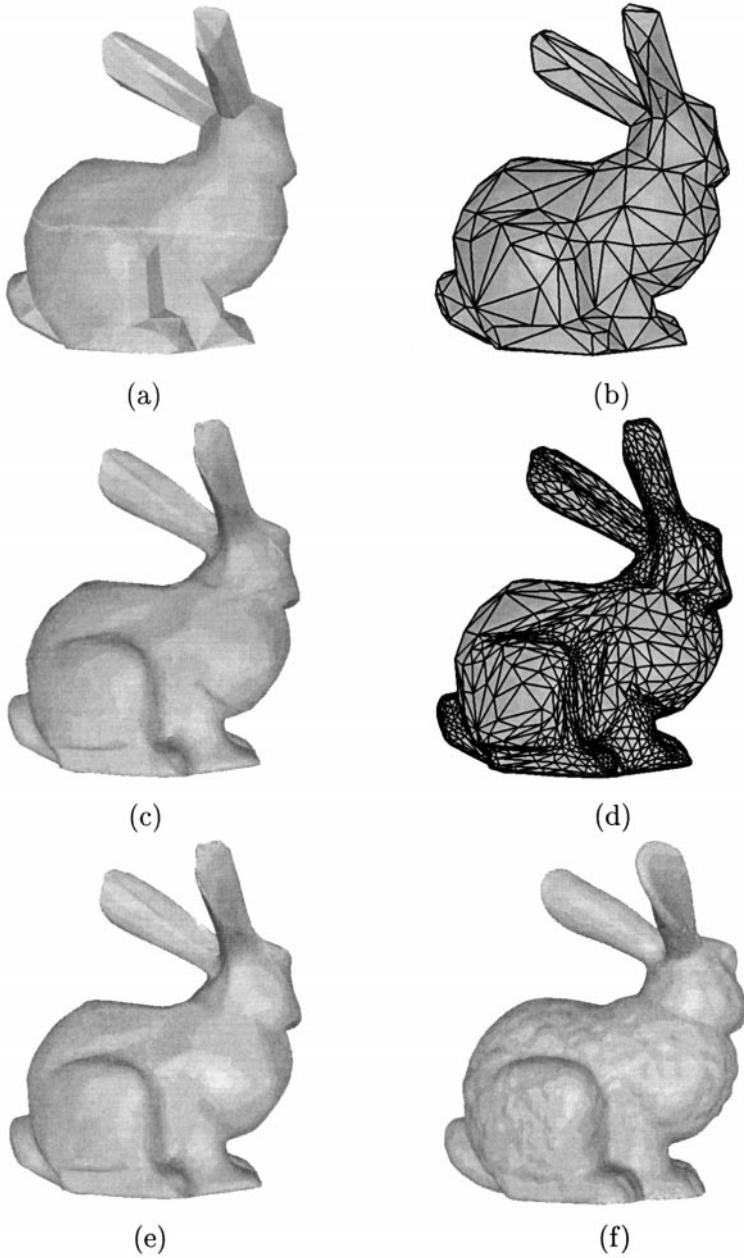


FIG. 8. Results for the Bunny model: (a) rendered with the simplified mesh, (b) the simplified mesh with wireframe, (c) rendered with the 2-level adaptive subdivision, (d) rendered with wireframe of (c), (e) rendered with the 2-level uniform subdivision, and (f) rendered with the original finest mesh.

We show two more graphic models, i.e., Bunny and Dinosaur, in Figs. 8 and 9, respectively. For each graphic model, the rendered image with the simplified mesh and the corresponding wireframe, the rendered images with the adaptive subdivided mesh and the corresponding wireframe, the rendered images with uniform subdivision and the original finest mesh are given. The postprocessing of the compressed Bunny model of compression ratio 138 : 1 was performed by taking $t = 0.75$ and 0.85 , respectively, for the first two levels

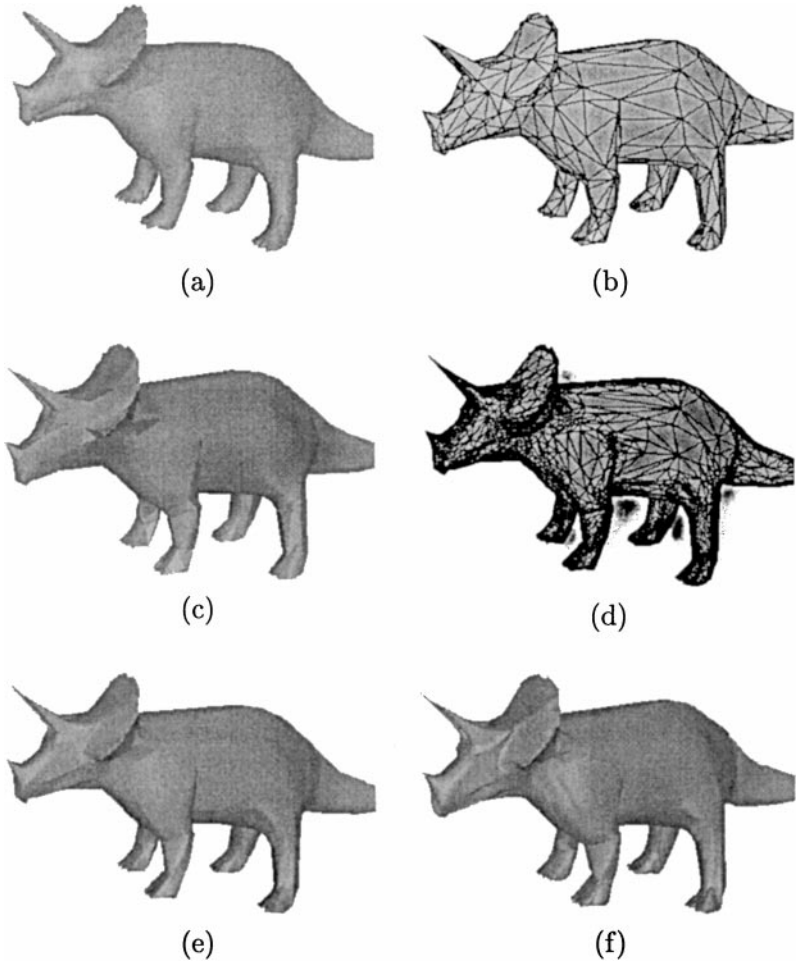


FIG. 9. Results for the Dinosaur model: (a) rendered with the original mesh, (b) the simplified mesh with wireframe, (c) rendered with the simplified mesh, (d) the adaptive subdivision with wireframe, (e) rendered with adaptive subdivision, and (f) rendered with uniform subdivision.

of subdivision. In other words, edges with incident dihedral angles greater than 154° are skipped. Note that triangles in flat regions such as on the back and at the side are skipped, while those at ears, legs, and feet are subdivided much more.

6. CONCLUSION AND FUTURE WORK

We provided a postprocessing technique for a generic 3-D mesh of triangular faces based on the MBS scheme [21, 22]. Adaptive subdivision was used and local flatness of edges was defined. We also introduced the tension factors to solve the problem of overshooting at sharp corners and to retain sharp features. Final results were obtained and compared with those with uniform subdivision. The number of vertices and faces can be saved from 40 to 80% with our new method.

We plan to extend our postprocessing to view-dependent rendering of 3-D meshes. The viewing information and lighting effects can be used to further reduce the number of faces

and vertices. We are currently working on models with only one attribute data, i.e., the vertex position. It is interesting to consider other attribute information such as colors and textures.

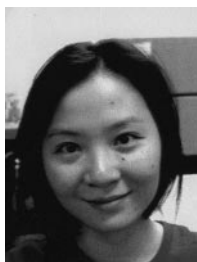
ACKNOWLEDGMENTS

This research has been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center with additional support from the Annenberg Center for Communication at the University of Southern California and the California Trade and Commerce Agency. The authors also thank the referees for their helpful comments.

REFERENCES

1. E. Catmull and J. Clark, Recursively generated B-spline surfaces on arbitrary topological meshes, *Comput. Aided Design* **10**, 1978, 350–355.
2. D. Doo, A subdivision algorithm for smoothing down irregularly shaped polyhedrons, in *Proceedings on Interactive Techniques in Computer Aided Design, Bologna, 1978*, pp. 157–165.
3. N. Dyn, D. Levin, and J. A. Gregory, A butterfly subdivision scheme for surface interpolation with tension control, *ACM Trans. Graphics* **9** 2, 1990, 160–169.
4. N. Dyn, S. Hed, and D. Levin, Subdivision schemes for surface interpolation, in *Workshop in Computational Geometry* (A. C. *et al.*, Eds.), pp. 97–118, World Scientific, Singapore, 1993.
5. M. Eck and H. Hoppe, Automatic reconstruction of B-spline surfaces of arbitrary topological type, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 325–334, 1996.
6. H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, Piecewise smooth surface reconstruction, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 295–302, 1994.
7. H. Hoppe, Progressive meshes, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 99–108, 1996.
8. H. Hoppe, View-dependent refinement of progressive meshes, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 189–198, 1997.
9. L. Kobbelt, Interpolatory subdivision on open quadrilateral nets with arbitrary topology, *Comput. Graphics Forum* **15**, 1996, 409–420. [Eurographics issue]
10. L. Kobbelt and P. Schröder, *Constructing Optimal Curves Using Subdivision*, Tech. Rep. CS-TR-97-05, Department of Computer Science, Caltech, 1997.
11. A. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, MAPS: Multiresolution adaptive parameterization of surfaces, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 95–104, 1998.
12. J. Li and C.-C. J. Kuo, Progressive coding on 3D graphic models, *Proc. IEEE* **86**, 1998, 1052–1063.
13. C. Loop, *Smooth Subdivision Surfaces Based on Triangles*, Master Thesis, Department of Mathematics, University of Utah, 1987.
14. C. Loop, Smooth spline surfaces over irregular meshes, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 303–310, 1994.
15. M. Lounsbery, *Multiresolution Analysis for Surfaces of Arbitrary Topological Type*, Ph.D. dissertation, University of Washington, 1993.
16. D. Luebke and C. Erikson, View-dependent simplification of arbitrary polygonal environments, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 199–208, 1997.
17. J. Peters, C^1 surface splines, *SIAM J. Numer. Anal.* **32**, 1995, 645–666.
18. W. J. Schroeder, Decimation of triangle meshes, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 65–70, 1992.
19. D. Taubin and J. Rossignac, Geometric compression through topological surgery, *Proc. IEEE* **86**, 1998, 1228–1243.
20. J. Xia and A. Varshney, Dynamic view-dependent simplification for polygonal models, in *Visualization '96 Proceedings*, IEEE, pp. 327–334, 1996.

21. D. Zorin, P. Schröder, and W. Sweldens, *Interpolating Subdivision for Meshes with Arbitrary Topology*, Tech. Rep. CS-TR-96-06, Department of Computer Science, Caltech, 1996.
 22. D. Zorin, P. Schröder, and W. Sweldens, Interpolating subdivision for meshes with arbitrary topology, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 189–192, 1996.
 23. D. Zorin, P. Schröder, and W. Sweldens, Interactive multiresolution mesh editing, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 259–268, 1997.
-



KA MAN CHEANG received the B.S. in mathematics and computer science from the University of Wisconsin—Madison in 1995 and the M.S. in applied mathematics and electrical engineering from the University of Southern California in 1998. Her current research interests include graphic compression, postprocessing of 3D graphic data, and multiresolution graphics editing.



WENLONG DONG received the B.S. in applied mathematics from Shanghai Jiao Tong University, Shanghai, China, in 1990 and the M.S. in probability and statistics in East China Normal University, Shanghai, China, in 1992. He worked as a lecturer in the Department of Applied Mathematics in Shanghai Jiao Tong University from 1992 to 1996. He is currently pursuing his Ph.D. in the Department of Electrical Engineering—Systems at the University of Southern California. His research interests include postprocessing of 3D graphics data, subdivision surfaces, mesh simplification, and 3D multiresolution editing.



JIANKUN LI received the B.S. in physics from the University of Science and Technology, China, in 1993. He received the Ph.D. in the Department of Electrical Engineering—Systems at the University of Southern California in 1998. His research interests include cartoon/image compression, graphics compression, animation and watermarking, and multiresolution graphics rendering.



C.-C. JAY KUO received the B.S. degree from the National Taiwan University, Taipei, Taiwan in 1980 and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, Massachusetts in 1985 and 1987, respectively, all in electrical engineering. Dr. Kuo was Computational and Applied Mathematics (CAM) Research Assistant Professor in the Department of Mathematics at the University of California, Los Angeles, from October 1987 to December 1988. Since January 1989, he has been with the Department of Electrical Engineering—Systems and the Signal and Image Processing Institute at the University of Southern California, where he currently has a joint appointment as professor of electrical engineering and mathematics. His research interests are in the areas of digital signal and image processing, audio and video coding, wavelet theory and applications, multimedia technologies, and Internet and wireless communications. He has authored more than 380 technical publications in international conferences and journals. Dr. Kuo is a member of SIAM, ACM, and a fellow of IEEE and SPIE. He is Editor-in-Chief for the *Journal of Visual Communication and Image Representation* and served as associate editor for *IEEE Transaction on Image Processing* during 1995–1998 and *IEEE Transaction on Circuits and Systems for Video Technology* during 1995–1997. Dr. Kuo received the National Science Foundation Young Investigator Award (NYI) and Presidential Faculty Fellow (PFF) Award in 1992 and 1993, respectively.