High-Boost Mesh Filtering for 3-D Shape Enhancement

Hirokazu Yagou* Alexander Belyaev† Daming Wei‡
*,†,‡ Shape Modeling Laboratory, University of Aizu, Aizu-Wakamatsu 965-8580 Japan
† Computer Graphics Group, Max-Planck-Institut für Infomatik, 66123 Saarbrücken, Germany
E-mail: {m5051134, belyaev, dm-wei}@u-aizu.ac.jp, belyaev@mpi-sb.mpg.de

Abstract

In this paper, we propose a new technique for enhancing 3-D shapes approximated by triangle meshes: high-boost mesh filtering. Concepts based on the high-boost filter in signal and image processing are used for the development of the mesh filtering. An algorithm of the high-boost mesh filtering consists of affecting the high-boost filter to face normals on a triangle mesh and updating mesh vertex positions to make them adapt to the boosted normals. An enhancement operation is achieved by a procedure which subtract smoothed face normals from the original ones. To apply the high-boost filter to face normals on a triangle mesh, we require a low-pass filter for smoothing them. We use mean filtering applied to face normals in numerical experiments. The results indicate that the high-boost mesh filtering is a technique effective for 3-D shape enhancement.

Keywords: *triangle meshes, mesh filtering, shape enhancement, high-boost filter, normal-based error minimization.*

Technical Area: *3-D Computer Graphics*

1 Introduction

The *high-boost filter* is a simple sharpening operator in signal and image processing. It is used for amplifying high frequency components of signals and images. The amplification is achieved via a procedure which subtracts an smoothed version of the media data from the original one. In image processing, we can sharpen edges of a image through the amplification and obtain a more clear image. If the filter is applied to a 3-D triangle mesh, its surface textures are enhanced. Such the enhancement operation gives us more attractive 3-D shape models.

Several techniques for enhancing 3-D shapes have been proposed in computer graphics so far. Guskov et al. [1] apply a similar algorithm directly to vertices. They developed

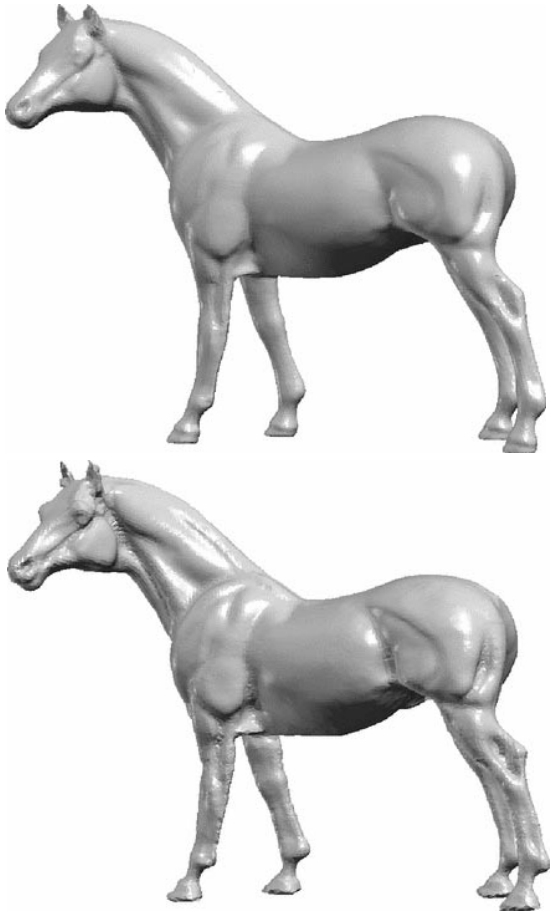
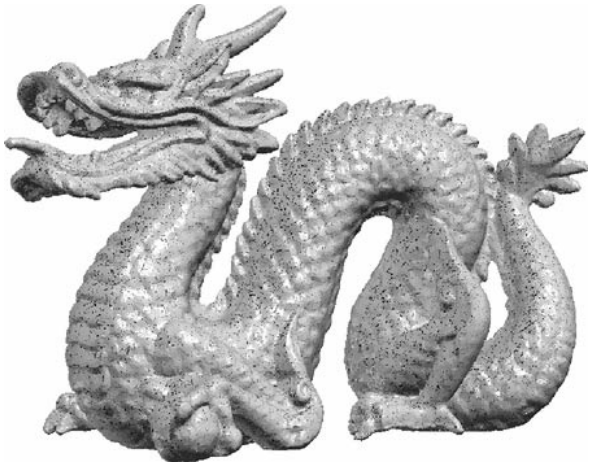
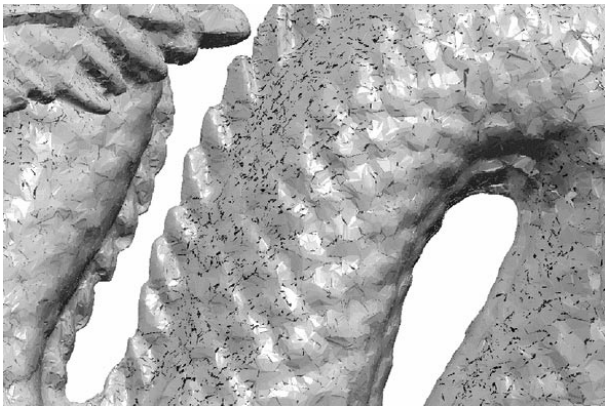


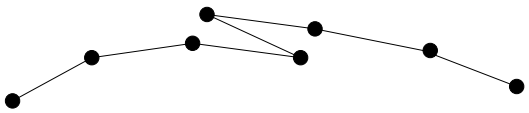
Figure 1. A horse model. Top: original one. Bottom: enhanced by the high-boost mesh filtering.



(a)



(b)



(c)

Figure 2. (a) Stanford dragon enhanced by vertex-based enhancement. (b) Black areas are flipping triangles. (c) A conceptual figure of the triangle flipping.

the algorithm from complex concepts based on multiresolution signal processing. Tolga et al. [6] applied high-boost filter to face normals on a surface represented as a level set of volumes. After the filtering, a level set method that is computationally expensive is used for manipulating the surface in order to fit it into the processed normals.

Our approach first applies the high-boost filter to face normals on a enhanced 3-D triangle mesh. Second, mesh evolution by updating vertex positions is performed via minimizing errors between the original and filtered normals [3]. The purpose of the mesh evolution is to adapt the mesh to the processed normals.

When a filtering technique is directly applied to mesh vertices, triangle flipping frequently occurs. We can prevent filtering results from the triangle flipping with a combination of filtering face normals and evolving mesh vertices via the normal-based error minimization. Figure 2 shows an example of the triangle flipping.

An enhancement with the high-boost mesh filtering is accomplished by a procedure that subtract smoothed face normals from the original ones. This means that we need a process for smoothing the face normal field on a triangle mesh. Our development realizes the smoothing process by mean filtering for averaging face normals at local neighborhood on a triangle mesh. Experimental results show that the high-boost mesh filtering method stably works without inducing triangle flipping.

This paper is organized as follows. In section 2, we introduce implementation details of the high-boost mesh filtering. We demonstrate experimental results in section 3, and they are discussed in section 4. Our conclusions are given in section 5.

2 Implementation Details

An implementation of the high-boost mesh filtering is discussed in this section. We first prepare smoothed face normals with mean filtering for them. Second, a new face normal field on a triangle mesh is defined using a boost threshold, the original normals, and the smoothed normals. Finally mesh vertex positions are updated via minimizing errors between the boosted normals and the original ones. A combination of the three steps achieves a 3-D shape enhancement.

It is first discussed how to smooth face normals using the mean filtering. Consider an oriented triangle mesh. Let T be a mesh triangle, $\mathbf{m}(T)$ be the unit normal of T, and $A(T)$ be the area of T. Denote by $\mathcal{N}(T)$ a set of all mesh triangles sharing the same edge or vertex with T. Figure 3 visually shows the $\mathcal{N}(T)$.

One iteration of the mean filtering on the face normal field is composed of the following two steps.

Step 1. For each mesh triangle T, we compute the face

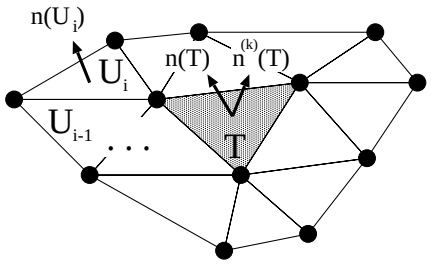


Figure 3. The center triangle T and neighboring triangles U_i.

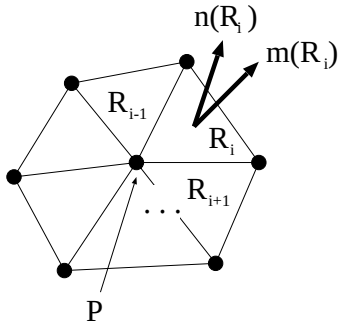


Figure 4. The 1-ring neighborhood of the P.

normal $\mathbf{n}(T)$ and perform the following area-weighted averaging operation:

$$\mathbf{n}^{(k)}(T) = \frac{1}{\sum A(U_i)} \sum_{i \in \mathcal{N}(T)} A(U_i)\mathbf{n}(U_i) \tag{1}$$

where $\mathbf{n}^{(k)}(T)$ is a face normal smoothed by k iterations.

Step 2. For each mesh triangle T, we normalize the averaged normals $\mathbf{n}^{(k)}(T)$:

$$\mathbf{n}^{(k)}(T) \longleftarrow \frac{\mathbf{n}^{(k)}(T)}{\|\mathbf{n}^{(k)}(T)\|}.$$

We can obtain the smoothed normals through the two steps.

Second, let us consider a high-boost filtering operation applied to face normals. Let $\mathbf{n}(T)$ be the original face normals and $\mathbf{n}^{(k)}(T)$ be face normals smoothed by k iterations. A operation of the high-boost filtering on the face normal field is defined by

$$\mathbf{m}(T) = \frac{(1+\alpha)\mathbf{n}(T) - \alpha\mathbf{n}^{(k)}(T)}{\|(1+\alpha)\mathbf{n}(T) - \alpha\mathbf{n}^{(k)}(T)\|} \tag{2}$$

where $\mathbf{m}(T)$ is the boosted normals, and α is the boost threshold. This operation defines a new unit vector field $\{\mathbf{m}\}$.

Finally, details of optimizing mesh vertex positions via normal-based error minimization [3] is presented. Let P be a updated vertex and R_i be a mesh triangle adjacent to the P. The normal-based error at the P is defined as the area-weighted summation of squared differences of $\mathbf{n}(R_i)$ and $\mathbf{m}(R_i)$:

$$E_n = \sum_{i \in \mathcal{F}_1(P)} A(R_i)\big(\mathbf{n}(R_i) - \mathbf{m}(R_i)\big)^2. \tag{3}$$

where $\mathcal{F}_1(P)$ is a set of all triangles at the 1-ring neighborhood of the P. Its partial derivative with respect P is given as follows:

$$\begin{aligned} \frac{\partial E_n}{\partial P} &= \sum_i 2\frac{\partial}{\partial P}A(R_i)\big(1 - \mathbf{n}(R_i)\cdot\mathbf{m}(R_i)\big) \\ &= \sum_i 2\left(\frac{\partial A(R_i)}{\partial P} - \frac{\partial A(S_i)}{\partial P}\right) \end{aligned} \tag{4}$$

where S_i is a triangle which is obtained by projecting R_i onto a plane defined by $\mathbf{m}(R_i)$. The summation is taken over all triangles in the $\mathcal{F}_1(P)$. The gradient of a triangle

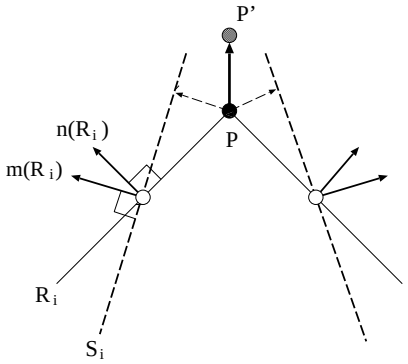


Figure 5. S_i is a projection of R_i onto a plane defined by a new face normal $\mathbf{m}(R_i)$.

area is computed by the following equation:

$$\frac{\partial A}{\partial P} = -\frac{1}{2}\big((P_1 - P)\cot\alpha + (P_2 - P)\cot\beta\big) \tag{5}$$

where A is an area of a triangle whose three vertices are $\{P, P_1, P_2\}$. This derivation of Equation (5) is described in [4]. The optimal vertex position P' is gotten by the following standard gradient descent method for minimizing E_n:

$$P' \longleftarrow P - \lambda \sum_i \left(\frac{\partial A(R_i)}{\partial P} - \frac{\partial A(S_i)}{\partial P}\right). \tag{6}$$

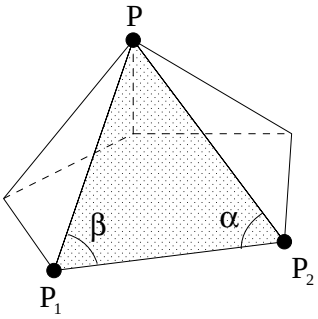


Figure 6. A triangle whose vertices are $\{P, P_1, P_2\}$ and its angles.

According to our experiments, Equation (6) stably works by setting $\lambda = 0.1$. When face normals on a triangle mesh are enhanced using the normals smoothed by k iterations, the updating operation should be performed by $10 \times k$ iterations. The evolution of vertex positions via normal-based error minimization requires many enough iterations.

3 Experimental Results

We check performance of the high-boost mesh filtering in this section. It is primary verified the filter's resistance to the triangle flipping.

The Stanford dragon model in Figure 7 is enhanced using the original face normals and ones smoothed by 100 iterations. The boost threshold is set to be 1.5. Its eyes and scales are well enhanced. Nevertheless, the triangle flipping does not occurs.

The Stanford bunny model in Figure 8 is enhanced by the original face normals and ones smoothed by 100 iterations. The boost threshold is set to be 1.5. In this case, mesh irregularization appears on the mesh model, as shown in Figure 8-(d). Further, aliasing occurs at edge parts, as seen in Figure 8-(f). We can suppress the aliasing and the mesh irregularization using the Laplacian smoothing [7, 2], as demonstrated in Figure 8-(e) and (g). In Figure 8-(e), a prominent aliasing part is marked by an ellipse.

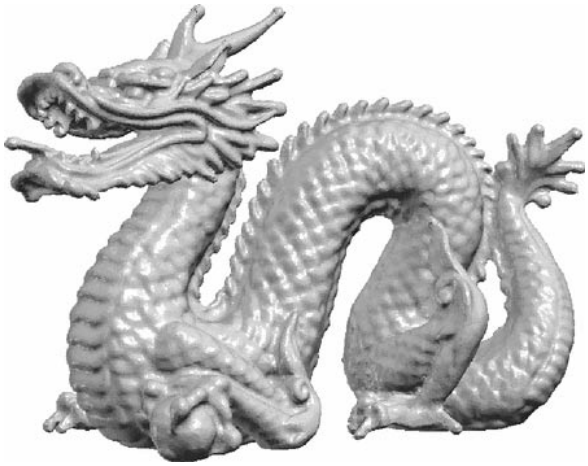
4 Discussion

When we apply the high-boost mesh filtering to some mesh models, the aliasing and the mesh irregularization occurs. Those problems are suppressed by the Laplacian smoothing, but the high-boost mesh filter should have functions preventing its filtering results from the aliasing and the mesh irregularization.

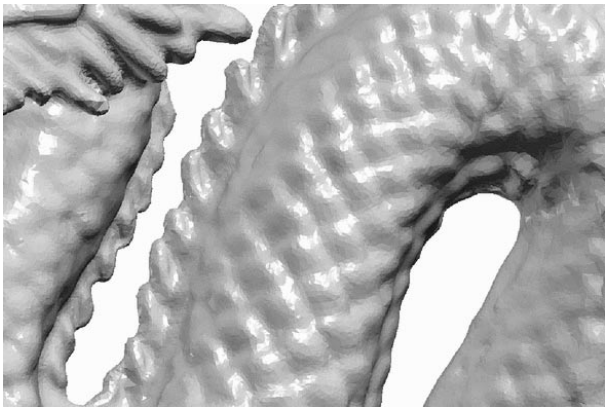
In this paper, we use the mean filtering applied to face normals to obtain the high frequency component of the normal field. For enhancing mesh models, we can consider



(a)



(b)



(c)

Figure 7. (a) Stanford dragon model. (b) Enhanced by the high-boost mesh filtering. (c) Triangle flipping does not occur.

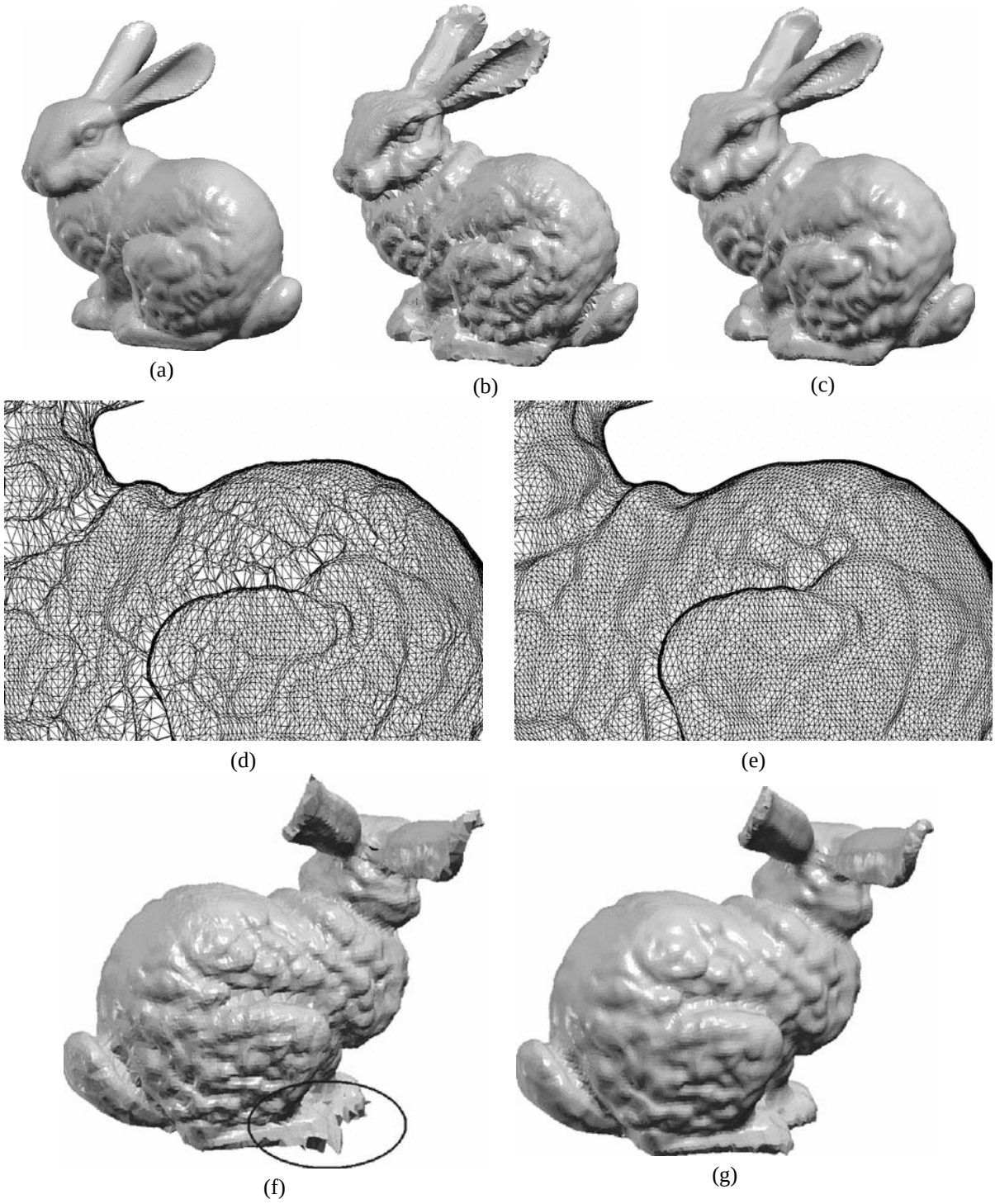


Figure 8. Stanford bunny model. (a) Original one. (b) Enhanced by the high-boost mesh filtering. (c) (b) smoothed by the Laplacian smoothing. (d) Mesh of (b). (e) Mesh of (c). (f) Aliasing. (g) Anti-aliasing by the Laplacian smoothing.

another approach such as the use of Laplacian on the face normal field. Taubin proposes Laplacian smoothing applied to face normals in [8]. In future research, we insert the smoothing method into the development of an improved high-boost mesh filtering algorithm.

5 Conclusion

In this paper, we applied the *high-boost mesh filtering* to 3-D mesh enhancement. The high-boost mesh filter is constructed by a combination of 1) the application of the high-boost filter to face normals on a triangle mesh and 2) the evolution of mesh vertex positions to fit them to the filtered normals. Numerical experiments show that the high-boost mesh filtering enhances 3-D triangle meshes with no significant triangle flipping.

References

[1] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. *Computer Graphics (Proceedings of SIGGRAPH99)*, pages 325-334, 1999.

[2] L. Kobbelt, S. Zamperori, J. Vorsatz, and H.-P. Seidel. Interactive multiresolution modeling on arbitrary meshes. *Computer Graphics (Proceedings of SIGGRAPH98)*, pages 105-114, 1998.

[3] Y. Ohtake. Mesh optimization and feature extraction. Ph.D Thesis, University of Aizu, March 2002.

[4] Y. Ohtake, A. Belyaev, and I. Bogaevski. Polyhedral surface smoothing with modified Laplacian and curvature flow. *The Journal of Three Dimensional Images*, Vol. 13, No. 3, September 1999, pages 19-24.

[5] Y. Ohtake, A. Belyaev, and A. Pasko. Dynamic meshes for accurate polygonization of implicit surfaces with sharp features. *International Conference on Shape Modeling and Applications*, Genova, Italy, May 2001, pages 134-141.

[6] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface processing via normal maps. Technical Report UUCS-02-003, University of Utah, January 2002.

[7] G. Taubin. A signal processing approach to fair surface design. *Computer Graphics (Proceedings of SIGGRAPH95)*, pages 351-358, 1995.

[8] G. Taubin. Linear Anisotropic Mesh Filtering. IBM Research Report RC22213 (W0110-051), IBM, October 2001.