

# An Iterative Method for Fast Mesh Denoising

Paper ID: 562

Category: Technical Paper

The 4th International Symposium on Visual Computing (ISCV08)

Las Vegas, Nevada, December 1-3, 2008

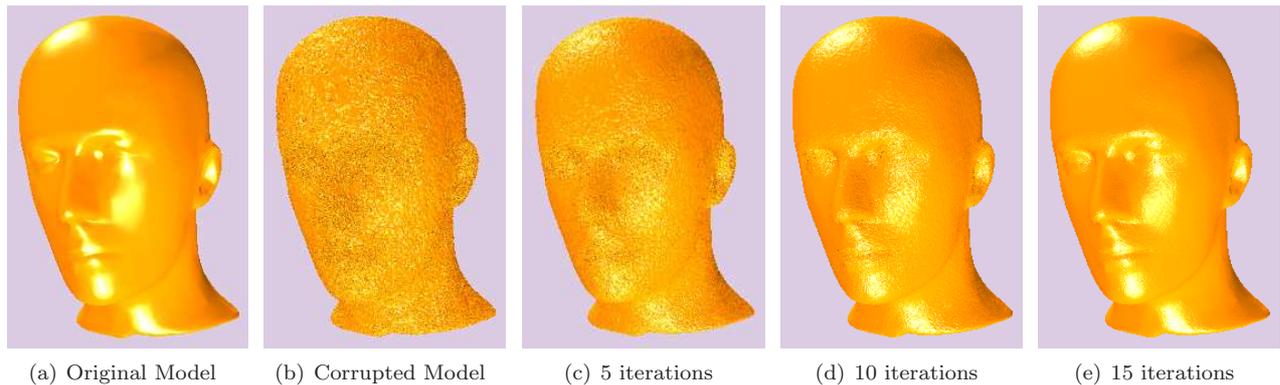


Figure 1: An example of noise removal using the new iterative method.

**Abstract** A new approach for removing noises from a corrupted 3D model (mesh or surface) of arbitrary topology is presented. The basic idea is to transform a space domain model into a frequency-like domain representation and achieve denoising by low pass filtering. The transformation from space domain to frequency domain is done by decomposing the 3D model into an infinite series of meshes of the same topology but less magnitude so that each mesh represents part of the information of the given model, with some meshes containing more information on overall shape while others containing more on subtle details. The transformation process does not require setting up any linear systems, nor any matrix computation, but is done by iteratively moving vertices of the given mesh locally until a smooth model with noises properly removed is reached. The iterative process converges at an exponential rate. Therefore the new iterative method is very fast and can be used for meshes with large number of vertices. The mesh decomposition scheme is obtained using the concept of Catmull-Clark subdivision surfaces, but the same idea can be applied to other subdivision schemes as well. Some test results obtained using this method are included. They show that the iterative method can achieve visually pleasant resulting models with noises properly removed.

## 1 Introduction

With the proliferation of 3D scanning devices, 3D shape reconstruction and visualization have become more and more important in computer graphics, reverse engineering, CAD, medical imaging, visualization, virtual reality, video games, and so on. However, data of those models are generally corrupted by measurement errors. Although fidelity of the scanners has improved dramatically over the past decade, noise is ever present in any practical system. For example, models extracted from CT or MRI scanning systems result in detailed models with significant amount of noises. Therefore, models containing noises need to be smoothed or denoised before any subsequent mesh processing such as surface collision or Boolean operations could be effectively applied. In the process of removing noises of a surface mesh, the shape of the input model could be changed. Hence it is very important to maintain the overall shape unchanged and keep the features of the surface itself as much as

possible.

Subdivision surfaces [1, 6, 8] have become popular recently in graphics, geometric modeling and computer animation because of their relatively high *visual quality*, *numerical stability*, *simplicity* in coding and, most importantly, their capability in modeling any complex shape with only one surface [4]. They are widely used for representing models of irregular topology. With multiresolution analyses [15, 16] for subdivision surfaces becoming available, subdivision surfaces are also used for dealing with corrupted sampled meshes [13, 14].

In this paper we describe a new iterative approach for removing noises from a noisy 3D model of arbitrary topology. The new method is easy to understand, easy to implement, and can achieve relatively good denoising results. The basic idea is to transform a space domain model into a frequency-like domain representation and achieve denoising by low pass filtering. The transformation from space domain to frequency domain is done by decomposing the 3D model into an infinite series of meshes such that each mesh represents low frequency or high frequency information of the shape of the given model. The transformation process does not require setting up any linear systems, nor any matrix computation, but is done simply by iteratively moving vertices of the given mesh locally until a smooth model with the noises properly removed is reached. The iterative process converges at an exponential rate. Therefore the new iterative method is very fast and can be used for meshes with large number of vertices. The mesh decomposition scheme is obtained using the concept of Catmull-Clark subdivision surfaces, but the same idea can be applied to other subdivision schemes as well. The capability of the new approach is demonstrated with test examples shown in the paper.

The remaining part of the paper is arranged as follows. A brief review of previous techniques on noise removal, multiresolution analysis and subdivision surfaces is given in Section 2. The mesh and surface decomposition approach is presented in Section 3. The fast iterative noise removal technique based on the mesh decomposition scheme is discussed in Section 4. Iteration convergence and exponential convergence rate are proved in Section 5. In Section 6, some test cases are shown and discussed. Concluding remarks and future work are presented in the Section 7.

## 2 Previous Work

### 2.1 Related Work on Smoothing/Denoising

With the recent advances in scanning and acquisition technologies, fairing, smoothing and denoising of noisy meshes have become more and more important. A wide variety of mesh smoothing algorithms have been proposed in recent years [9, 10, 11, 12, 14, 18]. In general, denoising algorithms can be roughly classified into two categories: linear and nonlinear smoothing. Linear smoothing treats features (large variations) and noises (small variations) indiscriminately, so it is not feature preserving and achieves smoothness at the expense of sharp features being blurred. Nonlinear smoothing updates each mesh point through local weighted averaging of its neighbors. A large weight usually need to be assigned to a point that involves low discontinuities, and vice versa. Because nonlinear smoothing treats features and noises differently, it can remove noises and meantime preserve features of a given model.

Taubin pioneered a signal processing approach [17] to mesh smoothing based on a basic and uniform approximation of the Laplacian. For meshes with irregular connectivity, Taubin's method leads to a variety of artifacts such as geometric distortion during smoothing and numerical instability. Desbrun et al. extended this approach to irregular meshes [10] using a geometric flow analogy, and introduced the use of a conjugate gradient solver that safely removes the stability condition, allowing for significant smoothing in reasonable time even on large meshes. These algorithms are isotropic in nature, i.e., they treat noise and salient features such as edges and corners identically. To address this shortcoming, anisotropic diffusion schemes have been recently proposed [11, 18]. The idea behind these approaches is to modify the diffusion equation to make it non-linear or anisotropic. The curvature tensor determines the local diffusion, thus preserving sharp features.

Multiresolution analyses [16] based on wavelet theory have started a new research direction on hierarchical methods for computer graphics. Lounsbery et al [15] made the connection between wavelets and subdivision surfaces to define different levels of resolution. Multiresolution analysis has recently proved its efficiency on dealing with irregularly sampled meshes by decomposing a mesh into coarser meshes which represent the initial model at different scales [15]. A set of detail coefficients are computed for each scale, defining the frequency content of the model, which can be used to build a denoising filter suitable for irregular meshes. Zorin et al

proposed a combination of subdivision and smoothing algorithms to construct a set of algorithms for interactive multiresolution editing of complex meshes with arbitrary topology [13].

## 2.2 Subdivision Surfaces

Given a control mesh, a subdivision surface is generated by iteratively refining (subdividing) the control mesh to form new and finer control meshes. The refined control meshes converge to a limit surface called a *subdivision surface*. So a subdivision surface is determined by the given control mesh and the mesh refining (subdivision) process. The control mesh of a subdivision surface can contain vertices whose *valences* (numbers of adjacent edges) are different from four. Those vertices are called *extra-ordinary vertices*. Popular subdivision surfaces include Catmull-Clark subdivision surfaces (CCSSs) [1], Doo-Sabin subdivision surfaces [2] and Loop subdivision surfaces [3].

Subdivision surfaces can model/represent complex shape of arbitrary topology because there is no limit on the shape and topology of the control mesh of a subdivision surface. Subdivision surfaces are intrinsically discrete. Recently it was proved that subdivision surfaces can also be parametrized [7]. Therefore, subdivision surfaces cover both *parametric forms* and *discrete forms*. Parametric forms are good for design and representation, discrete forms are good for machining and tessellation (including FE mesh generation). Hence, we have a representation scheme that is good for all graphics and CAD/CAM applications. Subdivision surfaces by far are the most general surface representation scheme. They include non-uniform B-spline and NURBS surfaces as special cases [6]. In this paper we only consider objects represented by CCSSs. But our approach works for other subdivision schemes as well.

## 3 Mesh & Surface Decomposition

Given a mesh  $M$  with noises corrupted at its vertices, our task is to find another mesh  $T$  which eliminates most of the noises in  $M$ . This can be achieved by decomposing the given mesh into an infinite series of meshes of the same topology but smaller magnitude so that each mesh represents part of the information of the given model, with some meshes containing more information on overall shape while others containing more on subtle details.

Let  $A$  be the matrix that calculates all the limit points of  $M$  when viewed as the control mesh of a subdivision surface (we assume the subdivision scheme considered here is the Catmull-Clark scheme. However, the techniques presented here work for other subdivision schemes as well), then  $A * M$  is a new mesh that has the same topology as  $M$  and whose vertices are the corresponding limit points of  $M$ . As we know, subdivision process satisfies the convex hull property and is a smoothing process. Hence  $A * M$  is a smoother version of  $M$ . Consequently  $A * M$  can be regarded as a model with some noises removed from  $M$ . If we repeat the process by applying  $A$  to the new meshes we get a sequence of meshes:

$$\{A^i * M, \quad 0 \leq i < \infty, \} \quad (1)$$

with each of them smoother than the ones ahead of it. However, this sequence cannot be used directly for noise removal because subdivision process is a shrinking process, i.e., it makes the resulting model thinner and smaller than the original mesh (this is the so called *volume lose* or *feature blurring* of the denoising process). As a result, at the end, the sequence would lose all the shape features of the given mesh and converges to a single point. Therefore, in addition to smoothing, feature preserving is also an important requirement of mesh denoising. There is a way, fortunately, for us to compensate the volume lose in the above smoothing and denoising process.

Note that the matrix  $A$  is invertible (see proof in the next section), Hence we have

$$A^{-1} = \sum_{i=0}^{\infty} (E - A)^i,$$

where matrix  $E$  is the identity matrix. With this expansion, a given mesh  $M$  can now be represented as

$$M = \sum_{i=0}^{\infty} (E - A)^i * A * M. \quad (2)$$

The above formula provides a way to expand a given mesh into a series of meshes. Just like Fourier series and multi-resolution representation [13, 15], any mesh (or surface) now can be decomposed into a series of meshes (or subdivision surfaces) of the same topology but smaller magnitude. For example, if  $\mathbf{S}(M)$  refers to the limit surface of mesh  $M$ , then subdivision surface  $\mathbf{S}(M)$  can be represented as an infinite series of subdivision surfaces as follows.

$$\mathbf{S}(M) = \sum_{i=0}^{\infty} \mathbf{S}((\mathbf{E} - \mathbf{A})^i * \mathbf{A} * M)$$

In the infinite series of eq. (2), each term  $(\mathbf{E} - \mathbf{A})^i \mathbf{A} M$  contains part of the information on  $M$ . Terms with smaller indices contain more information on overall shape of  $M$  while terms with bigger indices contain more subtle details on the shape of  $M$ . They can be regarded as low and high frequency information on mesh  $M$ . Hence the above two equations transform a space domain model into a frequency-like domain representation. Because each term itself is a mesh (or surface), it can be further decomposed using the above two equations to obtain more subtle details of the original model. Like Fourier series, this representation can be used for applications in other areas of graphics and modeling, such as fairing, smoothing, sharpening, low pass or high pass filtering etc. For example, for any  $n < \infty$ ,

$$M_n = \sum_{i=0}^n (\mathbf{E} - \mathbf{A})^i \mathbf{A} M \quad (3)$$

gives us a model smoother than  $M$ , and the smaller of  $n$ , the smoother of the resulting model. However, the smoothest model that can be achieved by Eq. (3) is when  $n$  is equal to 0, which is  $\mathbf{A} M$ . This is far from good enough. In the next section we show how to modify the above formula so that it can be used for model smoothing and denoising.

Note that  $M_n$  in the above formula should not be calculated directly because it requires costly matrix multiplications. Actually matrix  $\mathbf{A}$  is not needed at all in the process of computing  $M_n$ . Note that

$$\begin{aligned} M_{n+1} &= \sum_{i=0}^{n+1} (\mathbf{E} - \mathbf{A})^i \mathbf{A} M \\ &= M_n + (\mathbf{E} - \mathbf{A})^{n+1} \mathbf{A} M \\ &= M_n + (\mathbf{E} - \mathbf{A}) \sum_{i=0}^n (\mathbf{E} - \mathbf{A})^i \mathbf{A} M \\ &= M_n + \mathbf{A} M - \mathbf{A} M_n. \end{aligned} \quad (4)$$

Hence  $M_n$  can be calculated iteratively. The geometric meaning of the above formula is: in the iterative process, the new location of each mesh vertex can be obtained by moving that vertex by an offset, and the offset is the difference between the limit location of that vertex in the original mesh  $M$  and the limit point of that vertex in the current mesh  $M_n$ . Note that in eq. (4),  $\mathbf{A} * M$  and  $\mathbf{A} * M_n$  represent limit points of vertices in  $M$  and  $M_n$ , respectively. As we know, all limit points can be directly calculated according to Catmull-Clark subdivision rules [5], without the involvement of the matrix  $\mathbf{A}$  at all. Therefore the computation from  $M_n$  to  $M_{n+1}$  in eq. (4) does not require any costly matrix computation, but only a linear combination of vertices locally. Hence this is a linear local method which is very easy to implement and can deal with meshes of large number of vertices effectively.

## 4 Fast Iterative Noise Removal

As mentioned above, each term of the series in Eq. (2) contains some low or high frequency-like information of mesh  $M$ . Similar to a Fourier transform, Eq. (2) can be used for mesh fairing, smoothing, sharpening, low pass or high pass filtering etc. However, to obtain good denoising results, Eq. (2) has to be modified into an iterative form so that noises at all frequency levels can be removed and meanwhile, the overall shape of the given model still can be maintained.

Consider a finite portion of the series defined in eq. (2) and define a new series as follows.

$$\begin{cases} T_0 &= M \\ T_{k+1} &= \sum_{i=0}^m (\mathbf{E} - \mathbf{A})^i * \mathbf{A} * T_k, \quad k \geq 0 \end{cases} \quad (5)$$

where  $m$  is an integer and  $0 \leq m \leq \infty$ . When  $m = \infty$ ,  $T_{k+1}$  equals to  $T_k$ . But when  $m < \infty$ ,  $T_{k+1}$  is a smoother version of  $T_k$  because some high frequency information from  $T_k$  is not included in  $T_{k+1}$ . Through

simple computation, we have

$$T_{k+1} = (\mathbf{E} - (\mathbf{E} - \mathbf{A})^{m+1}) * T_k \quad (6)$$

Furthermore, if  $m$  is set to be the same for every  $k$ , we have

$$T_k = (\mathbf{E} - (\mathbf{E} - \mathbf{A})^{m+1})^k * T_0, \quad (7)$$

When  $m = 0$  in eq. (5),  $T_k = \mathbf{A}^k * T_0$ , which is the same as the formula in eq. (1). Because all the eigenvalues of  $\mathbf{A}$  are between 0 and 1 (see proof in the next section),  $\mathbf{A}^k * M$  converges to a point when  $k$  tends to infinity. Hence using matrix  $\mathbf{A}^k$  for smoothing would make the mesh shrink. Mesh shrinking leads to model details losing, hence  $\mathbf{A}^k$  should not be used for de-noising directly. However, by compensating  $\mathbf{A}^k$  with some details in each step, it would shrink much slower, consequently keep more details, and meanwhile smooth out undesired noises. For example, when  $m = 1$  we have,

$$T_k = (2\mathbf{E} - \mathbf{A})^k * \mathbf{A}^k * T_0$$

which compensates  $\mathbf{A}^k$  by a small scale factor of details  $(2\mathbf{E} - \mathbf{A})^k$  everywhere in the current mesh.

For any  $m$ , the biggest eigenvalue of matrix  $\mathbf{D} = (\mathbf{E} - (\mathbf{E} - \mathbf{A})^{m+1})$  is always 1 with corresponding eigenvector  $[1, 1, 1, \dots, 1]^T$  and all other eigenvalues are between 0 and 1 (see proof in the next section). Hence  $\mathbf{D}^k$  is convergent. Note that even though eventually  $\mathbf{D}^k * M$  shrinks to a single point when  $k$  approaches infinity, it shrinks at a speed much slower than  $\mathbf{A}^k * M$ . This is because their shrink speed is determined by the second biggest eigenvalue. If the second biggest eigenvalue of  $\mathbf{A}$  is  $\lambda$ , then the second biggest eigenvalue of  $\mathbf{D}$  is  $1 - (1 - \lambda)^{m+1}$ , which is much bigger than  $\lambda$  and becomes even bigger when  $m$  gets bigger. Because  $T_k$  shrinks slower, it maintains more details in the de-noising process while smoothing out undesired noises.

The number  $m$  in eq. (6) can be used as a parameter to control how much details should be kept in the de-noising process. The smaller of  $m$ , the smoother the resulting model and the less details are kept. When  $m$  gets bigger, more details, and possibly more noises are kept. In our test cases, setting  $m$  to 2 or 3 is good enough for fast and good de-noising (smoothing) while maintaining enough details of the input model.

Eq. (6) and eq. (7) should not be used to remove noises or smooth surfaces directly, because they require costly matrix multiplications. A iterative formula for eq. (6) and eq. (7) can be derived similarly to the one given in eq. (4) for each iterative step. And again, all vertex points can be directly calculated without using the matrix  $\mathbf{A}$  at all. Therefore one can use eq. (6) and eq. (7) to smooth a surface or mesh without using any costly matrix computation, but only using linear combination of vertices locally. Consequently, the new method is very easy to implement and can deal with meshes of large number of vertices effectively.

As is proven in the next section, eq. (6) and eq. (7) converges at an exponential rate. Hence the new method is very fast and good results of noise removal can be obtained in just a few iterations. Nevertheless, model differences can be explicitly calculated as  $\|T_{k+1} - \mathbf{D} * T_k\|$  and the iteration stop criterion can be determined based on some given tolerance or human interaction.

## 5 Proof of Convergence

### 5.1 Construction of Matrix $\mathbf{A}$

The matrix  $\mathbf{A}$  is not needed in the implementation. We show it here only for proof purpose. The matrix shown here is for the generalized Catmull-Clark subdivision scheme. The matrix for other schemes can be constructed similarly.

For generalized Catmull-Clark subdivision scheme, new face points and edge points are calculated the same way as the standard Catmull-Clark subdivision scheme, but the new vertex points are calculated differently, using the following formula

$$\mathbf{V}' = \frac{n-2}{n}\mathbf{V} + \frac{1}{n^2} \sum_{j=1}^n (\alpha\mathbf{V} + (1-\alpha)\mathbf{E}_j) + \frac{1}{n^2} \sum_{j=1}^n \mathbf{f}_j$$

where  $0 \leq \alpha \leq 1$  and  $\mathbf{f}_j$  are new face points after one subdivision [1]. When  $\alpha = 0$ , we get the standard Catmull-Clark subdivision scheme. The limit point [5] of a vertex  $\mathbf{V}_i$  of degree  $n_i$  can be calculated as:

$$\mathbf{V}_i^\infty = \frac{1}{n_i(n_i+5)} (b_{ii}\mathbf{V}_i + \sum_j b_{ij}\mathbf{E}_j + \sum_j b_{ij}\mathbf{F}_j)$$



contradicts the assumption  $X \neq 0$ . Hence if there exists at least one vertex whose valance is bigger than 3, then B is positive definite as well.

When all  $n_i$  are 3 in a mesh and  $\alpha = 0$ , B might not be positive definite. However, we can change the value of  $\alpha$  to convert the standard Catmull-Clark subdivision scheme into a general Catmull-Clark subdivision scheme, such that B is positive definite. It is easy to verify that when  $\alpha > 0$  and  $n_i \geq 3$ , we have  $\sigma_i > 0$ . Therefore let  $\alpha \in (0, 1]$ , say  $\alpha = 0.5$ , then B becomes positive definite.

Because B is positive definite, all the eigenvalues of A are positive. Therefore A is invertible and all its eigenvalues are in  $(0, 1]$ .

### 5.3 Rate of Convergence

Because all eigenvalues of A are  $\in (0, 1]$ , all the eigenvalues of  $(E - A)$  are  $\in [0, 1)$  and all the eigenvalues of  $D = E - (E - A)^{m+1}$  are  $\in (0, 1]$ . In addition, because the sum of each row in A is 1, the eigenvector of A corresponding to the eigenvalue 1 is  $[1, 1, \dots, 1]^T$ . It is also easy to verify that  $[1, 1, \dots, 1]^T$  is the eigenvector of matrix D corresponding to the eigenvalue 1. Hence  $D^k M$  converges to a single point when  $k$  tends to infinity. As a result,  $T_{k+1}$  is a smoother version of  $T_k$ .

It can also be proven that eq. (7) converges at an exponential rate. To prove this it is sufficient to show that  $\|T_{k+1} - T_k\|$  converges to 0 at an exponential rate.

$$\begin{aligned}
& \|T_{k+1} - T_k\| \\
&= \|D^{k+1}T_0 - D^kT_0\| \\
&= \|\mathbf{X}\Lambda^{k+1}\mathbf{X}^{-1}T_0 - \mathbf{X}\Lambda^k\mathbf{X}^{-1}T_0\| \\
&= \|\mathbf{X}(\Lambda^k(\Lambda - E))\mathbf{X}^{-1}T_0\| \\
&\leq \|\Lambda^k\| * \|\mathbf{X}\| * \|\Lambda - E\| * \|\mathbf{X}^{-1}\| * \|T_0\| \\
&= \|\Lambda^k\| * c
\end{aligned}$$

where  $c = \|\mathbf{X}\| * \|\Lambda - E\| * \|\mathbf{X}^{-1}\| * \|T_0\|$ , which is a constant and  $\Lambda$  is the diagonalized matrix of D. Suppose the biggest eigenvalue of matrix D is  $\lambda$ , then  $\|\Lambda^{n+1}\| \leq \lambda^{n+1}$ . As we know,  $0 \leq \lambda < 1$ . Hence we have

$$\|T_{k+1} - T_k\| \leq c * \lambda^{n+1},$$

which means  $(T_{k+1} - T_k)$  converges to 0 at an exponential rate.

## 6 Test Results

The proposed approach has been implemented in C++ using *OpenGL* as the supporting graphics system on the Windows platform. Quite a few examples have been tested with the method described here. All the examples have extra-ordinary vertices. Some of the tested results are shown in Figures 1 and 3. For all the test cases shown in this paper, the original model, the corrupted model and resulting models by several iterations of the new noise removal method are displayed. The number of iterations is also listed for each denoised model. The stop criterion of iteration used in our implementation is simply done by user interaction. Figures 1 and 3 demonstrate the capability of the new iterative method in mesh smoothing. From Figure 1 and 3 we can see that with very limited number of iterations, smooth and visually pleasant denoising results can be obtained, even for complicated models. Comparing the original model shown in Figure 1(e) with the figure shown in Figure 1(a), we can see that many subtle details are kept meanwhile noises are smoothed out.

The new denoising method can handle meshes with large number of vertices in a matter of seconds on an ordinary PC (3.2GHz CPU, 512MB of RAM). For example, the model shown in figures 3(a-e), has 23202 vertices and 46400 faces. It takes 6 seconds to obtain the model shown in figure 3(e). For simpler meshes, like models shown in figures 1(e), 3(j) and 3(o), the denoising process is done in about one second. Hence the new denoising method is suitable for interactive shape design, where simple shapes with small or medium-sized control vertex sets are used.

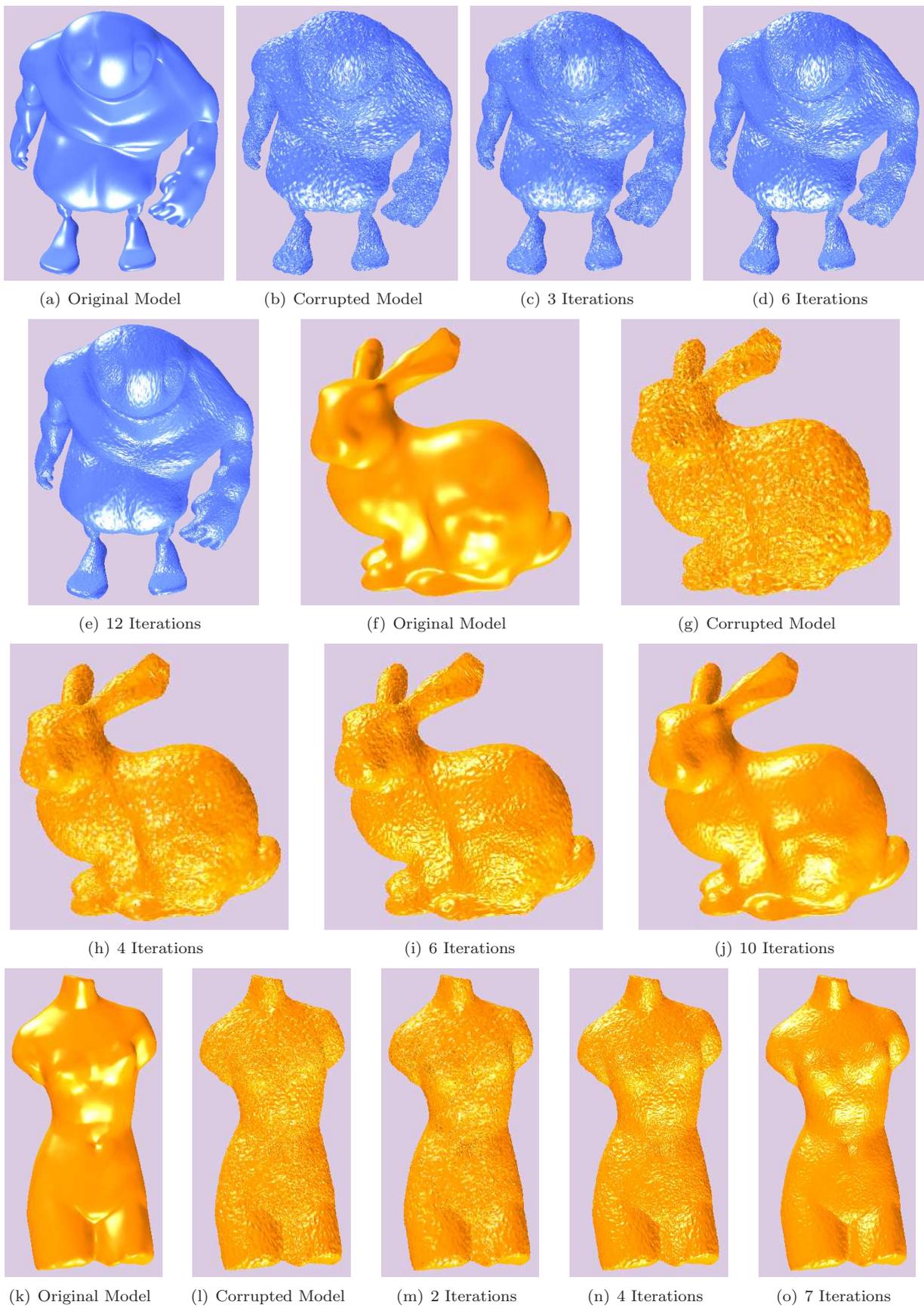


Figure 3: Some test example obtained using the iterative method.

## 7 Summary and Future Work

A fast iterative mesh denoising method using mesh decomposition technique is presented. The mesh decomposition technique decomposes a given mesh (surface) into an infinite series of meshes (surfaces) of the same topology but different magnitudes such that each mesh (surface) represents high or low frequency information of the given mesh (surface). Hence we can use decomposed items to control high-frequency and low-frequency information of the mesh (surface) and, consequently, overall shape or local details of the mesh (surface). Manipulating or balancing such pieces of information can basically achieve the same effects of low and high pass filtering. The new method is fast and does not require any matrix computation, or linear system solving in the process of denoising, hence it is very easy to implement.

One of our future research objectives is to investigate other possible applications of the mesh decomposition technique. Areas that will be considered include mesh compression, feature identification and mesh simplification. Another subject of our future research is to compare the performance of the new approach with other noise removal methods in the literature to study its effectiveness and possible improvements. In addition, the matrix  $A$  in eq. (2), instead of being a subdivision matrix, could be set to other matrices, as long as their eigen values are in  $(0, 1]$ . Hence it is possible to design an  $A$  and use eq. (2) to solve problems in computer graphics with specified requirements.

## References

- [1] Catmull E, Clark J, Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer-Aided Design*, 1978, 10(6):350-355.
- [2] Doo D, Sabin M, Behavior of recursive division surfaces near extraordinary points, *Computer-Aided Design*, 1978, 10(6):356-360.
- [3] Loop CT, Smooth Subdivision Surfaces Based on Triangles, MS thesis, Department of Mathematics, University of Utah, August, 1987.
- [4] DeRose T, Kass M, Truong T, Subdivision Surfaces in Character Animation, *Proceedings of SIGGRAPH*, 1998: 85-94.
- [5] Halstead M, Kass M, DeRose T, Efficient, fair interpolation using Catmull-Clark surfaces, *ACM SIGGRAPH*, 1993:35-44.
- [6] Sederberg T W, Zheng J, Sewell D, Sabin M, Non-uniform recursive subdivision surfaces, *Proceedings of SIGGRAPH*, 1998:19-24.
- [7] Stam J, Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values, *Proceedings of SIGGRAPH* 1998:395-404.
- [8] Adi Levin, Modified Subdivision Surfaces with Continuous Curvature, *Proceedings of SIGGRAPH*, 1035-1040, 2006.
- [9] S. Fleishman, I. D. Tel and D. Cohen-Or, Bilateral mesh denoising, *SIGGRAPH 2003*.
- [10] Mathieu Desbrun, Mark Meyer, Peter Schröder, Alan Barr, Implicit fairing of irregular meshes using diffusion and curvature flow, *SIGGRAPH 1999*, p.317-324.
- [11] T. Jones , F. Durand , M. Desbrun, Non-iterative, feature-preserving mesh smoothing, *ACM Transactions on Graphics*, v.22 n.3, 2003.
- [12] Leif Kobbelt, Discrete Fairing, In *Proc. of the 7th IMA Conf. on the Mathematics of Surfaces*, 1997.
- [13] Denis Zorin, Modeling with multiresolution subdivision surfaces, *SIGGRAPH* 2006, P.30-50.
- [14] Michael Roy; Sebti Fofou; Frederic Truchetet, Denoising 3D models with attributes using soft thresholding, *Wavelet Applications in Industrial Processing II*, Volume 5607, pp. 139-147 (2004).
- [15] M. Lounsbery, T. DeRose, and J. Warren, Multiresolution analysis for surfaces of arbitrary topological type, *ACM Transactions on Graphics* 16(1), pp. 34.73, 1997.

- [16] I. Guskov, W. Sweldens, and P. Schröder, Multiresolution signal processing for meshes, *Proceedings of ACM SIGGRAPH*, pp. 325-334, 1999.
- [17] G. Taubin, A Signal Processing Approach for Fair Surface Design, *Proceedings of ACM SIGGRAPH*, pp. 351-358, Aug. 1995.
- [18] U. Clarenz, U. Diewald, and M. Rumpf, Anisotropic geometric diffusion in surface processing, *Proceedings of IEEE Visualization*, pp. 397-405, 2000.