Bilateral Mesh Denoising

Shachar Fleishman Iddo Drori Daniel Cohen-Or
School of Computer Science, Tel Aviv University*

Abstract

We present an anisotropic mesh denoising algorithm that is effective, simple and fast. This is accomplished by filtering vertices of the mesh in the normal direction using local neighborhoods. Motivated by the impressive results of bilateral filtering for image denoising, we adopt it to denoise 3D meshes; addressing the specific issues required in the transition from two-dimensions to manifolds in three dimensions. We show that the proposed method successfully removes noise from meshes while preserving features. Furthermore, the presented algorithm excels in its simplicity both in concept and implementation.

CR Categories: I.3.0 [COMPUTER GRAPHICS]: General
Keywords: Mesh Denoising, Bilateral Filtering

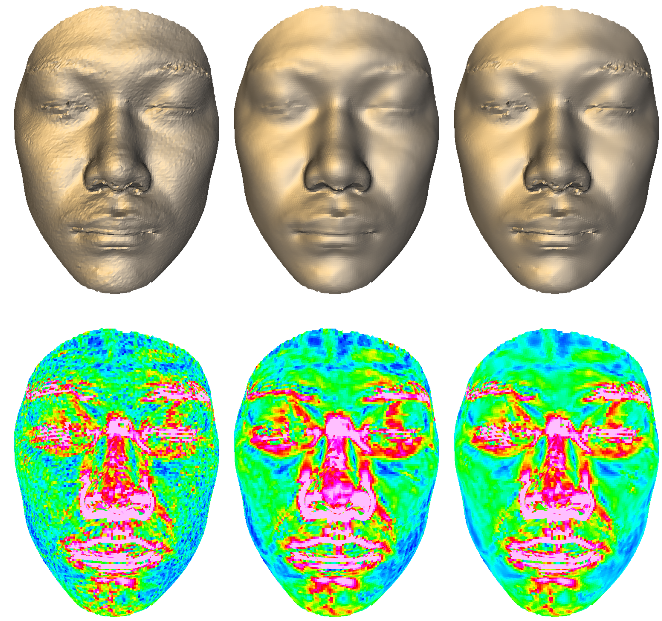

Figure 1: Denoising a scanned model: On the left is the input model, in the middle is the result of implicit fairing [Desbrun et al. 1999], and on the right is the result of our algorithm. The top row visualizes the details of the models, and on the bottom row is a mean curvature visualization. Data courtesy of Alexander Belyaev.

1 Introduction

With the proliferation of 3D scanning tools, interest in removing noise from meshes has increased. The acquired raw data of the sampled model contains additive noise from various sources. It remains a challenge to remove the noise while preserving the underlying sampled surface, in particular its fine features. Related techniques like mesh smoothing or mesh fairing alter the given surface to increase its degree of smoothness. The goal of these techniques is to create semi-uniform triangulation, often with subdivision connectivity. This paper focuses on mesh denoising, which is an important preprocess for many digital geometry applications that rely on local differential properties of the surface.

Denoising the sampled data can be applied either before or after generating the mesh. The advantage of denoising a mesh rather than a point-cloud, is that the connectivity information implicitly defines the surface topology and serves as a means for fast access to neighboring samples. The information in a mesh can be separated into two orthogonal components: a *tangential* and a *normal* component. The normal component encodes the geometric information of the shape, and the tangential component holds parametric information [Guskov et al. 1999]. In this formulation, moving vertices along their normal directions, modifies only the geometry of the mesh. Related to this notion are evolution curves [Osher and Sethian 1988], where points are shifted in the direction of the normal at a distance that is proportional to their curvature, to get smoother curves over time. Our denoising method is based on this idea, shifting mesh vertices along their normal direction.

The extensive research on image denoising serves as a foundation for surface denoising and smoothing algorithms. However, adapting these algorithms from the two dimensional plane to a surface in three dimensions is not straightforward for three main reasons: (i) **Irregularity**; unlike images, meshes are irregular both in connectivity and sampling, (ii) **Shrinkage**; image denoising algorithms are typically not energy preserving. While this is less noticeable in images, in meshes, this is manifested as shrinkage of the mesh, (iii) **Drifting**; naive adaptation of an image denoising technique may cause artifacts known as *vertex drifts*, in which the regularity of the mesh decreases.

The bilateral filter, introduced by Tomasi and Manduchi [1998], is a nonlinear filter derived from Gaussian blur, with a feature preservation term that decreases the weight of pixels as a function of intensity difference. It was shown that bilateral filtering is linked to anisotropic diffusion [Barash 2002], robust statistics [Durand and Dorsey 2002], and Bayesian approaches [Elad 2001]. Despite its simplicity, it successfully competes with image denoising algorithms in the above categories. The bilateral filtering of images and its adaptation to meshes has an intuitive formulation, which leads to a simple method for selecting the parameters of the algorithm.

The contribution of this paper is a mesh denoising algorithm that operates on the geometric component of the mesh. The origin of the denoising algorithm is the bilateral filter that has a simple and intuitive formulation, is fast and easy to implement, and adapting it to meshes, yields results that are as successful as its predecessor.

1.1 Previous work

Image denoising is part of on-going research in image processing and computer vision. The state-of-the-art approaches to image de-

*e-mail: {shacharf | idrori | dcor}@tau.ac.il

noising include: wavelet denoising [Donoho 1995], nonlinear PDE based methods including total-variation [Rudin et al. 1992], and bilateral filtering [Tomasi and Manduchi 1998]. These approaches can be viewed in the framework of basis pursuit [Chen et al. 1999].

Typically, mesh denoising methods are based on image denoising approaches. Taubin [1995] introduced signal processing on surfaces that is based on the definition of the Laplacian operator on meshes. Peng et al. [2001] apply locally adaptive Wiener filtering to meshes. Geometric diffusion algorithms for meshes was introduced by Desbrun et al. [1999], they observed that fairing surfaces can be performed in the normal direction. Anisotropic diffusion for height fields was introduced by Desbrun et al. [2000], and Clarenz et al. [2000] formulated and discretized anisotropic diffusion for meshes. Recently, Bajaj and Xu [2003] achieved impressive results by combining the limit function of Loop subdivision scheme with anisotropic diffusion. Tasdizen et al. [2002] apply anisotropic diffusion to normals of the level-set representation of the surface, and in the final step, the level-set is converted to a mesh representation. Guskov et al. [1999] introduced a general signal processing framework that is based on subdivision, for which denoising is one application.

2 Bilateral mesh denoising

We open with a description of our method for filtering a mesh using local neighborhoods. The main idea is to define a local parameter space for every vertex using the tangent plane to the mesh at a vertex. The heights of vertices over the tangent plane are synonymous with the gray-level values of an image, and the closeness components of the filter are the tangential components. The term *offset* is used for the heights. Let S denote the noise-free surface, and let M be the input mesh with vertices that sample S with some additive noise. Let $\mathbf{v} \in M$ be a vertex of the input mesh, d_0 its signed-distance to S, and $\mathbf{n_0}$ the normal to S at the closest point to $\mathbf{v}$. The noise-free surface S is unknown and so is d_0, therefore we estimate the normal to the surface as the normal $\mathbf{n}$ to the mesh, and d estimates d_0 as the application of the filter, updating $\mathbf{v}$ as follows:

$$\hat{\mathbf{v}} = \mathbf{v} + d \cdot \mathbf{n}. \tag{1}$$

Note that we do not have to define a coordinate system for the tangential component; as explained below, we apply a one-dimensional filter with a spatial distance as a parameter. The filter is applied to one vertex at a time, computing a displacement for the vertex and updating its position.

Bilateral filtering of images. Following the formulation of Tomasi and Manduchi [1998], the bilateral filtering for image $I(\mathbf{u})$, at coordinate $\mathbf{u} = (x, y)$, is defined as:

$$\hat{I}(\mathbf{u}) = \frac{\sum_{\mathbf{p} \in N(\mathbf{u})} W_c(\|\mathbf{p}-\mathbf{u}\|) W_s(|I(\mathbf{u}) - I(\mathbf{p})|) I(\mathbf{p})}{\sum_{\mathbf{p} \in N(\mathbf{u})} W_c(\|\mathbf{p}-\mathbf{u}\|) W_s(|I(\mathbf{u}) - I(\mathbf{p})|)}, \tag{2}$$

where $N(\mathbf{u})$ is the neighborhood of $\mathbf{u}$. The closeness smoothing filter is the standard Gaussian filter with parameter σ_c: $W_c(x) = e^{-x^2/2\sigma_c^2}$, and a feature-preserving weight function, which we refer to as a *similarity weight function*, with parameter σ_s that penalizes large variation in intensity, is: $W_s(x) = e^{-x^2/2\sigma_s^2}$. In practice, $N(\mathbf{u})$ is defined by the set of points $\{\mathbf{q_i}\}$, where $\|\mathbf{u} - \mathbf{q_i}\| < \rho = \lceil 2\sigma_c \rceil$.

Algorithm. We begin introducing the algorithm by describing how to compute the normal and tangential components that are assigned to Eq. 2, yielding a new offset d. Since S is unknown, and we wish to use the edge preservation property of the bilateral filter, we define $S_v \subset S$ as the smooth connected component of S that is closest to $\mathbf{v}$. For the normal component, we would like to compute the offsets of the vertices in the neighborhood of $\mathbf{v}$, denoted by $\{\mathbf{q_i}\}$, over the noise-free smooth component S_v. We use the tangent plane to $\mathbf{v}$ defined by the pair $(\mathbf{v}, \mathbf{n})$ as a first-order approximation to S_v. The offset of a neighbor q_i is the distance between q_i and the plane. The following is the pseudo-code for applying a bilateral filter to a single vertex:

```
DenoisePoint( Vertex v, Normal n)
  {q_i} = neighborhood(v)
  K = |{q_i}|
  sum = 0
  normalizer = 0
  for i := 1 to K
    t = ||v - q_i||
    h = <n, v - q_i>
    w_c = exp(-t^2/(2σ_c^2))
    w_s = exp(-h^2/(2σ_s^2))
    sum += (w_c · w_s) · h
    normalizer += w_c · w_s
  end
  return Vertex v̂ = v + n · (sum/normalizer)
```

The plane that approximates the noise-free surface should on one hand, be a good approximation of the local surface, and on the other hand, preserve sharp features. The first requirement leads to smoothing the surface, while the latter maintains the noisy surface. Therefore, we compute the normal at a vertex as the weighted average (by the area of the triangles) of the normals to the triangles in the 1-ring neighborhood of the vertex. The limited neighborhood average smoothes the local surface without over-smoothing. In some cases, for example, of a synthetic surface, the normal of an edge vertex will erroneously point to the average direction and lead to a rounded edge.

For the tangential component, the correct measure of distance between vertices on the smooth surface is the geodesic distance between points. Since we use local neighborhoods, we approximate the geodesic distance using the Euclidean distance. This approximation seems to introduce artifacts in the neighborhood of sharp features, since vertices that happen to be geodesically far from the smoothed vertex may be geometrically close. Furthermore, the assumption from differential geometry that a neighborhood of a point on a surface can be evaluated by a function over the tangent plane to that point may not be satisfied. Both apparent problems do not hinder our algorithm because any of the above offending vertices is penalized by the similarity weight function.

Mesh shrinkage and vertex-drift. Image denoising and smoothing algorithms that are based on (possibly weighted) averaging of neighborhood, result is shrinkage of the object. Taubin [1995] solves this problem for the Laplacian operator by alternating shrink and expand operations. Another common approach is to preserve the volume of the object as suggested by Desbrun et al. [1999].

Our algorithm, also shrinks the object. This can be observed when smoothing a vertex that is a part of a curved patch; the offset of the vertex approaches the average of the offsets in its neighborhood. Therefore, we follow the volume preservation technique.

Vertex-drift is caused by algorithms that change the position of the vertices along the tangent plane as well as the normal direction. The result is an increase in the irregularity of the mesh. Our algorithm moves vertices along the normal direction, and so, no vertex-drift occurs.

Handling boundaries. Often meshes, in particular scanned data sets, are not closed. There are two aspects to note here: first, the shape of the boundary curve, which is the related problem of "mesh fairing". Second, is that a filter is defined for a neighborhood of a point. However for boundary points, part of the neighborhood is

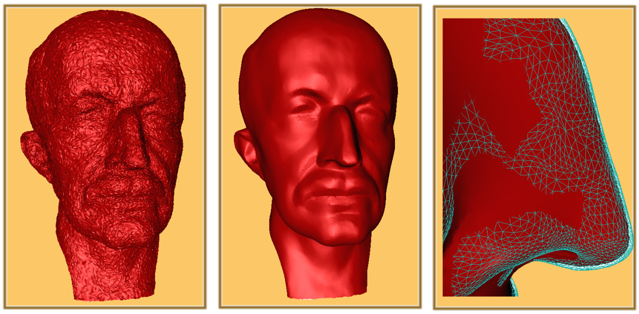

Figure 2: The shrinkage problem. On the left the model of Max Planck with heavily added random noise. In the middle the denoised model after four iterations of our algorithm without volume preservation. The Max-Planck model is courtesy of Christian Rössl from Max Planck Insitut für Informatik.

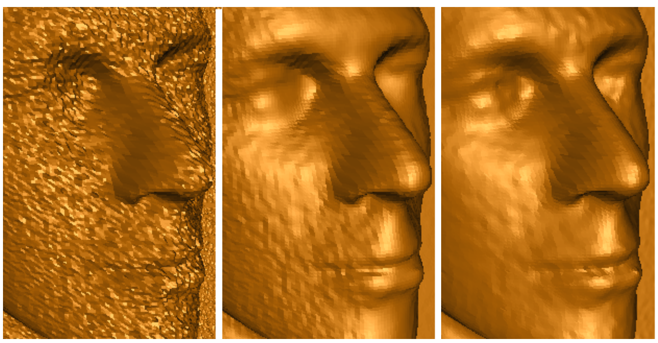

Figure 3: Comparison with AFP. On the left is the input model, in the middle is the result denoised by AFP, and on the right is the result of bilateral mesh denoising. Observe the difference in details in the area of the lips and eyes. The noisy model and the AFP denoised models are courtesy of Mathieu Desbrun.

not defined. One common solution to this problem is to define a virtual neighborhood by reflecting vertices over edges. Our filter inherently handles boundaries by treating them as sharp edges with virtual vertices at infinity. The similarity weight sets the weight of virtual vertices to zero, and thus, the normalization of the entire filter causes boundaries to be handled correctly.

Parameters. The parameters of the algorithm are: σ_c, σ_s, the kernel size ρ, and the number of iterations. We propose an intuitive user-assisted method for setting these parameters. Two parameters, σ_c and σ_s are interactively assigned: the user selects a point of the mesh where the surface is expected to be smooth, and then a radius of the neighborhood of the point is defined. The radius of the neighborhood is assigned to σ_c, and we set $\rho = 2\sigma_c$. Then σ_s is set to the standard deviation of the offsets in the selected neighborhood.

One may choose a large σ_c and perform a few iterations, or choose a narrow filter and increase the number of iterations. Multiple iterations with a narrow filter has the effect of applying a wider filter, and results in efficient computation. Using a small value for σ_c is sensible for two reasons: (i) large values may cross features as shown in, and (ii) smaller values result in a smaller neighborhood which leads to faster computation of every iteration.

In all the results shown in this paper, we used up to five iterations, we found a small number of iterations sufficient for our purposes, and advantageous both to the speed of computation and for the numerical stability of the filter.

Noisy data may lead to unstable computation of the normals if the 1-ring neighborhood of a vertex is used to compute the normals. For extremely noisy data, the normal to a vertex is computed using the k-ring neighborhood of the vertex, where k is defined by the user. For every scanned models that we denoised, the normals were computed using the 1-ring neighborhoods. Note that only for the Max Palanck (Figure 2) model, we were required to use the 2-ring neighborhood to compute normals.

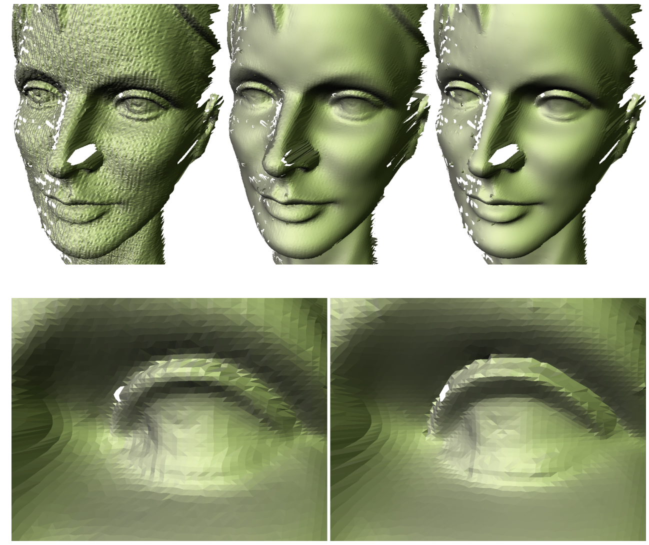

Figure 4: Results of denoising the face model. On the top row from left to right are the input noisy mode, the results of [Jones et al. 2003], and our result. On the bottom we zoom on the right eye of the model, where the bottom left image shows the results of Jones et al. , and on the bottom right is the result of our method. The face model is courtesy of Jean-Yves Bouguet.

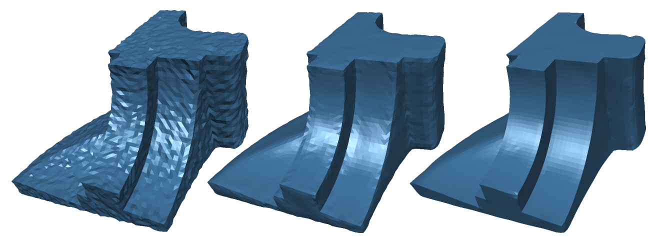

Figure 5: Results of denoising the Fandisk model. On the left is the input noisy model, in the middle is the results of [Jones et al. 2003], and on the right is our result.

3 Results

We have implemented the bilateral mesh denoising algorithm as described in the previous section and compared our results to the results of the anisotropic denoising of height fields algorithm *(AFP)* [Desbrun et al. 2000], Jones et al. [2003], and the *implicit fairing (IF)* algorithm [Desbrun et al. 1999] [1]. The times are reported on a 1GHz Pentium$^{\text{TM}}$ III. In Figure 3, we compare the AFP algorithm with our results. The mesh with 175K vertices is smoothed by three iterations in 10 seconds. Observe the preserved details near the mouth of the model denoised by our algorithm. Figure 1 shows a comparison with implicit fairing. The smoothness of the object can be appreciated from the visualization of the mean curvature in the second row. The model has 17K vertices, and it was denoised in three iterations, taking 1.8 seconds. In Figure 6 we show the denoising of a CAD object. Observe that the sharp features are preserved, but vertices with larger error are treated as outliers and thus are not smoothed out. For the Max Planck model (Figure2) (100k vertices), the timing for a single iteration is 5.75 seconds, and the total number of iterations was four.

[1] Implementation courtesy of Y. Ohtake

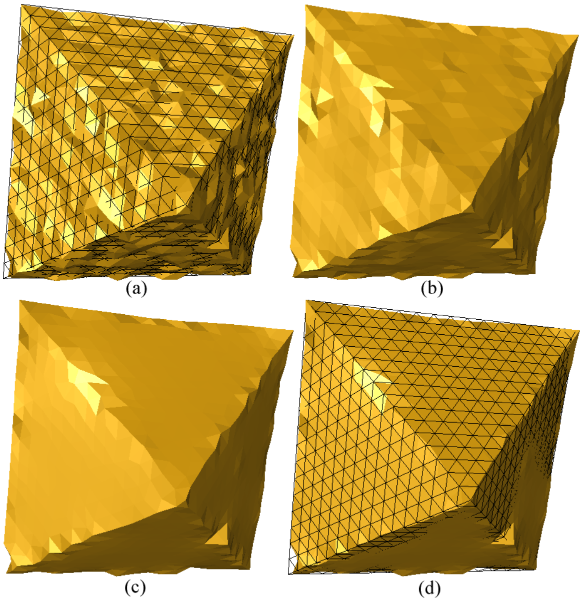

Figure 6: Denoising of CAD-like model. (a) is the input noisy model, (b) is the result of two iterations of our algorithm, (c) and (d) are the result of five iterations of our algorithm, where in (d) the clean model was superimposed on the denoised model.

Independently, Jones et al. [2003] present a similar algorithm that uses bilateral filtering for smoothing surfaces. The main difference between the two methods is in the surface predictor. More specifically, Jones et al. take the distance between the point and its projection onto the plane of a neighboring triangle, whereas our approach takes the distance between a neighboring point and the tangent plane. While we perform several filtering iterations, Jones et al. smooth a surface in a single pass. Figure 4 and 5 compare the results for two different types of models.

Volume preservation is a global operation, whereas denoising is a local operation. In Figure 2, we visualize the local change of volume by adding severe synthetic noise (of zero mean and variance of 0.05% of the diagonal of the bounding box of the model) to the clean model of Max Planck, and then denoised the model. On the right, we zoom on a feature of the model, superimposing the original mesh on top of the smoothed model, showing that the change in shape and volume is minimal.

4 Discussion and future work

Choosing the tangent plane as an approximation of S_v is an important component of our algorithm. Positioning the plane at the average of the offsets of neighboring points would improve our approximation of the smooth surface. However, this would nullify the effect of the similarity term of the bilateral filter. We expect that computing the offsets over a higher order function such as a polynomial of low degree will reduce the shrinkage problem. Finding a high-order, feature-preserving function for the noise-free surface is a difficult problem. In the future, we would like to investigate this possibility, by combining the bilateral filter with a rational function.

The key points of our algorithm are the choice of tangent plane and moving vertices along the normal direction. However this change in vertex position may lead to self-intersection of the denoised mesh. During the application of our algorithm to vertices on two sides of the edge, each vertex moves inwards, for sharp edges this will cause self-intersection after a number of iterations that is proportional to the angle of the edge.

The algorithm that we present assumes that the mesh is sampled regularly. This assumption is made when we fix the values of σ_c. Highly irregular meshes are uncommon in scanned data-sets. To handle irregular data-sets, the parameters must be adjusted locally.

We presented a mesh-denoising algorithm that modifies vertices in the normal direction. The bilateral filtering algorithm that we use is practical, clear and simple. The proposed method deals with irregular meshes and does not perform any reparameterization. In addition, the only property of the mesh that we use is the topological information, and therefore, the algorithm can be adapted to point-based representations.

Acknowledgements

We wish to thank Alexande Belyaev for his numerous remarks and suggestions. This work was supported in part by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities, and by the Israeli Ministry of Science, and by a grant from the German Israel Foundation (GIF).

References

BAJAJ, C. L., AND XU, G. 2003. Anisotropic diffusion of subdivision surfaces and functions on surfaces. *ACM Transactions on Graphics (TOG) 22*, 1, 4–32.

BARASH, D. 2002. A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 6.

CHEN, S. S., DONOHO, D. L., AND SAUNDERS, M. A. 1999. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing 20*, 1, 33–61.

CLARENZ, U., DIEWALD, U., AND RUMPF, M. 2000. Anisotropic geometric diffusion in surface processing. In *IEEE Visualization 2000*.

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of SIGGRAPH 99*, 317–324.

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 2000. Anisotropic feature-preserving denoising of height fields and bivariate data. In *Graphics Interface*, 145–152.

DONOHO, D. L. 1995. De-noising by soft-thresholding. *IEEE Transactions on Information Theory 41*, 3, 613–627.

DURAND, F., AND DORSEY, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics (TOG) 21*, 3, 257–266.

ELAD, M. 2001. On the bilateral filter and ways to improve it. *IEEE Transactions On Image Processing*.

GUSKOV, I., SWELDENS, W., AND SCHRÖDER, P. 1999. Multiresolution signal processing for meshes. In *Proceedings of SIGGRAPH 99*, 325–334.

JONES, T., DURAND, F., AND DESBRUN, M. 2003. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics*.

OSHER, S., AND SETHIAN, J. 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics 79*, 12–49.

PENG, J., STRELA, V., AND ZORIN, D. 2001. A simple algorithm for surface denoising. In *IEEE Visualization 2001*, 107–112.

RUDIN, L., OSHER, S., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D 60*, (1–4), 259–268.

TASDIZEN, T., WHITAKER, R., BURCHARD, P., AND OSHER, S. 2002. Anisotropic geometric diffusion in surface processing. In *IEEE Visualization 2002*.

TAUBIN, G. 1995. A signal processing approach to fair surface design. In *Proceedings of SIGGRAPH 95*, 351–358.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *ICCV*, 839–846.