A Simple Algorithm for Surface Denoising

Jianbo Peng, Vasily Strela, Denis Zorin

New York University

Abstract

We present a simple denoising technique for geometric data represented as a semiregular mesh, based on locally adaptive Wiener filtering. The degree of denoising is controlled by a single parameter (an estimate of the relative noise level) and the time required for denoising is independent of the magnitude of the estimate. The performance of the algorihm is sufficiently fast to allow interactive local denoising.

1 Introduction

The complexity of the models used in computer graphics, visualization and geometric modeling applications constantly increases. It becomes more and more difficult to create such models by hand, and 3D scanning is emerging as an attractive alternative. However, the raw data produced by 3D scanners (range images or point clouds) are usually far from usable in any application. Considerable number of algorithms were developed for processing such data. A typical processing pipeline includes several stages:

- registration of raw data to create a single point cloud;
- conversion of the point cloud to an arbitrary fine polygonal mesh;
- decimation and reparameterization of the resulting mesh.

While the reparameterization step is not essential for every application, it is often desirable when a complex model has to be modified, stored and displayed interactively. As it was recently shown [10, 9], reparameterization combined with correctly chosen compression techniques results in substantial reduction in error (by factor of four) for compressed geometry compared to methods preserving fine mesh connectivity.

This result is not surprising if we consider the geometric data represented by a mesh from the informational point of view. In contrast to images, there are three distinct types of information associated with a mesh: connectivity (which vertices are connected by edges), geometry (vertex positions sampled from the original surface), and topology (topological structure of the surface represented implicitly by connectivity). It should be noted that only geometry and topology carry information about the original surface. Connectivity is not explicitly present in the original model and is introduced as an artifact of the algorithms used to convert the point cloud to a mesh.

If we adopt this point of view, there are three types of noise present in the mesh data:

- Connectivity noise, which is the pretty much all of the connectivity information for surfaces of low genus. As the only information about the original surface carried by connectivity is the topological information, connectivity can be replaced by any other as long as topology is preserved. All topological information we theoretically need for a surface of genus 0 can be represented by a tetrahedron.
- Topological noise, which is created by the algorithms used to extract a mesh from the point cloud.
- Geometric noise, due to the errors in measurement and resampling of the data at various processing stages.

Topology-preserving reparameterization can be thought of as removing connectivity noise; recent work [8] addresses the problem of topological noise. We focus on geometric noise removal, assuming that the surface is already reparameterized. While our method can be potentially applied before reparameterization, it works best and is most natural for semiregular meshes.

Reparameterization greatly simplifies the problem, because the surface can be considered as a function, and simple and efficient signal processing approaches can be applied. If reparameterization is ultimately performed on geometric data we believe that denoising is best left to the last stage, because additional noise can be introduced at the resampling stage. This is the case when our approach applies. If reparameterization will not be performed, more complex techniques for denoising on arbitrary meshes [2, 6] are more appropriate.

The algorithm that we propose is based on recent work in image denoising which uses locally adaptive Wiener filtering [16, 17, 21, 22]. The subbands of a multiscale representation are modeled as a product of a Gaussian random vector with a hidden multiplier variable. Estimation of the multiplier leads to the estimation of the local variance and allows standard Wiener denoising. The resulting algorithm is quite efficient, as it requires only a single pass over the surface at each resolution level. It is controlled by a single user-defined parameter, namely, an estimate of the noise magnitude. The performance does not depend on the magnitude of this estimate, i.e. strong noise reduction takes exactly as much time as moderate amount of denoising. Given an infrastructure for supporting semiregular meshes it takes very little time to implement (several hundred lines of code) and can be used for interactive local denoising of a model.

1.1 Denoising

We start with formulating the problem more precisely. *Given a surface corrupted by geometric noise, our goal is to produce a new surface which is as close as possible to the original one*. This task requires implicit or explicit assumptions (model) about the noise and the surface.

It is useful to consider a simple 1D example to understand the problem more clearly. If nothing is known about a 1D signal, it cannot be denoised. However, if the signal contains no frequencies above ω and the source of noise produces only frequencies above ω, low-pass filtering is an ideal denoising procedure. This is a simple example of general pattern common for a wide class (but not all) of denoising approaches: apply a transformation to represent the signal in a domain (in our case, frequency domain) where the noise is well separated from the signal, use assumptions on the structure of the transform coefficients of the signal and noise in order to remove the noise, apply the inverse transform.

For real-world signals, the situation is more complex: these signals typically have spectra overlapping the spectrum of the noise, and low-pass filtering is likely to remove important parts of the signal together with noise. Surface smoothing does precisely that for surfaces [23, 4]. The way to achieve better results is to use additional information about the properties of the signal. A classical example is wavelet thresholding methods for image processing [7]. These methods take advantage of the fact that wavelet bases have good compression properties: in such bases, a typical non-noisy image will have mostly small coefficients, and only few large ones. Eliminating the small coefficients does not alter the reconstructed image much. One reason for this is that natural images often consist of large smooth areas (fine-level coefficients are small) separated by sharp boundaries (fine-level coefficients are large), with boundaries occupying only small area in images.

In contrast, the coefficient magnitude for noise is uniform and, as the signal energy is distributed over a large number of coefficients, each coefficient is likely to be small. This leads to the simple basic algorithm: apply a wavelet transform, threshold the coefficients and apply the inverse transform. For a restricted class of signals corrupted by white Gaussian noise a version of this procedure was proven to be optimal [7]. Note that this procedure is likely to preserve sharp transitions in a signal (edges for images, creases for surfaces).

Nevertheless, it is clear that a part of the useful signal is still removed. One can do better, however, by using additional assumptions. It was shown that Gaussian scale mixture (GSM) model is very suitable for than the statistics of wavelet coefficients of natural images [24, 22]. The combination of this model with Wiener filtering leads to better recovery of the original image. The resulting algorithm is not much more complex than the wavelet thresholding described above — the only additional step involved is local estimation of the signal variance. In the case of Gaussian noise the procedure is nearly optimal.

It turns out that GSM models also appear to reflect properly the statistics of multiscale representations of surfaces. Thus it is natural to apply the GSM-based denoising procedures to surfaces. The algorithm we propose is based on the general ideas of the image denoising algorithms but significantly differs from them in a number of aspects as detailed below.

2 Previous Work

Our algorithm primarily on work in image processing; relatively little has been done on surface denoising. Recent results include Clarenz et al. [2] on denoising of arbitrary meshes and Desbrun et al. [5] on denoising hight fields. In both cases, anisotropic curvature diffusion techniques are used. Our method is fundamentally different and difficult to compare directly to the diffusion-based approaches. Diffusion-based denoising is best regarded as a combination of smoothing and edge enhancement. It is relatively difficult to predict the scale of the noise that will be removed, and the amount of denoising depends on the algorithm running time. At the same time, as demonstrated in [2], curvature flow-based methods can be used on arbitrary meshes, while we assume reparameterization on semiregular meshes. For certain choices of parameters, our method produces results similar to anisotropic curvature diffusion. The methods are compared in greater detail in Section 7.

Recent developments in image denoising show that locally adaptive Wiener filtering is a very powerful technique. This approach was first developed in pixel domain [12, 11] and then extended to the multiresolution domain [16, 17] which allowed further improvement of the results. Local Wiener filtering uses a local estimate of the variance in either the spatial or the multiresolution domain. Wainwright and Simoncelli proposed a model that allows easy estimation of local variance and captures well the local statistical properties of wavelet coefficients of natural images [24]. This model is based on the class of random variables known as Gaussian scale mixtures (GSM). In the GSM model, groups of wavelet coefficients correspond to a product of a Gaussian random vector with a hidden multiplier variable. Similar models have been independently proposed in [15, 3]. The GSM approach combined with Wiener filtering was successfully implemented for image denoising [22]. We suggest a similar technique for noise removal on natural surfaces.

3 Overview of the algorithm

Our denoising procedure follows a common pattern described in Section 1.1. First, we apply a multiresolution transform described in Section 4 to a given noisy surface. We then use the GSM statistical model of the transform coefficients to distinguish the noise from the signal. The details of this step are given in Section 5. Finally, we reconstruct the surface from the denoised coefficients. See Section 6 for the complete description of the algorithm.

4 Multiresolution Surfaces

In this section, we describe in greater detail our assumptions about the parametric surface representation, and the specific representation we use.

It is generally sufficient to assume that the initial mesh was reparameterized on a mesh with semi-regular connectivity. The connectivity of such meshes can be obtained if we start with a relatively coarse mesh, and refine each face of such mesh regularly, in the simplest case, by recursive quadrisection of faces. The latter, however, not essential for our algorithm: any regular refinement can be used.

As a starting point, we use a Laplacian-pyramid multiresolution representation based on Loop subdivision . We refer the reader to [18, 25] for the details of implementation. The surface is represented by the coarsest level and the details at each level of resolution. The process of converting the finest-resolution data to the sequence of detail sets and the coarsest level mesh is called *analysis*. The process of reconstructing a surface from the coarse mesh and details is called *synthesis* The two processes are applied recursively, with analysis proceeding from finer to coarser levels, and synthesis from coarser to finer. A single step of both processes is illustrated in Figure 1.

For analysis, a smoothing filter is required in addition to subdivision rules. We use a simple Laplacian filter for smoothing.

It is important to note that the details at a finer level of resolution are represented in local frames computed from the previous coarser level. This is a valuable feature for surface editing and a natural way to represent surfaces: details are separated into tangential and normal components and become invariant witrh respect to rigid transforms. However, addition of the local frame makes the transform nonlinear. Our comparison of denoising with and without local frame transformations was inconclusive: it is still unclear if there is a substantial advantage in using local frames other than having a geometrically invariant result.

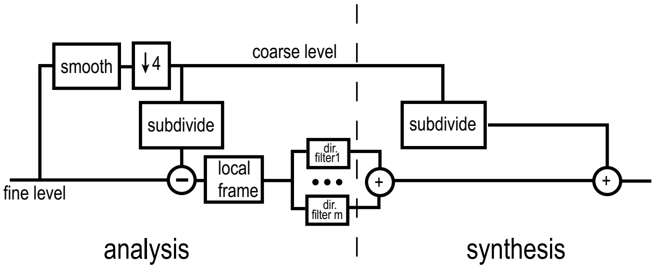


Figure 1: Synthesis and analysis diagrams for multiresolution surfaces.

We use an important modification to the pyramid based on the idea of steerable pyramids [20, 19]: a single detail band is decomposed into multiple directional bands, using directional filters (Figure 1). The number of directional bands m can be chosen arbitrarily by choosing the angular step θ_m. To reconstruct the signal the

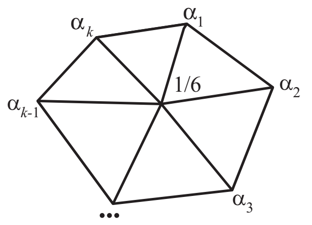


Figure 2: Directional decomposition of the details and the filter mask for a directional filter, $\alpha_i = \cos(\frac{2i\pi}{k} - \theta_m)$.

directional bands are simply added up to produce the detail. Introducing these directional bands is a crucial element of the algorithm. Clearly, the subbands are interdependent, and we need to store only two to be able to reconstruct the result of filtering in any direction.

5 GSM Model for Denoising Multiscale Data

As in the case of natural images, marginal distributions of the multiresolution coefficients of natural surfaces turn out to be sharp-peaked at zero and heavy-tailed (see Figure 3). The peak at zero is produced by the smooth regions, while heavy tails correspond to the slow decay of the coefficients at the edges. We propose to model this distribution by a Gaussian scale mixture process. The GSM random variables include several well-known sharp peaked and heavy tailed distributions such as generalized Gaussians, the α-stable family, and the Student t-variables [24]. One would expect a GSM model to be a good approximation in our case.

We now describe GSM in detail. A random vector X is said to be a Gaussian scale mixture if it is a product of two random variables: $X = \sqrt{z}U$, where z is a positive scalar random variable and U is a zero-mean Gaussian random vector with covariance C_u [1]. U and z are assumed to be independent. The probability density of a GSM variable is:

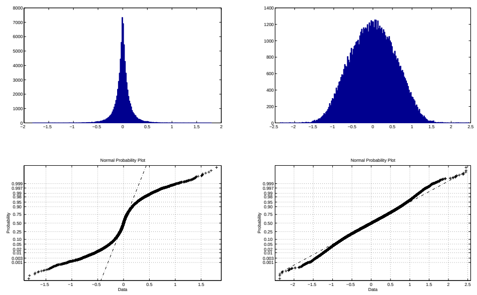
$$\mathcal{P}_x(X) = \int \frac{1}{(2\pi)^{N/2}\,|zC_u|^{1/2}} \exp\left(\frac{-X^T C_u^{-1} X}{2z}\right) \mathcal{P}_z(z)\,dz, \tag{1}$$

where N is the length of vectors U and X. Notice that normalized GSM variable $X/\sqrt{z}$ is Gaussian distributed which allows easy estimation of the statistical properties of the data. In particular, the Wiener filtering of the noisy GSM data should be close to optimal.

We assume that the directional detail coefficients in a single-ring neighborhood of a vertex on each level of a multiresolution mesh follows the GSM model. We also assume independence of the multipliers corresponding to different neighborhoods, even though the neighborhoods are overlapping. Moreover, in order to simplify the computations we treat both the coefficients of the noise and the signal in each neighborhood as uncorrelated (but not necessarily independent) and set their covariance matrices to be multiples of the identity $C_u = \sigma_u^2 I$, $C_w = \sigma_w^2 I$; we assume the variance of noise known (in practice, it is estimated by the user). While it is possible to vary σ_w, we use a single value for the whole surface which is a reasonable assumption for scanned models.

One can test how well the GSM model describes actual data. Let X be a vector corresponding to a single ring of pyramid coefficients around a vertex of a "clean" surface, with x the coefficient at the center of the ring. If the model is correct and there is a good estimate $\widehat{z}(X)$ of $z(X)$ then the distribution of the normalized coefficient $x' = x/\sqrt{\widehat{z}(X)}$ should be close to a Gaussian. We use the maximum likelihood estimator for the multiplier [24, 16, 22]:

$$\widehat{z}(X) = (X^T C_u^{-1} X)/N.$$



before dividing by $\sqrt{\widehat{z}}$ after dividing by $\sqrt{\widehat{z}}$

Figure 3: Histograms and normal probability plots of the pyramid coefficients before and after normalization of each coefficient by the estimate of hidden multiplier $\sqrt{\widehat{z}}$.

Within our assumptions $C_u = \sigma_u^2 I$ and $\sigma_u^2 \widehat{z}(X) = \frac{X^T X}{N}$ is just a local estimate of the variance of x. Figure 3 shows the marginal histograms and normal probability plots of x and x'. Presented data comes from one component of the third level of the multiresolution decomposition of the denoised surface (the model of a dog); similar results were obtained for a number of other scanned models. The histogram of the normalized coefficients is nearly Gaussian, and its corresponding normal probability plot lies nearly along a line. Thus the GSM model does a reasonable job approximating the data.

Our main goal is to estimate the multiplier z in the presence of noise. This will allow us to compute the variance of each element and use the Wiener filter to remove the noise. Suppose that vector Y is obtained from X by adding Gaussian white noise with variance σ_w^2 and mean 0, $Y = X + \sigma_w W$. If X is a GSM vector then each observed noisy coefficient can be represented as $y = \sqrt{z}u + \sigma_w w$, where σ_w^2 is the variance of the noise and w has Gaussian distribution with variance 1 and mean 0. If the value of z were known, then y would also be Gaussian distributed, and the optimal estimate of x would be the linear (Wiener) solution:

$$\widehat{x} = \frac{z\sigma_u^2}{z\sigma_u^2 + \sigma_w^2}\, y. \tag{2}$$

We use the maximum likelihood estimator in order to obtain $\widehat{z}(Y)$, $\widehat{z}(Y) = \left(1/\sigma_u^2\right)\left(Y^T Y/N - \sigma_w^2\right)$. The derivation of this result is given in [16]. When applying this formula to the real data, one often gets a small negative value for $\widehat{z}(Y)$. This happens because the neighborhood is not large enough to capture the statistics of the data or the estimated noise level is too large. In this cases we set $\widehat{z}(Y)$ to zero. We estimate the variance of the center of a neighborhood Y as

$$\widehat{z}\sigma_u^2 = \max\left(Y^T Y/N - \sigma_w^2, 0\right). \tag{3}$$

Equations (2), (3) are used in our denoising algorithm.

6 Denoising Algorithm

We implemented the results of previous sections in the denoising procedure. It consists of there steps: 1) multiresolution decomposition (Section 4). 2) Noise removal using formulas (2), (3) on each level of the decomposition. 3) Reconstruction.

To use formulas (2) and (3) one needs to know the variance of the noise σ_w^2. This is the parameter supplied by the user. In Section 7, we show the results for various values of σ_w. It is also possible to choose different numbers m of directional components for the filters, but, not surprisingly, 6 is the best choice for semiregular triangular meshes.

Denoising Algorithm

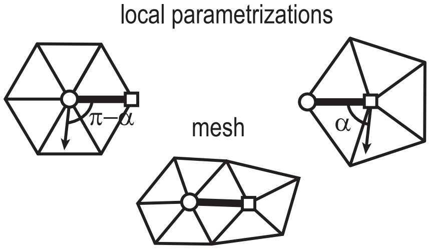


Figure 4: Local alignment of directions

1. Perform analysis of the mesh as described in Section 4.
2. For the details at each level of the decomposition
a) Compute the aligned directional components of the neighborhood of each vertex in the detail mesh in m directions.
b) Estimate the variance of each directional coefficient using formula (3); if the result is negative set the variance to zero.
c) Replace each directional coefficient by its Wiener estimate (2).
d) Replace the value of the center with the combination of the m denoised directional coefficients.
3. Reconstruct the surface mesh.

Unlike images, for which directional components can be chosen to be oriented consistently along global directions, for general surfaces this is not possible. However, as our algorithm is local, only local alignment is required. To choose the aligned directional components, we assume that the single-ring neighborhood of a vertex at a fixed refinement level is parameterized over a k-gon (Figure 4). As shown in the figure, for an arbitrary edge fixed as the zero direction one can pick corresponding directions for the filters for surrounding vertices. The result of filtering in one of these directions is simply a linear combination of two directional details.

7 Results and Discussion

Results of applying our algorithm to data with a high level of artificial noise added for several settings of σ_w are shown in Figure 5. Other denoising results are shown in Figures 9–10. The input parameter σ_w (estimate of the noise level) was chosen as a percentage of the average distance between the initial surface and the coarsest surface. Timings are provided for a relatively slow machine (200 MHz SGI Indigo2).

Comparison with anisotropic curvature diffusion. For comparison, we have implemented anisotropic curvature diffusion as described in [2]. Figure 9 demonstrates that for a certain choice of estimated noise value our algorithm produces results visually similar to anisotropic curvature diffusion [2, 5, 6]. The image was chosen to be as similar as possible to the one shown in [2]. The approaches based on statistical models (such as ours) and deterministic approaches (such as anisotropic curvature diffusion) are based on very different principles and, from mathematical point of view, solve different problems; hence it is difficult to compare the algorithms quantitatively.

The difference between the algorithms merits detailed discussion. The idea of anisotropic curvature diffusion can be summarized as follows: the denoised surface is obtained as the solution at some time τ of an anisotropic diffusion equation. The diffusion tensor is anisotropic near edges, with zero diffusion perpendicular to the edge and maximal along the edge. The edges are detected at each time step using a principal curvature threshold, applied to the curvature values obtained for a smoothed version of the surface. There are three parameters determining the result: the time τ, the constant ϵ controlling the amount of smoothing used before curvatures are computed and λ, the edge detection threshold; ϵ has less impact on the result, so we restrict our attention to τ and λ.

Feature preservation. Both algorithms attempt to preserve important surface features. Anisotropic curvature diffusion detects and attempts to preserve and sharpen edges [2, 6]. Our algorithm has implicit edge detection build in: if there is an edge passing through a point in a certain direction, in orthogonal direction the variance will be considerable and Wiener filtering will not reduce these coefficients by much if at all. The advantage of our approach is that there is no global threshold λ for curvature-controlling edge detection; this parameter is difficult to pick. This can be also regarded as disadvantage as there is no direct edge detection control. The best our algorithm can do is to preserve the noise perpendicular to the edge near the edge; anisotropic curvature diffusion can enhance edges. This is useful in the cases of man-made objects for which a collection of smooth surfaces with sharp edges is a good model. This is less useful and can be harmful for natural object, which seldom have sharp edges. For such objects increasingly sharp edges tend to appear at random locations.

Generality. Our algorithm relies on the multiresolution structure of the mesh, hence applies only to models that were reparameterized on semiregular meshes. In contrast, curvature diffusion works on an arbitrary mesh. while it might be possible to generalize our algorithm to hierarchies on irregular meshes, this would make it significantly more complex.

Running time. With anisotropic curvature diffusion, if a large amount of denoising is desired, large values of τ should be used and the algorithm takes longer, even if implicit integration and large time steps are used. Our algorithm takes exactly the same time no matter how much denoising is desired. For the specific example shown in Figure 9, our conjugate-gradient implementation of curvature diffusion is significantly slower than the GSM algorithm: (260 sec. vs. 16 sec.) However, an efficient multigrid solver is likely to make the times much closer. Our algorithm is fast enough to enable interactive applications.

Locality. Our algorithm can be easily applied locally (the video submitted with the paper shows an interactive application when the noise is removed locally). It is not clear how anisotropic diffusion will behave when applied locally.

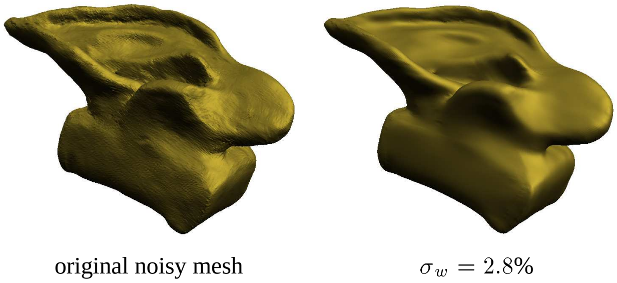
Ease of implementation. As we have mentioned, our algorithm takes very little effort to implement: it is a simple iteration over vertices with filters applied to the immediate neighbors. By comparison, curvature diffusion requires a good solver running on arbitrary meshes to be efficient, and even the basic algorithm requires more work.

8 Conclusions

We have presented a new algorithm for denoising of natural surfaces. It is based on a multiresolution steerable decomposition and utilizes a GSM statistical model of the transform coefficients. The results of our experiments are quite encouraging and compare favorably with other techniques; our algorithm ensures noise removal while preserving essential geometrical features.

Our results are just a first step in applying the GSM model for the denoising of surfaces. One can employ different multiresolution decompositions and extend the denoising algorithm along the lines suggested in [22]. In particular a non-trivial covariance structure of the coefficients within each neighborhood can be used.

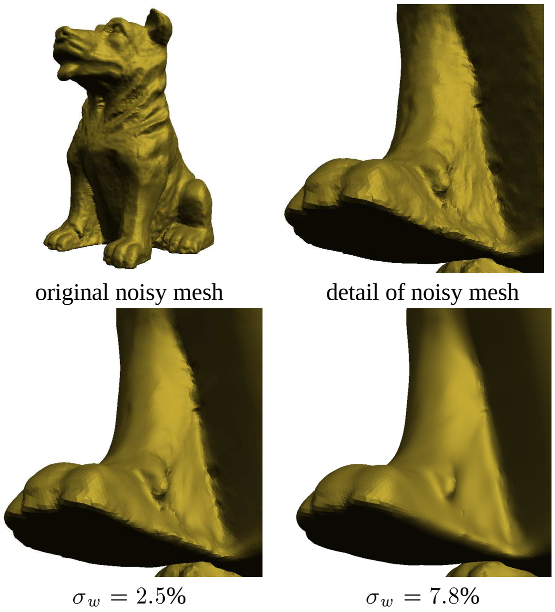
Our algorithm is based on the GSM assumption on the statistics of the multiscale coefficients. We have observed that it is a reasonable assumption for several models, but clearly more extensive studies are necessary.



original noisy mesh

$\sigma_w = 2.8\%$

Figure 9: Denoising a scanned and parameterized model of an ear (335 thousand triangles, 360 thousand after parameterization).



original noisy mesh

detail of noisy mesh

$\sigma_w = 2.5\%$

$\sigma_w = 7.8\%$

Figure 10: Denoising a scanned and parameterized model of a dog for different choices of σ_w (264 thousand triangles, 391 thousand triangles after parameterization). Denoising time: 59 secs.

Many questions which have well-understood answers for images (e.g. measure of difference between images) are much more difficult for surfaces and require further study to make it possible to compare algorithms in a more quantitative manner.

Note on Figure 7: We believe the small scale random texture visible on the surface is an artefact of the scanning and reconstruction process; it is similar in scale to the texture we have observed on other objects scanned using Cyberware scanners; according to Levoy et al. [13] the scale of the details on the surface of polished marble is less than 30 nm, which is far less than the characteristic texture size. On the other hand, the texture is too random and too closely spaced (0.5-1 mm is the characteristic scale) to be chisel marks which are more likely to be at least 2 mm wide [14].

References

[1] D Andrews and C Mallows. Scale mixtures of normal distributions. *J. Royal Stat. Soc.*, 36:99–, 1974.

[2] U. Clarenz, U. Diewald, and M. Rumpf. Anisotropic geometric diffusion in surface processing. In *Proceedings of Visualization 2000*, 2000.

[3] M S Crouse, R D Nowak, and R G Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Trans. Signal Proc.*, 46:886–902, April 1998.

[4] M Desbrun, M Meyer, P Schroder, and A Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Siggraph'99*, 1999.

[5] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Anisotropic feature-preserving denoising of height fields and bivariate data. In *Proceedings of Graphics Interface 2000*, 2000.

[6] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators in nd. submitted, 2000.

[7] D Donoho. Denoising by soft-thresholding. *IEEE Trans. Info. Theory*, 43:613–627, 1995.

[8] Igor Guskov and Zoë Wood. Topological noise removal. In *Proceedings of Graphics Interface 2001*, 2001.

[9] Andrei Khodakovsky and Igor Guskov. Normal mesh compression. submitted for publication.

[10] Andrei Khodakovsky, Peter Schröder, and Wim Sweldens. Progressive geometry compression. *Proceedings of SIGGRAPH 2000*, pages 271–278, July 2000. ISBN 1-58113-208-5.

[11] D T Kuan, A A Sawchuk, T C Strand, and P Chavel. Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Pat. Anal. Mach. Intell.*, PAMI-7:165–177, March 1985.

[12] J S Lee. Digital image enhancement and noise filtering by use of local statistics. *IEEE Pat. Anal. Mach. Intell.*, PAMI-2:165–168, March 1980.

[13] M. Levoy. Digital michalangelo project web site. `http://www.graphics.stanford.edu/projects/mich/other-body-parts/other-body-parts.html`.

[14] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The digital michelangelo project: 3d scanning of large statues. *Proceedings of SIGGRAPH 2000*, pages 131–144, July 2000. ISBN 1-58113-208-5.

[15] S M LoPresto, K Ramchandran, and M T Orchard. Wavelet image coding based on a new generalized gaussian mixture model. In *Data Compression Conf*, Snowbird, Utah, March 1997.

[16] M K Mihçak, I Kozintsev, K Ramchandran, and P Moulin. Low–complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Trans. on Signal Processing*, 6(12):300–303, December 1999.

[17] P Moulin and J Liu. Analysis of multiresolution image denoising schemes using a generalized Gaussian and complexity priors. *IEEE Trans. Info. Theory*, 45:909–919, 1999.

[18] K. Pulli and M. Lounsbery. Hierarchical editing and rendering of subdivision surfaces. Technical Report UW-CSE-97-04-07, Dept. of CS&E, University of Washington, Seattle, WA, 1997.

[19] E P Simoncelli and W T Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Second Int'l Conf on Image Proc*, volume III, pages 444–447, Washington, DC, October 1995. IEEE Sig Proc Society.

[20] E P Simoncelli, W T Freeman, E H Adelson, and D J Heeger. Shiftable multiscale transforms. *IEEE Trans Information Theory*, 38(2):587–607, March 1992. Special Issue on Wavelets.

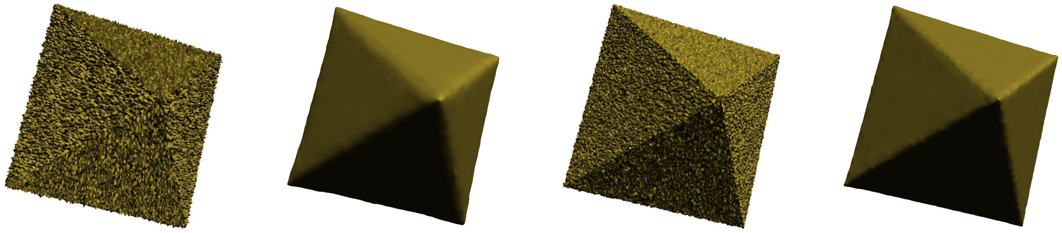
[21] V Strela. Denoising via block Wiener filtering in wavelet domain. In *3rd European Congress of Mathematics*, Barcelona, July 2000. Birkhäuser Verlag.

[22] V Strela, J Portilla, and E Simoncelli. Image denoising using a local gaussian scale mixture model in the wavelet domain. In *Proc SPIE, 45th Annual Meeting*, San Diego, July 2000.

[23] G Taubin. A signal processing approach to fair surface design. In *Siggraph'95*, pages 351–358, 1995.

[24] M J Wainwright and E P Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Adv. Neural Information Processing Systems*, volume 12, pages 855–861, Cambridge, MA, May 2000. MIT Press.

[25] Denis Zorin, Peter Schröder, and Wim Sweldens. Interactive multiresolution mesh editing. *Proceedings of SIGGRAPH 97*, pages 259–268, August 1997. ISBN 0-89791-896-7. Held in Los Angeles, California.



noisy mesh $\sigma_w = 12\%$ noisy mesh $\sigma_w = 4.3\%$

Figure 5: Denoising a simple mesh with artificial noise for different choices of estimated noise σ_w (98 thousand triangles).

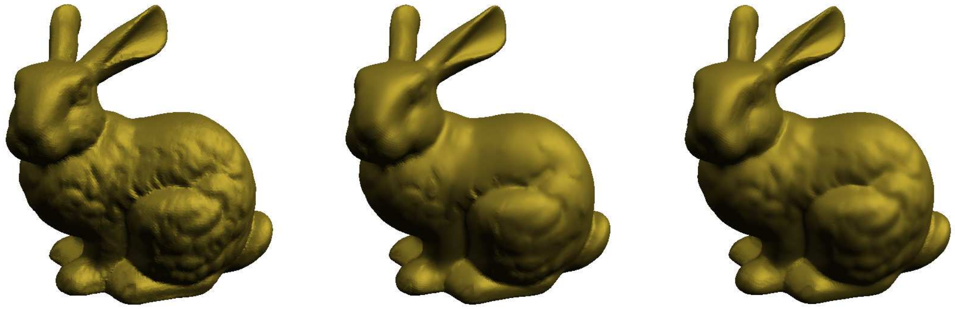
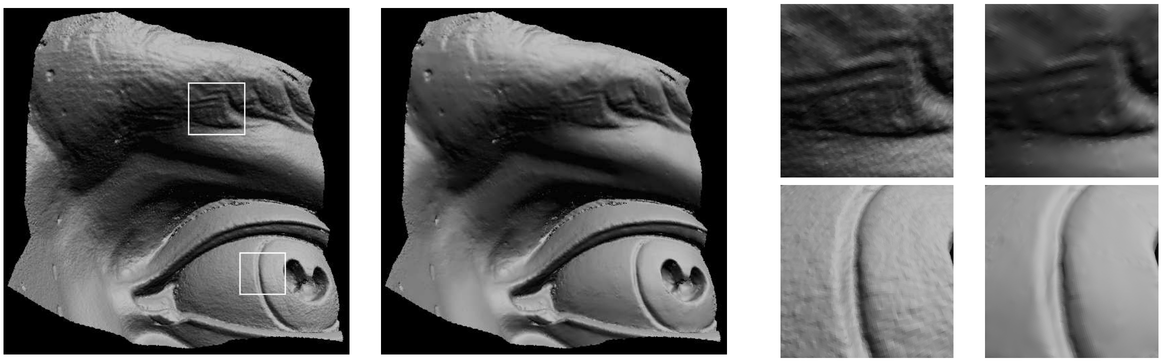


Figure 6: Denoising the reparameterized Stanford bunny mesh. (71 thousand triangles, 145 thousand after parameterization). From left to right: the reparameterized mesh; denoised by our algorithm; by anisotropic geometric diffusion method.



noisy mesh $\sigma_w = 1.4\%$ details

Figure 7: Denoising a part of the model of Michalangelo's David. (scanned at 0.29mm resolution; unstructured mesh 0.63 mln. triangles, 1,2 mln. triangles after parameterization). Denoising time: 167 secs. Original mesh courtesy of Marc Levoy, Stanford Computer Graphics Lab. From left to right: original mesh; denoised mesh; magnified views of two areas on the mesh before and after denoising. Note that chisle marks in the first area are preserved, while the small-scale noise is removed.

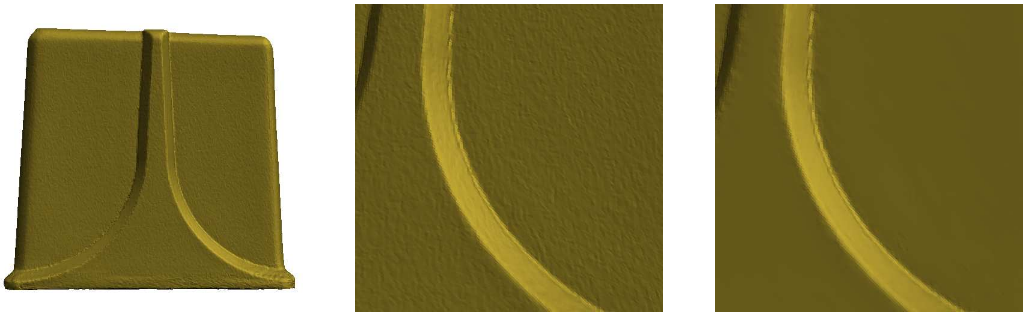


Figure 8: Denoising a scanned and parameterized model of a Ascension Technologies transmitter. From left to right: original model; magnified view of an area before and after denoising. Note that in the flat areas all small scale features were removed, with almost no change at the creases.