Non-Iterative, Feature-Preserving Mesh Smoothing

Thouis R. Jones*, Frédo Durand*, Mathieu Desbrun†
*Computer Science and Artificial Intelligence Laboratory, MIT
†Computer Science Department, USC

***Abstract*— With the increasing use of geometry scanners to create 3D models, there is a rising need for fast and robust mesh smoothing to remove inevitable noise in the measurements. While most previous work has favored diffusion-based iterative techniques for feature-preserving smoothing, we propose a radically different approach, based on robust statistics and local first-order predictors of the surface. The robustness of our local estimates allows us to derive a *non-iterative* feature-preserving filtering technique applicable to arbitrary "triangle soups". We demonstrate its simplicity of implementation and its efficiency, which make it an excellent solution for smoothing large, noisy, and non-manifold meshes.**

I. Introduction

With geometry scanners becoming more widespread and a corresponding growth in the number and complexity of scanned models, *robust* and *efficient* geometry processing becomes increasingly desirable. Even with high-fidelity scanners, the acquired 3D models are invariably noisy [1], [2], and therefore require smoothing. Similarly, shapes extracted from volume data (obtained by MRI or CT devices, for instance) often contain significant amounts of noise, be it topological [3], [4] or geometric [5], [6], that must be removed before further processing. Removing noise while preserving the shape is, however, no trivial matter. Sharp features are often blurred if no special care is taken. To make matters worse, scanned meshes often have cracks and non-manifold regions.

A. Previous Work

A wide variety of mesh smoothing algorithms have been proposed in recent years. Taubin [5] pioneered fast mesh smoothing by proposing a simple, linear and isotropic technique to enhance the smoothness of triangulated surfaces without resorting to expensive functional minimizations. Desbrun et al. [6] extended this approach to irregular meshes 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. Other improvements followed, such as a method combining geometry smoothing and parameterization regularization [7]. However, these efficient techniques are all isotropic, and therefore indiscriminately smooth noise *and* salient features: a noisy cube as input will become extremely rounded before becoming smooth. This lack of selectivity is limiting in terms of applications.

Feature-preserving surface fairing has also been proposed more recently [8], [9], [8], [10], [11], mostly inspired by image processing work on scale-space and anisotropic diffusion [12]. The idea behind these approaches is to modify the diffusion equation to make it non-linear and/or anisotropic. The curvature tensor determines the local diffusion, thus preserving (or even enhancing) sharp features. Although the results are of much higher quality, these methods rely on shock formation to preserve details, which affects the numerical conditioning of the diffusion equations. This can cause significant computational times, even after mollification of the data.

Other researchers have proposed diffusion-type smoothing on the normal field itself [13], [14], [15], [16]; fairing is achieved by first smoothing the normal field, and then evolving the surface to match the new normals. Here again, the results are superior to those from isotropic techniques, but with roughly similar computational cost as anisotropic diffusion on meshes.

Locally adaptive Wiener filtering has also been used with success for 3D meshes by Peng et al. [17], and for point-sampled surface by Pauly and Gross [18]. However, these methods rely on semi-regular connectivity or local parameterization, respectively. A different approach is taken by Alexa [19], similar to anisotropic diffusion, though with larger neighborhoods used for filtering. This also results in a fast method, and avoids some of the limitations discussed below, but still relies on a connected mesh and iterative application.

The diffusion-based feature-preserving techniques are, in essence, all *local and iterative*. From very local derivative approximations, geometry is iteratively updated until the noise has been diffused sufficiently. Numerical tools, such as preconditioned conjugate gradient solvers or algebraic multigrid solvers, can be used to improve efficiency by making the iterations more stable. Nevertheless, the diffusion-type setting that is the basis of these approaches requires manifoldness, not always present in raw scanned data. In order to address the need for robust and fast feature preserving smoothing, we propose to recast mesh filtering as a case of robust statistical estimation.

B. Robust Statistics

The field of *robust statistics* is concerned with the development of statistical estimators that are robust to the presence of outliers and to deviations from the theoretical distribution [20], [21]. Naïve estimators such as least-squares give too much influence to outliers, because the error function or norm they minimize is large for data points far from the estimator (quadratic in the case of least squares). In contrast, robust estimators are based on minimizing an energy that gives low weight to outliers, as illustrated by the Gaussian robust norm

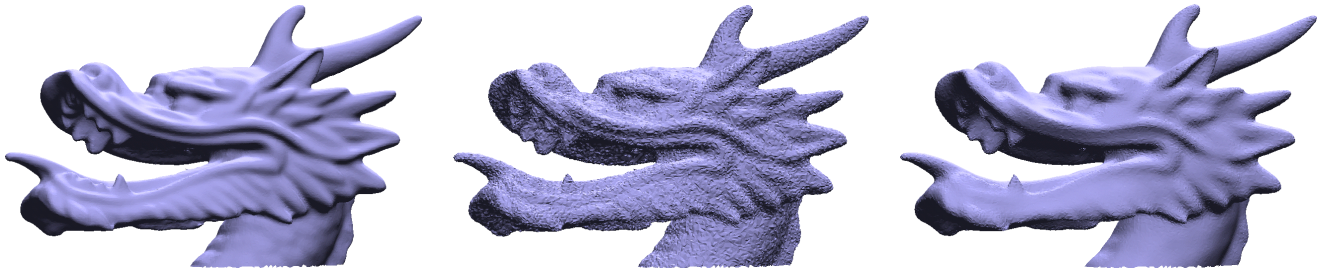


Fig. 1. The dragon model (left) is artificially corrupted by Gaussian noise ($\sigma = 1/5$ of the mean edge length) (middle), then smoothed in a single pass by our method (right). Note that features such as sharp corners are preserved.

in Fig. 2: after a certain distance from the estimator, controlled by a scale σ, an increasingly distant outlier has only limited effect.

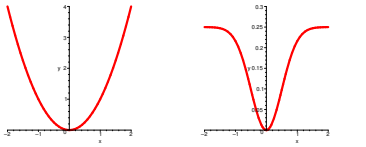


Fig. 2. Least-square vs. Gaussian error norm (after [22]).

Black et al. [22] showed that anisotropic diffusion can be analyzed in the framework of robust statistics. The edge-stopping functions of anisotropic diffusion [12] serve the same role as robust energy functions. Anisotropic diffusion minimizes such a function using an iterative method.

The bilateral filter is an alternative edge-preserving filter proposed by Smith and Brady [23] (see also [24]). The output $E(p)$ at a pixel p is a weighted average of the surrounding pixels in the input image I, where the weight of a pixel q depends not only on the spatial distance $||q - p||$, but also on the signal difference $||I(q) - I(p)||$:

$$E(p) = \frac{1}{k(p)} \sum_{q \in \Omega} I(q) \;\; f(q - p) \;\; g(I(q) - I(p)), \qquad (1)$$

where $k(p)$ is the normalization factor

$$k(p) = \sum_{q \in \Omega} f(q - p) \;\; g(I(q) - I(p)) \qquad (2)$$

In practice, a spatial Gaussian f and a Gaussian influence weight g are often used.

This dependence on the signal difference allows one to give less influence to outliers. Durand and Dorsey [25] show that bilateral filtering is a robust estimator and that a Gaussian influence weight corresponds to minimizing a Gaussian error norm. They also show that bilateral filtering is essentially similar to anisotropic diffusion. However, the bilateral filter is a *non-iterative* robust estimator, or *w-estimator* [20], which makes it more efficient than iterative schemes. In particular, this approach does not have to deal with shock formation at strong edges, and is therefore more stable than anisotropic diffusion. See also the work by Barash [26] and Elad [27].

C. Contributions

In this paper, we propose a novel feature-preserving fairing technique for arbitrary surface meshes based on non-iterative, robust statistical estimations[1]. Contrasting drastically with previous diffusion-based methods, our fast and stable approach relies on local robust estimations of shape. Moreover, our method does not require manifoldness of the input data, and can therefore be applied to "triangle soup".

One of our key insights is that feature preserving smoothing can be seen as estimating a surface in the presence of outliers. The extension from existing robust statistics techniques to surface filtering is, however, far from trivial because of the nature of the data: in a mesh, the *spatial location* and the *signal* are one and the same. This makes the definition of outliers and the control of their influence challenging. We propose to capture the smoothness of a surface by defining local first-order *predictors*. Using a robust estimator, we find the new position of each vertex as weighted sum of the predictions from the predictions in its spatial neighborhood. We will show that our method treats points on opposite sides of a sharp feature as *outliers* relative to one another. This limits smoothing across corners, which preserves features.

II. Non-Iterative, Feature-Preserving Mesh Smoothing

We cast feature-preserving mesh filtering as a robust estimation problem on vertex positions. The estimate for a vertex is computed using the prediction from nearby triangles. Moving each vertex to a robust estimate of its position removes noise and smoothes the mesh while preserving features.

A. Robust Estimation of Vertex Positions

To allow the proper definition of outliers, we must separate spatial location and signal. We capture surface smoothness using first-order predictors, i.e., tangent planes. In practice we use predictors based on triangles of the mesh as they represent natural tangent planes to the surface. The surface predictor Π_q defined by a triangle q is just the tangent plane of q (see Fig. 3a).

We use a method analogous to bilateral filtering for images [23], [24], but we form the estimate for the new position of a vertex p based on the *predictions* $\Pi_q(p)$ from its spatially nearby triangles. We employ a spatial weight f that depends

[1] In a contemporaneous work, Fleishman et al. [28] present a similar technique (cf. Section IV).

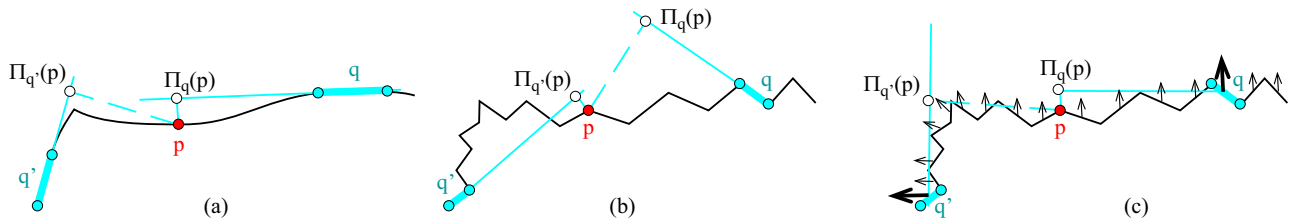


Fig. 3. (a) The prediction $\Pi_q(p)$ for a point p based on the surface at q is the projection of p to the plane tangent to the surface at q. Points across a sharp feature result in predictions that are farther away, and therefore given less influence. (b) Noisy normals can lead to poor predictors. (c) Mollified normal alleviate this problem. Note that corners are preserved because points are not displaced by the mollification: only the normals are smoothed.

on the distance $||p - c_q||$ between p and the centroid c_q of q. We also use an influence weight g that depends on the distance $||\Pi_q(p) - p||$ between the prediction and the original position of p. Finally we weight by the area a_q of the triangles to account for variations in the sampling rate of the surface. The estimate p' for a point on surface S is then:

$$p' = \frac{1}{k(p)} \sum_{q \in S} \Pi_q(p)\ a_q\ f(||c_q - p||)\ g(||\Pi_q(p) - p||), \quad (3)$$

where k is a normalizing factor (sum of the weights)

$$k(p) = \sum_{q \in S} a_q\ f(||c_q - p||)\ g(||\Pi_q(p) - p||), \quad (4)$$

Gaussians are used both for the spatial weight f and for the influence weight g in this paper. Other robust influence weights could also be used, but Gaussians have performed well in our experiments, as well as the work of others [23], [24], [25]. The amount of smoothing is controlled by the widths σ_f of the spatial and σ_g of the influence weight Gaussians. As can be seen in Fig. 3(a), predictions from across a sharp feature are given less weight because the distance between the prediction $\Pi_q(p)$ and p is large, and is penalized by the influence weight g.

Filtering a mesh involves evaluating Equation (3) for every vertex and then moving them as a group to their estimated positions. Note that no connectivity is required beyond triangles: we simply use the Euclidean distance to the centroid of surrounding triangles to find the spatial neighborhood of a vertex. A wider spatial filter includes a larger number of neighbors in the estimate, and can therefore remove a greater amount of noise, or smooth larger features. The influence weight determines when the predictions of neighbors are considered outliers (by according them less weight), and thereby controls the size of features that are preserved in the filtering.

As shown by Black et al.[22] and Durand and Dorsey [25], the evaluation of Equation (3) corresponds to approximately minimizing

$$E(p) = \int_{q \in S} f(||c_q - p||)\ \rho(||\Pi_q(p) - p||)\ dq, \quad (5)$$

where $\rho(||\Pi_q(p) - p||)$ is the distance between a vertex and its predicted position under a robust error norm ρ [21]. Robust error norms are bounded above by some maximum error, as discussed in Section I-B. In our case, we seek to minimize a Gaussian error norm (see Fig. 2); The relation between ρ and g is $g(x) \equiv \rho'(x)/x$ [22], [25], [21].

B. Mollification

Our predictors are based on the orientation of the tangent planes, as defined by the facet normals. Since the normals are first-order properties of the mesh, they are more sensitive to noise than vertex positions (Fig. 3(b)). Even so, the robust estimator performs well; we can however significantly improve the estimate with *mollification* [20], [29]. We mollify our estimators by smoothing the normals.

We first perform a pass of non-robust smoothing using Equation (3) without the influence weight, and with the simplest predictor, $\Pi_q(p) = c_q$, corresponding to simple Gaussian smoothing. We use a different width for the spatial filter during mollification, and in practice have always set this to $\sigma_f/2$. The normals of the mollified mesh are then copied to the facets of the original mesh before the robust filtering is performed. Notice that we do *not* alter the positions of the vertices at all: we only need to mollify the first-order properties (the normals), not the 0-order location (see Fig. 3(c)). Some normals might be improperly smoothed by mollification near corners. This is why it is important not to move the vertices during mollification in order to preserve these features. Fig. 4 shows a comparison of filtering with and without mollification. Without mollification, the facet normals of the mesh are much noisier, resulting in less effective smoothing.

C. Feature Preservation

The filtering method we have proposed preserves features through two combined actions. First is the use of a robust influence weight function, as discussed, while the second is our use of a predictor for vertex positions based on the tangent planes of the mesh. This predictor does not move vertices located at sharp features separating smooth areas of the mesh, since feature vertices are "supported" by the prediction from both sides. Neither of these actions is sufficient alone (see the discussion of Fig. 7 below for examples of how the influence weight affects the filter), but together they provide excellent feature-preserving behavior. Note the connection to bilateral filtering for images, which uses a prior of piecewise constant images. This is a special case of our formulation, corresponding to the predictor $\Pi_q(p) = c_q$. As well, the use of the existing mesh facets helps to simplify our formulation and its implementation, as they provide direct estimates for surface tangents.

In essence, our technique also relates to ENO/WENO methods [30], a class of finite-difference-based, shock capturing

(a) Initial mesh (b) isotropic smoothing [6] (c) our approach without mollification (d) our approach with no influence weight g (e) our complete approach

Fig. 4. Isotropic filtering vs. our method. Notice that details such as the upper and lower lids and the eye are better preserved, while flat regions are equivalently smoothed.

numerical techniques for hyperbolic PDE integration. In a nutshell, they seek to avoid dissipation of shocks –the equivalent of sharp features in our geometric setting. They base their local evaluation of differential quantities only on the local neighbors of similar field value. For example, the evaluation of a second derivative at a shock is not centered as this would use information on both sides; in contrast the evaluation is one-sided to prevent numerical dissipation. Robust statistics offers a principled framework to extend similar key concepts to geometry.

III. Results

We demonstrate our results in Figs. 1 and 4-8. In each case, we use Gaussians for the spatial (f), influence weight (g), and mollification filters with standard deviations of $\sigma_f, \sigma_g, \frac{\sigma_f}{2}$, respectively. This choice for g corresponds to a Gaussian error norm. All meshes are rendered with flat shading to show faceting. Table I summarizes our results and the parameters used to generate them, given in terms of the mean edge length ($||e||$) of the particular mesh. The cost of each evaluation of Equation (3) depends on the number of facets that lie within the support of f, so the time to filter a mesh grows approximately as the size of the mesh times σ_f^2.

Fig. 4 shows a portion of a mesh from a 3D scan of a head. We show the original mesh, the result of isotropic smoothing by Desbrun et al. [6], and our technique. We present this comparison to demonstrate the effectiveness of our approach for smoothing, even on noisy meshes with topological errors.

A comparison to the Wiener filtering approach of Peng et al [17] is shown in Fig. 6. The parameters for our method were chosen to visually match the smoothness in flat areas. Our method preserves features better for larger amounts of smoothing (compare 6(d) and 6(e)). Also, as noted previously, Wiener filtering requires resampling the input to a semi-regular mesh, and only operates on surfaces with manifold topology, while our method can be applied more generally, to non-regular and disconnected meshes. We estimate that their implementation would take about 15 seconds to filter this mesh on our machine, in comparison to 60 (or more, depending on the smoothing parameters) for our technique.

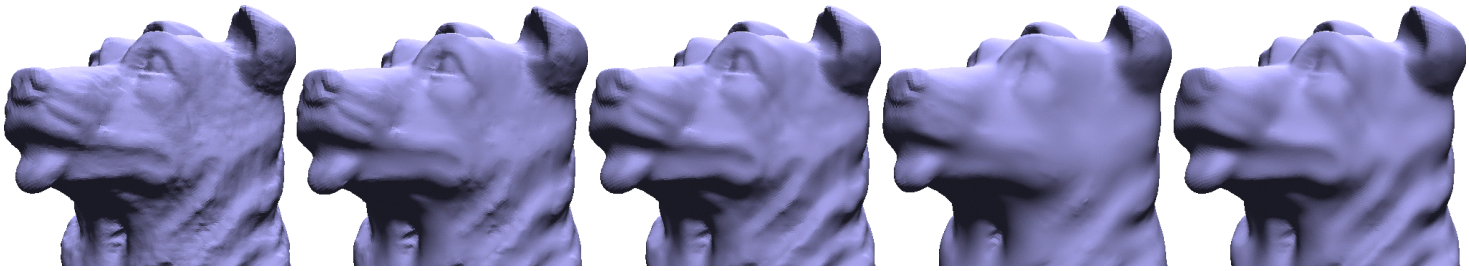
Model	Fig.	Verts.	Time	$\sigma_f/\|\|e\|\|/$		$\sigma_g/\|\|e\|\|$	
Dragon head	1	100k	80 s	4	(14)	1	(4)
Face	4(d)	41k	16 s	1.5	(9.2)	0.4	(2.4)
	(c)		10 s	1.5	(0.9)	0.5	(0.3)
Dog	6(c)	195k	82 s	2.7	(6.6))	0.4	(0.9)
	(e)		132 s	4	(9.9)	1.3	(3.3)
Bunny	7(b)	35k	11 s	2	(12)	0.2	(1.2)
	(c)		12 s	2	(1.2)	4	(24)
	(d)		23 s	4	(24)	4	(24)
Venus	10	134k	54 s	2.5	(8.1)	1	(3.3)
Dragon	8	100k	79 s	4	(14)	2	(7)

TABLE I: Results on a 1.4Ghz Athlon with 2GB of RAM. Times do not include the time to load meshes. The σs are expressed as ratios of the mean edges length $||e||$, and the numbers in parentheses are in thousandths of the bounding box diagonal for the particular meshes.

In Fig. 10 we compare our method to anisotropic diffusion smoothing by Clarenz et al. [9]. The original, noisy mesh is smoothed and smaller features removed by four iterations of diffusion. We have chosen the parameters of our method to match the result as closely as possible. One benefit of anisotropic diffusion is the ability to iteratively enhance edges and corners. Our method is not able to perform such enhancement in a single pass, resulting in a slightly different overall appearance, particularly in the hair, and slightly more noise around edges in the model.

We also show a false-color plot of the confidence measure k in Fig. 10. As can be seen, smoother areas (such as the cheek) and features bordered by smooth areas (such as the edges of the nose) have higher confidence, while curved areas or those near more complicated features have lower confidence. See also Fig. 8.

We show the effects of varying spatial and influence weight function widths in Fig. 7. For a wide spatial filter but nar-



Original Wiener (low noise) our method Wiener (high noise) our method

Fig. 6. Comparison of our method and Wiener filtering. Parameters for Wiener filtering from [17]. Parameters for our method chosen to approximately match surface smoothness in flat areas. (Original and Wiener filtered meshes courtesy of Jianbo Peng.)

original noise removal smooth small features smooth large features

narrow spatial, narrow influence narrow spatial, wide influence wide spatial, wide influence

$(\sigma_f = 2,\ \sigma_g = 0.2)$ $(\sigma_f = 2,\ \sigma_g = 4)$ $(\sigma_f = 4,\ \sigma_g = 4)$

Fig. 7. The effect of varying spatial and influence weight functions. Filter widths σ_f, σ_g given in terms of mean edge length in the mesh. (Mesh from the Stanford University Computer Graphics Laboratory 3D scanning repository.)

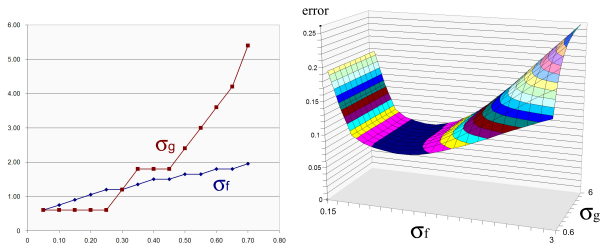


Fig. 5. (a) Values of σ_f and σ_g that yield the most accurate denoising for a mesh corrupted with Gaussian noise, as a function of the variance of the noise. (b) Evolution of the error for a given noise level as a function of σ_f and σ_g. All values are in terms of the mean edge length.

row influence weight, the mesh is smoothed only in mostly flat areas. In the converse, a narrow spatial filter and wide influence weight, small features are smoothed away but larger variations kept. Finally, for a wide spatial filter and wide influence weight, only the largest and strongest features are preserved. See Fig. 8 for a similar example of our method used to remove all but the most salient features of a mesh.

In order to facilitate denoising with our approach, we have performed experiments to find good values for σ_f and σ_g to smooth a model corrupted with a given amount of noise, such as might be produced by a scanner. If the amount of noise can be quantified, by examining an area on the model known to be flat, for example, then the plot in Fig. 5(a) shows the optimal values for σ_f, σ_g from our experiments. These values have been found effective on several models. The surface plot in Fig. 5(b) shows how, for a particular (representative) noise level, the post-filtering error changes. As can be seen, the error is most sensitive to σ_f. We compute the error as the L^2 distance between the original mesh before corruption and the filtered mesh [31].

In other applications, our general approach has been to increase σ_f and σ_g together until the filtered mesh is sufficiently smooth for our goals. We then decrease σ_g until features or noise that we are trying to remove begin to reappear.

All of our results demonstrate the effectiveness of our technique at feature preservation, due to a combination of a robust influence weight function and a first-order predictor, as discussed in Section II-C. In particular, the tips of the ears of the bunny are preserved, as are the head and extremities of the dragon. See also Fig. 9, part of a scan of an origami piece.

We apply our filtering method to a mesh corrupted with synthetic noise in Fig. 1. In the noisy mesh, each vertex is displaced by zero-mean Gaussian noise with $\sigma_{\text{noise}} = \frac{1}{5}$ of the mean edge length, along the normal. We filter the dragon mesh to recover an estimate of the original shape. For comparison, in the scanned mesh of Fig. 4 we estimate $\sigma_{\text{noise}} \approx \frac{1}{7}$. These results shows the ability of our method to smooth even in the

presence of extreme amounts of noise. Fig. 4 also indicates an area where our algorithm could be improved. Where a feature and noise coincide (e.g. in the nose), it is difficult to correctly separate the two. In Fig. 1, we have aimed for a smoother reconstruction, but lose some details in the process.

We have applied two basic optimizations to our implementation. We truncate the spatial filter at $2\sigma_f$ to limit the number of estimates that must be considered per vertex. This does not noticeably affect the results. We also group vertices and facets spatially for processing, to improve locality of reference.

As presented, our method is not necessarily volume preserving. We have not encountered a mesh where this is an issue. Adjusting the mesh after filtering to preserve its volume is a straightforward extension [6].

IV. Conclusion and Future Work

We have developed a novel, fast feature-preserving surface smoothing and denoising technique, applicable to triangle soups. We use a first-order predictor to capture the local shape of smooth objects. This decouples the signal and the spatial location in a surface, and allows us to use robust statistics as a solid base for feature preservation. A robust error norm is applied to the predictors and the vertex positions, from which we derive an efficient and stable one-step smoothing operator. Mollification is used to better capture shape and to obtain more reliable detection of outliers and features. We have demonstrated our algorithm on several models, for both noise removal and mesh smoothing.

Contemporaneous with this work, Fleishman et al. [28] have proposed an extension of bilateral filtering to meshes with similar goals as this work, but with a different approach. The main contrasts are that vertex normals are computed from the mesh to perform local projections, after which a vertex's updated position is computed as the bilateral filter of its neighborhood treated as a height field. In rough terms, their method is faster but requires a connected mesh. The speed increase is due to two factors: they do not mollify normals, and the density of triangles is roughly half that of vertices in a mesh. They require connectivity to estimate normals. They also apply their filter iteratively, while we have concentrated on a single-pass technique. There is also a fundamental difference in how the two methods form predictions for a vertex's filtered position. Our method projects the central vertex to the planes of nearby triangles, while that of Fleishman et al. projects nearby vertices to the plane of the central vertex. The relative costs and benefits of these two approaches merits further study.

There are several avenues for improvement of our method. The normalization factor k in Equation (3) is the sum of weights applied to the individual estimates from a point's neighborhood (see Fig. 10). It therefore provides a measure of the confidence that should be attached to the estimate of the point's new position, as noted by Durand and Dorsey [25]. We have not made use of the confidence measure k in this work, but feel that it could be a valuable tool in future approaches. In particular, we believe that it could be used to detect areas where a good estimate could not be formed, as on the sharp features in Fig. 1. Such areas could be processed further, perhaps by iterative filtering.

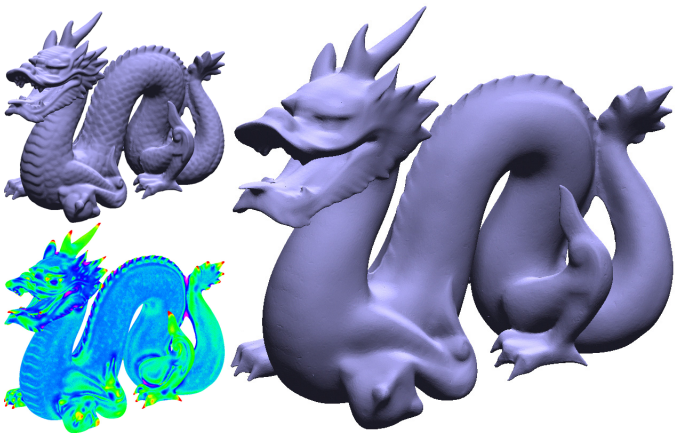


Fig. 8. Original and smoothed dragon, and the confidence k for the smoothed dragon. Note that sharp features are preserved while other details are removed. (Mesh from the Stanford University Computer Graphics Laboratory 3D scanning repository.)

In our experience, the $O(\sigma_f^2)$ growth rate of our algorithm has not been a limiting factor. If it were to become so, a promising approach is to subsample the mesh by simplifying it with some fast method, and then filter the original mesh vertices based on the simplified version. Our method should also extend easily to out-of-core evaluation, since it does not require connectivity information and since the computations are spatially local. This would allow our method to be applied to extremely large models.

Finally, the extension of robust statistics to meshes suggests other possibilities for their application. The influence weight could include other data on the mesh, such as color. It should also be straightforward to extend our filter to other shape representations, such as volume data or point-sample models [32]. In the latter case, where each sample includes a normal, the methods transfer directly, as should the results. We also plan to explore how robust statistics could be added to existing techniques for surface approximation, such as Moving Least Squares [33], to improve their robustness and sensitivity to noise.

a) Acknowledgments: Special thanks to Udo Diewald, Martin Rumpf, Jianbo Peng, and Denis Zorin for providing us with their meshes and results, to Erik Demaine for the origami,

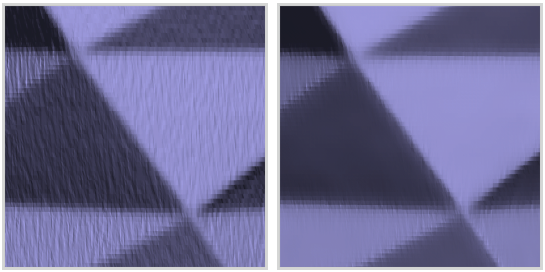


Fig. 9. Corners and straight edges are preserved by our method, as shown in a portion of a 3D scan of an origami sculpture.

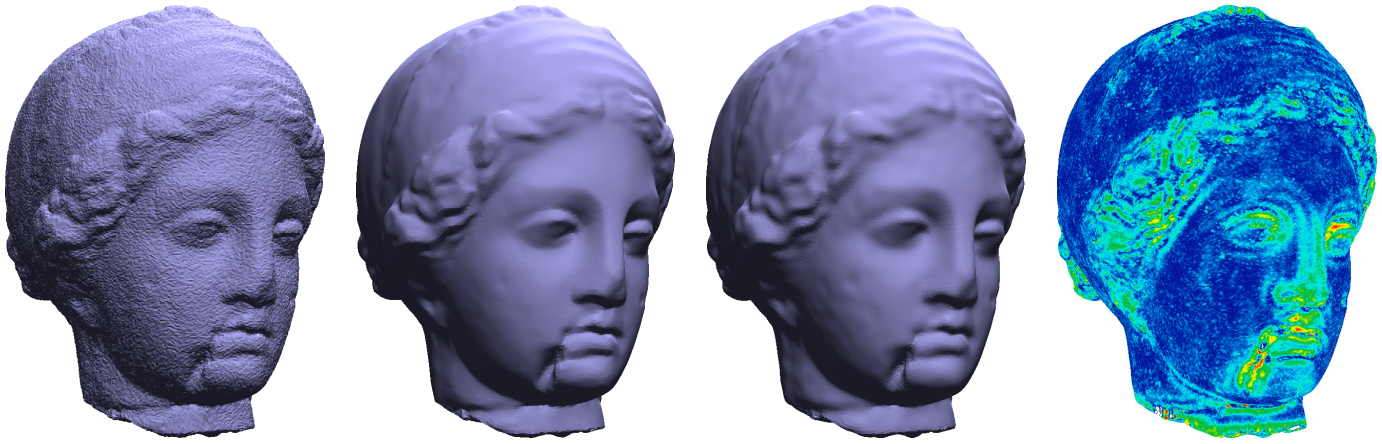


Fig. 10. Comparison of our method with anisotropic diffusion (four iterations) [9]. Parameters for our method chosen to match as best as possible. In the confidence plot, dark blue is high confidence, green and yellow lower confidence, and red least confidence. (Original and anisotropically smoothed mesh courtesy of Martin Rumpf.)

to the Stanford 3D Scanning Repository, and to the SigDraft and MIT pre-reviewers for their feedback. This work was funded in part by the NSF (CCR-0133983, DMS-0221666, DMS-0221669, EEC-9529152).

REFERENCES

[1] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D model acquisition," *ACM Trans. Gr.*, vol. 21, no. 3, pp. 438–446, July 2002.

[2] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," in *Proceedings of SIGGRAPH 2000*, ser. Computer Graphics Proceedings, Annual Conference Series. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, July 2000, pp. 131–144, iSBN 1-58113-208-5.

[3] I. Guskov and Z. Wood, "Topological noise removal," in *Graphics Interface 2001*, June 2001, pp. 19–26.

[4] Z. Wood, H. Hoppe, M. Desbrun, and P. Schröder, "Isosurface topology simplification," 2002, http://www.multires.caltech.edu/pubs/.

[5] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of SIGGRAPH 95*, Aug. 1995, pp. 351–358.

[6] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proceedings of SIGGRAPH 99*, Aug. 1999, pp. 317–324.

[7] Y. Ohtake, A. Belyaev, and I. Bogaeski, "Polyhedral Surface Smoothing with Simultaneous Mesh Regularization," in *Geometric Modeling and Processing*, 2000, pp. 229–237.

[8] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Anisotropic feature-preserving denoising of height fields and bivariate data," in *Graphics Interface*, 2000, pp. 145–152.

[9] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic geometric diffusion in surface processing," in *IEEE Visualization 2000*, Oct. 2000, pp. 397–405.

[10] H. Zhang and E. L. Fiume, "Mesh Smoothing with Shape or Feature Preservation," in *Advances in modeling, animation, and rendering, , editors*, J. Vince and R. Earnshaw, Eds., 2002, pp. 167–182.

[11] C. Bajaj and G. Xu, "Anisotropic diffusion on surfaces and functions on surfaces," *ACM Trans. Gr.*, vol. 22, no. 1, pp. 4–32, 2003.

[12] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE PAMI*, vol. 12, no. 7, pp. 629–639, 1990.

[13] G. Taubin, "Linear anisotropic mesh filtering, Tech. Rep. IBM Research Report RC2213, 2001. [Online]. Available: www.research.ibm.com/people/t/taubin/pdfs/Taubin-ibm22213.pdf

[14] A. Belyaev and Y. Ohtake, "Nonlinear Diffusion of Normals for Crease Enhancement," in *Vision Geometry X, SPIE Annual Meeting*, 2001, pp. 42–47.

[15] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normal," in *Vision, modeling and visualization*, 2002, pp. 203–210.

[16] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric surface smoothing via anisotropic diffusion of normals," in *Proceedings, IEEE Visualization 2002*, 2002, pp. 125–132.

[17] J. Peng, V. Strela, and D. Zorin, "A simple algorithm for surface denoising," in *Proceedings of IEEE Visualization 2001*, 2001, pp. 107–112. [Online]. Available: www.mrl.nyu.edu/ zorin

[18] M. Pauly and M. Gross, "Spectral processing of point-sampled geometry," in *Proceedings of ACM SIGGRAPH 2001*, Aug. 2001, pp. 379–386.

[19] M. Alexa, "Wiener Filtering of Meshes," in *Proceedings of Shape Modeling International*, 2002, pp. 51–57.

[20] P. J. Huber, *Robust Statistics*. John Wiley and Sons, 1981.

[21] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. John Wiley and Sons, 1986, iSBN 0471-63238-4.

[22] M. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 421–432, 1998.

[23] S. M. Smith and J. M. Brady, "SUSAN - a new approach to low level image processing," *IJCV*, vol. 23, pp. 45–78, 1997.

[24] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. on Computer Vision*, 1998, pp. 836–846.

[25] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Gr.*, vol. 21, no. 3, pp. 257–266, 2002.

[26] D. Barash, "Bilateral Filtering and Anisotropic Diffusion: Towards a Unified Viewpoint Danny Barash," in *Scale-Space 2001*, 2001, pp. 273–280.

[27] M. Elad, "On the bilateral filter and ways to improve it," *IEEE Trans. on Image Processing*, vol. 11, no. 10, pp. 1141–1151, Oct. 2002.

[28] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral Mesh Denoising," *ACM Trans. Gr. (Proceedings of ACM SIGGRAPH)*, pp. 950–953, 2003.

[29] D. A. Murio, *The mollification method and the numerical solution of ill-posed problems*. Wiley, 1993.

[30] S. Osher and R. P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, NY, 2002.

[31] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," in *Proceedings of SIGGRAPH 2000*, ser. Computer Graphics Proceedings, Annual Conference Series. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, July 2000, pp. 271–278, iSBN 1-58113-208-5.

[32] M. Zwicker, M. Pauly, O. Knoll, and M. Gross, "Pointshop 3D: An interactive system for point-based surface editing," *ACM Trans. Gr.*, vol. 21, no. 3, pp. 322–329, July 2002.

[33] D. Levin, "Mesh-independent surface interpolation," in *Advances in Computational Mathematics*, 2001.