Smoothing by Example: Mesh Denoising by Averaging with Similarity-based Weights

Shin Yoshizawa Alexander Belyaev Hans-Peter Seidel

Computer Graphics Group, MPI Informatik, Saarbrücken, Germany
E-mails: {shin.yoshizawa|belyaev|hpseidel}@mpi-inf.mpg.de

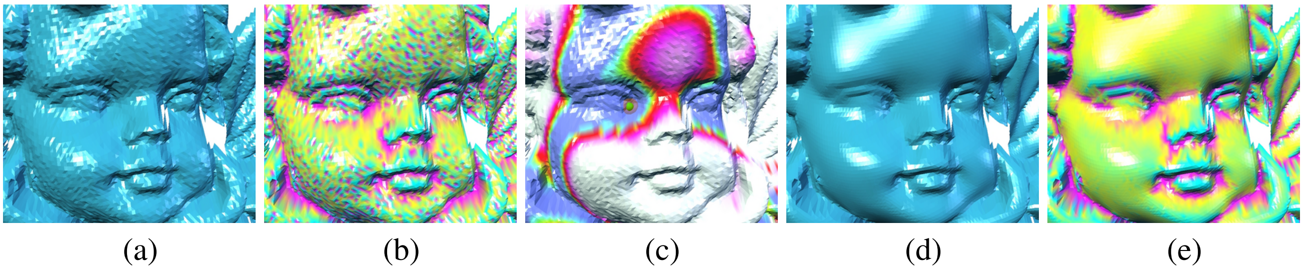

Figure 1: Denoising Angel model with Non-Local means. (a) Original noisy mesh (flat-shading is used). (b) Original noisy mesh colored by mean curvature; the curvature map helps us to identify surface defects and roughness. (c) Coloring by similarity. A mesh vertex is chosen at the left corner of the right eye of the original mesh. The mesh is colored according to a similarity with the shape of the model at the chosen vertex. The similarity increases from white to blue. The vertices with higher similarity values have a stronger contribution to the new (smoothed) position of the chosen vertex. (d) Mesh is smoothed by the similarity-based method developed in this paper (flat-shading is used). (e) Smoothed mesh colored by mean curvature; the curvature map indicates high quality of the smoothed surface.

Abstract

In this paper, we propose a new and powerful shape denoising technique for processing surfaces approximated by triangle meshes and soups. Our approach is inspired by recent non-local image denoising schemes and naturally extends bilateral mesh smoothing methods. The main idea behind the approach is very simple. A new position of vertex P of a noisy mesh is obtained as a weighted mean of mesh vertices Q with nonlinear weights reflecting a similarity between local neighborhoods of P and Q. We demonstrate that our technique outperforms recent state-of-the-art smoothing methods. We also suggest a new scheme for comparing different mesh/soup denoising methods.

1 Introduction

Real-world signals do not exist without noise. Denoising digital images and their 3D geometry counterparts, polygonal meshes and point clouds, remains to be an active area of research. In geometric modeling, recent advances in developing feature preserving smoothing techniques include diffusion-driven methods [23, 13], projection-based approaches [8, 20], and the so-called bilateral mesh filtering schemes [9, 14]. The latter were inspired by image processing techniques based on spatial-tonal normalized convolutions [22, 24] which in their turn can be considered as generalizations of the Yaroslavsky neighborhood filter [25].

Very recently, the so-called *Non-Local means* (or *NL-means*) concept, a natural and elegant extension of the image bilateral filtering paradigm was proposed by Buades, Coll, and Morel [5, 6, 4]. The NL-means method was inspired by the famous Texture-Synthesis-by-Example approach of Efros and Leung [7]. The method and its applications and extensions are currently a subject of intensive research in image and video processing [15, 17]. The basic idea behind NL-means is very simple: for a given pixel, its new (denoised) intensity value is computed as a weighted average of the other image pixels with weights reflecting the similarity between local neighborhoods of the pixel being processed and the other pixels. A similar idea was independently proposed in [2] where it was used for video enhancement purposes.

In this paper, we extend the NL-means concept to mesh denoising and develop a new mesh smoothing method which has a number of important advantages over the main state-of-the-art mesh denoising techniques. Fig. 1 demonstrated the idea and potential of our NL-means mesh smoothing method.

Recently semi-local similarity-based shape descriptors received a considerable attention in connection with shape matching, retrieval, and modeling applications [3, 10, 11, 21, 27] which are too expensive for practical mesh smooth-

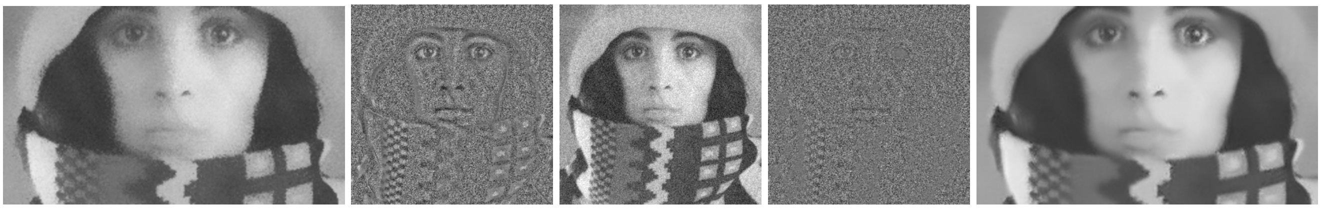

Figure 2: Center: ”Trui” image corrupted by noise. Left: smoothed by bilateral filtering; the difference between the original noisy and smoothed images contains important image structures. Right: smoothed by NL-means filter of Buades, Coll, and Morel; the difference between the noisy and NL-smoothed images contains much less features of the original image. Thus the NL-means filter does a much better denoising job than the bilateral filter.

ing. The approach we use in this paper is much simpler.

The rest of the paper is organized as follows. In Section 2, we briefly overview the Non-Local means approach of Buades, Coll, and Morel. We also explain their very simple and extremely useful SNR (Signal-to-Noise Ratio) analysis of the image smoothing process. In Section 3, we present our similarity-based method for mesh denoising. In Section 4, we compare our method with two state-of-the-art mesh denosing techniques and discuss advantages and limitations of the method. For the comparison, we extend the SNR analysis of Buades et al. from images to meshes. We conclude in Section 5.

2 Image Filtering with NL-means.

Consider a gray-scale image $I(\mathbf{x})$ defined over a bounded domain Ω (which is usually a rectangle). The NL-means filter is defined by

$$J(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{\Omega} w(\mathbf{x}, \mathbf{y}) I(\mathbf{y}) \, d\mathbf{y}, \qquad (1)$$

where the convolution kernel $w(\mathbf{x}, \mathbf{y})$ is given by

$$\exp\left\{ -\frac{1}{h^2} \int G_a(|\mathbf{t}|) \left| I(\mathbf{x}-\mathbf{t}) - I(\mathbf{y}-\mathbf{t}) \right|^2 d\mathbf{t} \right\}, \quad (2)$$

and measures a similarity between neighborhoods of pixels $\mathbf{x}$ and $\mathbf{y}$, $C(\mathbf{x}) = \int_{\Omega} w(\mathbf{x}, \mathbf{y}) \, d\mathbf{y}$ is a normalizing factor, and $G_a(\cdot)$ is a Gaussian kernel of standard deviation a. Here h and a are filtering parameters.

In practice, integration in (2) is performed over a a fixed-size small window. The typical window size varies from 5×5 to 9×9.

A pictorial explanation of the NL-means method is given in Fig. 3.

While the NL-means method is slow, it substantially outperforms the bilateral scheme and other similar filters. The advantages of the NL-means method are especially manifested by processing images with complex texture patterns. We compare the NL-means and bilateral filters in Fig. 2.

Buades, Coll, and Morel also suggested a simple and convenient technique for evaluating the quality of image smoothing methods [5, 6]. The idea is to consider and visualize the difference between the original noisy image $I(\mathbf{x})$ and its smoothed version $J(\mathbf{x})$. If the difference $I(\mathbf{x}) - J(\mathbf{x})$ does not contain geometric structures of the original image $I(\mathbf{x})$ and looks like a random signal, one can conclude that the tested smoothing method removes noise and do not destroy image features. (Of course, similar SNR-based techniques are widely used in image processing, see, for example, [12, 19] and references therein.) In Fig. 2, we apply the Buades et al. image difference technique to demonstrate that the NL-means filter substantially outperforms bilateral filtering in preserving salient image structures.

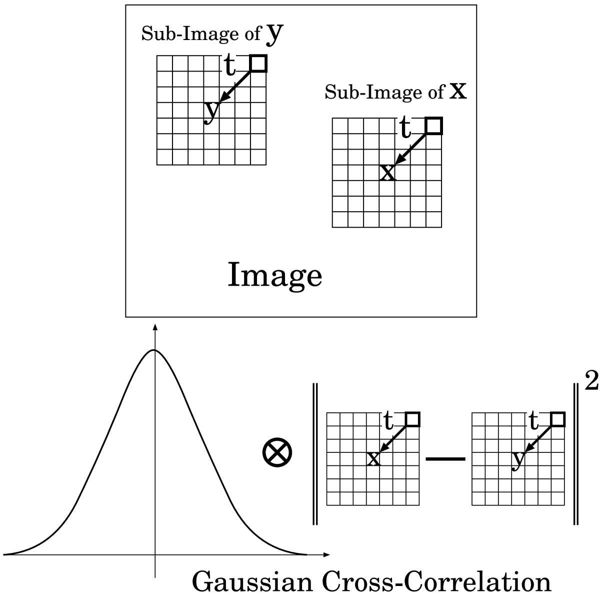

Figure 3: We measure similarity $w(\mathbf{x}, \mathbf{y})$ between two image windows centered at $\mathbf{x}$ and $\mathbf{y}$ by convolving the squared difference between the windows with a Gaussian kernel.

3 Mesh Filtering with NL-means

Given a triangle mesh $\mathcal{M}$, consider a mesh vertex $\mathbf{x}$ and denote by $\Omega_\sigma(\mathbf{x})$ the 2σ-neighborhood of $\mathbf{x}$ on $\mathcal{M}$: $\Omega_\sigma(\mathbf{x}) = \{\mathbf{y} \in \mathcal{M} \,:\, |\mathbf{x}-\mathbf{y}| \le 2\sigma\}$. We use bilateral mesh smoothing flow of [9] as a basis of our method and denoise $\mathcal{M}$ by updating repeatedly the position of each mesh vertex $\mathbf{x}$:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + k(\mathbf{x}^n)\mathbf{n}_\mathbf{x}^n, \tag{3}$$

where $\mathbf{n}_\mathbf{x}$ is the unit normal at $\mathbf{x}$,

$$k(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int\limits_{\Omega_{\sigma_2}(\mathbf{x})} w(\mathbf{x},\mathbf{y}) I(\mathbf{y})\, dS_\mathbf{y}, \tag{4}$$

$$C(\mathbf{x}) = \int\limits_{\Omega_{\sigma_2}(\mathbf{x})} w(\mathbf{x},\mathbf{y})\, dS_\mathbf{y}, \tag{5}$$

$$I(\mathbf{y}) = \langle \mathbf{n}_\mathbf{x}, \mathbf{y}-\mathbf{x}\rangle, \tag{6}$$

$$w(\mathbf{x},\mathbf{y}) = \exp\left\{-\, D(\mathbf{x},\mathbf{y})/\left(2\sigma_1^2\right)\right\}. \tag{7}$$

Here $S_\mathbf{y}$ stands for the area element of $\mathcal{M}$ at $\mathbf{y}$, $\langle \mathbf{a}, \mathbf{b}\rangle$ denotes the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$, and $D(\mathbf{x},\mathbf{y})$ is a similarity kernel.

Similarity Kernel. The main difficulty of extending the NL-means approach to meshes consists of defining an appropriate similarity kernel $D(\mathbf{x},\mathbf{y})$.

Consider mesh vertices $\mathbf{w} \in \Omega_{\sigma_3}(\mathbf{x})$, $\mathbf{z} \in \Omega_{\sigma_3}(\mathbf{y})$, and $\mathbf{y} \in \Omega_{\sigma_2}(\mathbf{x})$ as shown in the left-top image of Fig. 4.

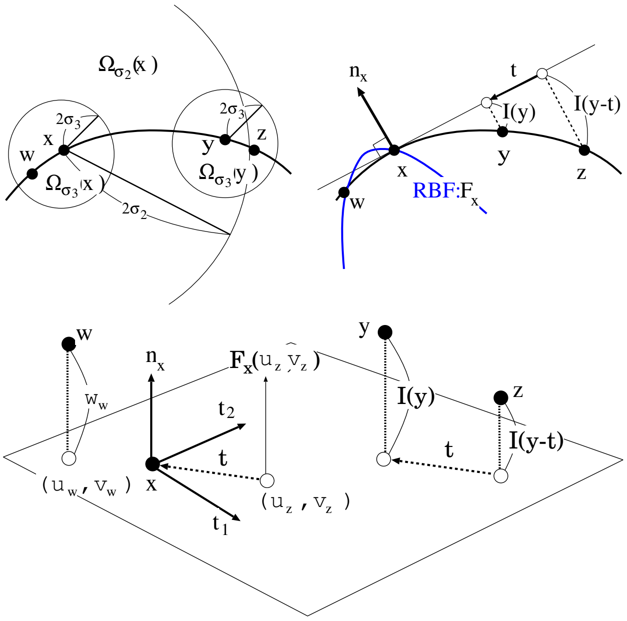

Figure 4: Neighbor and Local Coordinates for RBF.

First we choose a pair of unit tangent vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ in the tangent plane of each mesh vertex $\mathbf{x}$ (the tangent plane at mesh vertex $\mathbf{x}$ is the plane passing through $\mathbf{x}$ and orthogonal to mesh normal $\mathbf{n}_\mathbf{x}$). Let us define a translation vector $\mathbf{t}$, a mesh counterpart of the image translation vector $\mathbf{t}$ in (2), by

$$\mathbf{t} = -\left(u_\mathbf{z}, v_\mathbf{z}\right) = -\left(\langle \mathbf{t}_1, \mathbf{z}-\mathbf{y}\rangle, \langle \mathbf{t}_2, \mathbf{z}-\mathbf{y}\rangle\right).$$

Now let use radial basis functions (RBFs) to build a local approximation of the mesh in a neighborhood of $\mathbf{x}$. Let $(u_\mathbf{w}, v_\mathbf{w}, w_\mathbf{w})$ be the local coordinates of mesh vertex $\mathbf{w}$ w.r.t the basis $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{n}_\mathbf{x})$. The local RBF approximation near $\mathbf{x}$ is given by

$$F_\mathbf{x}(u,v) = p(u,v) + \sum_{\mathbf{w}\in\Omega_{\sigma_3}(\mathbf{x})} \lambda_\mathbf{w} \Phi(\sqrt{u^2+v^2}), \tag{8}$$

where $\Phi(\rho) = \rho^2 \log(\rho)$, $p(u,v)$ is a linear polynomial and RBF coefficients $\{\lambda_\mathbf{w}\}$ are obtained by solving a system of linear equations

$$F_\mathbf{x}(u_\mathbf{w}, v_\mathbf{w}) = w_\mathbf{w}, \qquad \sum_{\mathbf{w}\in\Omega_{\sigma_3}(\mathbf{x})} \lambda_\mathbf{w} p(u_\mathbf{w}, v_\mathbf{w}) = 0.$$

We approximate $I(\mathbf{x}-\mathbf{t})$ corresponding to $I(\mathbf{y}-\mathbf{t})$ by $F_\mathbf{x}(u_\mathbf{z}, v_\mathbf{z})$, as seen in Fig. 4. Finally we define the similarity kernel $D(\mathbf{x},\mathbf{y})$ by

$$D(\mathbf{x},\mathbf{y}) = \int\limits_{\Omega_{\sigma_3}(\mathbf{y})} G_{\sigma_3}(|\mathbf{t}|)\left|F_\mathbf{x}(u_\mathbf{z}, v_\mathbf{z}) - I(\mathbf{y}-\mathbf{t})\right|^2 d\mathbf{t}, \tag{9}$$

where $I(\mathbf{y}-\mathbf{t}) = \langle \mathbf{n}_\mathbf{x}, \mathbf{z}-\mathbf{x}\rangle$ and $G_\sigma(\cdot)$ is a Gaussian kernel.

4 Results and Discussion

In our numerical experiments, we use gcc 3.3.5 C++ compiler on a 1.7GHz Pentium 4 computer with 1GB of RAM. We use the N. Max weights [18] for computing the mesh normals.

Parameters. Four user-specified parameters are used in our method:

1. σ_1, the standard deviation of the similarity kernel (7);
2. σ_2 the size of the integration domain in (4) and (5);
3. σ_3, the size of the similarity domain in (9);
4. n, the number of iterations of (3).

Similar to [14], we make the parameters σs proportional to the average edge length e of the evolving mesh $\mathcal{M} = \mathcal{M}^n$:

$$\sigma_i = \eta_i e, \quad i = 1, 2, 3.$$

Ideally σ_1 represents the noise deviation, therefore similar to [9] it could be chosen as a standard deviation of the heights of vertices $\mathbf{y}$ for either a user-specified flat region or an average standard deviation of an entire mesh. The other two coefficients η_2 and η_3 are constant for the most of models, similar to the image case [6, 4, 15]. According to our experiments, setting $\eta_2 = \{1.0, 2.0\}$ and $\eta_3 = \{0.75, 1.0\}$ leads to good results.

Quality Evaluation and Comparison. We have implemented two recent state-of-the-arts mesh denoising techniques: the Anisotropic Mean Curvature Flow (AMCF) [13] and Bilateral Mesh Filter [9]. In our implementation of AMCF, the weight for ij edge is given by $G_\sigma(k_{ij})h_{ij}$, where h_{ij} is a cotangent-based weight associated with ij and k_{ij} is a directional curvature [16]. In our experiments, for both these methods we try to choose parameter settings producing the best results.

We use two visualization schemes to compare the techniques with our method. The first scheme consists of coloring by the mean curvature. The second scheme measures the difference between the original and smoothed meshes. More precisely, we visualize the differences in the positions of the corresponding vertices of the meshes $|\mathbf{x}_k^{\text{noisy}} - \mathbf{x}_k^{\text{smoothed}}|$.

We use three models in our comparison: a noisy Fandisk model (Fig. 5), a noisy Dragon-head model (Fig. 6), and the Angel model (Fig. 10). For these models, Table 1 presents timing results and parameter settings used for our method and our implementations of methods of [13] and [9].

Fig.	Method	n	η_1	η_2	η_3	Time
5	[13]	3	10	25		1.2s
	[9]	3	0.25	1	1	0.8s
	our	3	0.4	1	0.75	13.2s
9	[13]	1	2×10^4	100		14.7s
	[9]	2	1.5	4	1	126s
	our	3	0.35	1	0.75	606s
10	[13]	1	100	25		1.67s
	[9]	2	0.25	1	0.75	3.7s
	our	2	0.25	1	0.75	64.5s

Table 1: Parameter Setting and Timing results. Here n stands for the number of iterations. For AMCF [13], $\eta_1 e$ and $\eta_2 e$ denotes the step-size (implicit scheme) and the size σ of the Gaussian kernel, respectively. For bilateral filtering [9], the deviation of the hight Gaussian kernel is equal to $\eta_1 e$, the integration domain size is given by $\eta_2 e$, and the deviation of the spatial Gaussian kernel is set equal to $\eta_3 e$. Here e denotes the average edge length of the evolving mesh $\mathcal{M} = \mathcal{M}^n$.

As seen in Fig. 8 our method outperforms its rivals in restoring sharp edges and low-curvature regions. In addition, the max-norm and average errors produced by the method and measured w.r.t. the original clean Fandisk mesh are substantially smaller than those of the Anisotropic Mean Curvature Flow [13] and Bilateral Mesh Filter [9]. Fig. 9 demonstrates that our method delivers the best performance according the entropy of the differences between the original (noisy) and smoothed models. It also indicates that the method preserves fine geometric features better than two its competitors. Fig. 10 shows that our method produces lowest oversmoothing to compare with the two other smoothing techniques.

Finally in Fig. 11 we demonstrate how our method handles triangle soups. Denoising a complex Gargoyle model (about 98 K triangles) by our method is rather slow (five iterations took 31 minutes) but the result is worth seeing.

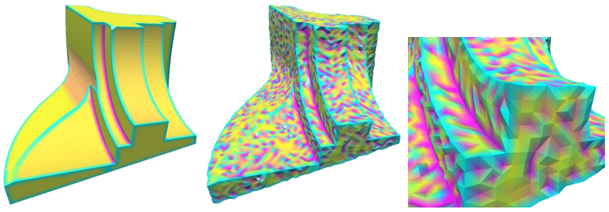

Figure 5: Left: initial Fandisk model colored by mean curvature. Center and Right: noisy Fandisk (Gaussian noise with $\sigma = 0.1e$ is added).

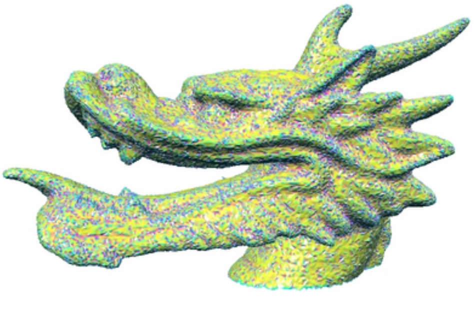

Figure 6: Noisy Dragon-head model (Gaussian noise with $\sigma = 0.2e$ is added) from [14] is colored by mean curvature.

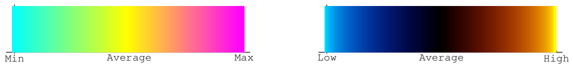

Figure 7: Left: mean curvature profile palette. Right: this palette is used for visualizing the differences in vertex positions of noisy and smoothed meshes.

Complexity. The average computational complexity of our method is given by $O(V_\mathbf{y} V_\mathbf{w} V_\mathbf{z} V + V \log V)$ where V is the number of vertices of $\mathcal{M}$, $V_\mathbf{y}$, $V_\mathbf{w}$ and $V_\mathbf{z}$ are the average numbers of vertices of local patches $\Omega_{\sigma_2}(\mathbf{x})$, $\Omega_{\sigma_3}(\mathbf{x})$ and $\Omega_{\sigma_3}(\mathbf{y})$. Retrieving $2\sigma_2$-neighborhood of $\mathbf{x}$ requires $O(\log V)$ operations by using a kd-tree, and evaluating the

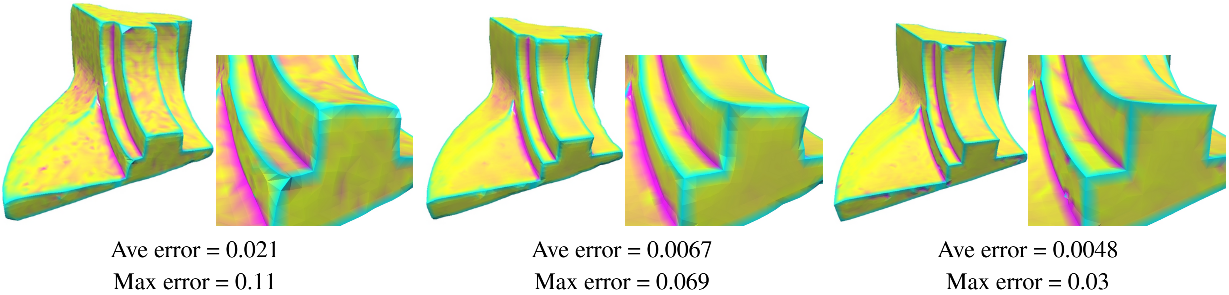

Figure 8: Smoothing noisy Fandisk model ($V = 6474$, $F = 12944$). Mean curvature coloring enhances surface defects and roughness of the smoothed meshes which can not be recognized by human eyes if we use a flat/smooth shading. Left: Anisotropic Mean Curvature Flow [13] is used. Middle: Bilateral Mesh Filter [9] is applied. Right: our method is employed.

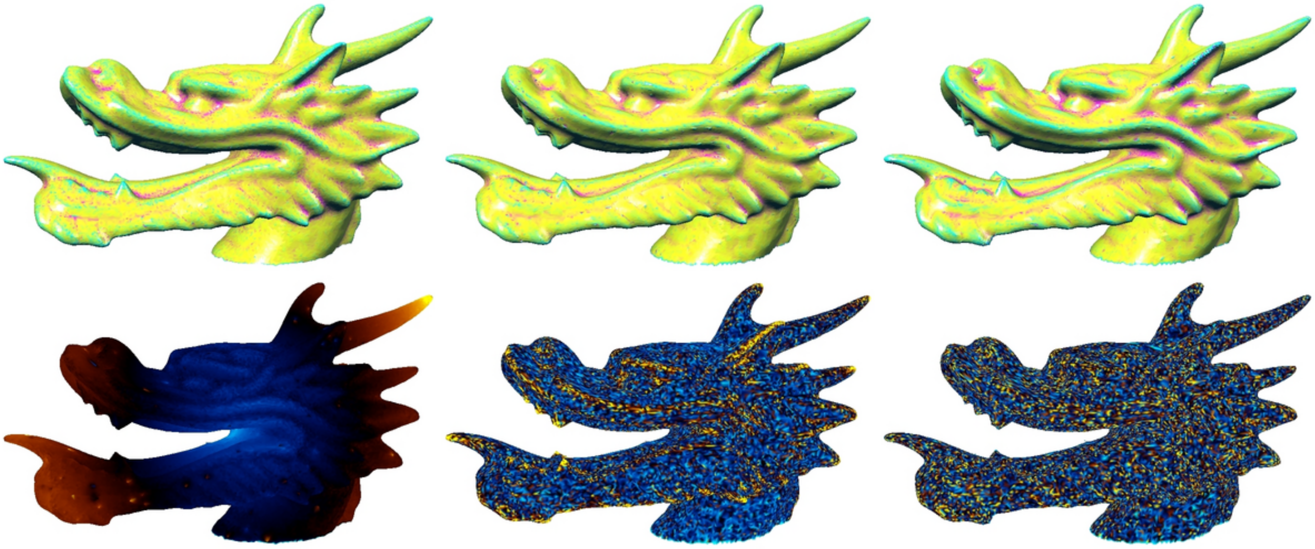

Figure 9: Smoothing noisy Dragon-head model ($V = 100056$, $F = 199924$). Left: Anisotropic Mean Curvature Flow [13] is used. Middle: Bilateral Mesh Filter [9] is applied. Right: our method is employed. Top: coloring by mean curvature indicates that our method outperforms its rivals in preserving fine surface features. Bottom: our method delivers the best performance according the entropy of the difference between the original (noisy) and smoothed models.

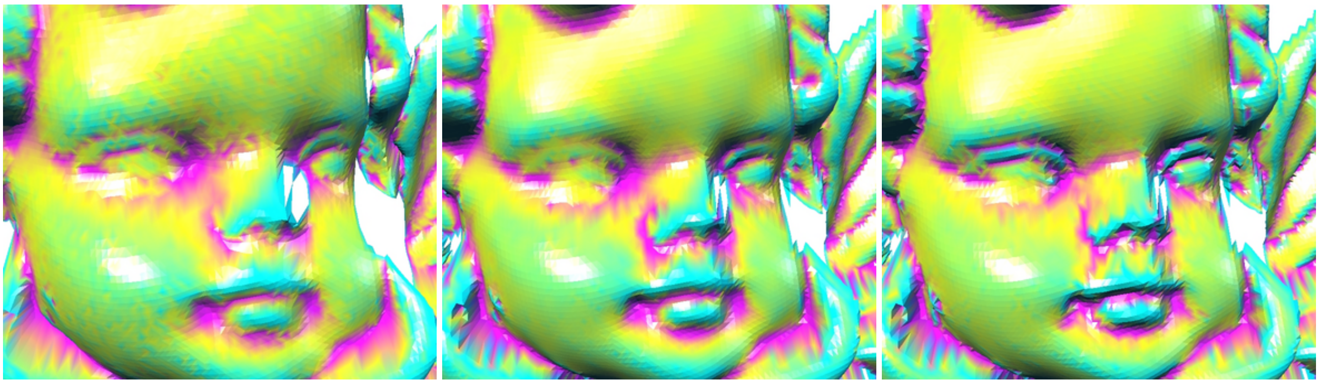

Figure 10: Smoothing noisy Angel model ($V = 24566$, $F = 48090$). Left: Anisotropic Mean Curvature Flow [13] is used. Middle: Bilateral Mesh Filter [9] is applied. Right: our method is employed. Our method produces lowest oversmoothing to compare with two other smoothing techniques.

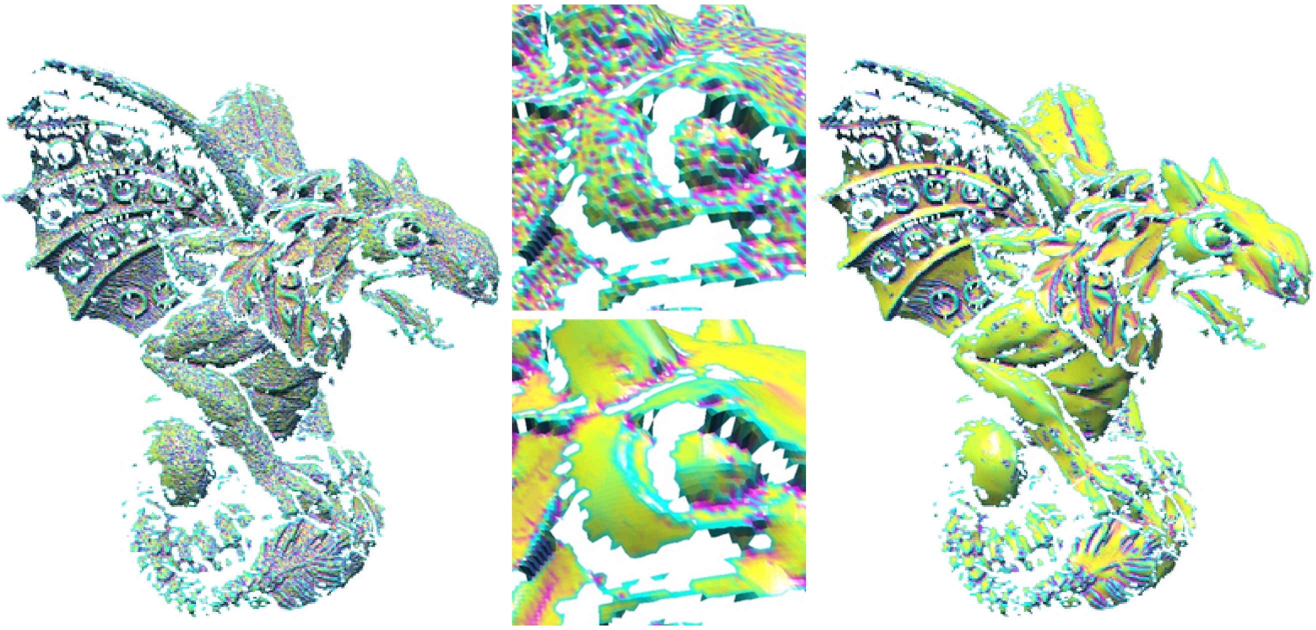

Figure 11: Denoising a complex Gargoyle model ($V = 54907$, $F = 97769$) by our method with $\{\eta_1, \eta_2, \eta_3\} = \{0.28, 2, 1\}$. Left and top-center: original data colored by mean curvature. Right and bottom-center: smoothed data colored by mean curvature; noise is gently removed and fine geometric features are accurately preserved.

similarity kernel (9) is done using $O(V_{\mathbf{w}}V_{\mathbf{z}})$ operations for each pair $\mathbf{x}$ and $\mathbf{y}$.

At the first glance, $O(V_{\mathbf{y}}V_{\mathbf{w}}V_{\mathbf{z}}V + V \ \log \ V)$ looks too large. However $V_{\mathbf{y}}$, $V_{\mathbf{w}}$, $V_{\mathbf{z}}$ are the number of vertices in local neighborhoods of mesh vertices $\mathbf{y}$, $\mathbf{w}$, $\mathbf{z}$ used in our method. For a typical uniformly dense mesh, we have $V_{\mathbf{y}} \approx 20\eta_2$ and $V_{\mathbf{w}} \approx 20\eta_3 \approx V_{\mathbf{z}}$. If η_2 is large, a fast implementation of RBFs [1] should be used.

Although the influence of each parameter σ_1, σ_2 and σ_3 is clear, an optimal selection of all of them is not trivial. Further work is required for a deeper understanding correlations between these parameters.

5 Conclusion

In this paper, we have extended the recent NL-means image filtering approach [5, 6, 4] to the 3D meshes and triangle soups approximating piecewise smooth surfaces. The extension is far from being straightforward, since the original NL-means approach relies heavily on the image structure regularity. We think we have found a simple and elegant solution to the problem by employing local RBF approximations.

We have demonstrated that our method outperforms two recent state-of-the-art smoothing techniques which are among best up-to-date mesh denoising schemes.

Finally we have suggested a new way to compare different mesh/soup denoising methods. We believe that statistical analysis (entropy measurements, etc.) of the difference between the original (noisy) and smoothed datasets will lead to developing new surface denoising techniques and new principles for a fair comparison of existing ones.

The source code of our method is available on the Web for evaluation [26].

Acknowledgements. Models are courtesy of Thouis Jones (Noisy Dragon Head, original from the Stanford University), EU AIM@Shape project (Gargoyle, provided from CNR), University of Washington (Fandisk), Caltech Vision Group (Angel). This work was supported in part by AIM@SHAPE, a Network of Excellence project (506766) within EU's Sixth Framework Programme.

References

[1] Rick K. Beatson and Garry N. Newsam. Fast evaluation of radial basis functions: I. *Computers Math. Applic.*, 24(12):7–19, 1992.

[2] E. P. Bennett and L. McMillan. Video enhancement using per-pixel virtual exposures. *ACM Transactions on Graphics*, 24(3):845–852, 2005. Proceedings of ACM SIGGRAPH 2005.

[3] P. Bhat, S. Ingram, and G. Turk. Geometric texture synthesis by example. In *Second Eurographics Symposium on Geometry Processing*, 2004.

[4] A. Buades, B. Coll, and J. M. Morel. Neighborhood filters and PDE's. *IEEE Transactions on Image Processing*. To appear.

[5] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition (CVPR'05)*, pages 60–65, 2005.

[6] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation (SIAM interdisciplinary journal)*, 4(2):490–530, 2005.

[7] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision (ICCV'99)*, pages 1033–1038, 1999.

[8] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics*, 24(3):544–552, 2005. Proceedings of ACM SIGGRAPH 2005.

[9] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transactions on Graphics*, 22(3):950–953, 2003. Proceedings of ACM SIGGRAPH 2003.

[10] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. In *Under revision for ACM TOG*, 2005.

[11] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka. Curvature maps for local shape comparison. In *Shape Modeling International (SMI'05)*, pages 244–256, 2005.

[12] G. Gilboa, N. Sochen, and Y. Y. Zeevi. Estimation of the optimal variational parameter via SNR analysis. In *Scale-Space 2005, LNCS 3459*, pages 230–241, 2005.

[13] K. Hildebrandt and K. Polthier. Anisotropic filtering of non-linear surface features. *Computer Graphics Forum*, 23(3):391–400, 2004. Proceedings of EUROGRAPHICS 2004.

[14] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics*, 22(3):943–949, 2003. Proceedings of ACM SIGGRAPH 2003.

[15] S. Kindermann, S. Osher, and P. W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling & Simulation (SIAM interdisciplinary journal)*, 4(4):1091–1115, 2005.

[16] C. Lange and K. Polthier. Anisotropic fairing of point sets. *Special Issue of Computer Aided Geometric Design. To appear*, 2005.

[17] M. Mahmoudi and G. Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *Signal Processing Letters*, 12(12):839–842, 2005.

[18] N. Max. Weights for computing vertex normals from facet normals. *Journal of Graphics Tools*, 4(2):1–6, 1999.

[19] P. Mrázek and M. Navara. Selection of optimal stopping time for nonlinear diffusion filtering. *International Journal of Computer Vision*, 52(2/3):189–203, 2003.

[20] Y. Ohtake, A. Belyaev, and M. Alexa. Sparse low-degree implicits with applications to high quality rendering, feature extraction, and smoothing. In *Third Eurographics Symposium on Geometry Processing*, pages 149–158, 2005.

[21] A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. *ACM Transactions on Graphics*, 23(3):878–887, 2004. Proceedings of ACM SIGGRAPH 2004.

[22] S. M. Smith and J. M. Brady. SUSAN - A new approach to low level image processing. *Int. J. Comput. Vision*, 23(1):45–78, 1997.

[23] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface processing via normal maps. *ACM Transactions on Graphics*, 22(4):1012–1033, 2003.

[24] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, pages 839–846, 1998.

[25] L. P. Yaroslavsky. *Digital Picture Processing - An Introduction*. Springer, 1985.

[26] S. Yoshizawa. www.mpi-inf.mpg.de/∼shin/Research/NL/NonLocal.html.

[27] S. Zelinka and M. Garland. Similarity-based surface modelling using geodesic fans. In *Second Eurographics Symposium on Geometry Processing*, pages 209–218, 2004.