Bilateral Normal Filtering for Mesh Denoising

Youyi Zheng, Hongbo Fu, Oscar Kin-Chung Au, and Chiew-Lan Tai

Abstract—Decoupling local geometric features from the spatial location of a mesh is crucial for feature-preserving mesh denoising. This paper focuses on first order features, i.e., facet normals, and presents a simple yet effective anisotropic mesh denoising framework via normal field denoising. Unlike previous denoising methods based on normal filtering, which process normals *defined on the Gauss sphere*, our method considers normals as a surface signal *defined over the original mesh*. This allows the design of a novel bilateral normal filter that depends on both spatial distance and signal distance. Our bilateral filter is a more natural extension of the elegant bilateral filter for image denoising than those used in previous bilateral mesh denoising methods. Besides applying this bilateral normal filter in a local, iterative scheme, as common in most of previous works, we present for the first time a global, noniterative scheme for an isotropic denoising. We show that the former scheme is faster and more effective for denoising extremely noisy meshes while the latter scheme is more robust to irregular surface sampling. We demonstrate that both our feature-preserving schemes generally produce visually and numerically better denoising results than previous methods, especially at challenging regions with sharp features or irregular sampling.

Index Terms—Mesh denoising, bilateral normal filtering, feature preserving, irregular surface sampling.

1 Introduction

Mesh denoising is a vital preprocessing tool for improving imperfect meshes obtained from scanning devices and digitization processes. Although there already exist a variety of mesh denoising methods, research on feature-preserving denoising remains active due to its challenging nature. On the one hand, local geometric features, either low-frequency or high-frequency, should be retained or even recovered during the denoising process. On the other hand, the features, especially those of high-frequency such as sharp edges and corners, are hard to distinguish from noises, which themselves are also often of high-frequency. Generally, feature-preserving denoising is achieved by locally adjusting vertex positions while respecting the underlying features. Anisotropic treatment is often needed to preserve features such as sharp edges and corners.

To preserve local geometric features during denoising, the features must first be identified, either implicitly or explicitly, and decoupled from the spatial location of a mesh defined in a global coordinate system. Although high-order differential properties, such as the shape operator, might contain desirable anisotropic information of features, they are not well defined at regions with sharp features and their computation might become not robust in the presence of noise. Therefore, many existing techniques rely on only first order features, i.e., facet normals. The key idea is to first denoise the normal field and then evolve the surface to match the denoised normals. Unlike high-order differential properties, facet normals are usually well defined anywhere on a surface, especially for triangular meshes. However, previous methods [2], [5], [6], [7] simply work on the normals defined on the Gauss sphere, completely ignoring the parametrization information of the input mesh.

We show that such parametrization is crucial for effective denoising of the normal field especially when the input meshes have highly irregular sampling. Thus we consider the facet normals as a surface signal defined over the original mesh instead of over the Gauss sphere. A new bilateral filter is designed to average the neighboring normals in an anisotropic manner. The averaging weight is determined by two main factors. One measures the signal difference, i.e., normal difference, and the other measures the spatial distance between the neighboring facets where normals are compared. Our carefully designed bilateral weighting guarantees that there is little interinfluence between normals which lie across sharp features or are far away, thus enabling feature-preserving denoising. We also take facet sizes into account in weight design, making our bilateral filter more robust to irregular surface sampling.

Due to the different levels of noise, locally applying a single step of our bilateral filter to the normal field often does not lead to satisfactory denoising results. Like most previous mesh denoising methods, we design a *local, iterative* scheme and allow users to control the degree of denoising by adjusting the number of iterations needed. At each iteration, similar to traditional convolution filters, our bilateral normal filter is independently applied to individual normals to obtain a new normal field as input to the next iteration.

Apart from our local, iterative scheme, we also investigate a *global, noniterative* denoising scheme. Unlike previous global smoothing techniques [8], [9], [10], which are all isotropic, to the best of our knowledge, ours is the first global denoising scheme which is anisotropic and thus able to preserve sharp features. The bilateral updating is formulated as a global optimization system which consists

- *Y. Zheng and C.-L. Tai are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kownloon, Hong Kong. E-mail: {youyi, taicl}@cse.ust.hk.*
- *H. Fu and O.K.-C. Au are with the School of Creative Media, The City University of Hong Kong, Tat Chee Avenue, Kownloon, Hong Kong. E-mail: {hongbofu, kincau}@cityu.edu.hk.*

Manuscript received 7 June 2009; revised 29 Jan. 2010; accepted 15 Nov. 2010; published online 16 Dec. 2010.
Recommended for acceptance by G. Taubin.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2009-06-0104.
Digital Object Identifier no. 10.1109/TVCG.2010.264.

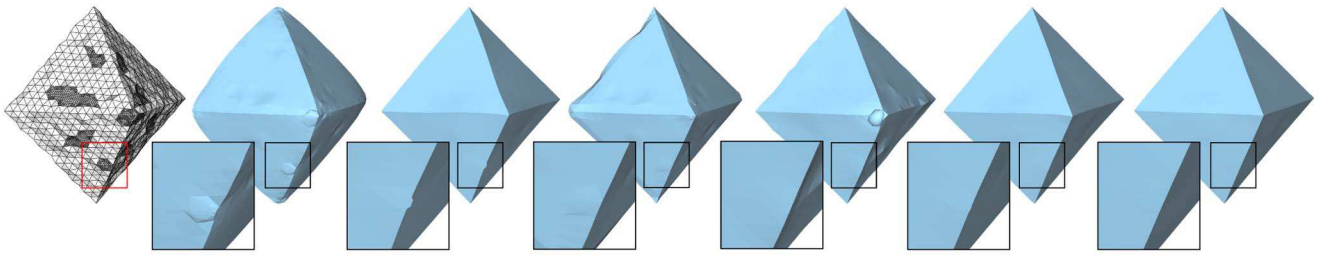


Fig. 1. Our mesh denoising schemes based on bilateral normal filtering produce better results than the state-of-the-art methods at challenging regions with sharp features or irregular surface sampling. From left to right: an input CAD-like model with random subdivision, denoising results with bilateral mesh filtering (vertex-based) [1], unilateral normal filtering [2], probabilistic smoothing [3], prescribed mean curvature flow [4], our local, iterative scheme, and our global, noniterative scheme. All the meshes in the paper are flat-shaded to show faceting.

of two energy terms, a smoothness term and a data term. The user adjusts a parameter for balancing between these two energy terms to control the degree of denoising. Solving this optimization gives a denoised normal field.

We apply our denoising schemes to a variety of meshes corrupted by significant noise, which is either synthetic or arises from imperfect measurement of scanning devices. We demonstrate that our schemes are able to faithfully retain geometric details and recover features corrupted by the noise. We show that our two schemes have their own advantages and disadvantages. The local, iterative scheme is generally faster and more effective for recovering the underlying surface from extremely noisy input meshes. The global, iterative scheme is more robust to irregular surface sampling. We also compare our schemes with the state-of-the-art denoising techniques, demonstrating that both of them consistently produce better results at challenging regions, e.g., flat regions near sharp edges/corners or regions with highly irregular sampling (see comparison examples in Figs. 1 and 2).

In summary, the main contributions of this paper are

- A bilateral filter to process a normal field defined over an input mesh. The filter is insensitive to surface sampling and can be easily extended to filter other types of vector fields defined over a manifold.
- Two practical anisotropic mesh denoising schemes which are efficient, robust, feature-preserving, and simple to implement.

2 Related Work

Our review here focuses only on existing works that are most related to ours. Particularly, we are interested more in anisotropic mesh denoising than isotropic denoising. Please refer to an insightful survey by Botsch et al. [11] on the general mesh smoothing/denoising problem.

Many mesh denoising methods have been extended from image denoising methods. For example, diffusion-based methods [12], [13], [14], [15] extend feature-preserving anisotropic diffusion in image processing to anisotropic geometric diffusion on surfaces. The diffusion process is mainly governed by the principle of heat transfer, expressed by a discretized PDE. Diffusion-based methods preserve or even sharpen geometric features during denoising by introducing anisotropic heat tensors [13], [14], [16], [17], [18]. Such approaches usually require the computation of energy gradients and Hessians, which typically involve complex implementation and are computationally expensive. An interesting fact is that when the diffusion tensor is locally constant, the diffusion process can be reduced to the mean curvature flow on the surface [4], [19], thus greatly simplifying implementation complexity. More specifically, a prescribed anisotropic mean curvature flow is introduced in [4], which can be seen as an effective discretization of Clarenz et al.'s anisotropic geometry diffusion equation [13]. It is also worth mentioning that there is a fundamental relationship between diffusion and bilateral filtering [20].

Recently, Ouafdi et al. [3] proposed a probabilistic smoothing method, which performs an anisotropic average of neighboring vertices weighted by the Riemannian distance according to a well-designed diffusion tensor. Since its computation is highly dependent on high-order differential properties, i.e., the shape operator, this method often causes denoising artifacts near regions with sharp features.

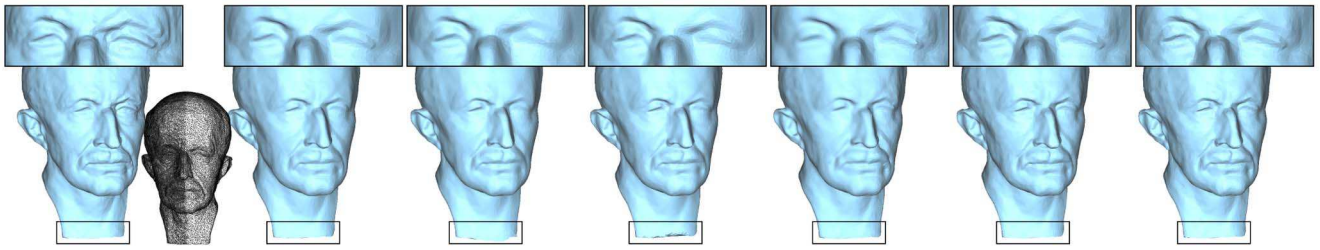


Fig. 2. Our method is able to faithfully recover geometric details corrupted by noise (see the corresponding input noisy model in Fig. 3). From left to right: ground truth, denoising results using bilateral mesh filtering [1], unilateral normal filtering [2], probabilistic smoothing [3], prescribed mean curvature flow [4], our local, iterative scheme, and our global, noniterative scheme. Our schemes are less influenced by irregular surface sampling, thus are consistently better at retaining/recovering features at regions with different sampling, e.g., eyes and the bottom part of the Max-Planck model. Note that the bottom part of the model is not an open boundary and thus is not fixed during denoising.

In view that facet normals are able to better represent local surface geometry than vertex positions, other researchers first filter the normal field and then reconstruct the denoised surface from the filtered normal field. For example, Yagou et al. proposed to use the mean, median [5], and alpha-trimming filters [7]. Simply averaging the neighboring normals isotropically, as done in the mean filter, destroys fine features. Since the sharpness of local features can be roughly measured by the difference of neighboring facet normals, the median filter chooses the normal with the median difference value as the new normal. However, as the new normal is always copied from the original normal field, it always carries the original noise to a certain extent, making the median filter perform poorly for highly noisy meshes. The alpha-trimming filter, as a compromise between mean and median filters, also does not guarantee feature-preserving denoising. Shen and Barner [6] introduce a more effective denoising method using a fuzzy vector median filter which first computes a vector median of the neighboring normals and then averages the neighboring normals weighted by their difference to this median. However, this method is at the cost of high time complexity. Instead of computing a vector median to compare, Sun et al. [2] simply ignore neighboring normals with too large difference to the current normal during averaging, leading to a more efficient feature-preserving denoising method. Sun et al. [21] also present a new method for reconstructing a denoised mesh from a filtered normal field, which we adopt in our vertex updating step. Recently, Sun et al. [21] adopt a random walk model to determine averaging weights. All the above methods process the normals without considering the original mesh parametrization. In other words, they essentially handle the normals defined over the Gauss sphere, making them unreliable to irregular surface sampling.

The bilateral filter [22] has proved to be a very effective edge-preserving filter for image processing. It has been extended to perform feature-preserving mesh denoising by Fleishman et al. [1] and Jones et al. [23]. The key idea behind the bilateral filter is to anisotropically average the signal within a neighborhood, weighted by a monotonously decreasing function in terms of both spatial difference and signal difference. Directly applying the bilateral filter to vertex positions fail to preserve features, since this would mean using the same set of vertex positions to measure both signal difference and spatial difference, reducing the bilateral filter to a unilateral filter. Fleishman et al. apply a bilateral filter to the signed distances of neighboring vertices to the tangent plane at a vertex and displace the vertex along its normal with the computed displacement from the bilateral filter. Jones et al. use the bilateral filter to average the positions obtained by projecting a vertex to the tangent planes determined by its neighboring triangles. Sun et al. [2] showed that Jones et al.'s method has a close connection to the vertex updating step used in denoising methods based on normal filtering. These two bilateral mesh denoising methods are able to effectively remove moderate noise but fail to recover features that are significantly corrupted by noise. A possible reason is that the bilateral filter is independently applied to *locally defined signals at vertices*, that is, individual vertices have different compact-support signals. In contrast, we apply the bilateral filter to a (global) surface signal, i.e., the normal field, defined over the whole mesh.

Most of the above-mentioned methods are local and iterative. In recent years, several global, noniterative mesh smoothing methods have been proposed. For example, Nealen et al. [8] present a global smoothing method by reconstructing the surface from the vanishing vertex Laplacians constrained by all the vertex positions. Instead of simply setting the Laplacians to zero, Su et al. [10] smooth the vertex Laplacians using a mean filter. Nehab et al. [9] introduce a global smoothing technique by taking advantage of the common error characteristics of measured positions and normals. Unlike the local, iterative methods, most of which do not guarantee convergence, all these global methods are numerically more robust. However, all of them are isotropic and easily blur high-frequency features. In contrast, our global, noniterative denoising method is anisotropic and retains all-frequency geometric features during denoising.

3 Bilateral Normal Filtering

In this section, we introduce a new bilateral filter for processing a normal field defined over an input mesh. We focus on triangular meshes and aim to filter the normals defined at every triangle f_i, denoted as $\mathbf{n}_i$.

The original bilateral filter for image denoising has the following form [22]:

$$g'(p) = K(p) \sum_{q \in N(p)} W_c(\|p - q\|)\, W_s(\|g(q) - g(p)\|)\, g(q), \quad (1)$$

where $N(p)$ defines the neighborhood of pixel position p, $K(p)$ is the normalization factor, and $g(p)$ is the signal to be processed at p, i.e., the colors defined over a uniform regular grid in the context of image denoising. The bilateral filter is essentially a weighted averaging filter with the weight consisting of two parts: W_c is a monotonically decreasing function in terms of the distance between p and q, and W_s is a monotonically decreasing function in terms of the signal difference at p and q. Gaussian functions are often used to represent both W_c and W_s in the literature [22].

As pointed out by Jones et al. [23], when extending the bilateral filter to mesh denoising, vertex positions cannot simply be considered as the signal to be processed. Otherwise, we would have $p = g(p)$, reducing the bilateral filter to a unilateral filter. Instead, Jones et al. and Fleishman et al. apply the filter to the signal *locally* defined at each vertex, which is obtained by either projecting a vertex to the first order surface predictors at its neighbors [23] or projecting its neighboring vertices to the predictor at the vertex [1]. Although the bilateral filter itself is robust to outliers, these methods heavily depend on the approximation of the local predictors and signals, which is sensitive to noise. The mollification step introduced by Jones et al. only partially solves this problem, since features might be blurred in this step and can no longer be recovered [23].

We consider a *global* discrete signal defined over an input mesh: $\mathbf{n}(\mathbf{c}_i) = \mathbf{n}_i$, where $\mathbf{c}_i$ is the centroid of triangle f_i. With this configuration, we can directly apply the traditional bilateral filter to the normal field. Specifically, we formulate bilateral normal filtering as follows:

$$\mathbf{n}'(\mathbf{c}_i) = K(\mathbf{c}_i) \sum_{j \in N(i)} \zeta_{ij} W_c(\|\mathbf{c}_i - \mathbf{c}_j\|) W_s(\|\mathbf{n}(\mathbf{c}_i) - \mathbf{n}(\mathbf{c}_j)\|)\ \mathbf{n}(\mathbf{c}_j), \quad (2)$$

or simply

$$\mathbf{n}'_i = K_i \sum_{j \in N(i)} \zeta_{ij}\ W_c(\|\mathbf{c}_i - \mathbf{c}_j\|)\ W_s(\|\mathbf{n}_i - \mathbf{n}_j\|)\ \mathbf{n}_j, \quad (3)$$

where $K_i = K(\mathbf{c}_i) = 1/\sum_{j \in N(i)} \zeta_{ij} W_c(\|\mathbf{c}_i - \mathbf{c}_j\|)\ W_s(\|\mathbf{n}_i - \mathbf{n}_j\|)$ is the normalization factor, $N(i)$ is the one-ring face neighborhood of a face f_i, and ζ_{ij} is the weight to account for the influence from surface sampling rate. Following [2], we tested two types of one-ring face neighborhood. The first type, denoted by $N_I(i)$, is a set of faces that share edges with f_i and the second type, denoted by $N_{II}(i)$, is a set of faces that share common vertices with f_i. The choice of these two types of neighborhood will be discussed in Section 5.

When the normals at two neighboring facets differ significantly, it often implies that these two facets lie on different sides of a sharp edge (e.g., ridge or valley). To avoid blending sharp features, we use the signal weighting term W_s to penalize the signal difference, simply measured by $\|\mathbf{n}_i - \mathbf{n}_j\|$. Specifically, we define

$$W_s(\|\mathbf{n}_i - \mathbf{n}_j\|) = \exp\left(-\|\mathbf{n}_i - \mathbf{n}_j\|^2 / 2\sigma_s^2\right), \quad (4)$$

where σ_s is the standard deviation, with which we adjust the denoising power (see its detailed setting in Section 5). Apart from a Gaussian function, we also tested the truncating weighting function proposed in [2] and found no significantly noticeable difference in terms of denoising results.

It is natural to assume that the influence of two normals should be inversely proportional to the distance between their corresponding facets and gradually vanish with increasing distance. There are several ways to measure the distance between two facets, e.g., geodesic distance along the surface or Euclidian distance. We adopt the euclidian distance for simplicity, i.e., $\|\mathbf{c}_i - \mathbf{c}_j\|$, and define the spatial weighting term as

$$W_c(\|\mathbf{c}_i - \mathbf{c}_j\|) = \exp\left(-\|\mathbf{c}_i - \mathbf{c}_j\|^2 / 2\sigma_c^2\right). \quad (5)$$

To reduce the number of user-specified parameters, our algorithm automatically sets the parameter σ_c. We have tried several ways to determine a desired value for σ_c (e.g., as the average distance of facets in $N(i)$) and found that using the average distance of all adjacent facets in an input mesh generally works the best in our experiments.

Unlike images, which are always defined over uniform regular grids, meshes are largely irregular in terms of both connectivity and sampling. Since the normals defined at larger facets are more likely to be faithful normals of the underlying surface, those facets demand larger averaging weights. This is reflected by our sampling weighting term ζ_{ij}. We have experimented different weighting functions for ζ_{ij} and found $\zeta_{ij} = S_j$ is a good choice, where S_j is the area of facet f_j.

4 Mesh Denoising via Bilateral Normal Filtering

In this section, we present two mesh denoising schemes based on bilateral normal filtering.

4.1 Local and Iterative Scheme

Like most previous denoising methods, our first scheme is local and iterative. It is a *two-stage* iterative scheme. In the first stage, we iteratively update the normal field; in the second stage, we iteratively update the vertex positions which reflect the normals computed in the first stage.

Stage 1: Normal updating. In a noisy mesh, the facet normals are corrupted by the noise, though their computation is well defined. Applying (3) to individual normals decreases the level of noise in the whole mesh. However, it also possibly brings the noise from a normal to its neighbors and vice versa. To some extent it is like an anisotropic diffusion process. Therefore a straightforward but suitable way to increase the degree of denoising is to apply (3) multiple iterations in an explicit manner:

$$\mathbf{n}_i^{t+1} = K_i \sum_{j \in N(i)} \omega_{ij}\ \mathbf{n}_j^t, \quad (6)$$

where $\omega_{ij} = \zeta_{ij} W_c W_s$ is the averaging weight in (3) measured on the input mesh. We normalize the new normals after each iteration. Theoretically, we need to update the weighting terms, ζ_{ij}, W_c, and W_s, with respect to the newly denoised normal field at the current iteration. However, we found that keeping the averaging weight measured on the input mesh lead to no noticeable visual difference, thus reducing computational cost.

Multiple iterations of the normal updating increase the influence of the bilateral filter from a one-ring neighborhood to a wider region, leading to a smoother mesh. Since input meshes often have different levels of noise and automatically estimating the noise level is an ill-posed problem, like other denoising methods, we let the user control the number of iterations to reduce the noise to a desired level.

Stage 2: Vertex updating. After obtaining the denoised normal field, we evolve the mesh to match the new normal field using the iterative vertex updating method proposed by Sun et al. [2]. Our current implementation does not handle vertices on open boundaries and simply leaves them fixed during denoising. Like [2], we usually perform 10 or 20 iterations of vertex updating in our experiments.

We qualitatively evaluated the contributions of individual weighting terms in (3). In Figs. 3 and 4 we show different denoising effects with the local, iterative scheme when some weighting terms are omitted. The comparisons show that the filter with both the spatial weighting term W_c and the signal weighting term W_s enables better recovering of fine details than the filter with W_s only. For example, note the denoising difference in the left eye of the Max-Planck model between the middle and middle right images in Fig. 3 and in the mouth and eye regions in Fig. 4. Fig. 3 also shows that by introducing the sampling weighting term ζ_{ij}, our denoising result is consistent even at regions with significantly different sampling rates, e.g., the highlighted bottom part of the Max-Planck model. The sampling term ζ_{ij} also helps to recover fine details, e.g., the left eye of the Max-Planck model.

4.2 Global and Noniterative Scheme

Instead of applying our local bilateral filter iteratively to mimic a wider filter, an alternative solution is to solve for all the new normals in a single pass by minimizing

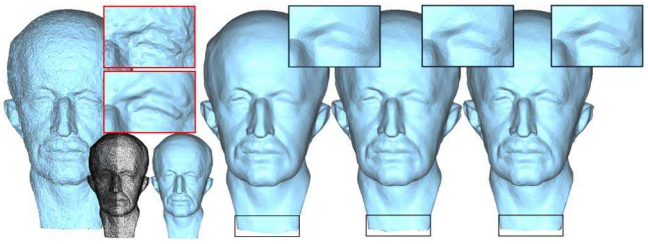


Fig. 3. Contributions of individual weighting terms in (3). The Max-Planck model is resampled (by decreasing the density of the right half of the model) and artificially corrupted by Gaussian noise with standard deviation $\sigma = 0.1$ mean edge length (Left). The denoising comparison results with our local, iterative scheme show that the filter with both the spatial weighting term W_c and the signal weighting term W_s (Middle Right) is able to recover fine details better than the filter with only the signal weighting term W_s (Middle Left). The further introduction of the sampling weighting term ζ_{ij} gives the best result (Right) and makes the result less influenced by irregular surface sampling.

$$E_s = \sum_i A_i \left\| \mathbf{n}'_i - K_i \sum_{j \in N(i)} \omega_{ij} \mathbf{n}'_j \right\|^2, \tag{7}$$

where $\mathbf{n}'_i$ are the unknown normals for the denoised mesh. This can be regarded as an implicit updating compared to the explicit updating performed in (6) (cf. explicit diffusion [24] and implicit diffusion [25]). When updating the normals explicitly using (6), multiplying both sides of the equation by a constant has no effect on the solution. However, when solving the normals in a least-squares sense, which will be described shortly, changing the weight of each equation (i.e., $\mathbf{n}'_i - K_i \sum_{j \in N(i)} \omega_{ij} \mathbf{n}'_j = 0$) does affect the solution. To make the optimization avoid bias toward regions with dense sampling, each equation is weighed by $A_i = S_i / \bar{S}$, where S_i is the area of facet f_i and $\bar{S}$ is the average triangle area over the entire mesh. Note that ω_{ij} alone (i.e., without A_i) cannot address the bias problem, since ω_{ij} is only used to average neighboring normals *inside* each equation and thus cannot balance the relative importance of each equation during optimization.

It can be shown that (7) is closely related to Laplacian optimization over the normal field. By simple derivation, we have

$$E_s = \sum_i A_i \left\| K_i \sum_{j \in N(i)} \omega_{ij} (\mathbf{n}'_j - \mathbf{n}'_i) \right\|^2 = \sum_i A_i \|\mathbf{L}(\mathbf{n}'_i)\|, \tag{8}$$

where $\mathbf{L}$ is the Laplace operator [8] with our bilateral weighting. The so-called Laplacian coordinates, $\mathbf{L}(\mathbf{x})$, have been extensively used to represent local geometric details in the literature of differential-based mesh deformation (see [26] and the references therein). Since minimizing E_s leads to vanishing Laplacian coordinates defined over the normal field, this process completely removes the geometric details, i.e., the high-frequency information, from the normal field. Note that unlike the traditional Laplace operator, where the weighting schemes (e.g., uniform weighting or cotangent weighting) are all isotropic, our bilateral weighting makes E_s result in an anisotropic smoothness term.

The minimization of E_s alone is under-constrained. More importantly, we intend to let the user control the degree of denoising, retaining the signal of the original normal field to some extent. Therefore, similar to the position-based Laplacian optimization framework [8], we introduce a data term by using all the original normals as the soft constraints:

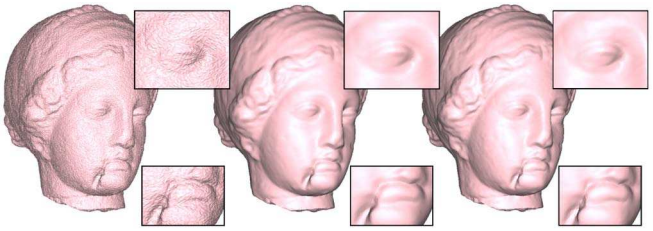


Fig. 4. Left: an noisy input model. The local, iterative scheme, with both spatial weighting term W_c and signal weighting term W_s (Right) gives better denoising results than with only signal weighting term W_s (Middle).

$$E_d = \sum_i A_i \|\mathbf{n}'_i - \mathbf{n}_i\|^2. \tag{9}$$

Our final optimization is formulated as

$$\arg\min_{\{\mathbf{n}'_i\}} (1 - \lambda)\, E_s + \lambda\, E_d, \tag{10}$$

where $\lambda \in [0, 1]$ is a parameter to balance the smoothness and data terms. In other words, the user can adjust λ to control the degree of denoising: smaller values of λ give more power to the smoothness term, thus leading to smoother meshes (see Fig. 6). Note that the above optimization is effectively equivalent to performing one step of implicit updating of $\mathbf{n}_i$ with step size related to λ. The resulting optimization is essentially a linear least-squares optimization problem, which can be efficiently solved using many standard numerical solvers [27], e.g., conjugate gradient. Multigrid algorithms [28] can also be adopted for faster computation. The differences between our optimization and Laplacian mesh optimization proposed by Nealen et al. [8] are as follows: First, our Laplacian optimization is performed over the normal field instead of the vertex positions. Second, we use bilateral weighting rather than uniform/cotangent weighting, making our method faithfully preserve features.

5 Results and Discussion

We have tested our denoising schemes on a variety of models with either raw or synthetic noise. Like previous works, we mostly use models with synthetic noise for quantitative analysis of the effectiveness of our method. Most of our synthetic noise is generated by a zero-mean Gaussian function with standard deviation σ proportional to the mean edge length of the input mesh. Figs. 1, 2, 3, 5, 6 (Bottom), and 11, 12, 13 show some denoising results using our schemes for models with synthetic noise, and Figs. 4, 6 (Top), and 14, 15, 16 for models with raw noise. All the models are flat-shaded to show faceting effect. Observe that both our local and global schemes effectively remove noise while preserving features, e.g., sharp features in CAD-type models (Figs. 1, 6 (Bottom), 11 and 13) or fine details in nonCAD models (Figs. 2, 6 (Top), 14, 15 and 16).

Parameters. Like previous denoising methods, we fine tune the parameters to produce the best results. The parameters used for some denoising examples are listed

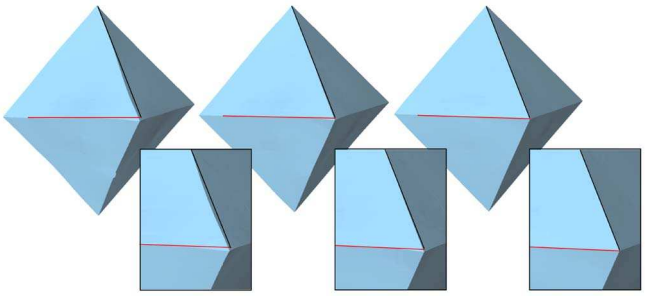


Fig. 5. Our denoising schemes are more robust to irregular surface sampling than previous denoising methods like [2] (Left). Our global, noniterative scheme (Right) outperforms our local, iterative scheme (Left) at regions with highly different sampling, thus better preserving global structures (highlighted by overlaid lines). The noisy input model is shown in Fig. 1.

in Table 1. BMF, UNF, PS, PMC, Scheme I and Scheme II stand for bilateral mesh filtering [1], unilateral normal filtering [2], probabilistic smoothing [3], prescribed mean curvature flow [4], our local, iterative scheme and our global, noniterative scheme, respectively. The parameter sets for these methods are: BMF (vertex iterations); UNF (normal updating iterations, feature detection threshold, vertex updating iterations, neighborhood size NI, or NII); PS (vertex iterations, time step, feature detection threshold); PMC (integration scheme, number of steps, step width, feature detection parameter); Scheme I (normal updating iterations, σ_s, vertex updating iterations, neighborhood size); Scheme II (λ, σ_s, vertex updating iterations, neighborhood size). The column E_v lists the vertex-based errors between the denoised meshes and the corresponding ground-truth models, with the smallest errors among these methods highlighted. All the timings are measured on a notebook with Duo CPU 2.2 GHz and 2 GB RAM. Note that the implementation we use for PMC comes with a simple CG-solver for the implicit flow integration and using direct solvers is expected to bring an enormous speedup.

Among the parameters, the number of normal updating iterations for our local, iterative scheme, λ for our global, noniterative scheme, and σ_s for both schemes influence the degree of denoising the most significantly. Fig. 6 shows that larger values of σ_s or λ lead to smoother results. In our experiments, σ_s typically lies in the range of [0.2-0.6], with higher values for higher level of noise. We found that the first type of face neighborhood N_I works well for nonCAD models and N_{II} is more suitable for CAD-like models. We speculate that more facets involved in N_{II} are able to better characterize sharp features, especially sharp edges. In addition, given the same number of normal updating iterations or the same value of λ, N_{II} generally produces smoother results.

TABLE 1
Parameter Settings and Timings

Objects	Methods	Parameters	E_V ($\times 10^{-3}$)	Time
Pyramid	BMF	(10)	4.949	0.1124s
($\|V\| = 2,948$	UNF	(50, 0.8, 100, NII)	4.154	1.2750s
$\|F\| = 5,892$)	PS	(20, 0.001, 0.85)	4.580	0.7374s
Figure 1	PMC	(implicit, 20,0.01,0.0059)	4.473	24.766s
	Scheme I	(50, 0.23, 100, NII)	4.128	2.0185s
	Scheme II	(0.0001, 0.23, 100, NII)	**3.837**	1.7220s
Fandisk	BMF	(5)	2.559	0.1227s
($\|V\| = 6,475$,	UNF	(10, 0.55, 10, NII)	2.439	0.4806s
$\|F\| = 12,946$)	PS	(5, 0.01, 0.85)	2.500	0.6537s
Gaussian noise	PMC	(explicit, 10,0.01,0.039)	2.449	12.070s
Figure 11	Scheme I	(5, 0.3, 10, NII)	2.364	0.5784s
	Scheme II	(0.07, 0.3, 10, NII)	**2.358**	0.6686s
Fandisk	BMF	(5)	3.642	0.1214s
($\|V\| = 6,475$,	UNF	(20, 0.55, 30, NII)	3.561	0.8804s
$\|F\| = 12,946$)	PS	(8, 0.001, 0.85)	3.168	0.6223s
impulse noise	PMC	(explicit, 15,0.01,0.0122)	3.535	17.100s
Figure 11	Scheme I	(15, 0.35, 20, NII)	**3.136**	1.1483s
	Scheme II	(0.01, 0.4, 20, NII)	3.390	2.4056s
Octa-flower	BMF	(3)	0.727	0.1025s
($\|V\| = 7,919$	UNF	(6, 0.8, 20, NII)	0.651	0.3899s
$\|F\| = 15,834$)	PS	(3, 0.001, 0.7)	0.609	0.8070s
Figure 13	PMC	(explicit, 5,0.01,0.0053)	0.676	8.023s
	Scheme I	(6, 0.25, 20, NII)	0.417	0.6327s
	Scheme II	(0.1, 0.33, 20, NII)	**0.383**	2.9237s
Max Planck	BMF	(3)	0.642	0.3874s
($\|V\| = 30,942$	UNF	(3, 0.3, 20, NII)	0.399	1.2120s
$\|F\| = 61,880$)	PS	(3, 0.01, 0.85)	0.674	1.5291s
Figure 2	PMC	(explicit, 10,0.01,0.009)	0.381	18.400s
	Scheme I	(3, 0.35, 20, NII)	0.352	1.5892s
	Scheme II	(0.15, 0.4, 20, NII)	**0.346**	24.367s
iH-bunny	BMF	(5)	2.441	0.7574s
($\|V\| = 34,834$	UNF	(5, 0.5, 20, NII)	1.809	1.7722s
$\|F\| = 69,451$)	PS	(3, 0.01, 0.85)	1.960	1.4085s
Figure 12	PMC	(explicit, 10,0.01,0.006)	2.103	45.302s
	Scheme I	(4, 0.5, 20, NII)	1.720	2.9048s
	Scheme II	(0.1, 0.5, 20, NII)	**1.707**	28.308s

Comparisons with previous methods. To demonstrate the effectiveness of our denoising schemes, we compare with four exemplary denoising techniques, namely, bilateral mesh filtering (vertex-based) [1], unilateral normal filtering [2], probabilistic smoothing [3], and prescribed mean curvature flow [4]. For all these methods and ours, we chose the parameters that produce visually the best denoising results, as summarized in Table 1. Figs. 1, 2, and 11, 12, 13, 14, 15, 16 show some of comparison examples, with magnified views clearly showing the differences. Both our schemes usually outperform the previous methods at challenging regions

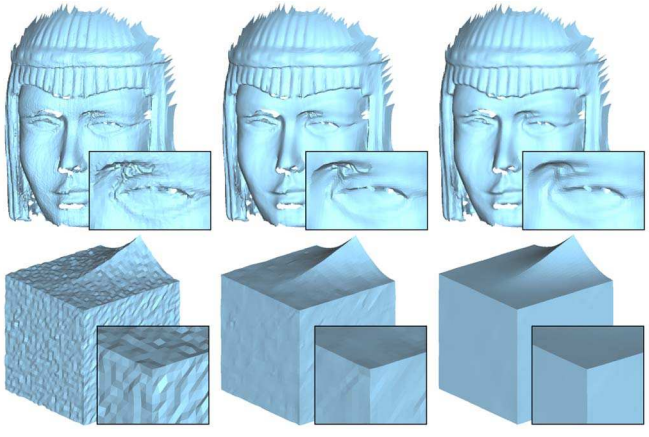


Fig. 6. Top: our local, iterative scheme produces different degrees of denoising results from an noisy input model (Left) with respect to different values of σ_s (Middle: $\sigma_s = 0.3$ and Right: $\sigma_s = 0.9$). Bottom: our global, noniterative scheme generates smoother denoising results with smaller values of λ (Left: input noisy model; Middle: denoising with $\lambda = 0.15$; Right: $\lambda = 0.03$).

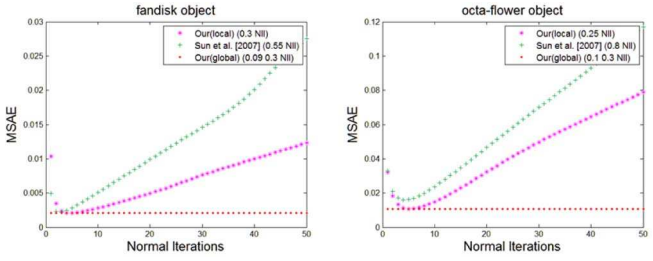


Fig. 7. Normal errors resulting from our schemes and unilateral normal filtering [2]. Our schemes consistently lead to smaller normal errors than unilateral normal filter.

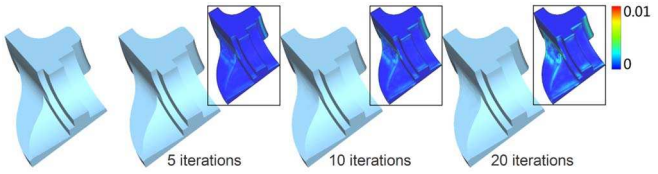


Fig. 8. Our local, iterative scheme ($\sigma_s = 0.2$) is applied to a noise-free model (Left). The colored models show the visualization of errors measured as the difference of individual vertex positions between the processed models and the original model.

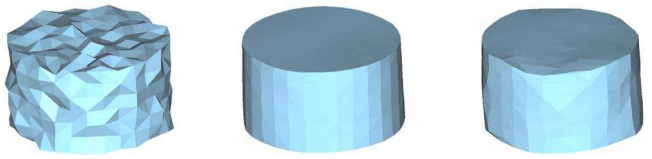


Fig. 9. Denoising a noisy cylinder model (Left). Our local scheme (Middle) better recovers geometric features than the global scheme (Right) when the level of noise is high.

with either irregular surface sampling or sharp features. As seen from these figures, bilateral mesh filtering often blurs sharp edges, e.g., the blurred fandisk model (see per-vertex errors visualized in Fig. 10). Unilateral normal filtering consistently does not perform well at regions with significantly different sampling, as shown in Fig. 5. Since both the probabilistic smoothing method and the mean curvature flow are highly dependent on the approximation of curvature tensors, whose computation is not robust at regions with sharp features or irregular tessellations, these methods perform unreliably near such regions (Figs. 1, 11, and 13).

We have also quantitatively analyzed the differences between our schemes and previous approaches. Specifically,

Fig. 10. The errors between the denoised results (Fig. 11 Top) and the ground truth, measured as the positional difference between corresponding vertices.

we measure the L^2 vertex-based error E_v between the denoised mesh and the ground-truth model, as shown in Table 1. Please refer to [2] for the formulation of E_v ((29) in [2]). Note that although we did not deliberately adjust the parameters toward small values of E_v, both our schemes consistently give lower errors than the compared methods. Since the signal we are processing is a normal field, we also measure the difference between the normal field of the original model (before addition of noise) and the denoised normal field. Following [2], we use the mean square angular error (MSAE) as the error metric and compare the normal errors between our local, iterative scheme and unilateral normal filtering in terms of number of iterations (Fig. 7). For comparison, we also plot the normal errors resulting from our global scheme in the same diagrams, though this scheme is noniterative. It can be seen that the global, noniterative scheme generally leads to smaller normal errors. For iterative methods, including our local, iterative scheme and unilateral normal filtering, the normal errors generally form a V-shaped curve with the smallest error usually around at five iterations in our experiments. Although our local, iterative scheme generally does not converge to a noise-free model, by applying our scheme to a noise-free

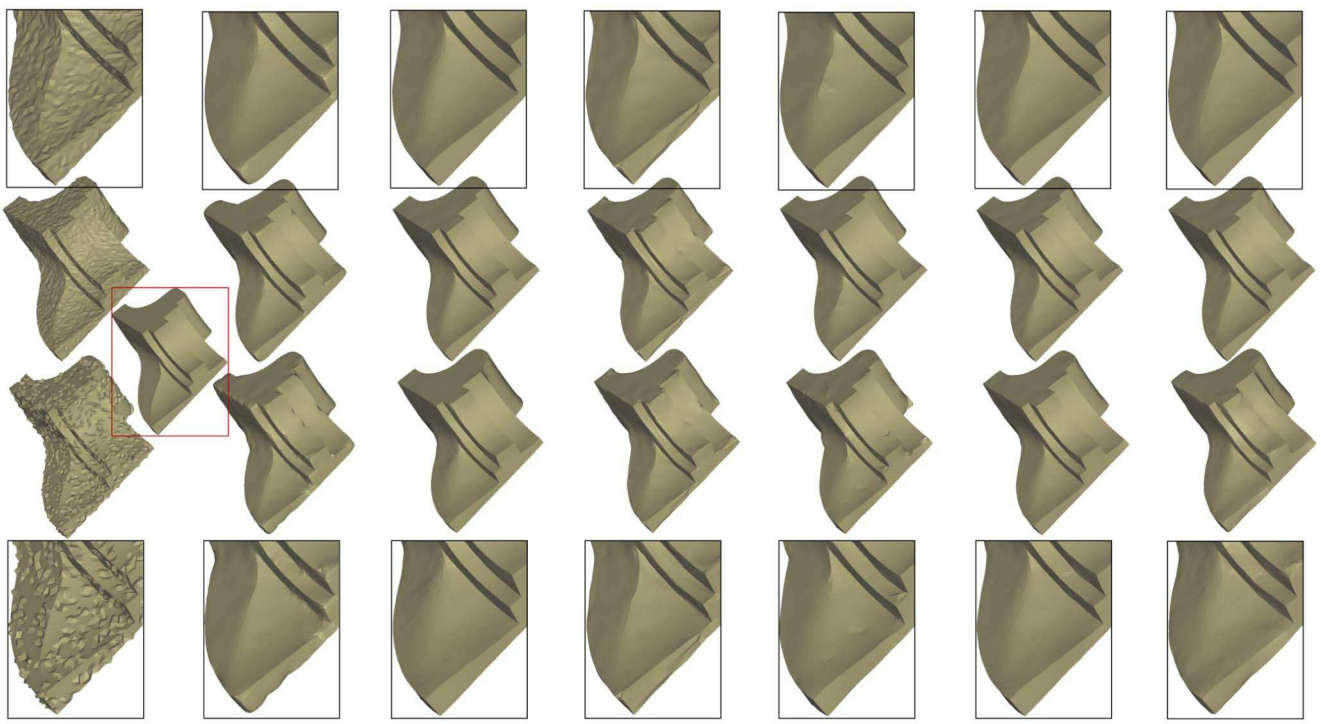


Fig. 11. From left to right columns: input noisy model with ground truth highlighted, denoising results with [1], [2], [3], [4], and our local and global schemes. The top fandisk model is corrupted with 0.1 mean edge length gaussian noise while the bottom one is corrupted with 0.5 mean edge length impulsive random noise.

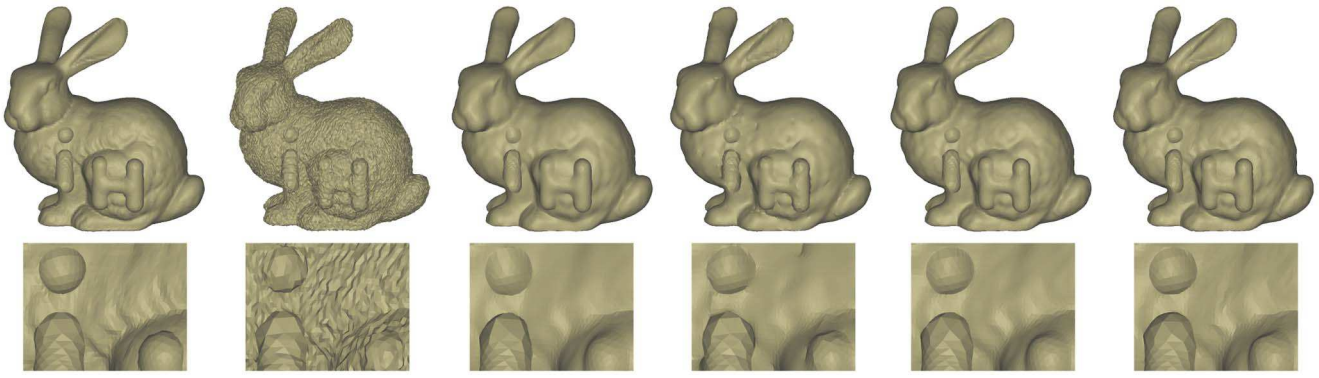


Fig. 12. From left to right columns: ground truth, input noisy model (by introducing 0.2 mean edge length gaussian noise into the ground truth), denoising results with [2], [4], and our local and global schemes.

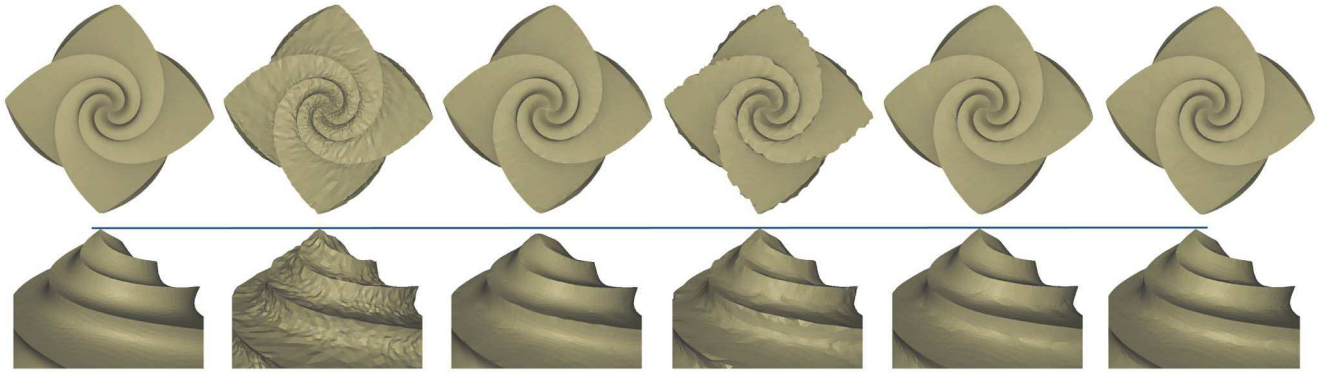


Fig. 13. From left to right columns: ground truth, input noisy model (by introducing 0.1 mean edge length gaussian noise into the ground truth), denoising results with [1], [3], and our local and global schemes.

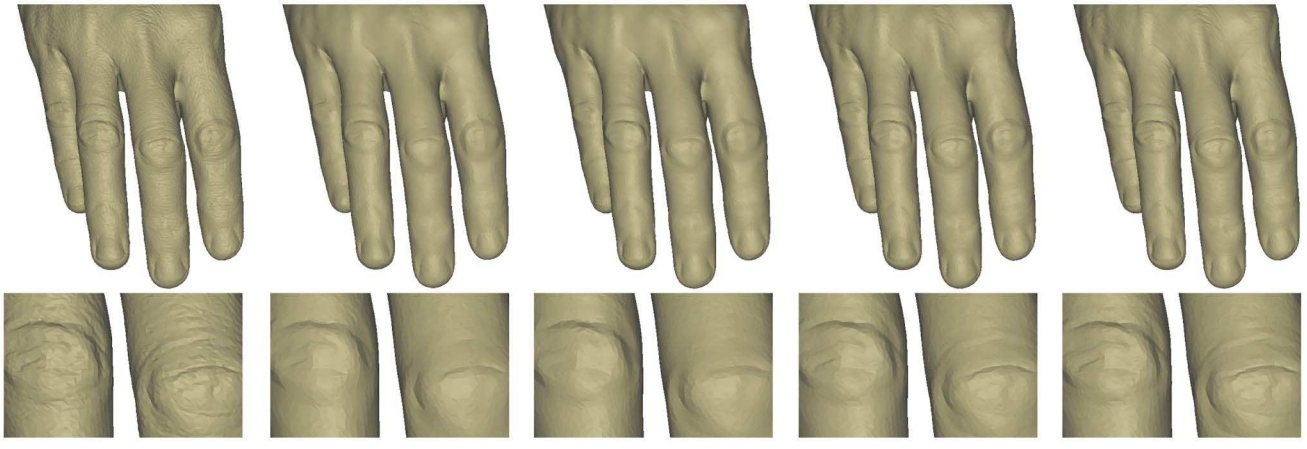


Fig. 14. Denoising results of raw hand. From left to right columns: original noisy model, denoising results with [1], [3], our local, iterative scheme, and our global, noniterative scheme.

model (Fig. 8) we show that geometric features can still be well preserved even after a moderate number of iterations.

Local scheme versus global scheme. Experiments show that our local and global denoising schemes have their own advantages and disadvantages. The global scheme is more robust to irregular surface sampling, as clearly shown by the denoising example of a CAD-like model in Fig. 5. This model has a very different sampling rate at one of its sharp corners, making the local scheme behave slightly differently when recovering the geometry at the highlighted ridge near that corner. In contrast, our global scheme better retains the global structure of the ridges, possibly because it distributes the influence from the irregular sampling to the whole surface through optimization. In addition, we found that the global scheme is able to better preserve fine details, as shown in Figs. 2, 12, 13, 14, 15, and 16. However, the better feature-preservation property of our global scheme may become undesirable when denoising high level of noise. Since features in highly noisy meshes are seriously corrupted, iterative methods are usually better in recovering the features gradually. Fig. 9 shows that our local, iterative scheme recovers the whole shape of the cylinder model much better than the global, noniterative scheme. In terms of computational cost, the local, iterative scheme is faster and

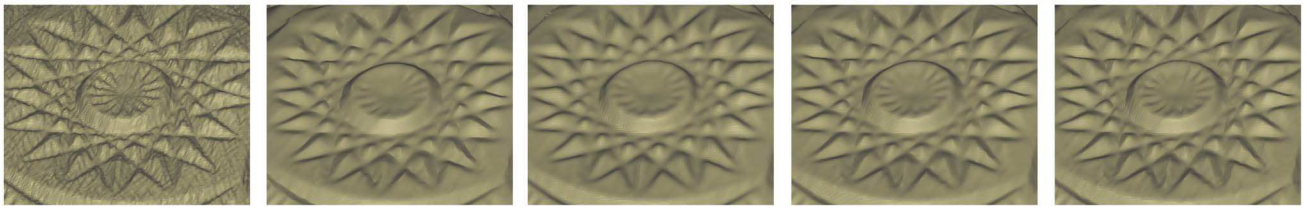


Fig. 15. Denoising results of raw vase. From left to right columns: original noisy model, denoising results with [2], [4], our local, iterative scheme, and our global, noniterative scheme.

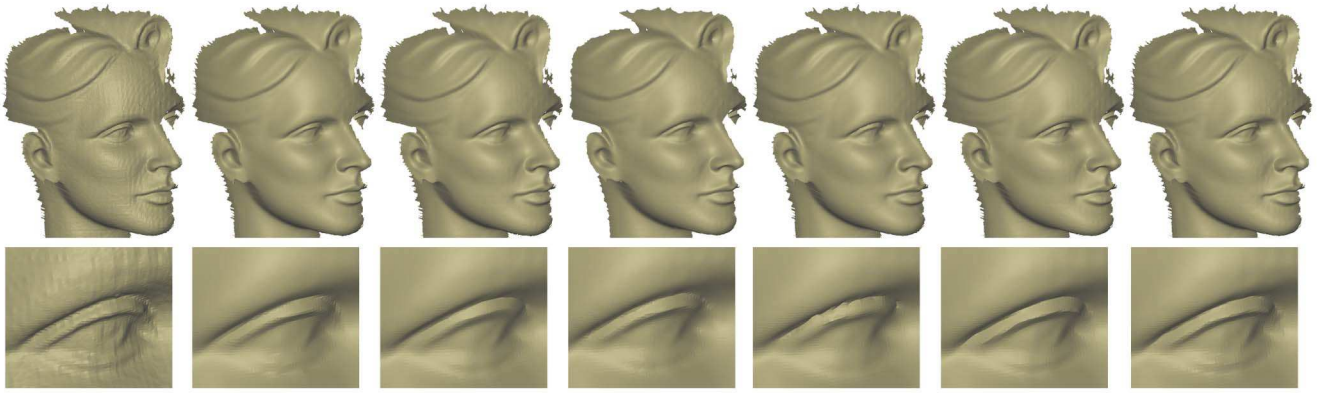


Fig. 16. Denoising results of raw head. From left to right columns: original noisy model, denoising results with [1], [2], [3], [4], our local, iterative scheme, and our global, noniterative scheme.

has lower memory consumption (see Table 1). Similar to previous local, iterative methods, our local scheme is also highly scalable to models of large size.

Limitations. Like most previous denoising methods, our local, iterative method cannot guarantee convergence, as shown in Fig. 7. In other words, more iterations lead to smoother normal fields (i.e., possibly larger normal errors), no matter what level of noise there is essentially in an input model. Therefore, the user needs to choose an appropriate number of iterations (usually around five in our experiments) to achieve the desired results. In contrast, our global, noniterative scheme is unconditionally stable. In addition, theoretically both our schemes cannot guarantee volume preservation, though we did not observe any noticeable shrinking case in practice. Finally, our current implementation simply fixes open boundaries during denoising, which might be undesirable for some scenarios.

6 Conclusion

We have presented two efficient and effective denoising schemes for irregular triangular meshes. Our denoising schemes are based on a novel bilateral normal filter. We regard the facet normals as a surface signal parameterized on an input mesh and formulate the influence of both spatial difference and signal difference into bilateral weighting. Both our local, iterative scheme and global, noniterative schemes are robust to irregular surface sampling and are able to retain/recover geometric details, achieving better results than the state-of-the-art methods. Our bilateral normal filtering can be easily extended to filter other types of vector fields defined over a manifold, e.g., surface normals from photometric stereo or raw normals from scanning devices defined over point clouds.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by grants from the Hong Kong Research Grant Council (Project No. GRF619908 and GRF9041562) and the City University of Hong Kong (Project No. 7002533 and 7200148). Some models used in our experiments are courtesy of [2], [9], and [29].

References

[1] S. Fleishman, I. Drori, and D. Cohen-or, "Bilateral Mesh Denoising," *ACM Trans. Graphics,* vol. 22, pp. 950-953, 2003.

[2] X. Sun, P.L. Rosin, R.R. Martin, and F.C. Langbein, "Fast and Effective Feature-Preserving Mesh Denoising," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 5, pp. 925-938, July 2007.

[3] A.F.E. Ouafdi, D. Ziou, and H. Krim, "A Smart Stochastic Approach for Manifold Smoothing," *Computer Graphics Forum,* vol. 27, no. 5, pp. 1357-1364, 2008.

[4] K. Hildebrandt and K. Polthier, "Anisotropic Filtering of Non-Linear Surface Features," *Computer Graphics Forum,* vol. 23, pp. 391-400, 2004.

[5] H. Yagou, Y. Ohtake, and A. Belyaev, "Mesh Smoothing via Mean and Median Filtering Applied to Face Normals," *Proc. Geometric Modeling and Processing '02,* pp. 124-131, 2002.

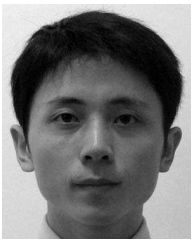
[6] Y. Shen and K.E. Barner, "Fuzzy Vector Median-Based Surface Smoothing," *IEEE Trans. Visualization and Computer Graphics,* vol. 10, no. 3, pp. 252-265, Mar. 2004.

[7] H. Yagou, Y. Ohtake, and A. Belyaev, "Mesh Denoising via Iterative Alpha-Trimming and Nonlinear Diffusion of Normals with Automatic Thresholding," *Proc. Int'l Computer Graphics '03,* pp. 28-33, 2003.

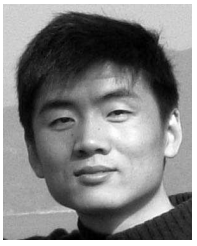
[8] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian Mesh Optimization," *Proc. Conf. ACM GRAPHITE,* pp. 381-389, 2006.

[9] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, "Efficiently Combining Positions and Normals for Precise 3D Geometry," *ACM Trans. Graphics,* vol. 24, pp. 536-543, 2005.

[10] Z.-X. Su, H. Wang, and J.-J. Cao, "Mesh Denoising Based on Differential Coordinates," *Proc. Conf. Int'l Shape Modeling (SMI '09),* 2009.
[11] M. Botsch, M. Pauly, L. Kobbelt, P. Alliez, B. Levy, S. Bischoff, and C. Rössl, "Geometric Modeling Based on Polygonal Meshes," *SIGGRAPH Course Notes,* 2007.
[12] M. Desbrun, M. Meyer, P. Schröder, and A.H. Barr, "Anisotropic Feature-Preserving Denoising of Height Fields and Images," *Proc. Graphics Interface,* pp. 145-152, 2000.
[13] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic Geometric Diffusion in Surface Processing," *Proc. Conf. Visualization (VIS '00),* pp. 397-405, 2000.
[14] R.L. Bajaj and G. Xu, "Anisotropic Diffusion of Subdivision Surfaces and Functions on Surfaces," *ACM Trans. Graphics (TOG),* vol. 22, pp. 4-32, 2003.
[15] A.F. El Ouafdi and D. Ziou, "A Global Physical Method for Manifold Smoothing," *Proc. Conf. Int'l Shape Modeling (SMI '08),* pp. 11-17, 2008.
[16] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric Surface Processing via Normal Maps," *ACM Trans. Graph.,* vol. 22, no. 4, pp. 1012-1033, 2003.
[17] G. Taubin, "Linear Anisotropic Mesh Filtering," *IBM Research Report RC22213(W0110-051),* 2001.
[18] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric Surface Smoothing via Anisotropic Diffusion of Normals," *Proc. Conf. Visualization(VIS '02),* pp. 125-132, 2002.
[19] M. Meyer, M. Desbrun, P. Schröder, and A.H. Barr, *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds,* pp. 35-57. Springer-Verlag, 2002.
[20] D. Barash, "A Fundamental Relationship between Bilateral Filtering, Adaptive Smoothing, and the Nonlinear Diffusion Equation," *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI),* vol. 24, no. 6, pp. 844-847, Aug. 2002.
[21] X. Sun, P.L. Rosin, R.R. Martin, and F.C. Langbein, "Random Walks for Feature-Preserving Mesh Denoising," *Computer Aided Geometric Design,* vol. 25, no. 7, pp. 437-456, 2008.
[22] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. Conf. Int'l Conf. Computer Vision (ICCV '98),* pp. 839-846, 1998.
[23] T.R. Jones, F. Durand, and M. Desbrun, "Non-Iterative, Feature-Preserving Mesh Smoothing," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 943-949, 2003.
[24] G. Taubin, "A Signal Processing Approach to Fair Surface Design," *Proc. Conf. ACM SIGGRAPH,* pp. 351-358, 1995.
[25] M. Desbrun, M. Meyer, P. Schröder, and A.H. Barr, "Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow," *Proc. Conf. ACM SIGGRAPH,* pp. 317-324, 1999.
[26] O. Sorkine, "Differential Representations for Mesh Processing," *Computer Graphics Forum,* vol. 25, no. 4, pp. 789-807, 2006.
[27] M. Botsch, D. Bommes, and L. Kobbelt, *Efficient Linear System Solvers for Mesh Processing,* pp. 62-83. Lecture Notes in Computer Science, Springer, 2005.
[28] L. Shi, Y. Yu, N. Bell, and W.-W. Feng, "A Fast Multigrid Algorithm for Mesh Deformation," *ACM Trans. Graphics,* vol. 25, no. 3, pp. 1108-1117, 2006.
[29] S. Yoshizawa, A. Belyaev, and H.-P. Seidel, "Smoothing by Example: Mesh Denoising by Averaging with Similarity-Based Weights," *Proc. Conf. Int'l Shape Modeling (SMI '06),* pp. 38-44, 2006.



Youyi Zheng received the BSc and MSc degrees in mathematics both from the Zhejiang University. Currently, he is working toward the PhD degree at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include digital geometric processing, computer graphics, and interactive techniques.



Hongbo Fu received the BS degree in information sciences from Peking University, China, in 2002 and the PhD degree in computer science from the Hong Kong University of Science and Technology in 2007. Currently, he is working as an assistant professor in the School of Creative Media, City University of Hong Kong. Before joining City University, he had postdoctoral research trainings at the Imager Lab, University of British Columbia, Canada and the Department of Computer Graphics, Max-Planck-Institut Informatik, Germany. His primary research interests fall in the field of computer graphics with an emphasis on digital geometry processing.

Oscar Kin-Chung Au received the BEng, MPhil, and PhD degrees, all from the Department of Computer Science and Engineering at Hong Kong University of Science and Technology. He is an assistant professor of school of creative media at the City University of Hong Kong. His research interests include geometric modeling and processing, computer graphics, and interactive techniques.

Chiew-Lan Tai received the BSc degree in mathematics from the University of Malaya, the MSc degree in computer and information sciences from the National University of Singapore, and the DSc degree in information science from the University of Tokyo. Currently, she is working as an associate professor of computer science at the Hong Kong University of Science and Technology. Her research interests include geometry modeling and processing, computer graphics, and character animations.