Bi-Normal Filtering for Mesh Denoising

Mingqiang Wei, Jinze Yu, Wai-Man Pang, *Member, IEEE*, Jun Wang, Jing Qin, Ligang Liu, and Pheng-Ann Heng, *Senior Member, IEEE*

Abstract—Most mesh denoising techniques utilize only either the facet normal field or the vertex normal field of a mesh surface. The two normal fields, though contain some redundant geometry information of the same model, can provide additional information that the other field lacks. Thus, considering only one normal field is likely to overlook some geometric features. In this paper, we take advantage of the piecewise consistent property of the two normal fields and propose an effective framework in which they are filtered and integrated using a novel method to guide the denoising process. Our key observation is that, decomposing the inconsistent field at challenging regions into multiple piecewise consistent fields makes the two fields complementary to each other and produces better results. Our approach consists of three steps: *vertex classification*, *bi-normal filtering*, and *vertex position update*. The classification step allows us to filter the two fields on a piecewise smooth surface rather than a surface that is smooth everywhere. Based on the piecewise consistence of the two normal fields, we filtered them using a piecewise smooth region clustering strategy. To benefit from the bi-normal filtering, we design a quadratic optimization algorithm for vertex position update. Experimental results on synthetic and real data show that our algorithm achieves higher quality results than current approaches on surfaces with multifarious geometric features and irregular surface sampling.

Index Terms—Mesh denoising, bi-normal filtering, feature preserving, piecewise consistence, piecewise smooth surface

1 Introduction

Mesh surfaces are widely used in computer graphics and applications such as computer-aided industrial design, interactive virtual reality and medical diagnosis and treatment. Even with high-fidelity scanners, the obtained surfaces still contain noise from various sources [1]. Noise not only degrades visualization quality, but also causes troubles in downstream processing [2]. Though some recent techniques [2], [3], [4], [5] have yielded promising results, denoising models with features at various scales and/or with irregular surface sampling is still a challenging task.

Mesh denoising aims to eliminate noise or spurious information while simultaneously preserving genuine information at all frequencies. Nowadays, the two-stage process is commonly adopted [6], [7], [8], which first adjusts the facet normal field and then updates vertex positions to match the adjusted facet normal field. This is based on the observation that the first-order normal variations can better describe surface variations rather than vertex position variations [9]. For models with challenging regions of features at various scales and/or of irregular surface sampling, performing vertex position update with only the facet normal field, however, would lead to artifacts in denoising results, including feature blurring, shape distortion, and vertex drifts. This is because the geometry information of one single normal field is insufficient in guiding the optimization of vertex position, needless to say that for most techniques the genuine information will at least be partially lost in the normal filtering stage.

There are two commonly used normal fields of a mesh surface, i.e., the facet normal field and the vertex normal field. These two normal fields sometimes contain redundant geometry information of the same mesh surface. This is because if the underlying surface of a mesh is smooth everywhere, the two fields would be consistent with each other. The consistence between the two fields can explain why they can be recovered from each other through simple interpolation and why most of the two-stage methods employing only single normal filtering [3], [8] can still achieve quality results, especially at smooth regions.

However, the two fields also have some differences: the facet normal field tends to reflect the global geometric variations of a mesh surface, while the vertex normal field more illustrates local details of mesh vertices. Thus if the underlying surface is piecewise smooth, rather than smooth everywhere, the two fields are inconsistent at feature regions (also called discontinuous points or outliers in some papers), where one normal field cannot be directly estimated from the other with simple interpolation. Therefore, depending on the geometry property of a feature, each normal field may contain distinct geometry information at a region. This motivates us to investigate the possibility of employing both normal fields in the denoising process to complement each other at regions with features and/or with irregular surface sampling.

- *M. Wei and P.-A. Heng are with the Department of Computer Science and Engineering, the Chinese University of Hong Kong, Hong Kong. E-mail: {mqwei, pheng}@cse.cuhk.edu.hk.*
- *J. Yu is with the School of Computing, National University of Singapore, Singapore. E-mail: jinze.yu@comp.nus.edu.sg.*
- *W.-M. Pang is with the Department of Computer Science, Caritas Institute of Higher Education, Hong Kong. E-mail: wmpang@ieee.org.*
- *J. Wang is with College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu. E-mail: davis.wjun@gmail.com.*
- *J. Qin is with the Human-Computer Interaction Center, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, 1068 Xueyuan Avenue, Shenzhen, Guangdong. E-mail: jing.qin@siat.ac.cn.*
- *L. Liu is with the School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui. E-mail: lgliu@ustc.edu.cn.*

Manuscript received 8 Sept. 2013; revised 5 May 2014; accepted 10 May 2014. Date of publication 28 May 2014; date of current version 26 Nov. 2014. Recommended for acceptance by S.-M. Hu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2014.2326872

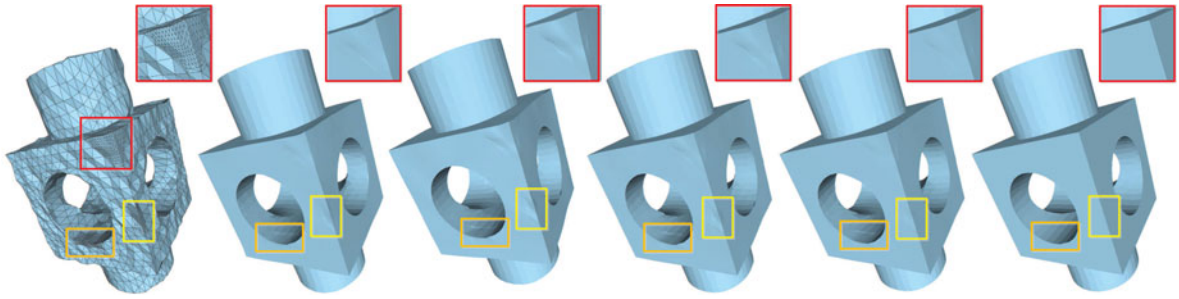


Fig. 1. Denoising results of the Block model. From left to right columns: noisy model which is first locally subdivided at three regions (highlighted) using the middle-point subdivision strategy (two iterations) to produce the irregular sampling and then corrupted with noise (Gaussian noise with $\sigma = 0.15$ of mean edge length), denoising results of Sun et al.'s unilateral normal filtering ($T = 0.38, n_1 = 10, n_2 = 12$) [8], Fan et al.'s second-order bilateral filtering ($k_1 = 35, k_2 = 8, Eps = 2, MinPts = 6$) [10], Wang et al.'s cascaded filtering ($n_1 = 12, \tau = 0.005, n_2 = 14$) [11], He and Schaefer area-based edge filtering ($\mu = \sqrt{2}, \alpha = 0.1\overline{\gamma}, \lambda = 0.02\ell_e^2\overline{\gamma}$) [2], and our bi-normal filtering ($\sigma_{S1} = 0.3, n_1 = 12, \sigma_{S2} = 0.3, n_2 = 12$). Our approach yields a better result than other techniques at challenging regions with features and irregular surface sampling.

However, it is difficult to estimate the vertex and facet normals accurately at these feature regions due to the existence of anisotropic patches at these regions. The situation becomes even worse in the presence of noise. We propose a novel strategy, namely bi-normal filtering, to more accurately compute the two normal fields and employ them to guide the optimization of vertex position for preserving features and dealing with irregular surface sampling.

Before going into the process of bi-normal filtering, we classify vertices of an input noisy mesh into feature vertices and non-feature vertices, which tells us which vertices require special consideration during the filtering process. In the bi-normal filtering, we first initialize the facet normal field. Existing normal filters can produce accurate facet normals at smooth regions, but result in less reliable normals at feature regions. Then we cluster each feature vertex's neighboring facets into multiple piecewise smooth patches by formulating an energy function derived from the initialized facet normal field. Thanks to the neighboring facets clustering, the vertex normals at the feature regions can be estimated by fitting these piecewise smoothed patches. Meanwhile, we further refine the facet normal field by adding the piecewise consistence at feature regions as constraints into a global system [5], which can fully prevent negative effects from other piecewise smooth patches during the refinement. Finally, we formulate the vertex update process as a quadratic optimization problem exploiting both the vertex normal field and the facet normal field, which can more accurately update vertices to approximate the underlying surface of the mesh.

As shown in Fig. 1, the utilization of the two normal fields is paramount in making denoising results visually more appealing. Extensive experiments reported in Section 7 further demonstrate that our algorithm can achieve higher quality results than previous approaches on noisy surfaces with multifarious geometric features and irregular surface sampling. The main contributions of this paper are

- we take advantage of the piecewise consistent property between the two normal fields on mesh surfaces and design a novel bi-normal filtering strategy to address the inherent difficulties in estimating the two normal fields of a noisy mesh at feature regions;
- we formulate the vertex position update as a quadratic optimization problem based on the two normal fields. Since one normal field can provide geometric information that the other field might lack, combining them to guide the optimization of vertex positions leads to higher fidelity of the underlying surface.

2 Related Work

Existing surface noise removal approaches are either isotropic (i.e., surface smoothing) or anisotropic (i.e., surface denoising). Here, we mainly focus on existing techniques which are closely related to our work. An all-around overview is beyond the scope of this paper; readers can refer to [12], [13], [14] and references therein for more details about this topic.

Early methods for noise removal are isotropic, which means the filters applied in these methods are surface geometry independent. Laplacian smoothing [15] is the fastest and simplest scheme among surface smoothing techniques. To solve the shrinkage problem of Laplacian operator, Taubin [6] presented a two-step Laplacian operator to expand the mesh after smoothing. Desbrun et al. [16] extended this approach to irregular meshes by using geometric flow analogy to re-scale the mesh. Meyer et al. [17] further extended this idea to handle anisotropy for feature preserving. Later, Kim and Rossignac [18] refined these three approaches to design a lowpass/highpass filtering framework with exaggeration and attenuation options. Recently, several global, noniterative surface smoothing approaches [19], [20], [21] have been proposed, which are usually based on differential properties. In general, these isotropic approaches are numerically robust, and could generate smoothing results. However, they inevitably wipe away high-frequency features due to the intrinsic characteristic of isotropic filters.

Compared with surface smoothing techniques, surface denoising techniques usually better preserve geometric features while removing noise. Many denoising approaches [22], [23], [24], [25], [26] are inspired from scale space and anisotropic diffusion in image processing. They adopted feature-preserving anisotropic diffusion for 2D grids to anisotropic geometric diffusion on 3D surfaces. Although these approaches can yield high-quality results, they suffer from the numerical instability of the diffusion equations, as they are heavily based on shock formation [27]. Hildebrandt and

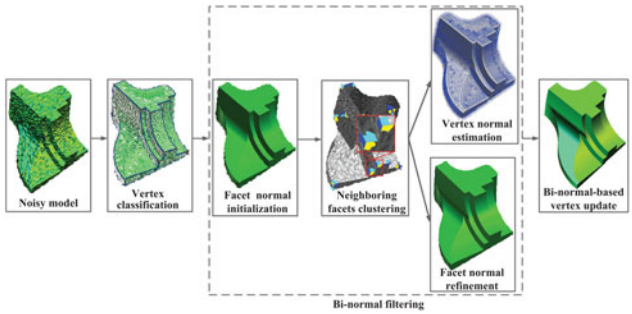


Fig. 2. The framework of the proposed algorithm.

Polthier [12] adopted prescribed mean curvature flow to preserve surface feature when removing noise. He and Schaefer [2] proposed a denoising approach by minimizing the curvature of a surface except at sharp features.

Some denoising techniques are inspired from the bilateral filtering [28] which was initially employed in image denoising. Fleishman et al. [3] and Jones et al. [27] independently extended it to denoise 3D meshes. Fleishman et al. used an iterative one-stage scheme which is relatively fast but cannot always accurately retain features. Jones et al. used a non-iterative two-stage scheme which is slow because it treats normal smoothing and vertex update as global problems. El Ouafdi et al. [29] designed a Riemannian distance based on diffusion tensor to filter neighboring vertices which formed a probabilistic denoising algorithm.

In recent years, many researchers, e.g., Taubin [30], Ohtake et al. [7], Yagou et al. [31], Shen and Barner [32], [13], Sun et al. [8], [4], Zheng et al. [5] and Zhu et al. [33] have adopted the multi-stage framework which first filters facet normals and then updates vertex positions based on facet normals. Compared with one-stage approaches, these multi-stage approaches are generally more effective for recovering features contaminated by reasonable amount of noise. However, these filters cannot eliminate interinfluence of different geometric structures which leads to feature blurring, especially the blurring of shallow features as shown in the result section.

More recently, several researchers, e.g., Fan et al. [10], Bian and Tong [34], and Wang et al. [11], classified vertices on the noisy mesh into different types before denoising process. Wang et al. [35] simultaneously decoupled noises and features on 3D surfaces for denoising. Generally, an accurate estimation of the differential geometric properties is vital for these methods. However, higher order derivatives are sensitive to noise, which usually leads to poor results.

3 Overview

How to produce the two normal fields of higher accuracies with the help from each other for mesh denoising is the main focus of this study. We adopt a cascaded operation on the two normal fields, where methods used are built on the geometry assumption that the underlying surface of a noisy mesh is piecewise smooth and a feature lies on the intersection of multiple smooth surface patches [36]. Fig. 2 illustrates the framework of the proposed algorithm. It is composed of three steps: *vertex classification*, *bi-normal filtering* and *vertex position update* (also called *mesh evolution*). We first employ a vertex classification technique [37] to label mesh vertices as features and non-features (Section 4). In bi-normal filtering, we first obtain a relatively accurate initial facet normal field via a local facet normal filtering [5] to reduce the effects of noise on the following steps (Section 5.1). We then cluster each feature vertex's facet neighbors into distinct piecewise smooth patches (with different colors in Fig. 2) according to similarity of their facet normals (Section 5.2). Next, the vertex normal field is estimated based on the clustering results (Section 5.3). In detail, clusters of different piecewise smooth patches are fitted into different planes to determine normals of feature vertices. Thus, we can obtain a more accurate vertex normal field than traditional methods that indiscriminately average a vertexs 1-ring facet normals for all vertices. Meanwhile, the representative normals of those consistent patches are added as constraints into a global equation system [5] for further optimizing the facet normal field (Section 5.4). Finally, noisy vertices are adjusted to new positions by solving a quadratic optimization based on the two normal fields (Section 6).

4 Vertex Classification

We employ the normal tensor voting algorithm [37] to classify vertices of the input noisy mesh according to the piecewise smooth assumption. In detail, according to the eigenvalues of the voting tensor, vertices can be classified into three types, i.e., corner, sharp edge, and face, in which sharp edges are considered as intersections of two piecewise smooth patches and corners as intersections of three or more piecewise smooth patches. We call vertices of sharp edges and corners as *feature vertices*, and faces as *non-feature vertices*. Due to the existence of noise, the normal tensor voting algorithm may result in misclassifications in some cases [37]. For example, a non-feature vertex may be wrongly classified into sharp edge or corner. Fortunately, this will not influence the performance of our algorithm, as in the following bi-normal filtering, these misclassifications can be eliminated, which will be described in Section 5.2.

5 Bi-Normal Filtering

If the underlying surface of a mesh is smooth everywhere, the two normal fields are consistent with each other. To be specific, one of the normal fields can be directly recovered from the other through weighted interpolation. However, if the underlying surface is piecewise smooth, the two fields are not consistent in the neighborhood around a feature, where multiple smooth surface regions exist. In order to faithfully recover features during denoising, we propose a bi-normal filtering approach to estimate both the facet and vertex normal fields in turns with a careful handling of the inconsistence at feature regions.

5.1 Facet Normal Field Initialization

In order to reduce the effects of noise in the following steps, we first initially estimate the facet normal field. Another reason we choose to start with the facet normal field is that vertex normals are commonly less accurate or even missing in raw data. Many existing filters can be used in the initialization to obtain a relatively accurate facet normal field, as we

found no significant differences in our experiments. Here, we adopt Zheng et al.'s local bilateral normal filter [5], in which the weight function depends on spatial distance, signal distance and sampling ratio simultaneously.

5.2 Neighboring Facets Clustering

In order to more accurately estimate the vertex normal field and refine the facet normal field at feature regions, we attempt to figure out the underlying surface geometry around a feature vertex, and use its geometric information to estimate its normal. We first cluster the vertexs neighboring facets into a number of groups. The neighboring facets clustering is trying to find consistent grouping of connected facets around the vertex, so that the facets in each group have similar or smoothly varying normals. In other words, we are seeking sub-regions around the vertex with as few geometrical variations as possible. Notice that it is only necessary to apply the clustering to feature vertices, as non-feature vertices will always result in one single group.

We formulate the clustering problem into an optimization framework with the energy function defined as

$$E = \sum_{i=1}^{k} \int_{M_i} \rho(x)\mathbf{n} \cdot (\mathbf{n} - \overline{\mathbf{n}}_i)dx, \tag{1}$$

where k is the number of clusters (groups); M_i denotes an arbitrary cluster; $\rho(x)$ gives the density function (usually a constant function); $\mathbf{n}$ represents the normal of arbitrary points on a facet in the cluster M_i; and $\overline{\mathbf{n}}_i$ is the representative normal computed by normalizing the weighted average of all facet normals within the cluster. We can find that the energy E encodes the total offsets of the facet normals within a cluster to the representative normal of that cluster. If we assume that normals are the same in arbitrary points within a facet, and specify $s_j = \int_{f_j} \rho(x)dx$ of facet f_j, the E can be discretized as

$$E = \sum_{i=1}^{k} \left[\sum_{f_j \in M_i} s_j \|\mathbf{n}_j\|^2 - \left\| \sum_{f_j \in M_i} s_j \mathbf{n}_j \right\| \right], \tag{2}$$

where $\mathbf{n}_j$ is the unit normal of facet f_j. In our case, we set $\rho(x) = 1$ and hence s_j is the area of facet f_j. We propose a simple and efficient iterative clustering procedure to solve this optimization problem.

Initialization. The number of groups k to be clustered is decided by the results of vertex classification. For sharp edge vertex, we suppose the vertex is at an intersection between 2 consistent regions, and k is set to be 2. For corner vertex, we set $k \geq 3$. To initialize the groups, we select k seed facets, so that each facet is assigned to a different group at the beginning. The seed facets are selected from the vertex's one-ring facets, and they have maximal normal difference.

Optimization. We adopt an iterative strategy to adjust the boundary edges in all the groups, so that the energy E decreases progressively. In each iteration, we process every boundary edge of all groups. For each boundary edge e of M_i, with facets f_k and f_h incident to it, we are going to determine whether f_k and f_h should be clustered into M_i or not. Without loss of generality, supposing that f_k is already in group M_i, there are two ways to deal with facet f_h:

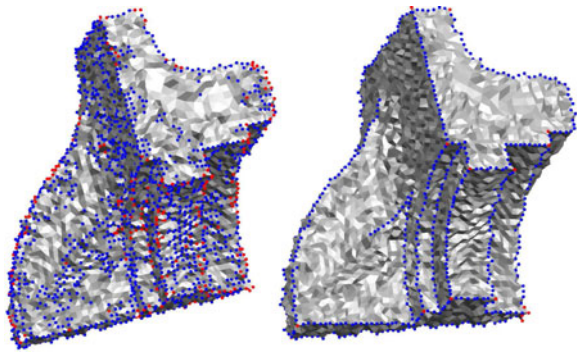


Fig. 3. Vertex classification results can be refined by the neighboring facets clustering algorithm. Left: the classification results of the normal tensor voting. Right: the results are refined after performing the clustering. Red points denote corner vertices and blue points denote sharp edge vertices.

1) if f_h has not been clustered into any facet group, we add it to M_i;
2) in case that f_h has already belonged to a facet group (e.g., M_j), we test whether a change of facet group leads to a decrease of E. If the test is positive, we would move either f_h to M_i or f_k to M_j, otherwise keep the grouping unchanged.

The boundary test mentioned involves the computation and comparison of three possible E values:

1) E_1: $f_k \in M_i$ and $f_h \in M_j$;
2) E_2: $f_k \in M_i$ and $f_h \in M_i$;
3) E_3: $f_k \in M_j$ and $f_h \in M_j$.

By checking with $E_{min} = \min(E_1, E_2, E_3)$, we can decide if a change of facet group is necessary.

When all the boundary edges have been tested, an iteration ends. The process repeats until no further decrease of E is found, and we suppose E has arrived its minimum. The convergence of E in the optimization can be guaranteed by a prove derived from Eq. (2).

In the case of corner vertices, k usually ranges between 3 and 6, we calculate E_k for each value of k, and then minimize E_k to find the optimal k. As there is only a few number of candidate values for k and a small percentage of corner vertices in a mesh, the proposed approach to solve the optimization is acceptable in terms of computation complexity (see Table 2 in Section 7.5). There is no significant increase of computational time observed when solving an optimal clustering for a corner vertex.

It is worth noticing that, our clustering algorithm can also help in further distinguishing the nature of features. In fact, it is common that normal tensor voting or other existing classification techniques will misclassify features because of the existence of noise. During our clustering process, when there happens to have empty clusters, it means that the current vertex is misclassified. For example, if the clustered result contains an empty group in a corner vertex, this vertex should be misclassified, and should be re-classified as a sharp edge or a non-feature vertex. Fig. 3 shows an example that the vertex classification results can be refined by the neighboring facets clustering algorithm. Fig. 4 shows clustering results for a noisy Fandisk model. It is observed that a set of consistent facet groups are produced which demonstrates the effectiveness of the clustering technique.

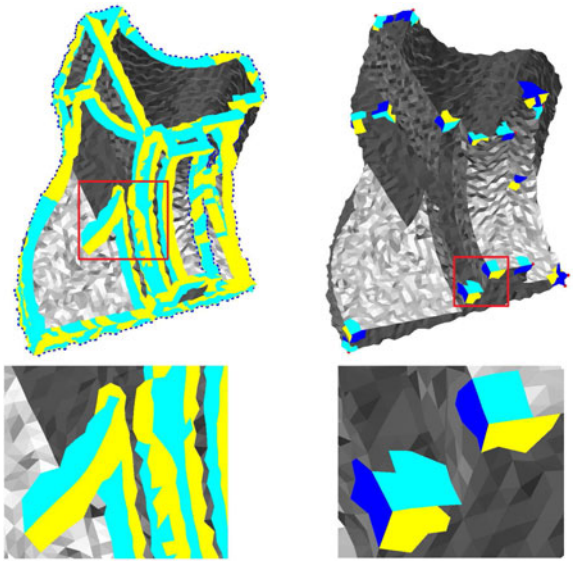


Fig. 4. Neighboring facets clustering of the noisy Fandisk model. Left column shows clustering results for sharp edge vertices; right column shows clustering results for corner vertices. Each color represents a distinct segment.

5.3 Vertex Normal Field Estimation

Based on the results of vertex classification and neighboring facets clustering, we estimate the vertex normal field. For a non-feature vertex, as the variation of surface geometry is relatively small, vertex and facet normals are usually consistent to each other. Therefore, we obtain its vertex normal simply from its surrounding facet normals through weighted interpolation. The estimation is performed with a weighted average of all normals from its one-ring neighboring facets, where the weighting factor is selected to be the area of each facet.

For a feature vertex, the estimation based on simple average of closest facet normals will suffer from significant errors due to the inconsistence. The estimation of its vertex normal is performed with a plane-fitting operation on each of the clustered facet group. The optimal plane is obtained by fitting the vertices in a least-squares manner. We take the normal of each optimal plane as the representative normal of the facet group, and the vertex normal can be easily computed by averaging the representative normals. The process repeats for all feature vertices to form the vertex normal field. Thus, we can obtain a more accurate vertex normal field than traditional methods that indiscriminately average a vertex's one-ring facet normals for all vertices.

An accurate vertex normal field is important for mesh denoising. Employing inaccurate vertex normal field may lead to vertex drifts. Fig. 5 illustrates the vertex drifting problem caused by an inaccurate vertex normal field. Most of the existing denoising approaches lead to vertex drifts, especially when irregular surface sampling exists. This phenomenon also happens in our approach if inaccurate vertex normals are used. However, once more accurate vertex normals are obtained, such artifacts can be avoided (see the last column of Fig. 5).

5.4 Facet Normal Field Refinement

The local bilateral filter used for initializing the facet normal field is performed on the entire neighborhood of each facet, where multiple piecewise smooth regions probably exist. This would result in less accurate facet normals, yet inevitably lead to blurred sharp features and elimination of shallow features during denoising.

As mentioned, the vertex normal at a feature vertex is computed by averaging the representative normals of the clustered groups of its facet neighbors. This means the representative normals can be considered as the decomposed components of the vertex normal. Each component represents the normal of a piecewise smooth region around the vertex, and the consistence of the two normal fields is strong within each piecewise smooth region. To further optimize the facet normal field, we improve Zheng et al.'s global bilateral filter [5] by adding the piecewise consistence around feature vertices as constraints.

In order to give readers an intuitive impression, we briefly introduce the original global scheme here. Zheng et al. formulated a global bilateral filter function E_s to encode the sum of squared error of each facet normal to the normals of its neighboring facets. To be faithful to the raw facet normal field, they formulated another function E_d to encode the squared deviation of the new facet normal field to its raw one. Though adding the raw normals as the constraints, Zheng et al.'s global scheme still suffers from feature blurring, since the bilateral weight factor in the Laplace operator is defined on the entire neighborhood.

Our improved method, therefore, add one more constraint term:

$$E_l = \sum_{i \in T} \sum_{K \in M} A_K \left\| \sum_{j \in K} \mathbf{n}'_j / |K| - \mathbf{n}_{i,K} \right\|^2, \qquad (3)$$

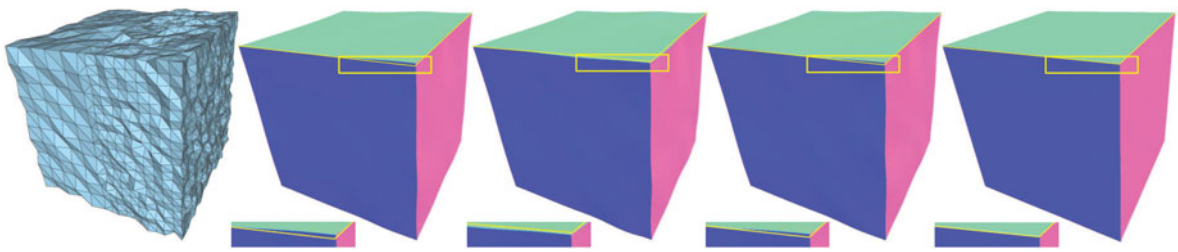


Fig. 5. From left to right: the cube model which is of sparse sampling on the left half part and dense sampling on the right part, and is corrupted by Gaussian noise; denoising results by Sun et al.'s method [8], Zheng et al.'s method (local scheme) [5], our method with inaccurate vertex normals obtained by traditional methods that indiscriminately average a vertex's one-ring facet normal field, and our method with vertex normals obtained by the proposed method. From the zoomed views we notice that the three preceding methods cause the edge drifts except our approach with well estimated vertex normals.

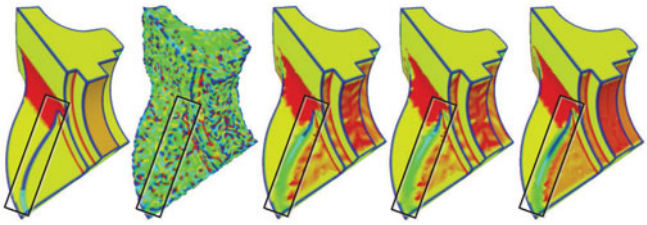


Fig. 6. Mean curvature visualization of the Fandisk model with shallow features highlighted. From left to right: the original Fandisk model, the noisy model (Gaussian noise with $\sigma = 0.2$ of mean edge length), denoising results of Zheng et al.'s local scheme ($\sigma_S = 0.3, n_1 = 10, n_2 = 12$) [5], Zheng et al's global scheme ($\lambda = 0.02, \sigma_{S2} = 0.35, n_2 = 12$) [5], Zheng et al.'s global scheme with the extra energy term E_l ($\sigma_{S2} = 0.35, n_2 = 12$). The global filter combining the extra energy term is more useful to recover the shallow geometry.

where T is the set of the feature vertices; M is the cluster set of each considered neighborhood of each feature vertex; A_K is the ratio of the average triangle area of the cluster K to the average triangle area of the entire mesh; $\mathbf{n}'_j$ are the updated facet normals in cluster K; $|K|$ is the number of triangles in K; $\mathbf{n}_{i,K}$ denotes the decomposed component of the vertex normal representing cluster K. The constraint term E_l encodes the total squared errors between each facet normal in a cluster and the cluster's representative normal around all feature vertices. This new constraint term can guarantee that the face normals do not diffuse themselves outside of the boundary of piecewise smooth patches, and thus, the negative effects from other piecewise smooth patches can be completely avoided during the optimization.

Finally, the optimization is formulated as

$$\arg\min_{n'_i}(1-\alpha-\beta)E_s + \alpha E_d + \beta E_l, \tag{4}$$

where α and β are two positive variables with the range of $\alpha + \beta \in [0, 1]$. Here α is used to control the degree of retaining the original normal field, while β is used to more accurately recover the facet normals around the feature vertices. Since the facet normal field has been initialized in Section 5.1, α can be chosen relatively large. β can also be set relatively large to preserve features, benefitting from the rectified vertex normal field. We empirically set $\alpha = 0.4$ and $\beta = 0.2$ which work well in our experiments.

Fig. 6 demonstrates the advantages of using the one more energy term E_l. From the mean curvature visualization of the Fandisk model with shallow feature highlighted, it is observed that both Zheng et al.'s local and global filters wipe away the shallow features more seriously. When the extra energy term is combined in Zheng et al.'s global filter, the result is improved obviously.

6 Vertex Position Update

After the bi-normal filtering, we update vertex positions according to both the facet and vertex normal fields. There exists a simple geometric fact that every mesh vertex lying within tangent planes of its neighboring facets gives the mesh a better approximation of the underlying surface [38]. Inspired by this idea, we formulate vertex updating as a quadratic optimization problem.

We define the first *quadric* as

$$E_1(v') = \sum_{f \in N_v(f)} \left[\mathbf{n}_f^T (v' - c_f)\right]^2, \tag{5}$$

which encodes the sum of squared distances from vertex v' to its one-ring facets, where $\mathbf{n}_f$ is the refined unit normal of facet f, and c_f is its triangle barycenter. Minimizing Eq. (5) probably yields a nice approximation if all mesh facets are tangent to the underlying surface at certain points. However, existing normal filters, including our method of embedding the piecewise consistence to the global bilateral filter, cannot guarantee convergence [8] and produce mesh facets exactly tangent to the underlying surface. In this regard, only minimizing the above quadric derived from filtered facet normals inevitably leads to feature blurring and vertex drifts to some extent. Therefore, we define the second *quadric* using the estimated vertex normals as

$$E_2(v') = (v' - v)^T (v' - v) - \left[\mathbf{n}_v^T (v' - v)\right]^T \left[\mathbf{n}_v^T (v' - v)\right], \tag{6}$$

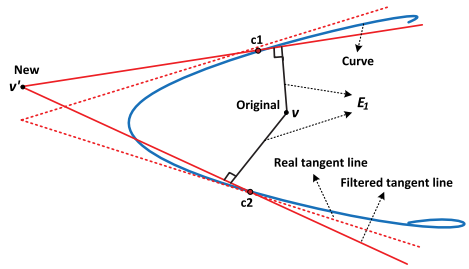
where $\mathbf{n}_v$ represents the unit normal of vertex v. In Eq. (6), the term $\mathbf{n}_v^T(v' - v)$ describes the projection from vector $(v' - v)$ to vertex normal $\mathbf{n}_v$; this quadric, in turn, encodes the squared distance from the vertex normal line to the optimized vertex v'. We encourage the vertex to move along its normal direction in the optimization. Ideally, when vector $(v' - v)$ is parallel to $\mathbf{n}_v$, $E_2(v')$ equals zero.

Accordingly, we minimize the two quadrics together to generate the new vertex position as

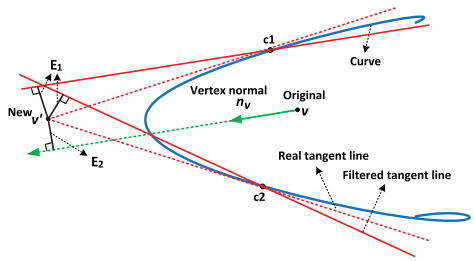
$$\arg\min_{v'}[E_1(v') + \theta E_2(v')], \tag{7}$$

which only involves a simple system of linear equations. The θ serves for balancing the effects of $E_1(v')$ and $E_2(v')$. If θ is relatively small, the resulting mesh tends to have smoother patches, which is caused by the definition of $E_1(v')$; if θ is relatively big, $E_2(v')$ has more effect in Eq. (7), yet is more capable of preserving geometry details and avoiding vertex drifts especially in feature vertices. Here, we empirically set $\theta = 1.0$ which assigns equal importance to the two normal fields.

Fig. 7 demonstrates the behaviors of our quadratic optimization in 2D. Assume a curve with two points c_1, c_2 on it, v' represents the optimized position of v, $\mathbf{n}_v$ represents the estimated normal at v, the two solid red lines individually denote the filtered tangent lines at c_1 and c_2, while the dashed red lines denote their corresponding real tangent lines at the two points, E_1 encodes the total squared distances from a point to its filtered neighboring tangent lines and E_2 encodes the squared distance from a point to its filtered vertex normal line. Due to the deviation of current normal filters, the filtered tangent lines (see solid red lines) would not exactly match the noise-free ones (see dashed red lines). The tangent lines have the capability of describing the local structures of the curve. If these lines cannot be exactly estimated, they probably lead to side effects, such as vertex drifts and feature blurring. We notice from Fig. 7a that, directly minimizing the first quadric we would obtain undesirable 'optimized' vertex v' because of inaccurate tangent lines used. Whereas, from Fig. 7b, we notice that, one more



(a) Minimizing E_1 to obtaining v'.



(b) Minimizing $(E_1 + E_2)$ to obtaining v'.

Fig. 7. A 2D example of vertex position update by the proposed quadratic optimization (in this case, three points, v, c_1 and c_2 are used to approximate the local curve): (a) the updating of vertex v by the first quadric leads to undesirable updated position (see v') due to applying inaccurate tangent lines, and (b) the updating of v by the two quadrics guides v to converge to an appropriate position.

quadric is added as the compensation of information lost for the facet normal filtering. As a result, minimizing the two quadrics can correct the optimization along the normal line, where it produces the better result. The vertex adjustment by minimizing the two quadrics is a strong characteristic of this method which allows feature preservation while removing noise very well.

Fig. 8 demonstrates the advantages of incorporating the vertex normal field in the vertex position update. From the example of the first row, we can notice that vertex drifts can be avoided by the collaboration of the two normal fields. From the example of the second row, it is observed that more fined detailed can be preserved in the presence of the vertex normal field. More experimental results can be found in Section 7.

7 Results and Discussion

We have performed our approach on multifarious mesh models with either synthetic or raw noise. Several of them are even factitiously resampled to simulate irregular surface sampling. These models are applied to validate the approach in dealing with challenging regions with features at various sizes and/or irregular surface sampling. The synthetic noise used in this paper is generated by a zero-mean Gaussian with standard deviation σ proportional to the mean edge length of the mesh.

We have compared our approach with six exemplary denoising techniques: Fleishman et al.'s bilateral mesh filter [3], Sun et al.'s unilateral normal filter [8], Fan et al.'s second-order bilateral filter [10], Zheng et al.'s local bilateral normal filter and its global representation [5], Wang et al.'s cascaded filter [11], and He and Schaefer's area-based edge filter [2]. All these approaches are anisotropic filters which, in general, can better preserve features than isotropic ones.

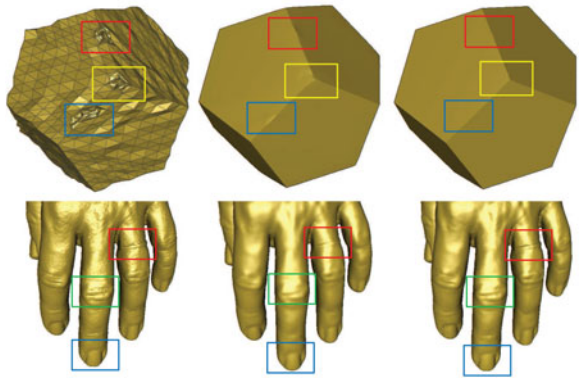


Fig. 8. Collaboration of the two normal fields for vertex position update. The upper row is a CAD-like model with irregular surface sampling at three special regions (one around a corner, and two around sharp edges), and the bottom row is a real scanned model with rich details. Form left to right: the original noisy models, the denoising results with only the facet normal field, and the denoising results with the collaboration of the two normal fields.

Apart from visual comparison, we employ three quantitative criteria, namely, the mean square angular error (MSAE), the Hausdorff distance histogram, and the L^2 vertex-based error to assess the fidelity of denoised results to ground truths. Finally, we evaluate the time performance of the proposed method.

To allow for fair comparisons, we carefully adjust the parameters of each approach to produce visually best results. The parameters of our approach include the standard deviation σ_{S1} of normal difference and the iteration number n_1 used in the facet normal field initialization; the weights α, β and the standard deviation σ_{S2} in the facet normal field optimization; and the weight θ and the iteration number n_2 used in vertex position update. In the following experiments, if not specified, we set $\alpha = 0.4$, $\beta = 0.2$, and $\theta = 1.0$, the σ_{S1} in the local representation and the σ_{S2} in the global representation of the facet normal filter are all within the range of [0.2-0.6] as suggested in [5]; the n_1 is within [3-20] and the n_2 is within [1-15].

7.1 Denoising with Two Normal Fields

To illustrate the advantages of using the two normal fields in denoising, we first compare our results with those of Zheng et al.'s local and global bilateral filters, which use only facet normal field in denoising. We perform the comparison on the Eros model. It is observed in the first column of Fig. 9, this model has varieties of features and abundant small details. It is ruined by synthetic noise as shown in the second column of Fig. 9. It is found that Zheng et al.'s two schemes usually over-smooth some of the weak features, such as those in the eye region and hair region. In contrast, our method can better preserve these features as shown in the blow-up images. These is because we simultaneously employ the two normal fields that are estimated according to the piecewise consistence during denoising.

We further compare our method with widely-used or start-of-the-art mesh denoising approaches. Fig. 10 shows the denoising results of the CAD-like Octa-flower model. This model has very sharp corners (the petal region) and many curved edges (the helix region), which are difficult to be

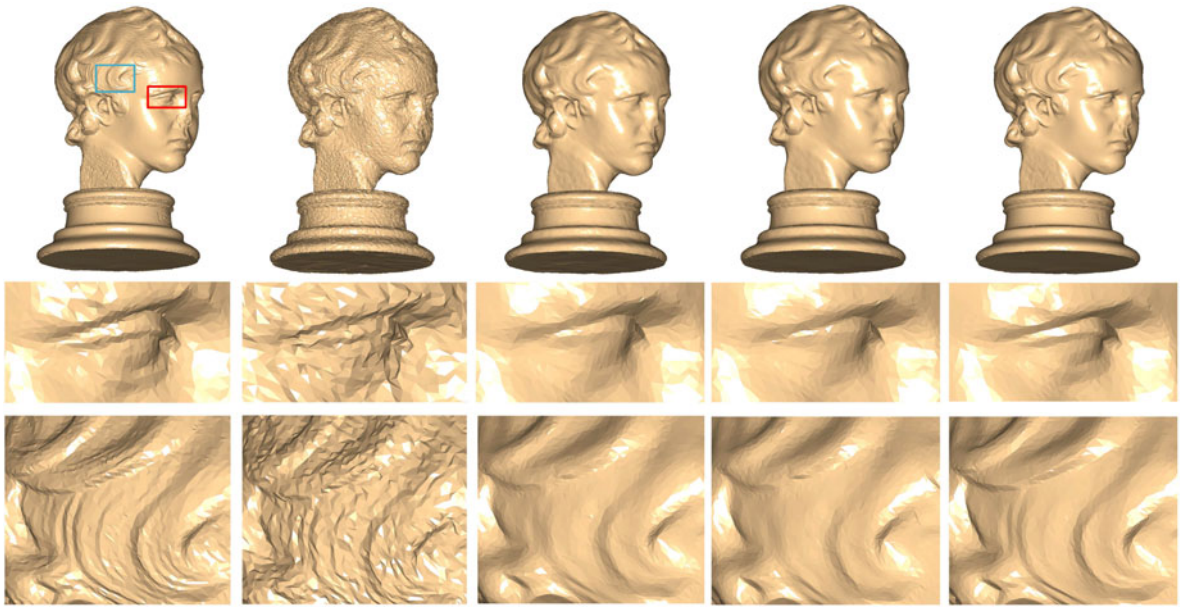


Fig. 9. Denoising of the Eros model. From left to right: the original model, the noisy model (Gaussian noise with $\sigma = 0.1$ of mean edge length) and denoising results with Zheng et al.'s local scheme [5] ($\sigma_{S1} = 0.35, n_1 = 5, n_2 = 5$), Zheng et al.'s global scheme ($\lambda = 0.15, \sigma_{S2} = 0.4, n_2 = 8$) [5], and our method ($\sigma_{S1} = 0.35, n_1 = 5, \sigma_{S2} = 0.3, n_2 = 4$). The use of the two normal fields performs better in preserving weak features as in the blow-up regions shown in the middle and bottom rows.

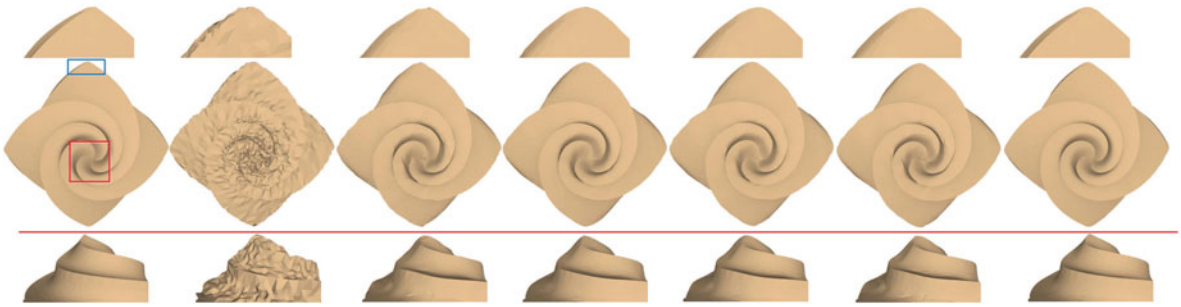


Fig. 10. Denoising of the Octa-flower model. From left to right: the original model, the noisy model (Gaussian noise with $\sigma = 0.2$ of mean edge length) and denoising results of Sun et al.'s method ($T = 0.4, n_1 = 5, n_2 = 10$) [8], Fan et al.'s method ($k_1 = 35, k_2 = 8, Eps = 2, MinPts = 6$) [10], Wang et al.'s method ($n_1 = 5, \tau = 0.008, n_2 = 9$) [11], He and Schaefer's method ($\mu = \sqrt{2}, \alpha = 0.1\overline{\gamma}, \lambda = 0.02\ell_e^2\overline{\gamma}$) [2], and our method ($\sigma_{S1} = 0.35, n_1 = 8, \sigma_{S2} = 0.3, n_2 = 8$). The top row zooms in at the petal region (see the blue rectangle) and the bottom row is the zoomed side view at the helix region (see the red rectangle).

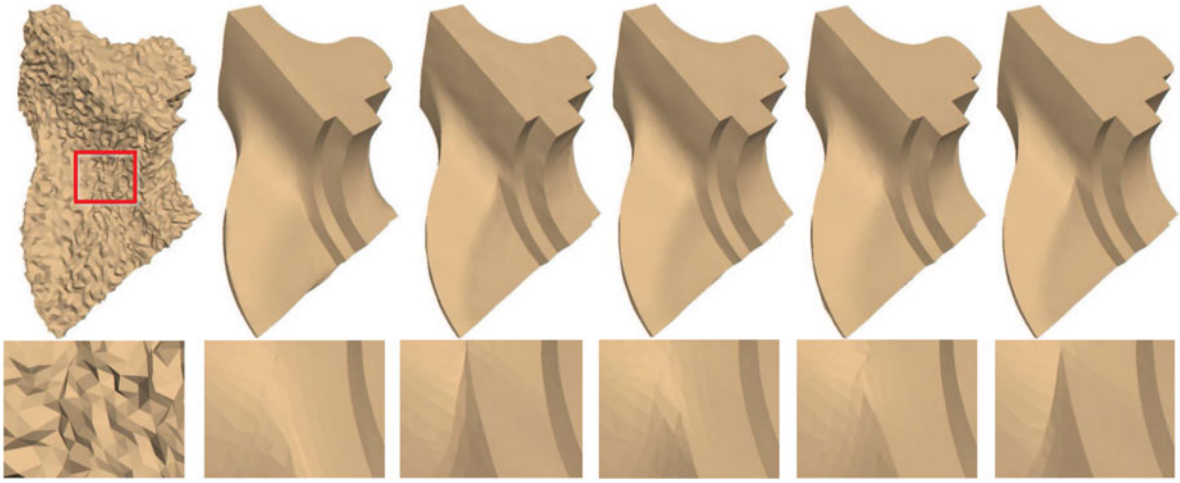


Fig. 11. Denoising of the Fandisk model. From left to right: the input noisy model (Gaussian noise with $\sigma = 0.3$ of mean edge length), denoising results of Sun et al.'s method ($T = 0.35, n_1 = 20, n_2 = 40$) [8], Fan et al.'s method ($k_1 = 30, k_2 = 8, Eps = 2, MinPts = 6$) [10], Wang et al.'s method ($n_1 = 6, \tau = 0.01, n_2 = 9$) [11], He and Schaefer's method ($\mu = \sqrt{2}, \alpha = 0.1\overline{\gamma}, \lambda = 0.02\ell_e^2\overline{\gamma}$) [2], and our method ($\sigma_{S1} = 0.35, n_1 = 20, \sigma_{S2} = 0.3, n_2 = 14$). The bottom row shows magnified fragments of a shallow feature on the model. Our method outperforms other techniques in preserving the shallow feature.

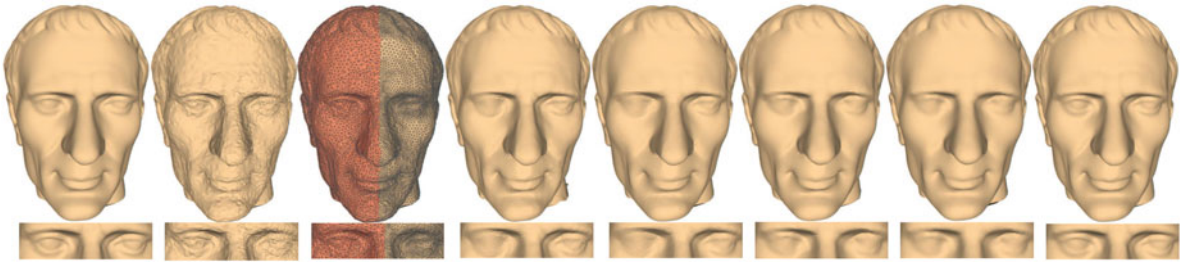


Fig. 12. Denoising of the Julius model. From left to right: the original model, the model which is first resampled by simplifying the left half part of the model and then corrupted by Gaussian noise with 0.2 of mean edge length, the wireframe display for showing the irregular sampling, denoising results of Fleishman et al.'s method ($n = 5$) [3], Sun et al.'s method ($T = 0.6, n_1 = 3, n_2 = 4$) [8], Wang et al.'s method ($n_1 = 3, \tau = 0.01, n_2 = 4$) [11], He and Schaefer's method ($\mu = \sqrt{2}, \alpha = 0.1\overline{\gamma}, \lambda = 0.02\ell_c^2\overline{\gamma}$) [2], and our method ($\sigma_{S1} = 0.35, n_1 = 3, \sigma_{S2} = 0.3, n_2 = 2$).

completely preserved when removing noise. It is obvious that, in our result, noise around sharp edges is more nicely eliminated and the corner and sharp edges are better preserved, when compared with other methods (see the blow-ups).

Fig. 11 shows the denoising results on the Fandisk model corrupted by relatively high level of noise. One of the most challenging tasks when handling this model is preserving its shallow features (see the blow-up region) during denoising. It is observed that, the first four methods can preserve sharp features while effectively removing noise, but the shallow feature is almost completely eliminated in the results. Our result can well preserve both sharp and shallow features, and more faithfully recover the shallow curved surface as the original model (see the blow-up region of our result). These experiments demonstrate that our method is capable of well preserving both sharp features and shallow features in a mesh model.

7.2 Denoising Irregularly Sampled Mesh

It is a challenging task to denoise irregularly sampled surfaces, as most previous approaches do not adapt well to variably sampling density in the mesh. Fig. 1 shows the denoising results of the Block model which is first subdivided at three specially chosen regions and then corrupted by Gaussian noise. The first column shows the wireframe of the noisy and irregularly sampled mesh, and we can clearly see that the corner regions are more densely sampled. When different denoising approaches are applied on this model, it is observed that the other four methods can preserve sharp corners while removing noise to some extent. However, with a careful investigation of magnified fragments, we can observe that these approaches all unavoidably introduce unevenness at the transition between irregular and regular sampled regions and introduce vertex drifts. In contrast, our method can avoid the artifacts at the transitional region, and thus better preserve the corners and sharp edges.

Fig. 12 shows another experiment performed on a more general model with irregular sampling and less shape corners. While all approaches can well remove noise to some extent, our result is less influenced by the irregular surface sampling at the eyes region (see the blow-ups) of the model and more faithful to the ground truth.

7.3 Denoising Scanned Data

Apart from synthetic cases, we also verify the effectiveness of our approach on two scanned models with raw noise as shown in Figs. 13 and 14. It is observed that our approach also outperforms other approaches in terms of preserving fine details and sharp features for raw scanned models. For example, at the wing and eye regions of the Angel model, we can see that weak features remain after denoising in the result of our approach but are lost in the results of other approaches. Similar phenomena are also found in the results of the Decorative pattern embossed model.

7.4 Quantitative Evaluation

From the above comparisons, we find that our approach can produce visually better results than some state-of-the-art

Fig. 13. Denoising of the Angel model. The Leftmost is the original model, and from middle-topmost to bottom-rightmost: denoising results of Fleishman et al.'s method ($\mathrm{n} = 15$) [3], Sun et al.'s method ($T = 0.55, n_1 = 5, n_2 = 10$) [8], Wang et al.'s method ($n_1 = 5, \tau = 0.05, n_2 = 9$) [11], He and Schaefer's method ($\mu = \sqrt{2}, \alpha = 0.1\overline{\gamma}, \lambda = 0.02\ell_c^2\overline{\gamma}$) [2], and our method ($\sigma_{S1} = 0.4, n_1 = 5, \sigma_{S2} = 0.35, n_2 = 4$).

Fig. 14. Denoising of the Decorative pattern embossed model. The left most is the original model, and from middle-topmost to bottom-rightmost: denoising results of Fleishmann et al.'s method $(n = 7)$ [3], Sun et al.'s method $(T = 0.5, n_1 = 3, n_2 = 4)$ [8], Wang et al.'s method $(n_1 = 4, \tau = 0.03, n_2 = 6)$ [11], He and Schaefer's method $(\mu = \sqrt{2}, \alpha = 0.1\overline{\gamma}, \lambda = 0.02\ell_e^2\overline{\gamma})$ [2], and our method $(\sigma_{S1} = 0.35, n_1 = 3, \sigma_{S2} = 0.3, n_2 = 2)$.

approaches in the experimental cases. To give a more objective comparison, we further evaluate these denoising results using three widely-used quantitative metrics. The first metric is the mean square angular error, which is used to compare the denoised facet normals in terms of the number of iterations with those of the ground-truth models [8], [5], [11]. The second metric is the Hausdorff distance histogram, which is applied to evaluate fidelity of the denoising results (after vertex position update) to the underlying surface [39], [11]. The third is the L^2 vertex-based mesh-to-mesh error metric [8], [4], [5], which is used to compare our vertex updating scheme with traditional gradient descent based (GDM) method [8].

Fig. 15 plots MSAE of Zheng et al.'s local and global bilateral filters [5], and our bi-normal filtering scheme. In order to clearly observe the differences, we perform Zheng et al.'s local scheme up to 50 times, though it is not necessary in practice. Zheng et al.'s global scheme and the improved global scheme in our bi-normal filtering scheme are non-iterative, which are also plotted in the same diagrams. Because we employ the piecewise consistence in the bi-normal filtering, the facet normals near the feature vertices are better recovered. Thus, in general, our constrained global scheme can consistently result in lower MSAE.

Fig. 16 shows a detailed plotting of the Hausdorff distances between the denoising mesh surface and the underlying surface. We can notice that, regardless of CAD-like models or general models, our approach consistently leads to smaller Hausdorff distances than other approaches. This indicates that our approach yields a closer mesh surface relative to the underlying surface, in addition to its high effectiveness of noise elimination and feature preservation at various sizes.

To validate the efficacy of the proposed vertex position update scheme based on two normal fields, we compare our quadratic optimization-based method (QOM) with the gradient descent based methods used in [8], [4], [5]. The GDM is designed to match the facet normal field, while the proposed QOM is to simultaneously match the facet and vertex normal fields. For the fair comparison, we employ the normal field obtained by our bi-normal filtering scheme as the input of both QOM and GDM, and then measure the L^2 vertex-based error E_v between the denoised meshes and the ground truth models. In addition, as QOM and GDM may take different iteration numbers to obtain visually best results, we calculate E_v at the two different iteration numbers for each model. Table 1 shows the results. It is observed that the E_v of QOM is always lower than that of GDM, demonstrating the results of the proposed QOM are more faithful to the ground truth models.

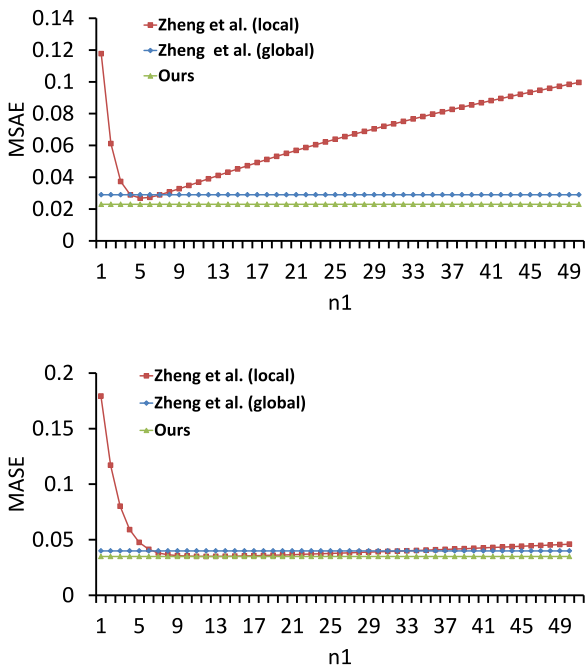


Fig. 15. Comparison of the MSAE resulting from Zheng et al.'s local bilateral filtering, Zheng et al.'s global bilateral filtering and our bi-normal filtering on the the Fandisk model (up) and the Eros model (bottom). The horizonal axis represents the number of normal filtering iterations, and the vertical axis shows the corresponding MSAE.

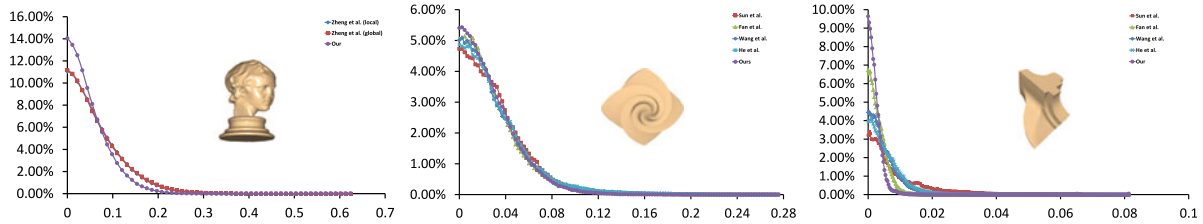


Fig. 16. Comparison of the Hausdorff distances. The horizontal axis denotes the error (absolute distance value) between the denoising mesh and the noise-free mesh, and the vertical axis represents the corresponding percentage in terms of each error value. The denoising results displayed here are from Figs. 9, 10, and 11, respectively.

7.5 Computational Cost

We record time performance of each step in our experimental cases and the results are shown in Table 2. Our method is implemented using C++ and the experiments are performed on a PC with a 2.90 GHz Intel core i5 and 8.0 GB of RAM. It is observed that the vertex classification and the vertex position update are quite fast while the bi-normal filtering is relatively time-consuming, especially when the model contains a large number of vertices and facets (such as the Eros Model). This is because we take a lot of time to optimize the facet normal field. Although our method is a little more computationally intensive than existing methods that just apply facet normal field to guide the denoising, the time performance is still acceptable. For example, it just takes around 4s to denoise a complicated model with $45{,}001$ vertices and $89{,}998$ facets. In the future, we will investigate how to accelerate the bi-normal filtering by GPUs.

Limitations. First, we employ normal tensor voting algorithm for vertex classification, which is able to handle reasonably large noise, e.g., Gaussian noise with $\sigma = 0.3$ of mean edge length. When higher level of noise is added, a large number of pseudo-features would be detected, which will degrade the performance of our method. However, such large noise is not common in practice. Second, our algorithm can not handle models with extremely irregular surface sampling, as in this case, it is difficult to recover the two normal fields from each other. Fig. 17 shows an example, where shape edges and corners are formed by a lot of irregular triangles, making it difficult to estimate the two normal fields and achieve a good denoising result. Last, we empirically fix some parameters and they work well for the experimental models. However, they may not be suitable for all models with different structures and/or different levels of noise. But we have provided readers the principles on how to choose these parameters.

TABLE 1
L^2 Vertex-Based Mesh-to-Mesh Error Comparison

Models	iterations	Methods	$E_v(\times 10^{-3})$
Block	$n_2 = 12$	GDM	4.397
		QOM	3.425
	$n_2 = 4$	GDM	2.439
		QOM	1.746
Octa-flower	$n_2 = 8$	GDM	3.617
		QOM	3.069
	$n_2 = 10$	GDM	3.431
		QOM	3.246
Fandisk	$n_2 = 14$	GDM	5.483
		QOM	4.662
	$n_2 = 40$	GDM	4.917
		QOM	4.822

8 Conclusion

In this paper, we present an effective mesh denoising algorithm which can better preserve multifarious features than existing techniques when removing noise. The effectiveness of our approach comes from the bi-normal filtering of the vertex normal field and the facet normal field and the novel quadratic optimization based on the two fields. By enhancing the piecewise consistence of the two normal fields at feature regions, we can well preserve both sharp and shallow features and produce visually and quantitatively better results. Extensive experiments on various synthetic and real data demonstrate the capability of the

TABLE 2
Time Performance of Our Method

Models	Stage 1	Stage 2	Stage 3	Total
Block		(0.0235 s+		
$\|V\|$: 2,627	0.0007 s	0.0069 s+	0.0125 s	0.0636 s
$\|F\|$: 5,262		0.0001 s+		
Fig. 1		0.0200 s)		
Eros		(0.2393 s+		
$\|V\|$: 45,001	0.0174 s	0.1241 s+	0.1362 s	4.1811 s
$\|F\|$: 89,998		0.0003 s+		
Fig. 9		3.6641 s)		
Octa-flower		(0.0730 s+		
$\|V\|$: 7,919	0.0034 s	0.0854 s+	0.0760 s	0.5751 s
$\|F\|$: 15,834		0.0001 s+		
Fig. 10		0.3373 s)		
Fandisk		(0.0777 s+		
$\|V\|$: 6,475	0.0014 s	0.0400 s+	0.0401 s	0.2490 s
$\|F\|$: 12,946		0.0001 s+		
Fig. 11		0.0898 s)		

In the table, stage 1 means vertex classification, stage 2 means bi-normal filtering, and stage 3 means vertex position update.

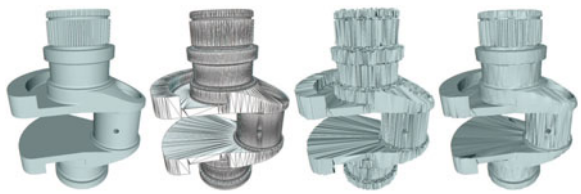


Fig. 17. An example for which our algorithm fails to recover its features while removing noise. From left to right: the ground truth with an extreme triangulation, the wireframe display, the noisy input, and our result.

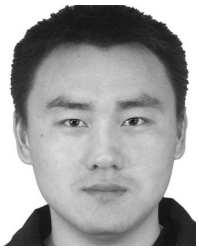
proposed method. In the future, we will attempt to implement our approach on GPU for acceleration and solve the problems mentioned in the limitation part.

ACKNOWLEDGMENTS

The authors thank the reviewers for their valuable comments. They also thank Xianfang Sun for his constructive suggestions, and Xianfang Sun and Youyi Zheng for providing executable files. Mingqiang Wei and Jinze YU both contributed equally to this paper, and should be considered equal first authors. This work was supported by the grants from the Research Grants Council of Hong Kong (Project No. 412412 and 412513) and the grants from the National Natural Science Foundation of China (Project No. 61233012 and 61222206). The models used in this paper were from the AIM Shape Repository, the Stanford 3D Scanning Repository, and Lei He.

REFERENCES

[1] S. Liu, K.-C. Chan, and C. C. Wang, "Iterative consolidation of unorganized point clouds," *IEEE Comput. Graph. Appl.*, vol. 32, no. 3, pp. 70–83, May. 2012.
[2] L. He, and S. Schaefer, "Mesh denoising via l0 minimization," in *Proc. ACM SIGGRAPH*, 2013, pp. 64:1–64:8.
[3] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. Graph.*, vol. 22, pp. 950–953, Jul. 2003.
[4] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Random walks for feature-preserving mesh denoising," *Comput. Aided Geometric Des.*, vol. 25, no. 7, pp. 437–456, 2008.
[5] Y. Zheng, H. Fu, O. K.-C. Au, and C.-L. Tai, "Bilateral normal filtering for mesh denoising," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 10, pp. 1521–1530, Oct. 2011.
[6] G. Taubin, "A signal processing approach to fair surface design," in *Proc. ACM SIGGRAPH*, 1995, pp. 351–358.
[7] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski, "Mesh regularization and adaptive smoothing," *Comput.-Aided Des.*, vol. 33, no. 11, pp. 789–800, 2001.
[8] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Fast and effective feature-preserving mesh denoising," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 5, pp. 925–938, Sep./Oct. 2007.
[9] T. Tasdizen, R. T. Whitaker, P. Burchard, and S. Osher, "Geometric surface processing via normal maps," *ACM Trans. Graph.*, vol. 22, no. 4, pp. 1012–1033, 2003.
[10] H. Fan, Y. Yu, and Q. Peng, "Robust feature-preserving mesh denoising based on consistent subneighborhoods," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 2, pp. 312–324, Mar./Apr. 2011.
[11] J. Wang, X. Zhang, and Z. Yu, "A cascaded approach for feature-preserving surface mesh denoising," *Comput.-Aided Des.*, vol. 44, no. 7, pp. 597–610, 2012.
[12] K. Hildebrandt, and K. Polthier, "Anisotropic filtering of nonlinear surface features," *Comput. Graph. Forum*, vol. 23, pp. 391–400, Sep. 2004.
[13] C.-Y. Chen and K.-Y. Cheng, "A sharpness dependent filter for mesh smoothing," *Comput. Aided Geometric Des.*, vol. 22, no. 5, pp. 376–391, 2005.
[14] M. Botsch, M. Pauly, L. Kobbelt, P. Alliez, and B. uno L¦vy, "Geometric modeling based on polygonal meshes," in *Proc. Eurograph. Tuts.*, 2008.
[15] J. Vollmer, R. Mencl, and H. Mueller, "Improved laplacian smoothing of noisy surface meshes," *Comput. Graph. Forum*, vol. 18, pp. 131–138, 1999.
[16] M. Desbrun, M. Meyer, P. Schroder, and A. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. ACM SIGGRAPH*, 1999, pp. 317–324.
[17] M. Meyer, M. Desbrun, P. Schroder, and A. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and Mathematics III*. New York, NY, USA: Springer, 2003, vol. 12, no. 7, pp. 35–57.
[18] B. Kim and J. Rossignac, "Geofilter: Geometric selection of mesh filter parameters," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 295–302, 2005.
[19] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, "Efficiently combining positions and normals for precise 3d geometry," *ACM Trans. Graph.*, vol. 24, pp. 536–543, 2005.
[20] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *Proc. 4th Int. Conf. Comput. Graph. Interactive Techn. Australasia Southeast Asia*, 2006, pp. 381–389.
[21] L. Liu, C.-L. Tai, Z. Ji, and G. Wang, "Non-iterative approach for global mesh optimization," *Comput.-Aided Des.*, vol. 39, no. 9, pp. 772–782, 2007.
[22] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski, "Polyhedral surface smoothing with simultaneous mesh regularization," in *Proc. Geometric Model. Process.*, 2000, pp. 229–237.
[23] C. Bajaj and G. Xu, "Anisotropic diffusion on surfaces and functions on surfaces," *ACM Trans. Graph.*, vol. 22, no. 1, pp. 4–32, 2003.
[24] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Anisotropic feature-preserving denoising of height fields and bivariate data," in *Proc. Graph. Interface*, 2000, pp. 145–152.
[25] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic geometric diffusion in surface processing," in *Proc. IEEE Conf. Vis.*, 2000, pp. 397–405.
[26] A. E. Ouafdi and D. Ziou, "A global physical method for manifold smoothing," in *Proc. IEEE Int. Conf. Shape Model. Appl.*, 2008, pp. 11–17.
[27] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 943–949, 2003.
[28] C. Tomasi, and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. Int. Conf. Comput. Vis.*, 1998, pp. 839–846.
[29] A. F. E. Ouafdi, D. Ziou, and H. Krim, "A smart stochastic approach for manifolds smoothing," *Comput. Graph. Forum*, vol. 27, no. 5, pp. 1357–1364, 2008.
[30] G. Taubin, "Linear anisotropic mesh filtering," IBM T.J. Watson Research Center, Ossining, NY, USA, IBM Res. Rep. RC22213 (W0110-051) 2001.
[31] H. Yagou, Y. Ohtake, and A. G. Belyaev, "Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding," in *Proc. Comput. Graph. Int.*, 2003, pp.28–33.
[32] Y. Shen and K. Barner, "Fuzzy vector median-based surface smoothing," *IEEE Trans. Vis. Comput. Graph.*, vol. 10, no. 3, pp. 252–265, May./Jun. 2004.
[33] L. Zhu, M. Wei, J. Yu, W. Wang, J. Qin, and P.-A. Heng, "Coarse-to-fine normal filtering for feature-preserving mesh denoising based on isotropic subneighborhoods," *Comput. Graph. Forum*, vol. 32, no. 7, pp. 371–380, 2013.
[34] Z. Bian and R. Tong, "Feature-preserving mesh denoising based on vertices classification," *Comput. Aided Geometric Des.*, vol. 28, pp. 50–64, 2011.
[35] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, "Decoupling noises and features via weighted l1-analysis compressed sensing," *ACM Trans. Graphics*, vol. 33, no. 2, pp. 1–12, 2014.
[36] G. Taubin, "Introduction to geometric processing through optimization," *IEEE Comput. Graph. Appl.*, vol. 32, no. 4, pp. 88–94, Jul./Aug. 2012.
[37] H. S. Kim, H. K. Choi, and K. H. Lee, "Feature detection of triangular meshes based on tensor voting theory," *Comput.-Aided Des.*, vol. 41, no. 1, pp. 47–58, 2009.
[38] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. ACM SIGGRAPH*, Aug. 1997, pp. 209–216.
[39] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," *Comput. Graph. Forum*, vol. 17, no. 2, pp. 167–174, 1998.



Mingqiang Wei is currently working toward the PhD degree in the Department of Computer Science and Engineering, the Chinese University of Hong Kong. His research interests include geometric modeling and its applications in biomedical systems.

Jinze Yu received the BSc degree in mathematics from Tsinghua University in 2008 and the MEng degree from Ecole Polytechnique in 2012. He was in the Virtual Reality, Visualization and Imaging Research Centre in the Department of Computer Science and Engineering, the Chinese University of Hong Kong in 2013. His research interests include mesh processing, image processing, and low-level domain of computer vision.

Wai-Man Pang received the PhD degree in computer science and engineering from the Chinese University of Hong Kong in 2008. He joined the University of Aizu's Computer Arts Lab as an assistant professor in 2010. He is currently an assistant professor in the Department of Computer Science at the Caritas Institute of Higher Education, Hong Kong. His research interests include nonphotorealistic rendering and GPU computing. He is a member of the IEEE.

Jun Wang received the PhD degree in CAD from the Nanjing University of Aeronautics and Astronautics (NUAA) in 2007. From 2008 to 2010, he conducted research as a postdoctoral scholar at the University of California, Davis and the University of Wisconsin, Milwaukee. From 2010 to 2013, he was a senior engineer at Leica Geosystems Inc.. In October 2013, he joined NUAA as a professor. His research areas include geometry processing and 3D laser scanning.

Jing Qin received the PhD degree in the Department of Computer Science and Engineering at the Chinese University of Hong Kong (CUHK) in 2009. Currently, he is an associate professor at Shenzhen Institute of Advanced Technology. His research interests include virtual reality-based surgical simulation, multisensory human-computer interaction, and biomechanical modeling.

Ligang Liu received the BSc degree in 1996 and the PhD degree in 2001 from Zhejiang University, China. He is a professor at the University of Science and Technology of China. Between 2001 and 2004, he was at Microsoft Research Asia. Then he was at Zhejiang University during 2004 and 2012. He paid an academic visit to Harvard University during 2009 and 2011. His research interests include geometric processing and image processing. His research works could be found at his research website: http://staff.ustc.edu.cn/∼lgliu.

Pheng-Ann Heng received the PhD degree in computer science from Indiana University Bloomington in 1992. He is a professor at the Chinese University of Hong Kong's Department of Computer Science and Engineering. His research interests include VR applications in medicine, visualization, medical imaging, human-computer interaction, and computer graphics. He is a senior member of the IEEE.