SIAM J. SCI. COMPUT.
Vol. 41, No. 1, pp. B1–B26

TRIANGULATED SURFACE DENOISING USING HIGH ORDER REGULARIZATION WITH DYNAMIC WEIGHTS*

ZHENG LIU†, RONGJIE LAI‡, HUAYAN ZHANG§, AND CHUNLIN WU¶

Abstract. Recovering high quality surfaces from noisy triangulated surfaces is a fundamental important problem in geometry processing. Sharp features including edges and corners cannot be well preserved in most existing denoising methods except the recent total variation (TV) and ℓ_0 regularization methods. However, these two methods have suffered producing staircase artifacts in smooth regions. In this paper, we first introduce a second order regularization method for restoring a surface normal vector field, and then propose a new vertex updating scheme to recover the desired surface according to the restored surface normal field. The proposed model can preserve sharp features and simultaneously suppress the staircase effects in smooth regions, which overcomes the drawback of the first order models. In addition, the new vertex updating scheme can prevent ambiguities introduced in existing vertex updating methods. Numerically, the proposed high order model is solved by the augmented Lagrangian method with a dynamic weighting strategy. Intensive numerical experiments on a variety of surfaces demonstrate the superiority of our method visually and quantitatively.

Key words. triangulated surface denoising, total variation, high order regularization, augmented Lagrangian method

AMS subject classifications. 65K10, 65D25, 65D18, 68U05

DOI. 10.1137/17M115743X

1. Introduction.

Triangulated surfaces are used in a variety of fields, such as computer graphics [5], computer-aided design [2], computer vision [8], and many others [32, 31, 33]. Triangulated surfaces are usually generated by some digital scanner devices or triangulation algorithms [38]. However, even with high-fidelity scanners, the scanning process inevitably produces noise due to local measurement errors [29]. Such noise affects the quality of surfaces and usually causes errors in downstream geometry applications, such as surface reconstruction, segmentation, and visualization [55]. Thus, how to effectively remove noise to recover high quality surfaces is one of the most fundamental tasks in geometry processing. In practice, it is difficult to distinguish noise and sharp features as they are of high frequency information. Meanwhile, it is also important to preserve smooth regions such as quadratic patches. Therefore, it is still quite challenging to remove noise while preserving sharp features and smoothly curved regions.

*Submitted to the journal's Computational Methods in Science and Engineering section November 20, 2017; accepted for publication (in revised form) October 8, 2018; published electronically January 10, 2019.

http://www.siam.org/journals/sisc/41-1/M115743.html

Funding: The first author's work was supported by NSF of China (61702467). The second author's work was supported in part by NSF DMS-1522645 and NSF CAREER award (DMS-1752934). The fourth author's work was supported by NSF of China (11301289, 11531013, and 11871035), Recruitment Programm of Global Young Expert, and the Fundamental Research Funds for the Central Universities.

†National Engineering Research Center of Geographic Information System, China University of Geosciences, Wuhan 430074, China (liu.zheng.jojo@gmail.com).

‡Department of Mathematics, Rensselaer Polytechnic Institute, Troy, NY 12180 (lair@rpi.edu).

§School of Computer Science and Software, Tianjin Polytechnic University, Tianjin 300384, China (zhy101@mail.ustc.edu.cn).

¶Corresponding author. School of Mathematical Sciences, Nankai University, Tianjin 300071, China (wucl@nankai.edu.cn).

Filtering schemes, which can be roughly classified into two categories (isotropic and anisotropic methods), are widely applied in surface denoising. The isotropic methods [24, 53, 18, 42] are classical and simple, among which Laplacian smoothing [24] is typical. Laplacian smoothing is the process of reducing the surface area. It smoothes the surface to remove the noise without considering surface geometric features. Thus, it, as well as other isotropic methods, suffers surface shrinkage and blurs geometric features. Later, a variety of anisotropic methods [16, 19, 1, 25, 30, 67, 50] were proposed to provide geometric features preservation. Compared to the isotropic methods, the anisotropic methods are more effective for preserving geometric features. However, when the noise level increases, the anisotropic methods usually fail to produce satisfactory results. Especially, this drawback is more severe for surfaces containing sharp features.

Variational methods are another kind of technique for triangulated surface denoising proposed recently. To keep the sharp features, the variational models use sparsity regularization terms. Inspired by the great success of total variation (TV) regularization in image processing [46], several researchers extended it to triangluated surface denoising. The authors of [23] presented an analogue of TV by minimizing the absolute value of Gauss curvature. Very recently, Zhang et al. proposed in [64] a vectorial TV-based model on a face normal field over triangulated surfaces. This method achieved impressive results for preserving sharp features. Another sparsity regularization is ℓ_0 quasi-norm. Indeed, He and Schaefer [27] extended ℓ_0 minimization [59] to triangulated surfaces for preserving sharp features. These methods achieve impressive results for surfaces consisting of flat regions and sharp features, e.g., polyhedron surfaces. However, if a surface has smoothly curved regions, they tend to flatten the smooth regions. The reason is the staircase effect of the sparsity regularization in the gradient field. The staircase effect of TV in image processing has been studied both from theoretical and experimental points in previous works; see [12, 13, 39, 40, 41] and references therein. To overcome this disadvantage of TV, high order PDEs [48, 39, 28, 3, 35] and combination methods of TV and high order models [12, 63, 13, 39, 40, 28, 14, 45] have been used in the image processing community. However, to the best of our knowledge, very few of the high order models or combinations are known over triangulated surfaces.

Wavelet frame methods have been successfully applied in image restoration [9, 10]. Recently, Dong et al. [22, 21, 20] extended the wavelet frame methods to triangulated surfaces. Their tight wavelet frame systems are potentially effective in many geometry applications, such as denoising and semisupervised clustering. Especially, for surface denoising, Dong et al. [21] proposed multiscale representation of surfaces using wavelet frames, which can achieve impressive denoising results for piecewise smooth surfaces with multiscale details. Yang and Wang [61] proposed a wavelet frame based variational model in [61]. Their method can effectively remove mixed Gaussian and impulse noise for the $\ell_1 + \ell_2$ fidelity term of their model. However, the existing wavelet frame based methods have difficulty recovering surfaces consisting of sharp features.

Among the methods mentioned above, there are some methods belonging to two-stage methods, i.e., face normal filtering followed by updating vertices [54, 60, 49, 36, 51, 52, 67, 64, 65]. The difference between these two-stage methods is in their normal filtering strategies, e.g., a mean and median normal filter was applied in [60, 51] to adopted trimmed quadratic weights for averaging the normals, while Zhang et al. in [64] used a TV based model to filter a face normal field. All of the normal filtering strategies can either deal with smooth regions or sharp features well. Moreover, all of

these two-stage methods use almost the same vertex updating model, which originated from Taubin [54] and has a beautiful implementation by Sun et al. [51]. When the noise level is low, the approach by Sun et al. [51] can achieve good results. However, when the noise level increases, the recovered vertex positions deviate far from those of the clean surface. In this situation, the method of Sun et al. [51] suffers producing frequent foldovers. Moreover, large scale noise in random directions make the matter even worse. This is due to the method [51] neglecting the orientations of triangle face normals, which leads to the vertex updating ambiguity problem; see section 5 for an explanation of this ambiguity.

As we can see, the aforementioned surface denoising schemes, including the filtering, variational, and wavelet frame methods, can properly handle either smooth regions or sharp features separately. However, it is still quite challenging to handle simultaneously smooth regions and sharp features well. In this paper, we propose a high order regularization model by introducing a new second order difference operator over triangulated surfaces. The proposed model with a well-designed weighting function is applied to the surface face normal field, which has a crucial advantage in handling surfaces consisting of both smooth regions and sharp features. It preserves sharp features well and substantially suppresses the staircase effect. It is numerically solved by the operator splitting and augmented Lagrangian method. The weighting function enhances the sparsity of the proposed high order model and is implemented by a dynamic weights strategy. After restoring the face normals, the surface vertices should be updated to match the filtered face normals. Last but not least, a new vertex updating method is presented. Compared to the traditional vertex updating method [51], our new method can eliminate ambiguities and reconstruct much better triangulated surfaces. To summarize, the contributions of the paper are listed as follows:

- We introduce a new second order difference operator and its adjoint operator in piecewise constant function space over triangulated surfaces. To the best of our knowledge, this second order operator is the first defined over triangulated surfaces.
- We introduce a novel normal filtering model using the second order regularization with a well-designed weighting function, which can preserve sharp features and simultaneously prevent the staircase effect in smooth regions.
- We propose a new vertex updating method to recover surface vertices. The proposed method significantly reduces foldovers compared to the existing vertex updating methods.

The rest of this paper is organized as follows. In section 2, we briefly review TV based models in image processing and reweighted ℓ_1 minimization. Section 3 provides the definitions of a new second order difference operator and two high order regularization models in piecewise constant function spaces. The differences of this second order operator and a Laplace-like operator are discussed at the end of section 3. In section 4, we present a high order regularization normal filtering model with a well-designed weighting function. An augmented Lagrangian method is applied to solve the variational model with a dynamic weights strategy. In section 5, a new vertex updating method is introduced for recovering the vertex positions with respect to the filtered face normals. Our two-stage denoising method is discussed and compared to typical existing methods both qualitatively and quantitatively in section 6. Section 7 concludes the paper.

2. TV based models and reweighted ℓ_1 minimization. In this section, we present TV based models and reweighted ℓ_1 minimization, since they are closely related to our approach.

2.1. TV, vectorial TV, and high order models for images. Since the pioneering work of Rudin, Osher, and Fatemi [46], TV has been proven very successful in image processing for its excellent edge-preserving property [46, 39, 40, 35]. The TV denoising model (ROF) aims at solving

$$\min_u \int_\Omega |\nabla u| + \frac{\alpha}{2}\int_\Omega (u-f)^2, \tag{1}$$

where f is an observed noisy image, $\int_\Omega |\nabla u|$ is the TV regularization, and α is a positive fidelity parameter. For $\mathfrak{N}$-channel images $\mathbf{u}, \mathbf{f} : \Omega \to \mathbb{R}^{\mathfrak{N}}$, where $\mathbf{u} = (u_1, u_2, \ldots, u_{\mathfrak{N}})$ and $\mathbf{f} = (f_1, f_2, \ldots, f_{\mathfrak{N}})$, the model (1) can be naturally extended to its vectorial version as follows:

$$\min_{\mathbf{u}} \int_\Omega \left(\sum_{i=1}^{\mathfrak{N}} |\nabla u_i|^2 \right)^{\frac{1}{2}} + \frac{\alpha}{2}\int_\Omega |\mathbf{u}-\mathbf{f}|^2. \tag{2}$$

This model can be used for multispectral image processing with a special case $\mathfrak{N} = 3$ for color image processing. The regularization of model (2), referred to as vectorial TV, has been discussed in [47, 4, 15, 7]. Both of the objective functionals are coercive, proper, continuous, and strictly convex. Thus, problems (1) and (2) have, respectively, a unique minimizer.

A well-known drawback of the above TV and vectorial TV models is the staircase effect [12, 13, 39, 40]. To overcome this, high order models such as the Lysaker–Lundervold–Tai (LLT) model [39] and the total generalized variation (TGV) model [6] have been studied [48, 39, 28, 3, 6]. The idea is essentially to introduce high order derivatives to the energy regularization. High order models in general perform well in recovering smooth regions, but they cannot compete with TV in dealing with discontinuous edges. A natural solution is to combine TV and high order models [12, 63, 13, 39, 40, 28, 14, 45]. For examples, Lysaker and Tai [40] used a convex combination of TV with LLT [39]. In [13], Chan, Marquina, and Mulet presented a model combining a TV term with a weighted Laplacian term to reduce the staircase effect while preserving sharp edges. A model using infimal-convolution of the TV and high order term was proposed by Chambolle and Lions in [12], in which the TV term was used to keep sharp edges while the high order term preserves smooth regions. The key to these methods is to balance the contribution of the TV and high order term. The balance is usually implemented by a weighting parameter or function, which needs to be tuned carefully.

Almost all these variational models, after discretization over image pixel grids, penalize the sparsity of various orders of differences of the image by ℓ_1 related penalization functions.

2.2. Reweighted ℓ_1 minimization. The reweighted ℓ_1 minimization was first presented by Candès, Wakin, and Boyd in [11] to enhance the sparsity in sparse signal recovery. It outperforms ℓ_1 minimization in situations where substantially fewer measurements are used to recover a signal.

The key of the reweighted ℓ_1 minimization is to solve a sequence of weighted ℓ_1 minimization problems,

$$x^{(k)} = \arg\min_{x\in\mathbb{R}^n} \|W^{(k)}x\|_1 \quad \text{s.t.} \quad Ax = b, \tag{3}$$

where $W^{(k)} = \mathrm{diag}(w_1^k, \ldots, w_n^k)$ is updated according to $x^{(k-1)}$. Although there are a variety of reweighted ℓ_1 algorithms proposed to update the weights [11, 56, 66], as a rough rule of thumb the weights should be inversely proportional to signal magnitudes [11]. For example, the reweighted method proposed by Candès, Wakin, and Boyd in [11] is as follows:

$$w_i^k = \frac{1}{|x_i^{k-1}| + \epsilon}, \ i = 1, \ldots, n \quad \text{for } \epsilon > 0.$$

3. Discrete high order regularization models in piecewise constant function spaces over triangulated surfaces. In this section, we introduce some notation followed by definitions of piecewise constant function spaces and difference operators over triangulated surfaces. The discrete high order models in piecewise constant spaces are presented and discussed.

3.1. Notation. Let M be a compact triangulated surface of arbitrary topology with no degenerate triangles in $\mathbb{R}^3$. The set of vertices, edges, and triangles of M are denoted as $\{v_i : i = 0, 1, \ldots, \mathrm{V} - 1\}$, $\{e_i : i = 0, 1, \ldots, \mathrm{E} - 1\}$, and $\{\tau_i : i = 0, 1, \ldots, \mathrm{T} - 1\}$, respectively. Here V, E, and T are the numbers of vertices, edges, and triangles of M, respectively. If v is an endpoint of an edge e, then we write it as $v \prec e$. Similarly, $e \prec \tau$ denotes that e is an edge of a triangle τ; $v \prec \tau$ denotes that v is a vertex of a triangle τ.

Denote the 1-ring of the triangle τ_i as $D_1(\tau_i)$, which is the set of the triangles sharing some common edges with τ_i indicated as green triangles in Figure 1(a). Let $B_1(\tau_i) = \{l_{j,\tau_i} : i = 0, 1, \ldots, \mathrm{T} - 1; j = 0, 1, 2\}$ be the set of lines connecting the barycenter and vertices of τ_i, where j counterclockwise marks the vertex contained in τ_i. Namely, l_{j,τ_i} is the line connecting the barycenter of τ_i and the vertex marked as j in τ_i. Let $B_2(\tau_i)$ be the set of lines connecting vertices of τ_i and barycenters of triangles in $D_1(\tau_i)$, indicated as blue lines in Figure 1(b). Write the 1-disk of the vertex v_i as $M_1(v_i)$ denoting the indices of triangles containing v_i, indicated as gray triangles in Figure 1(c). We write the 1-neighborhood of vertex v_i as $N_1(v_i)$, which is the set of vertices connecting to v_i, indicated as orange vertices in Figure 1(d).

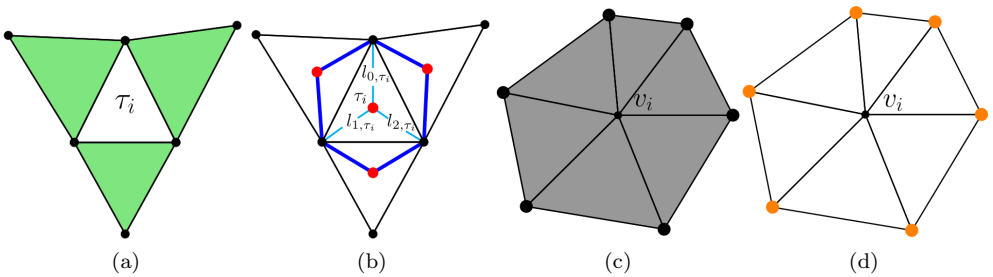


FIG. 1. *The illustration of $D_1(\tau_i), B_1(\tau_i), B_2(\tau_i), M_1(v_i)$, and $N_1(v_i)$. The elements contained in $D_1(\tau_i), B_1(\tau_i), B_2(\tau_i), M_1(v_i)$, and $N_1(v_i)$ are plotted in* green, cyan, blue, gray, *and* orange, *respectively.* (a) *$D_1(\tau_i)$ is the* 1*-ring of the triangle τ_i, which refers to three triangles;* (b) *$B_1(\tau_i)$ is the set of lines connecting the barycenter and vertices of τ_i, which refers to three lines, and $B_2(\tau_i)$ is the set of lines connecting vertices of τ_i and barycenters of triangles contained in $D_1(\tau_i)$, which refers to six lines;* (c) *$M_1(v_i)$ is the* 1*-disk of the vertex v_i, which refers to six triangles;* (d) *$N_1(v_i)$ is the* 1*-neighborhood of v_i, which refers to six vertices. (Figure is in color online.)*

We further introduce the relative orientation of an edge e to a triangle τ, which is denoted by $\mathrm{sgn}(e,\tau)$, as follows. First, we assume that all triangles are with counterclockwise orientation and all edges are with randomly chosen fixed orientations. For an edge $e \prec \tau$, if the orientation of e is consistent with the orientation of τ, then $\mathrm{sgn}(e,\tau) = 1$; otherwise, $\mathrm{sgn}(e,\tau) = -1$.

3.2. Piecewise constant function spaces and operators. To describe the piecewise constant data field, we present the concept of piecewise constant function space. Compared to piecewise linear function space which is suitable to deal with vertex-based problems, we find that, for feature preserving geometry processing [64, 37], the piecewise constant function space is more suitable, which is related to a piecewise constant finite element method in numerical PDE. For normal-based triangulated surface denoising, the piecewise linear function space requires the input to be vertex normals, while the input of the piecewise constant function space is face normals. The vertex normals are averaged from face normals. The second order geometry information of this smoothed vertex normal field is much less sparse than that of the face normal field. Thus, it is more appropriate to discretize our high order regularization model in the piecewise constant function space for preserving sharp features. We should point out that, over triangulated surfaces, the second order difference operator is newly defined in this paper.

We denote the space $V_M = \mathbb{R}^{\mathrm{T}}$, which is isomorphic to the piecewise constant function space over M. $u = (u_0, u_1, \ldots, u_{\mathrm{T}-1}) \in V_M$ means that the value of u restricted on the triangle τ is u_τ, which is written as $u|_\tau$ sometimes. The inner product and norm in V_M are as follows:

$$(u^1, u^2)_{V_M} = \sum_\tau u^1|_\tau u^2|_\tau s_\tau, \quad \|u\|_{V_M} = \sqrt{(u,u)_{V_M}} \quad \forall u^1, u^2, u \in V_M, \tag{4}$$

where s_τ is the area of triangle τ. For any $u \in V_M$, the jump of u over an edge e is

$$[u]_e = \begin{cases} \sum\limits_{\tau, e\prec\tau} u|_\tau \mathrm{sgn}(e,\tau), & e \not\subset \partial M, \\ 0, & e \subset \partial M. \end{cases} \tag{5}$$

It is then natural to define the first order difference operator [64, 37] by

$$\delta : V_M \to Q_M, u \mapsto \delta u, \quad \delta u|_e = [u]_e \quad \forall e \text{ for } u \in V_M, \tag{6}$$

where $Q_M = \mathbb{R}^{\mathrm{E}}$ is the range of δ. The Q_M space is equipped with the following inner product and norm:

$$(q^1, q^2)_{Q_M} = \sum_e q^1|_e q^2|_e \mathrm{len}(e), \quad \|q\|_{Q_M} = \sqrt{(q,q)_{Q_M}} \tag{7}$$

for $q^1, q^2, q \in Q_M$, where $\mathrm{len}(e)$ is the length of the edge e.

It is straightforward to derive the adjoint operator of δ, namely $\delta^\star : Q_M \to V_M, q \mapsto \delta^\star q$, using the above inner products in V_M and Q_M. For $q \in Q_M$, $\delta^\star q$ is given by

$$(\delta^\star q)|_\tau = \frac{1}{s_\tau} \sum_{\substack{e\prec\tau, \\ e\not\subset\partial M}} q|_e \mathrm{sgn}(e,\tau)\mathrm{len}(e) \quad \forall \tau. \tag{8}$$

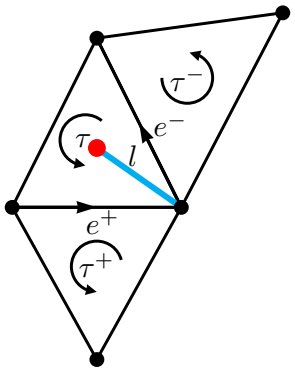


FIG. 2. *The illustration of* $[[u]]_l$ *over the line* l *plotted in* cyan *in triangle* τ *with the barycenter plotted in* red. *(Figure is in color online.)*

In the following section, we define a second order difference operator which will be used to construct our high order regularization models. Let l be a line connecting the barycenter and one vertex of the triangle τ. The two edges of triangle τ, which share the common vertex of l, are denoted as e^+ and e^-, respectively. The two triangles, contained in $D_1(\tau_i)$ and sharing these two edges, are denoted as τ^+ and τ^-, respectively. All aforementioned descriptions are indicated in Figure 2. Then, we define $[[u]]_l$ over the line l in τ as

$$\begin{aligned}
[[u]]_l &= [u]_{e^+}\mathrm{sgn}(e^+,\tau^+) + [u]_{e^-}\mathrm{sgn}(e^-,\tau^-)\\
&= \big((u_\tau \mathrm{sgn}(e^+,\tau) + u_{\tau^+}\mathrm{sgn}(e^+,\tau^+))\mathrm{sgn}(e^+,\tau^+)\big)\\
&\quad + \big((u_\tau \mathrm{sgn}(e^-,\tau) + u_{\tau^-}\mathrm{sgn}(e^-,\tau^-))\mathrm{sgn}(e^-,\tau^-)\big)\\
&= \big(u_{\tau^+} - u_\tau\big) + \big(u_{\tau^-} - u_\tau\big)\\
&= u_{\tau^+} + u_{\tau^-} - 2u_\tau,
\end{aligned} \tag{9}$$

which is written as $[[u]]_{l,\tau}$ sometimes. For simplicity, with Neumann boundary condition, we define, for any $u \in V_M$,

$$[[u]]_l = \begin{cases} u_{\tau^+} + u_{\tau^-} - 2u_\tau, & e^+ \text{ and } e^- \not\subset \partial M,\\ 0, & e^+ \text{ or } e^- \subset \partial M. \end{cases} \tag{10}$$

From (9), we can obviously see that the definition of $[[u]]_l$ is invariant under the choice of the orientation of edge e.

Then, the second order difference operator is defined by

$$\delta^2 : V_M \to P_M, u \mapsto \delta^2 u, \quad (\delta^2 u)|_\tau = ([[u]]_{l_{0,\tau}}, [[u]]_{l_{1,\tau}}, [[u]]_{l_{2,\tau}}) \ \ \forall \tau, \text{ for } u \in V_M, \tag{11}$$

where $P_M = \mathbb{R}^{\mathrm{T}} \times \mathbb{R}^{\mathrm{T}} \times \mathbb{R}^{\mathrm{T}}$ is the range of δ^2. The P_M space is equipped with the following inner product and norm:

$$(p^1, p^2)_{P_M} = \sum_\tau \sum_{l \in B_1(\tau)} p^1|_l p^2|_l \mathrm{len}(l), \quad \|p\|_{P_M} = \sqrt{(p,p)_{P_M}} \tag{12}$$

for $p^1, p^2, p \in P_M$, where $\mathrm{len}(l)$ is the length of line l.

LEMMA 3.1. *The adjoint operator of δ^2, that $(\delta^2)^\star : P_M \to V_M, p \mapsto (\delta^2)^\star p$, has the following form:*

$$((\delta^2)^\star p)|_\tau = \frac{1}{s_\tau}\Big(\sum_{\substack{l\in B_2(\tau),\\ e^+ \text{ and } e^- \not\subset \partial M}} p|_l \mathrm{len}(l) - \sum_{\substack{l\in B_1(\tau),\\ e^+ \text{ and } e^- \not\subset \partial M}} 2p|_l \mathrm{len}(l)\Big) \quad \forall \tau, \text{ for } p \in P_M. \tag{13}$$

Proof. As the definition of adjoint operator, we have

$$\langle \delta^2 u, p\rangle_{P_M} = \langle u, (\delta^2)^\star p\rangle_{V_M}. \tag{14}$$

By using the inner products (12) and (4) in P_M and V_M, (14) can be reformulated as

$$\sum_\tau \sum_{l\in B_1(\tau)} [[u]]_l p|_l \mathrm{len}(l) = \sum_\tau u_\tau ((\delta^2)^\star p)|_\tau s_\tau. \tag{15}$$

By using (10), the left-hand side of (15) is actually

$$\begin{aligned}\sum_\tau \sum_{l\in B_1(\tau)} [[u]]_l p|_l \mathrm{len}(l) &= \sum_\tau \Big(\sum_{\substack{l\in B_1(\tau),\\ e^+ \text{ and } e^- \not\subset \partial M}} (u_{\tau^+} + u_{\tau^-} - 2u_\tau) p|_l \mathrm{len}(l)\Big)\\ &= \sum_\tau \Big(\sum_{\substack{l\in B_2(\tau),\\ e^+ \text{ and } e^- \not\subset \partial M}} u_\tau p|_l \mathrm{len}(l) - \sum_{\substack{l\in B_1(\tau),\\ e^+ \text{ and } e^- \not\subset \partial M}} 2u_\tau p|_l \mathrm{len}(l)\Big)\\ &= \sum_\tau u_\tau \Big(\sum_{\substack{l\in B_2(\tau),\\ e^+ \text{ and } e^- \not\subset \partial M}} p|_l \mathrm{len}(l) - \sum_{\substack{l\in B_1(\tau),\\ e^+ \text{ and } e^- \not\subset \partial M}} 2p|_l \mathrm{len}(l)\Big).\end{aligned}$$

Therefore, we have

$$\sum_\tau u_\tau \Big(\sum_{\substack{l\in B_2(\tau),\\ e^+ \text{ and } e^- \not\subset \partial M}} p|_l \mathrm{len}(l) - \sum_{\substack{l\in B_1(\tau),\\ e^+ \text{ and } e^- \not\subset \partial M}} 2p|_l \mathrm{len}(l)\Big) = \sum_\tau u_\tau \big(((\delta^2)^\star p)|_\tau s_\tau\big).$$

Then, the assertion follows immediately. ∎

To handle vectorial data, we extend the above concepts to vectorial cases. Two vectorial spaces $\mathbf{V}_M, \mathbf{P}_M$ are as follows:

$$\mathbf{V}_M = \underbrace{V_M \times \cdots \times V_M}_{\mathfrak{N}}, \quad \mathbf{P}_M = \underbrace{P_M \times \cdots \times P_M}_{\mathfrak{N}}$$

for $\mathfrak{N}$-channel data. The inner products and norms in $\mathbf{V}_M$ and $\mathbf{P}_M$ are as follows:

$$(\mathbf{u}^1, \mathbf{u}^2)_{\mathbf{V}_M} = \sum_{1\le i\le \mathfrak{N}} (u_i^1, u_i^2)_{V_M}, \quad \|\mathbf{u}\|_{\mathbf{V}_M} = \sqrt{(\mathbf{u},\mathbf{u})_{\mathbf{V}_M}}, \mathbf{u}^1, \mathbf{u}^2, \mathbf{u} \in \mathbf{V}_M,$$

$$(\mathbf{p}^1, \mathbf{p}^2)_{\mathbf{P}_M} = \sum_{1\le i\le \mathfrak{N}} (p_i^1, p_i^2)_{P_M}, \quad \|\mathbf{p}\|_{\mathbf{P}_M} = \sqrt{(\mathbf{p},\mathbf{p})_{\mathbf{P}_M}}, \mathbf{p}^1, \mathbf{p}^2, \mathbf{p} \in \mathbf{P}_M.$$

We mention that $\delta\mathbf{u}, \delta^2\mathbf{u}$ and their adjoint operators can be computed channel by channel.

3.3. Two discrete high order models over triangulated surfaces. Assume $f \in V_M$ to be an observed noisy scalar field on M. A high order regularization model reads

$$\min_{u\in V_M}\left\{E_{\mathrm{ho}}(u) = R_{\mathrm{ho}}(\delta^2 u) + \frac{\alpha}{2}\|u-f\|^2_{V_M}\right\}, \tag{16}$$

where

$$R_{\mathrm{ho}}(\delta^2 u) = \sum_{\tau}\sum_{l\in B_1(\tau)} \big|[[u]]_l\big|\mathrm{len}(l) = \sum_{l}\big|[[u]]_l\big|\mathrm{len}(l)$$

is the second order variation of u, and α is a positive fidelity parameter. The minimization problem (16) has a unique minimizer under the assumption of no degenerate triangles on M.

Our second discrete high order regularization model is for vector field denoising over surfaces. Suppose $\mathbf{f} = (f_1, f_2, \ldots, f_{\mathfrak{N}})$ be an observed noisy vector field. The vectorial version of (16) reads

$$\min_{\mathbf{u}\in \mathbf{V}_M}\left\{E_{\mathrm{vho}}(\mathbf{u}) = R_{\mathrm{vho}}(\delta^2\mathbf{u}) + \frac{\alpha}{2}\|\mathbf{u}-\mathbf{f}\|^2_{\mathbf{V}_M}\right\}, \tag{17}$$

where

$$R_{\mathrm{vho}}(\delta^2\mathbf{u}) = \sum_{\tau}\sum_{l\in B_1(\tau)}\left(\sum_{i=1}^{\mathfrak{N}}\big|[[u_i]]_l\big|^2\right)^{\frac{1}{2}}\mathrm{len}(l) = \sum_{l}\left(\sum_{i=1}^{\mathfrak{N}}\big|[[u_i]]_l\big|^2\right)^{\frac{1}{2}}\mathrm{len}(l)$$

is the vectorial high order seminorm.

3.4. A discussion on the second order difference and Laplace-like operator. The Laplacian is the mostly frequently used high order operator in geometry processing. Of course, it can also be used to construct a high order regularization model. In this subsection, we discuss differences between the second order difference operator (11) and Laplace-like operator in piecewise constant function space over triangulated surfaces. The high order regularization models using these two high order operators are also compared.

For clarity, we first give the discretization of a Laplace-like operator in piecewise constant function space. By using the first order difference operator (6) and its adjoint operator (8), the Laplace-like operator $\delta^\star\delta : V_M \to V_M, u \mapsto \delta^\star\delta u$, can be derived as

$$(\delta^\star\delta u)|_\tau = \frac{1}{s_\tau}\sum_{\substack{e\prec\tau,\tau\cap\tau_i=e,\\ e\not\subset\partial M}}(u_\tau - u_{\tau_i})\mathrm{len}(e) \quad \forall\tau. \tag{18}$$

For a noisy scalar field, an ℓ_1-norm Laplacian-like regularization model reads

$$\min_{u\in V_M}\left\{E_{\mathrm{lap}}(u) = R_{\mathrm{lap}}(\delta^\star\delta u) + \frac{\alpha}{2}\|u-f\|^2_{V_M}\right\}, \tag{19}$$

where the regularization term is defined as

$$R_{\mathrm{lap}}(\delta^\star\delta u) = \sum_{\tau}\big|(\delta^\star\delta u)|_\tau\big|s_\tau.$$

As we can see, discretizations of the second order operator (11) and Laplace-like operator (18) are totally different. Our second order operator can be seen as a set of second order central differences defined over l along three different directions in one triangle. For uniformly sampled triangulated surfaces, the second order regularization term of high order model (16) is an analogue of the regularization term that depicts

$$\int_\Omega |u_{xx}| + |u_{yy}| dxdy$$

in a two-dimensional (2D) domain, while the Laplacian-like regularization term of model (19) can be seen as an analogue to

$$\int_\Omega |u_{xx} + u_{yy}| dxdy.$$

The regularization term using second order operator (11) can be computed separately in different directions, while that using the Laplace-like operator (18) cannot. Compared to the ℓ_1-norm Laplacian-like model (19), the second order regularization model (16) is more effective for recovering sharp signals over triangulated surfaces; see the comparison of vectorial implementations of these two models in Figure 13. Moreover, the second order regularization model overcomes the staircase effect introduced by first order models.

4. Normal filtering using high order model with dynamic weights. The recent TV [64] and ℓ_0 [27] based minimization methods use the concept of sparsity of first order information to remove noise from triangulated surfaces. These methods preserve sharp features well, but suffer from the staircase effect in smooth regions inevitably. In this section, a high order normal filtering model with dynamic weighs is proposed for preserving sharp features and removing the staircase effect in smooth regions simultaneously. The dynamic weights are applied in the proposed model to significantly improve effectiveness for preserving sharp features.

4.1. High order normal filtering model with dynamic weights. For a given noisy surface M^{in}, we write the face normals as $\mathbf{N}^{in}$. To remove noise in $\mathbf{N}^{in}$ through our vectorial high order model (17) with multiple spherical constraints, we propose the following variational model:

$$\min_{\mathbf{N}\in C_{\mathbf{N}}} \left\{ E(\mathbf{N}) = R_{\text{vhow}}(\delta^2 \mathbf{N}) + \frac{\alpha}{2} \left\| \mathbf{N} - \mathbf{N}^{in} \right\|^2_{\mathbf{V}_M} \right\}, \tag{20}$$

where

$$C_{\mathbf{N}} = \{\mathbf{N} \in \mathbf{V}_M : \|\mathbf{N}_\tau\| = 1 \ \forall \tau\},$$
$$R_{\text{vhow}}(\delta^2 \mathbf{N}) = \sum_l w_l \left(\sum_{i=1}^{3} (\delta^2 N_i|_l)^2 \right)^{\frac{1}{2}} \text{len}(l).$$

Note that $\mathbf{V}_M$ denotes 3-channel V_M here. The dynamic weight w_l on each l of triangle is defined as

$$w_l = \exp(-\|\mathbf{N}_{\tau^+} + \mathbf{N}_{\tau^-} - 2\mathbf{N}_\tau\|^4). \tag{21}$$

See Figure 2 for the relations of l, τ^+, τ^-, and τ in (21). This function is expected to be large when the norm of second order difference defined on l is small, and vice

versa. Thus, it can offer large weights to smooth regions (smoothly curved regions and flat regions), and small weights to sharp features, and therefore smoothes nonfeatures regions and preserves sharp features.

For most surfaces, the proposed vectorial high order regularization model (17) can achieve good denoising results. However, in rare cases, where the noise level is increased, the proposed model (17) may smooth some sharp features a little. Thus, we use the dynamic weights, updated with respect to the face normals in each iteration, to enhance the sparsity of our high order model for improving sharp features reconstruction. The dynamic weights scheme is inspired by Candès, Wakin, and Boyd in [11]. These dynamic weights penalize smooth regions (smoothly curved regions and flat regions) more than sharp features, which can be applied to achieve the lower-than-ℓ_1-sparsity effect. In general, the combination of the high order model and the dynamic weights is able to preserve sharp features well and at the same time recover smooth regions without staircase effects.

4.2. Augmented Lagrangian method for solving the high order normal filtering model. It is challenging to solve the normal filtering model (20) due to the nondifferentiability and nonlinear constraints. Recently, the variable splitting and augmented Lagrangian method (ALM) have attracted intensive attention for their efficiency in many ℓ_1 related optimization problems [43, 44, 58]. Hence, we introduce an auxiliary variable and use ALM to handle the regularization term of (20). Moreover, in each iteration of ALM, the weight (21) is updated dynamically.

We first introduce a new variable $\mathbf{p} \in \mathbf{P}_M$ and rewrite the problem (20) as

$$\min_{\mathbf{N}\in\mathbf{V}_M,\mathbf{p}\in\mathbf{P}_M} \left\{ R_{\text{vhow}}(\mathbf{p}) + \frac{\alpha}{2}\left\|\mathbf{N}-\mathbf{N}^{in}\right\|^2_{\mathbf{V}_M} + \sigma_{C_{\mathbf{N}}}(\mathbf{N}) \right\} \quad \text{s.t.} \quad \mathbf{p} = \delta^2\mathbf{N}, \tag{22}$$

where

$$\sigma_{C_{\mathbf{N}}}(\mathbf{N}) = \begin{cases} 0, & \mathbf{N} \in C_{\mathbf{N}}, \\ +\infty, & \mathbf{N} \notin C_{\mathbf{N}}. \end{cases}$$

Accordingly, we define the following augmented Lagrangian function:

$$\begin{aligned} L(\mathbf{N},\mathbf{p};\lambda_{\mathbf{p}}) = {} & R_{\text{vhow}}(\mathbf{p}) + \frac{\alpha}{2}\left\|\mathbf{N}-\mathbf{N}^{in}\right\|^2_{\mathbf{V}_M} + \sigma_{C_{\mathbf{N}}}(\mathbf{N}) \\ & + (\lambda_{\mathbf{p}}, \mathbf{p}-\delta^2\mathbf{N})_{\mathbf{P}_M} + \frac{r_{\mathbf{p}}}{2}\|\mathbf{p}-\delta^2\mathbf{N}\|^2_{\mathbf{P}_M}, \end{aligned} \tag{23}$$

where $\lambda_{\mathbf{p}}$ is a Lagrange multiplier and $r_{\mathbf{p}}$ is a positive real number. This primal variables update procedure can be separated into two subproblems:

- The $\mathbf{N}$-sub problem: given $\mathbf{p}$

$$\min_{\mathbf{N}\in\mathbf{V}_M} \frac{\alpha}{2}\|\mathbf{N}-\mathbf{N}^{in}\|^2_{\mathbf{V}_M} + (\lambda_{\mathbf{p}}, -\delta^2\mathbf{N})_{\mathbf{P}_M} + \frac{r_{\mathbf{p}}}{2}\|\mathbf{p}-\delta^2\mathbf{N}\|^2_{\mathbf{P}_M} + \sigma_{C_{\mathbf{N}}}(\mathbf{N}). \tag{24}$$

- The $\mathbf{p}$-sub problem: given $\mathbf{N}$

$$\min_{\mathbf{p}\in\mathbf{P}_M} R_{\text{vhow}}(\mathbf{p}) + (\lambda_{\mathbf{p}}, \mathbf{p})_{\mathbf{P}_M} + \frac{r_{\mathbf{p}}}{2}\|\mathbf{p}-\delta^2\mathbf{N}\|^2_{\mathbf{P}_M}. \tag{25}$$

The **N**-sub problem is a quadratic minimization with orthogonality constraints. It is challenging to find the exact solution of the **N**-sub problem due to the nonlinear and nonconvex constraints. Although some iterative methods [26, 57, 34] were proposed to efficiently approximate the exact solution of this type of problem, it is not possible to get the exact solution in finite precision arithmetic. Moreover, it is never practical to iterate for a long time. Fortunately, due to error forgetting and cancellation properties of ALM for the ℓ_1 minimization problem [62], the suboptimization problem (24) does not have to be solved very accurately. Here we adopt an approximate strategy to balance the precision and computational efficiency. We first ignore $\sigma_{C_{\mathbf{N}}}(\mathbf{N})$ and solve a quadratic programming and then project the minimizer onto a unit sphere. The quadratic problem (without constraints) has the first order optimality condition

$$r_{\mathbf{p}}((\delta^2)^\star\delta^2\mathbf{N}) + \alpha\mathbf{N} = r_{\mathbf{p}}((\delta^2)^\star\mathbf{p}) + (\delta^2)^\star\lambda_{\mathbf{p}} + \alpha\mathbf{N}^{in}. \tag{26}$$

This equation can be reformulated into a sparse and positive semidefinite linear system, which can be solved by various well-developed numerical packages. Here we use the conjugate gradient (CG) method to solve the problem. The maximum number of iterations and the tolerance threshold of CG method are empirically set to be 10 and 1e-2, respectively. The iteration procedure terminates when one of these two stopping criteria is satisfied. Then, we directly project the solution onto the unit sphere.

Next, we solve the **p**-sub problem (25). By (12), this problem can be written as

$$\min_{\mathbf{p}} \sum_l w_l|\mathbf{p}_l|\mathrm{len}(l) + \sum_l (\lambda_{p_l}, \mathbf{p}_l)\mathrm{len}(l) + \sum_l \frac{r_{\mathbf{p}}}{2}|\mathbf{p}_l - (\delta^2\mathbf{N})|_l|^2\mathrm{len}(l). \tag{27}$$

The problem (27) is decoupled and can be solved line-by-line. For each line l connecting the barycenter and one vertex of one triangle, we need to solve

$$\min_{\mathbf{p}_l} w_l|\mathbf{p}_l| + (\lambda_{p_l}, \mathbf{p}_l) + \frac{r_{\mathbf{p}}}{2}|\mathbf{p}_l - (\delta^2\mathbf{N})|_l|^2,$$

which has a closed form solution

$$\mathbf{p}_l = \begin{cases} (1 - \frac{w_l}{r_{\mathbf{p}}|\xi_l|})\xi_l, & |\xi_l| > \frac{w_l}{r}, \\ 0, & |\xi_l| \le \frac{w_l}{r}, \end{cases} \tag{28}$$

with

$$\xi = \delta^2\mathbf{N} - \frac{\lambda_{\mathbf{p}}}{r_{\mathbf{p}}}.$$

In summary, the algorithm of high order normal filtering model (20) is given in Algorithm 1. Based on the variable splitting and ALM, this algorithm solves the nondifferentiability problem with nonconvex constraints by iterating several simple operations. We should point out that, in the conventional reweighted ℓ_1 minimization (3), the minimization problem with fixed weights is usually solved exactly. Therefore, the reweighted strategy is time-consuming. In contrast, Algorithm 1 updates the weights in each iteration. It can be regarded as an inexact but more efficient version of the conventional reweighted minimization algorithm. Although we currently cannot give a rigorous proof of convergence for Algorithm 1, our numerical experiments strongly validate it in practice. A theoretical analysis of this algorithm is worthy of future research.

Algorithm 1 Augmented Lagrangian Method for High Order Normal Filtering with Dynamic Weights.

Initialization: $\lambda_{\mathbf{p}}^0 = 0$, $\mathbf{N}^{-1} = 0$, $\mathbf{P}^{-1} = 0$, $k = 0$, $K = 100$, $\epsilon = 1e-4$
do
  1. Compute $\mathbf{N}^k$ from (26) for fixed $(\lambda_{\mathbf{p}}^k, \mathbf{p}^{k-1})$; **Normalize** $\mathbf{N}^k$
  2. Compute $\mathbf{p}^k$ from (28) for fixed $(\lambda_{\mathbf{p}}^k, \mathbf{N}^k)$
  3. Update Lagrange multiplier $\lambda_{\mathbf{p}}^{k+1} = \lambda_{\mathbf{p}}^k + r_{\mathbf{p}}(\mathbf{p}^k - \delta^2 \mathbf{N}^k)$
  4. Update each weight w_l through (21) with respect to $\mathbf{N}^k$
while $\|\mathbf{N}^k - \mathbf{N}^{k-1}\|_{\mathbf{V}_M} < \epsilon$ or $k > K$
return $\mathbf{N}^k$

5. Ambiguity free vertex updating method. After restoring the face normal field by Algorithm 1, the positions of vertices need to be reconstructed to match the updated face normals. As mentioned in section 1, all of the existing two-stage methods [54, 60, 49, 36, 51, 52, 67, 64, 65] use the same vertex updating model,

$$\min_v \sum_\tau \sum_{(v_i, v_j) \in \tau} s_\tau (\mathbf{N}_\tau \cdot (v_i - v_j))^2, \tag{29}$$

where s_τ is the area and $\mathbf{N}_\tau$ is the filtered normal of τ. The gradient descent method is used to minimize this optimization problem, where its initialization is the restored face normal field. This optimization problem is to penalize the nonorthogonality between the filtered face normal and the three edges at each face over the surface. However, when a surface is corrupted by noise in random directions, the vertex updating method [51] usually produces foldings, even with the exact (ground truth) face normals. In addition, large scale noise makes this phenomenon even worse; see the second column of Figure 3. The reason is that the model (29) only penalizes the nonorthogonality and cannot distinguish between $-\mathbf{N}_\tau$ and $\mathbf{N}_\tau$. Thus, the model neglects the orientations of triangle face normals and leads to updating ambiguities. In other words, a vertex v_i of triangle τ may be updated along the direction $-\mathbf{N}_\tau$ instead of $\mathbf{N}_\tau$. These trianglewise orientation ambiguities cause inconsistent normal vectors crossing different triangles.

To address the orientation ambiguity problem, we propose a new vertex updating method, which reconstructs the surface from a given normal vector field by solving the following minimization problem:

$$\min_v \left\{ E(v) = -\sum_{\tau = (v_i, v_j, v_k)} s_\tau \mathbf{N}_\tau \cdot \left(\frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|} \right) + \frac{\eta}{2} ||v - v^{in}||^2 \right\}, \tag{30}$$

where (v_i, v_j, v_k) are vertices of τ with counterclockwise order, v^{in} is the noisy surface, and η is a small positive parameter. The first term of (30) is used to solve the orientation ambiguity problem. This term not only considers the orthogonality between the triangle face and its corresponding normal direction, but also takes into account the orientation of the face. Thus, compared to (29), the energy of model (30) poses no ambiguity. The second term of (30) is a fidelity term.

The partial derivatives of the energy $E(v)$ with respect to v_i is as follows:

$$
\begin{aligned}
&\nabla_{v_i} E(v) \\
&= -\sum_{\substack{\tau \in M_1(v_i),\\ \tau = (v_i, v_j, v_k)}} s_\tau \Bigg(\frac{\mathbf{N}_\tau \times (v_k - v_j)}{\|(v_j - v_i) \times (v_k - v_i)\|} + \frac{\mathbf{N}_\tau \times (v_k - v_j)}{\|(v_k - v_j) \times (v_i - v_j)\|} + \frac{\mathbf{N}_\tau \times (v_k - v_j)}{\|(v_i - v_k) \times (v_j - v_k)\|} \\
&\qquad - \frac{\mathbf{N}_\tau \cdot [(v_j - v_i) \times (v_k - v_j)]\,[(v_j - v_i) \times (v_k - v_i) \times (v_k - v_j)]}{\|(v_j - v_i) \times (v_k - v_i)\|^3} \\
&\qquad - \frac{\mathbf{N}_\tau \cdot [(v_k - v_j) \times (v_i - v_j)]\,[(v_k - v_j) \times (v_i - v_j) \times (v_k - v_j)]}{\|(v_k - v_j) \times (v_i - v_j)\|^3} \\
&\qquad - \frac{\mathbf{N}_\tau \cdot [(v_i - v_k) \times (v_j - v_k)]\,[(v_i - v_k) \times (v_j - v_k) \times (v_k - v_j)]}{\|(v_i - v_k) \times (v_j - v_k)\|^3} \Bigg) + \eta(v_i - v_i^{in}).
\end{aligned}
$$

Using the two facts that

$$
\|(v_j - v_i) \times (v_k - v_i)\| = \|(v_k - v_j) \times (v_i - v_j)\| = \|(v_i - v_k) \times (v_j - v_k)\| = 2\mathcal{S}_\tau,
$$

$$
\frac{(v_j - v_i) \times (v_k - v_i)}{2\mathcal{S}_\tau} = \frac{(v_k - v_j) \times (v_i - v_j)}{2\mathcal{S}_\tau} = \frac{(v_i - v_k) \times (v_j - v_k)}{2\mathcal{S}_\tau} = \mathcal{N}_\tau,
$$

where $\mathcal{S}_\tau$ and $\mathcal{N}_\tau$ are updating area and normal of triangle τ according to the updated vertices v, we arrive at

$$
\nabla_{v_i} E(v) = \sum_{\substack{\tau \in M_1(v_i),\\ \tau = (v_i, v_j, v_k)}} \frac{3 s_\tau ((\mathbf{N}_\tau \cdot \mathcal{N}_\tau)\mathcal{N}_\tau - \mathbf{N}_\tau) \times (v_k - v_j)}{2\mathcal{S}_\tau} + \eta(v_i - v_i^{in}). \tag{31}
$$

With the given gradient information (31) and the vertex positions of the initial noisy surface, many popular optimization techniques, such as accelerated gradient descent and quasi-Newton methods, can be used to solve our model (30). In this paper, we choose the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [17], one of the most commonly used methods for solving nonconstrained problems like (30). In each iteration, BFGS algorithm uses only the energy and gradient evaluated at the current and previous iterations.

We compare our vertex updating method (30) with the method proposed by Sun et al. [51] in Figures 3 and 4. Both methods have two inputs: face normals ($\mathbf{N}_\tau$) and vertex positions of the noisy surface (v^{in}). The accuracy of the input normals will influence the final reconstructed result. In Figure 3, we use normals of the clean surfaces and vertex positions of the noisy surfaces as inputs. This configuration can eliminate the influence produced by possible incorrect face normals, and show the essential difference of these two vertex updating methods more clearly. More practically, we then use filtered normals produced by our normal filtering, Algorithm 1, and vertex positions of the noisy surfaces as inputs in Figure 4. As we can see in these two figures, our method (30) can greatly reduce foldovers and help to improve the quality of the reconstructed surfaces, compared to the method proposed by Sun et al. [51]. We should point out that if the normal filtering step provides an exactly correct normal vector field, our vertex updating method can perfectly recover the surface without any foldovers, whereas the method proposed by Sun et al. [51] still produces lots of foldovers; see Figure 3 for an example. When the noise level is high, the previous normal filtering step may produce some incorrect normals. Although these

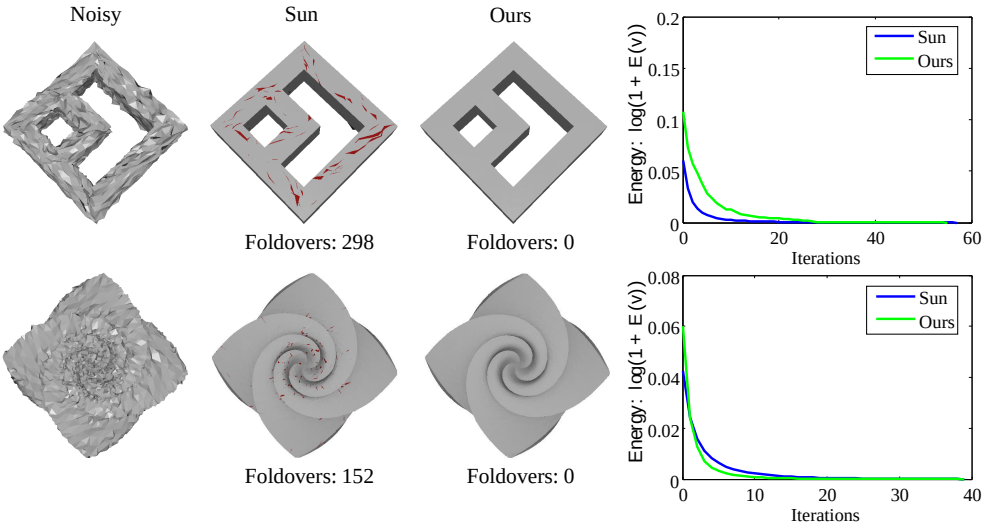


FIG. 3. *Comparisons of vertex updating methods (the method* (29) *proposed by Sun et al.* [51] *and ours* (30)*). The first column shows the triangulated surfaces corrupted by Gaussian noise with standard deviation* $\sigma = 0.4$ *mean edge length and* $\sigma = 0.3$ *mean edge length in random directions. The second and third columns are results produced by* [51] *and ours, respectively. The foldovers are highlighted in red. The last column illustrates the energy evolution curves via iteration numbers. (Figure is in color online.)*

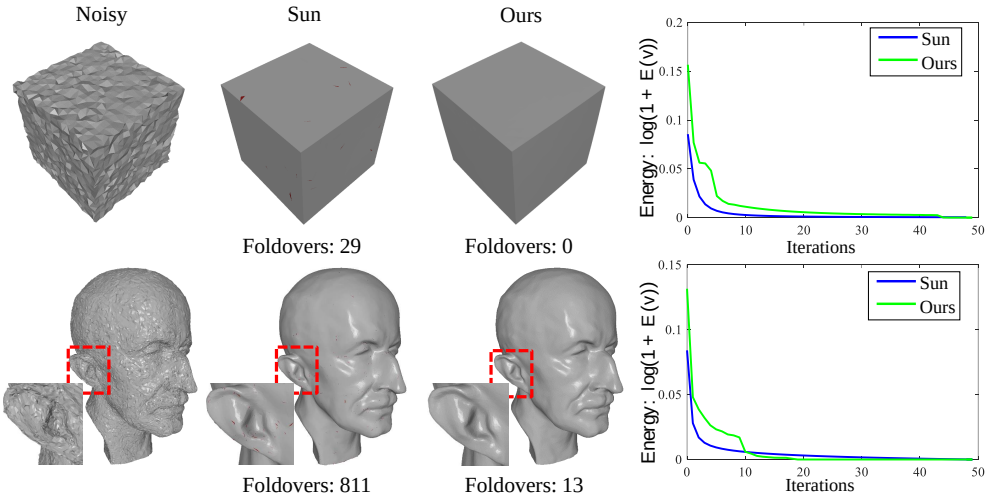


FIG. 4. *Comparisons of vertex updating methods (the method* (29) *proposed by Sun et al.* [51] *and ours* (30)*). The first column shows the triangulated surfaces corrupted by Gaussian noise with standard deviation* $\sigma = 0.2$ *mean edge length and* $\sigma = 0.2$ *mean edge length in random directions. The second and third columns are results produced by* [51] *and ours, respectively. The foldovers are highlighted in red. The last column illustrates the energy evolution curves via iteration numbers. (Figure is in color online.)*

incorrect normals may cause foldovers in our vertex updating method, the number of foldovers produced by our method is much smaller than the number of the method

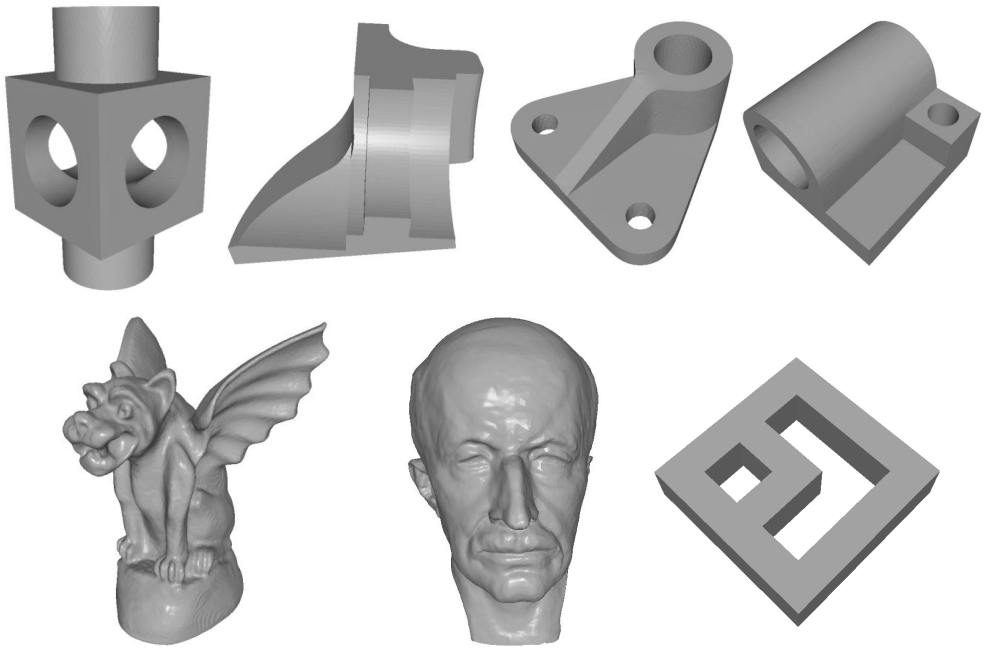


FIG. 5. *The clean surfaces tested in section* 6. *From top left to bottom right: Block, Fandisk, Part, Joint, Gargoyle, Max-Planck, and Doubletorus.*

proposed by Sun et al. [51] (see the last row of Figure 4). From the energy evolution curves in Figure 3, we observe that both methods are convergent and the iteration numbers of these two are close. However, the results produced by [51] suffer from severe foldovers and are inaccurate, while our method produces much better results.

6. Numerical experiments. We verify the effectiveness of our two-stage denoising method on a variety of triangulated surfaces with either synthetic or raw noise. The synthetic noise added in random directions is produced by a zero-mean Gaussian function with standard deviation σ proportional to the mean edge length of the clean surface. The clean surfaces tested in this section are listed in Figure 5.

To verify the robustness of our denoising method to the quality of surface triangles, we use two quantities as in [37]. These quantities are defined as follows:

$$D^{global} = \frac{\min_\tau \text{ area of } \tau}{\max_\tau \text{ area of } \tau},$$

$$D^{local} = \min_\tau \frac{\min_{e \prec \tau} \text{ length of } e}{\max_{e \prec \tau} \text{ length of } e}.$$

D^{global} stands for the smallest largest triangle area ratio used for globally describing the distribution of triangles. D^{local} denotes the smallest one of ratios of shortest and longest edge lengths in triangles, which can be used to locally measure the quality of triangles. The information of the clean surfaces are listed in Table 1. Although several surfaces including Gargoyle, Max-Planck, and Embossment do not feature very regular meshes as D^{global} and D^{local} indicated in Table 1, our method can still effectively handle all of these surfaces and produce satisfactory results.

TABLE 1
Information of surfaces tested in this paper.

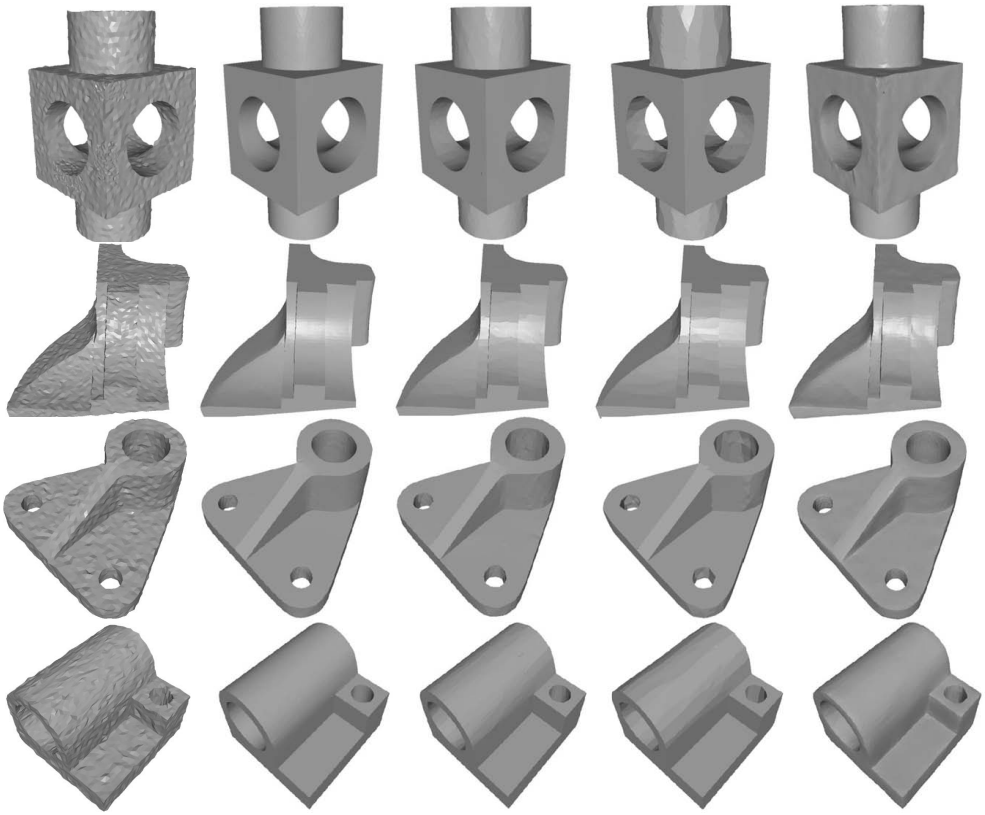
Surface	#vertices	#triangles	D^{global}	D^{local}
Block	8771	17500	0.066312	0.369025
Fandisk	6475	12946	0.0202721	0.333103
Part	4261	8530	0.0596278	0.235325
Joint	5636	11276	0.000489636	0.0508141
Gargoyle	25002	50000	0.000194802	0.0814815
Max-Planck	30942	61880	9.28547e-006	0.0135542
Rabbit	37394	73679	0.0805432	0.0991931
Angel	24566	72690	0.041708	0.0824533
Shell	58354	174031	0.00645787	0.106432
Embossment	65988	195095	0.00735766	0.0106189
Doubletorus	2686	5376	0.00439037	0.109461

For fair comparisons, we have implemented all of the algorithms tested in this paper using C++ and run all examples on a notebook with a Intel dual core 2.10 GHz processor and 8GB RAM. All of the surfaces are rendered in flat-shading model to show faceting effect. Our algorithm is compared qualitatively and quantitatively to state-of-art methods, respectively. We also discuss our algorithm from various aspects, including influences of parameters and algorithm convergence.

6.1. Qualitative comparisons. In this subsection, we compare our surface denoising method w-HO with other methods including the TV normal filtering method [64], ℓ_0 minimization [27], and bilateral weighting Laplacian optimization [67], abbreviated as TV, ℓ_0 and bw-Laplacian, respectively. For all of these methods, we carefully tuned the parameters to get the visually best denoising results.

In Figure 6, we compare the results for surfaces containing both sharp features (including sharp edges and corners) and smooth regions (including smoothly curved regions and flat regions). As we can see, bw-Laplacian keeps smooth regions well but blurs sharp features, while our w-HO method, TV, and ℓ_0 preserve most sharp features well. Furthermore, TV and ℓ_0 both suffer from staircase effects in smoothly curved regions indicated in Figures 6(c) and (d), and this phenomenon is extremely serious for ℓ_0, which produces false edges in the first and last row of (d) of Figure 6. However, our w-HO method does not produce the staircase effect while preserving sharp features well. As we know, both sharp features and noise belong to high frequency information. The bw-Laplacian cannot distinguish them strictly, especially for small scale features. Thus, it may treat some features as noise and blur them. In addition, as stated in compressed sensing, both the ℓ_0 norm and ℓ_1 norm have sparse property, which can be used for preserving sharp features. However, as ℓ_0 and TV use low order information of surfaces, they tend to produce staircase effects in smooth regions, especially for ℓ_0 for its high sparsity requirement. Consequently, the compared three methods can deal either with smooth regions or sharp features well, but not both. In contrast, our w-HO method can suppress the staircase effects in smooth regions and simultaneously preserve sharp features. Overall, for CAD-like surfaces, visual comparisons in Figure 6 show that our w-HO method is noticeably better than all of the other three methods in terms of smooth regions and sharp features recovery.

Figure 7 shows results of surfaces with fine features. As can be seen, TV and ℓ_0 tend to flatten some details, and ℓ_0 performs even worse. Our w-HO method and bw-Laplacian both can generate visually better denoising results. However, from numerical metrics (which will be introduced in subsection 6.2), we observe that errors



(a) Noisy (b) w-HO (c) TV (d) ℓ_0 (e) bw-Laplacian

FIG. 6. *Denoising results of Block, Fandisk, Part, and Joint (corrupted by Gaussian noise, standard deviation = 0.15 mean edge length). From left to right: input noisy surfaces, denoising results produced by our proposed w-HO method, TV method* [64], ℓ_0 *minimization* [27], *and bilateral weighting Laplacian method* [67], *respectively.*

of our method are always lower than those of bw-Laplacian. This demonstrates that our method is better than bw-laplacian. In general, for non-CAD surfaces, our w-HO method can also yield satisfactory results containing more details than other methods.

To further demonstrate the validity of our w-HO method, we test it on real scanned surfaces; see Figure 8. We can see that our method can yield very good denoising results preserving most features well.

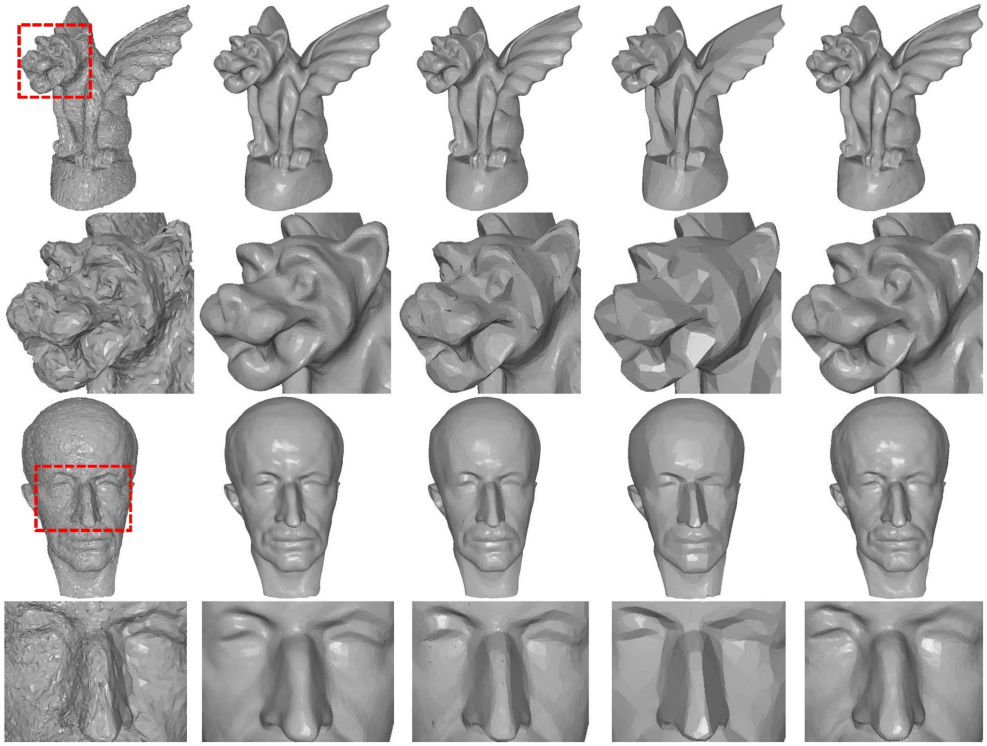
6.2. Quantitative comparisons. From the above comparisons, we find that our w-HO method generates visually better results than those compared methods. In this subsection, we further compare them quantitatively.

We use two error metrics [51, 52, 67] to measure the deviation of the denoised surface from the clean one, which are defined as followed:

- Mean square angular error (MSAE):

$$\text{MSAE} = \text{average}(\angle(\mathbf{N}^{'}, \mathbf{N})),$$

where $\angle(\mathbf{N}^{'}, \mathbf{N})$ is the square angle between the normal of the denoising result and the clean surface, average$(\angle(\mathbf{N}^{'}, \mathbf{N}))$ is the square angle averaged over



(a) Noisy (b) w-HO (c) TV (d) ℓ_0 (e) bw-Laplacian

FIG. 7. *Denoising results of Gargoyle (corrupted by Gaussian noise, standard deviation* $= 0.25$ *mean edge length) and Max-Planck (corrupted by Gaussian noise, standard deviation* $= 0.2$ *mean edge length). From left to right: input noisy surfaces, denoising results produced by our proposed w-HO method, TV method* [64], ℓ_0 *minimization* [27], *and bilateral weighting Laplacian method* [67], *respectively. The second and fourth rows show magnified views of Gargoyle and Max-Planck.*

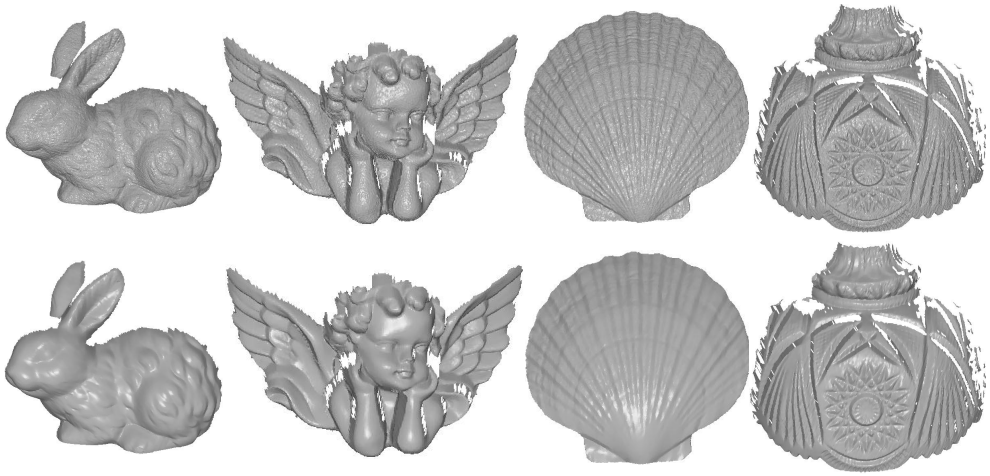


FIG. 8. *Our denoising results for four real scanned surfaces. From left to right: Rabbit, Angel, Shell, and Embossment.*

TABLE 2

Quantitative evaluation results of Figures 6 *and* 7 *for our proposed w-HO method, TV method* [64], ℓ_0 *minimization* [27], *and bilateral weighting Laplacian method* [67]. σ *is the standard deviation of the Gaussian noise added to the clean surface.* N*-step records CPU costs in the normal filtering step, and* V *step records CPU costs in the vertex updating step.*

Models	σ	Methods	MSAE ($\times 10^{-3}$)	$\mathcal{E}_{v,2}$ ($\times 10^{-3}$)	CPU costs (in seconds) N step	V step	Total	Foldovers
Block	0.15	w-HO	**2.40**	**0.79**	7.35	0.203	7.553	0
		TV	3.61	0.98	2.13	0.047	2.177	12
		ℓ_0	4.31	1.81	–	16.57	16.57	5
		bw-Laplacian	5.66	1.01	1.45	0.078	1.528	37
Fandisk	0.15	w-HO	**1.48**	0.88	1.68	0.141	1.821	0
		TV	1.53	**0.86**	0.84	0.032	0.872	3
		ℓ_0	3.85	1.12	–	7.53	7.53	3
		bw-Laplacian	2.50	1.01	0.76	0.047	0.807	5
Part	0.15	w-HO	**1.29**	**0.94**	1.84	0.094	1.934	0
		TV	2.51	1.22	0.67	0.016	0.686	7
		ℓ_0	8.1	2.32	–	8.01	8.01	1
		bw-Laplacian	4.22	1.23	0.53	0.047	0.577	7
Joint	0.15	w-HO	**2.88**	**0.76**	2.80	0.125	2.925	4
		TV	4.21	1.20	1.98	0.031	2.011	17
		ℓ_0	12.7	2.37	–	12.84	12.84	4
		bw-Laplacian	6.77	2.01	0.81	0.063	0.873	33
Bunny	0.2	w-HO	**12.9**	0.92	18.78	0.859	19.639	5
		TV	16.1	0.89	8.57	0.234	8.804	61
		ℓ_0	27.5	2.16	–	60.72	60.72	10
		bw-Laplacian	13.4	**0.88**	7.89	0.375	8.265	68
Gargoyle	0.25	w-HO	**13.5**	0.77	23.32	0.688	24.008	11
		TV	17.3	0.83	12.34	0.203	12.543	197
		ℓ_0	31.0	1.79	–	57.36	57.36	15
		bw-Laplacian	16.1	**0.75**	5.12	0.344	5.464	86
Max-Planck	0.2	w-HO	**10.8**	**0.85**	31.81	0.734	32.544	13
		TV	16.6	1.11	13.41	0.171	13.581	809
		ℓ_0	33.5	1.86	–	74.05	74.05	58
		bw-Laplacian	12.1	0.91	9.41	0.281	0.691	649

all faces.

- L_2 vertex-based surface-to-surface error:

$$\mathcal{E}_{v,2} = \sqrt{\frac{1}{3\sum_{\tau} s_\tau} \sum_{i=0}^{\mathrm{V}-1} \Big(\sum_{M_1(v_i)} s_\tau \Big) \mathrm{dist}(v_i^{'}, M)^2},$$

where $\mathrm{dist}(v_i^{'}, M)$ is the distance between the updated vertex $v_i^{'}$ and a triangle of the clean surface M which is closest to $v_i^{'}$.

Then, we compare our w-HO method to the other three methods using the above two error metrics for the examples shown in Figures 6 and 7. The evaluation results are listed in Table 2. As can be seen, our w-HO method outperforms the other methods in the sense that angular errors (MSAE) from w-HO are significantly smaller than all of the other methods, especially for CAD-like surfaces. It is also observed that the results of w-HO have the least L_2 vertex-based errors ($\mathcal{E}_{v,2}$) in most cases. This demonstrates that the results produced by w-HO are more faithful to the ground truth surfaces.

The CPU costs of all the tested methods are recorded in the second-to-last column of Table 2. For our w-HO, TV, and bw-Laplacian, the CPU costs of the normal

filtering step and vertex updating step are recorded separately. Moreover, w-HO uses our vertex updating method, whereas TV and bw-Laplacian use the method proposed by Sun et al. [51] to recover the surfaces. ℓ_0 is the one-step denoising method that directly updates vertex positions; thus we only record the CPU costs in the vertex updating step for this method. For our w-HO method, the most time-consuming part is solving the **N**-sub problem. As mentioned in subsection 4.2, due to the error forgetting property [62] of our ALM algorithm, we use a fast approximate strategy to solve this subproblem. As can be seen, the bilateral weighting Laplacian method [67] is the fastest method, while ℓ_0 minimization [27] is the slowest. Although our w-HO method is a little more computationally intensive than the TV method [64], the CPU cost is still acceptable.

In terms of the CPU costs in the vertex updating step as shown in Table 2, we find that our vertex updating method is a little slower than the method proposed by Sun et al. [51]. However, our vertex updating method can significantly improve the quality of reconstructed surfaces and reduce the foldovers.

6.3. Influence of parameters. To the best of our knowledge, most triangulated surface denoising methods have parameters which need to be manually tuned. Among the compared state-of-the-art methods (TV [64], ℓ_0 [27], and bw-Laplacian [67]), TV has two parameters, and ℓ_0 and bw-Laplacian have three parameters. Our w-HO also has two parameters, i.e., α and $r_{\mathbf{p}}$. These two parameters need to be tuned for producing prominent results. The first parameter is used to balance the fidelity and regularization term of the normal filtering model (20). The second one is introduced by the augmented Lagrangian method.

We use α to control the degree of denoising and smoothness of the result surface. Figure 9 illustrates results of different α with fixed $r_{\mathbf{p}}$. As can be seen, if α is too large, noise cannot be effectively removed as indicated in Figure 9(b); and if α is too small, surfaces will be oversmoothed and fine features will be lost as illustrated in Figure 9(e). For each noisy surface, there exist a range of α for our w-HO producing visually well denoising results; see Figures 9(c) and (d). In order to generate both quantitatively and visually well results, α is suggested in the range of $[40, 1100]$, with greater values for non-CAD or scanned surfaces and with smaller values for CAD surfaces. Moreover, α should be set with a smaller value for a higher level of noise, and vice versa. The results of our algorithm changes little when α has small perturbations.

(a) Noisy (b) $\alpha = 3000$ (c) $\alpha = 1500$ (d) $\alpha = 600$ (e) $\alpha = 200$

FIG. 9. *Denoising results for α with fixed $r_{\mathbf{p}}$. From left to right: input noisy surface (corrupted by Gaussian noise, standard deviation = 0.1 mean edge length), and results with different α.*

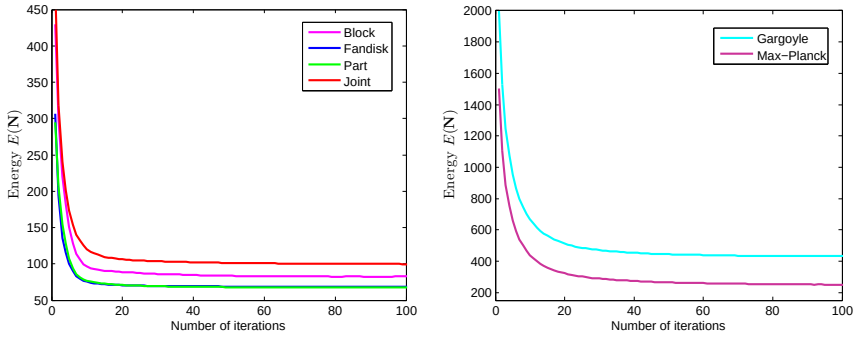
The parameter $r_{\mathbf{p}}$ also has influence on denoising results. Figure 10 shows results of different $r_{\mathbf{p}}$ with fixed α. As we can see, too small $r_{\mathbf{p}}$ will leave some noise on the

surface as indicated in Figure 10(b), and too large $r_{\mathbf{p}}$ should oversmooth the result as illustrated in Figure 10(e). Again, for each noisy surface, there exist a range of $r_{\mathbf{p}}$ for our algorithm producing visually well results as shown in Figures 10(c) and (d). For almost all the CAD surfaces, $r_{\mathbf{p}}$ can be set to 1 to produce both quantitatively and visually well results. For non-CAD and scanned surfaces, $r_{\mathbf{p}}$ is suggested in the range of $[0.1, 1.2]$, with a greater value for a higher level of noise.

(a) Noisy (b) $r_{\mathbf{p}} = 0.1$ (c) $r_{\mathbf{p}} = 1$ (d) $r_{\mathbf{p}} = 5$ (e) $r_{\mathbf{p}} = 100$

FIG. 10. *Denoising results for $r_{\mathbf{p}}$ with fixed α. From left to right: input noisy surface (corrupted by Gaussian noise, standard deviation = 0.15 mean edge length) and results with different $r_{\mathbf{p}}$.*

6.4. Algorithm convergence and effect of dynamic weights. Due to nonlinear and nonconvex constraints of the proposed high order normal filtering model (20), it is a challenge to achieve the convergence analysis of Algorithm 1. However, we can verify the convergence using numerical experiments. From the energy evolution in Figure 11, we observe that the energy always decreases in each iteration. This verifies the numerical convergence of Algorithm 1.

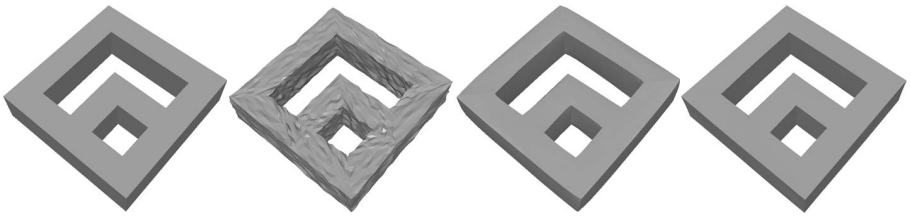


(a) (b)

FIG. 11. *Energy evolution via iteration numbers of* (a) *surfaces in Figure* 6 *and* (b) *surfaces in Figure* 7.

Dynamic weights in the proposed normal filtering model (20) play a large role in recovering sharp features of surfaces; see their effect in Figure 12. As we can see, without dynamic weights, some sharp edges are smoothed a little in the denoising procedure. In contrast, the result with these dynamic weights is better.

6.5. Comparison to ℓ_1-norm Laplacian-like normal filtering model. In this subsection, we compare our normal filtering model (20) with the ℓ_1-norm Laplacian-like normal filtering model to show the advantage of our second order difference (11) over the Laplace-like operator (18) in surface denoising application. The ℓ_1-norm



(a) ground truth (b) noisy (c) result without dynamic weights (d) result with dynamic weights

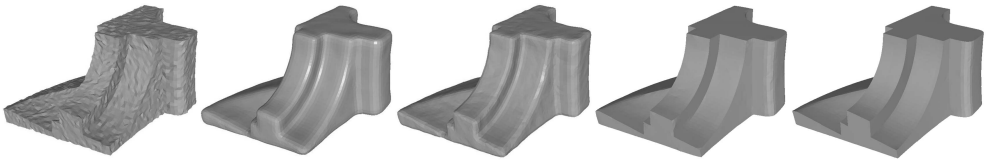
FIG. 12. *Denoising results of Doubletorus (corrupted by Gaussian noise, standard deviation* $= 0.15$ *mean edge length). From left to right: the clean surface, input noisy surface, denoising result produced by the proposed high order normal filtering model* (20) *without and with dynamic weights, respectively.*

Laplacian-like normal filtering model is given as

$$\min_{\mathbf{N}\in C_{\mathbf{N}}} \left\{ E(\mathbf{N}) = R_{\text{vlap}}(\delta^\star \delta \mathbf{N}) + \frac{\alpha}{2} \left\| \mathbf{N} - \mathbf{N}^{in} \right\|^2_{\mathbf{V}_M} \right\}, \tag{32}$$

where

$$C_{\mathbf{N}} = \{\mathbf{N} \in \mathbf{V}_M : \|\mathbf{N}_\tau\| = 1 \ \forall \tau\},$$
$$R_{\text{vlap}}(\delta^\star \delta \mathbf{N}) = \sum_\tau \mathrm{w}_\tau \left(\sum_{i=1}^{3} (\delta^\star \delta N_i|_\tau)^2 \right)^{\frac{1}{2}} s_\tau.$$



(a) (b) (c) (d) (e)

FIG. 13. *Denoising results of Fandisk (corrupted by Gaussian noise, standard deviation* $= 0.15$ *mean edge length).* (a) *is the noisy surface;* (b) *and* (c) *are results produced by* ℓ_1*-norm Laplacian-like normal filtering model* (32) *without and with dynamic weights, respectively;* (d) *and* (e) *are results produced by the high order normal filtering model* (20) *without and with dynamic weights, respectively.*

The dynamic weight w_τ on each triangle is defined as

$$\mathrm{w}_\tau = \exp\left(- \left\| \sum_{\tau_j \in D_1(\tau_i)} (\mathbf{N}_\tau - \mathbf{N}_{\tau_j}) \right\|^4 \right),$$

which is used to enhance the sparsity of the proposed model (32). For the sake of fairness, our normal filtering model (20) is compared with the Laplacian-like one (32) without and with dynamic weights, respectively. As we can see in Figure 13, although our normal filtering model (20) and the Laplacian-like model (32) both remove the staircase effect, our model (20) can preserve sharp features well while the model (32) with the Laplace-like operator cannot.

7. Conclusion. In this paper, we propose a triangulated surface denoising method using a newly defined discrete high order regularization. The method applies the high order regularization to the normal vector field with a well-designed weighting function. The variational model is solved by the augmented Lagrangian method with dynamic weights strategy. Moreover, a new vertex updating scheme is presented to overcome the orientation ambiguities introduced by previous vertex updating methods. We also compare our method to several denoising methods on a variety of triangulated surfaces both qualitatively and quantitatively. Conventional methods either smooth sharp features or generate staircase artifacts. Since our method preserves sharp features well and produces no staircase effect, it outperforms the other three compared methods. Thus it can be applied to more general surfaces containing both sharp features and smoothly curved regions.

REFERENCES

[1] C. L. Bajaj and G. Xu, *Anisotropic diffusion of surfaces and functions on surfaces*, ACM Trans. Graph., 22 (2003), pp. 4–32.

[2] R. E. Barnhill, *Surfaces in computer aided geometric design: A survey with new results*, Comput. Aided Geom. Design, 2 (1985), pp. 1–17.

[3] M. Bergounioux and L. Piffet, *A second-order model for image denoising*, Set-Valued Var. Anal., 18 (2010), pp. 277–306.

[4] P. Blomgren and T. F. Chan, *Color TV: Total variation methods for restoration of vector-valued images*, IEEE Trans. Image Process., 7 (1998), pp. 304–309.

[5] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy, *Polygon Mesh Processing*, AK Peters, CRC Press, Boca Raton, FL, 2010.

[6] K. Bredies, K. Kunisch, and T. Pock, *Total generalized variation*, SIAM J. Imaging Sci., 3 (2010), pp. 492–526, https://doi.org/10.1137/090769521.

[7] X. Bresson and T. F. Chan, *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, Inverse Probl. Imaging, 2 (2008), pp. 455–484.

[8] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical Geometry of Non-Rigid Shapes*, Springer, New York, 2008.

[9] J. F. Cai, B. Dong, S. Osher, and Z. Shen, *Image restoration: Total variation, wavelet frames, and beyond*, J. Amer. Math. Soc., 25 (2012), pp. 1033–1089.

[10] J. F. Cai, B. Dong, and Z. Shen, *Image restoration: A wavelet frame based model for piecewise smooth functions and beyond*, Appl. Comput. Harmon. Anal., 41 (2015), pp. 94–138.

[11] E. J. Candès, M. B. Wakin, and S. P. Boyd, *Enhancing sparsity by reweighted ℓ_1 minimization*, J. Fourier Anal. Appl., 14 (2008), pp. 877–905.

[12] A. Chambolle and P. L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.

[13] T. Chan, A. Marquina, and P. Mulet, *High-order total variation-based image restoration*, SIAM J. Sci. Comput., 22 (2000), pp. 503–516, https://doi.org/10.1137/S1064827598344169.

[14] T. F. Chan, S. Esedoglu, and F. Park, *A fourth order dual method for staircase reduction in texture extraction and image restoration problems*, in Proceedings of the 2010 IEEE International Conference on Image Processing, 2010, pp. 4137–4140.

[15] T. F. Chan, S. H. Kang, and J. Shen, *Total variation denoising and enhancement of color images based on the CB and HSV color models*, J. Visual Commun. Image Represent., 12 (2001), pp. 422–435.

[16] U. Clarenz, U. Diewald, and M. Rumpf, *Anisotropic geometric diffusion in surface processing*, in Processdings of the 2000 Conference on Visualization, 2000, pp. 397–405.

[17] J. E. Dennis, Jr., and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89, https://doi.org/10.1137/1019005.

[18] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, *Implicit fairing of irregular meshes using diffusion and curvature flow*, in Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999, pp. 317–324.

[19] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, *Anisotropic feature-preserving denoising of height fields and bivariate data*, in Processdings of the 2000 Graphics Interface, Vol. 11, 2000, pp. 145–152.

[20] B. Dong, *Sparse representation on graphs by tight wavelet frames and applications*, Appl. Comput. Harmon. Anal., 42 (2017), pp. 452–479.
[21] B. Dong, Q. Jiang, C. Liu, and Z. Shen, *Multiscale representation of surfaces by tight wavelet frames with applications to denoising*, Appl. Comput. Harmon. Anal., 41 (2016), pp. 561–589.
[22] B. Dong, Y. Mao, I. D. Dinov, Z. Tu, Y. Shi, Y. Wang, and A. W. Toga, *Wavelet-based representation of biological shapes*, in Proceedings of the 2009 International Symposium on Visual Computing, 2009, pp. 955–964.
[23] M. Elsey and S. Esedoğlu, *Analogue of the total variation denoising model in the context of geometry processing*, Multiscale Model. Simul., 7 (2009), pp. 1549–1573, https://doi.org/10.1137/080736612.
[24] D. A. Field, *Laplacian smoothing and Delaunay triangulations*, Commun. Appl. Numer. Methods, 4 (1988), pp. 709–712.
[25] S. Fleishman, I. Drori, and D. Cohen-Or, *Bilateral mesh denoising*, ACM Trans. Graph., 22 (2003), pp. 950–953.
[26] D. Goldfarb, Z. Wen, and W. Yin, *A curvilinear search method for p-harmonic flows on spheres*, SIAM J. Imaging Sci., 2 (2009), pp. 84–109, https://doi.org/10.1137/080726926.
[27] L. He and S. Schaefer, *Mesh denoising via ℓ_0 minimization*, ACM Trans. Graph., 32 (2013), pp. 1–8.
[28] W. Hinterberger and O. Scherzer, *Variational methods on the space of functions of bounded hessian for convexification and denoising*, Computing, 76 (2006), pp. 109–133.
[29] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, *Surface reconstruction from unorganized points*, in Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, 1992, pp. 71–78.
[30] T. R. Jones, F. Durand, and M. Desbrun, *Non-iterative, feature-preserving mesh smoothing*, ACM Trans. Graph., 22 (2003), pp. 943–949.
[31] M. R. Kaus, J. Von Berg, J. Weese, W. Niessen, and V. Pekar, *Automated segmentation of the left ventricle in cardiac MRI*, Med. Image Anal., 8 (2004), pp. 245–254.
[32] R. Kimmel and J. A. Sethian, *Computing geodesic paths on manifolds*, Proc. Natl. Acad. Sci. USA, 95 (1998), pp. 8431–8435.
[33] R. Lai and T. F. Chan, *A framework for intrinsic image processing on surfaces*, Comput. Vis. Image. Und., 115 (2011), pp. 1647–1661.
[34] R. Lai and S. Osher, *A splitting method for orthogonality constrained problems*, J. Sci. Comput., 58 (2014), pp. 431–449.
[35] R. Lai, X.-C. Tai, and T. F. Chan, *A ridge and corner preserving model for surface restoration*, SIAM J. Sci. Comput., 35 (2013), pp. A675–A695, https://doi.org/10.1137/110846634.
[36] K.-W. Lee and W.-P. Wang, *Feature-preserving mesh denoising via bilateral normal filtering*, in Proceedings of the 2005 International Conference on Computer Aided Design and Computer Graphics, 2005, pp. 275–280.
[37] Z. Liu, H. Zhang, and C. Wu, *On geodesic curvature flow with level set formulation over triangulated surfaces*, J. Sci. Comput., 70 (2017), pp. 631–661.
[38] W. E. Lorensen and H. E. Cline, *Marching cubes: A high resolution 3d surface construction algorithm*, in Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, 1987, pp. 163–169.
[39] M. Lysaker, A. Lundervold, and X. C. Tai, *Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time*, IEEE Trans. Image Process., 12 (2003), pp. 1579–1590.
[40] M. Lysaker and X. C. Tai, *Iterative image restoration combining total variation minimization and a second-order functional*, Int. J. Comput. Vis., 66 (2006), pp. 5–18.
[41] G. D. Maso, I. Fonseca, G. Leoni, and M. Morini, *A higher order model for image restoration: The one-dimensional case*, SIAM J. Math. Anal., 40 (2007), pp. 2351–2391, https://doi.org/10.1137/070697823.
[42] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, *Efficiently combining positions and normals for precise 3d geometry*, ACM Trans. Graph., 24 (2005), pp. 536–543.
[43] T. Oden, R. Glowinski, and P. L. Tallec, *Augmented Lagrangian and operator splitting method in non-linear mechanics*, Math. Comput., 58 (1992).
[44] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489, https://doi.org/10.1137/040605412.
[45] K. Papafitsoros and C. B. Schönlieb, *A combined first and second order variational approach for image reconstruction*, J. Math. Imaging Vision, 48 (2014), pp. 308–338.

[46] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[47] G. Sapiro and D. L. Ringach, *Anisotropic diffusion of multivalued images with applications to color filtering*, IEEE Trans. Image Process., 5 (1996), pp. 1582–1586.
[48] O. Scherzer, *Denoising with higher order derivatives of bounded variation and an application to parameter estimation*, Computing, 60 (1998), pp. 1–27.
[49] Y. Shen and K. E. Barner, *Fuzzy vector median-based surface smoothing*, IEEE Trans. Vis. Comput. Graph., 10 (2004), pp. 252–265.
[50] J. Solomon, K. Crane, A. Butscher, and C. Wojtan, *A General Framework for Bilateral and Mean Shift Filtering*, preprint, https://arxiv.org/abs/1405.4734, 2014.
[51] X. Sun, P. Rosin, R. Martin, and F. Langbein, *Fast and effective feature-preserving mesh denoising*, IEEE Trans. Vis. Comput. Graph., 13 (2007), pp. 925–938.
[52] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, *Random walks for feature-preserving mesh denoising*, Comput. Aided Geom. Design, 25 (2008), pp. 437–456.
[53] G. Taubin, *A signal processing approach to fair surface design*, in Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, 1995, pp. 351–358.
[54] G. Taubin, *Linear Anisotropic Mesh Filters*, IBM Research Report RC22213 (W0110-051), IBM T. J. Watson Research, 2001.
[55] J. Wang, X. Zhang, and Z. Yu, *A cascaded approach for feature-preserving surface mesh denoising*, Comput. Aided Geom. Des., 44 (2012), pp. 597–610.
[56] Y. Wang and W. Yin, *Sparse signal reconstruction via iterative support detection*, SIAM J. Imaging Sci, 3 (2010), pp. 462–491, https://doi.org/10.1137/090772447.
[57] Z. Wen and W. Yin, *A feasible method for optimization with orthogonality constraints*, Math. Program., 142 (2013), pp. 397–434.
[58] C. Wu and X.-C. Tai, *Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial tv, and high order models*, SIAM J. Imaging Sci., 3 (2010), pp. 300–339, https://doi.org/10.1137/090767558.
[59] L. Xu, C. Lu, Y. Xu, and J. Jia, *Image smoothing via ℓ_0 gradient minimization*, ACM Trans. Graph., 30 (2011), 174.
[60] H. Yagou, Y. Ohtake, and A. Belyaev, *Mesh smoothing via mean and median filtering applied to face normals*, in Proceedings of Geometric Modeling and Processing, 2002, pp. 124–131.
[61] J. Yang and C. Wang, *A wavelet frame approach for removal of mixed gaussian and impulse noise on surfaces*, Inverse Probl. Imaging, 11 (2017), pp. 783–798.
[62] W. Yin and S. Osher, *Error forgetting of Bregman iteration*, J. Sci. Comput., 54 (2013), pp. 684–695.
[63] Y. L. You and M. Kaveh, *Fourth-order partial differential equations for noise removal*, IEEE Trans. Image Process., 9 (2000), pp. 1723–1730.
[64] H. Zhang, C. Wu, J. Zhang, and J. Deng, *Variational mesh denoising using total variation and piecewise constant function space*, IEEE Trans. Vis. Comput. Graph., 21 (2015), pp. 873–86.
[65] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu, *Guided mesh normal filtering*, Comput. Graph. Forum, 34 (2015), pp. 23–34.
[66] Y.-B. Zhao and D. Li, *Reweighted ℓ_1 minimization for sparse solutions to underdetermined linear systems*, SIAM J. Optim., 22 (2012), pp. 1065–1088, https://doi.org/10.1137/110847445.
[67] Y. Zheng, H. Fu, K. C. Au, and C. L. Tai, *Bilateral normal filtering for mesh denoising*, IEEE Trans. Vis. Comput. Graph., 17 (2011), pp. 1521–1530.