This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2939362, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

High Order Mesh Denoising via ℓ_p-ℓ_1 Minimization

ZHENG LIU[1,2], MINGQIANG GUO[1], ZHONG XIE[1,2], JINQIN LIU[1], AND BOHONG ZENG[1]
[1]School of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China
[2]National Engineering Research Center of Geographic Information System, China University of Geosciences, Wuhan 430074, China

Corresponding author: Zhong Xie (xiezhong@cug.edu.cn)

This work was supported in part by the National Science Foundation of China under Grants 61702467, 41671400, 41971356, and 41701446, and in part by the National Key Research and Development Program under Grant 2018YFB0505500.

ABSTRACT Mesh denoising is crucial for improving the quality of meshes required by scanning devices. The main challenge is to maximally preserve geometric features while removing different kinds of noise. In this paper, we propose a novel normal filtering model that incorporates a high order ℓ_p regularization term and an ℓ_1 fidelity term. Then, vertex positions of the mesh can be reconstructed according to the filtered face normals. Thanking to the proposed ℓ_p-ℓ_1 normal filtering model, our method has crucial advantage in preserving geometric features and simultaneously is robust against different kinds of noise. Numerically, we develop an efficient algorithm based on iteratively reweighted ℓ_1 minimization and augmented Lagrangian method to solve the problem. We testify effectiveness of our mesh denoising method on synthetic meshes and a broad variety of scanning data produced by the laser scanner and Kinect sensors. We compare our method to state-of-the-art methods and demonstrate the superiority of our method in various cases.

INDEX TERMS Augmented lagrangian method, iteratively reweighted ℓ_1 minimization, mesh denoising, 3D geometry processing.

I. INTRODUCTION

In recent years, due to the rapid development of digital scanning devices (e.g., Microsoft Kinect, Xtion Pro, Google Project Tango, and Intel RealSense), more and more triangulated meshes can be easily acquired from the real world. Unfortunately, scanned meshes are usually corrupted by different kinds of noise introduced in physical scanning and reconstruction processes. The noise will degrade the quality of meshes and cause errors in downstream geometry applications, such as segmentation, parameterization, reconstruction, and visualization. Thus, the task of recovering high quality meshes from noisy inputs becomes increasingly important. The main challenge of mesh denoising is to preserve underlying geometric features of the surface as much as possible while removing different kinds of noise including Gaussian, impulsive, and mixed noise (impulsive noise is called outliers sometimes).

Mesh denoising has been widely studied in the last two decades. It is beyond our scope to review all existing mesh denoising methods, and we only review several notable methods and methods that are most relevant to this work. Interested readers can refer to the works [1]–[3] for a comprehensive review of mesh denoising. In our opinion, mesh denoising methods can be rough classified into three categories: filter-based, optimization-based, and data-driven methods. Early filter-based methods [4]–[7] directly applied isotropic smoothing on mesh vertices for removing noise. Although these methods are simple and can remove noise from the mesh, they also smooth geometric features evidently. Later on, many anisotropic filtering methods [8]–[13] were proposed to recover geometric features while removing noise. Compared to the isotropic methods, the anisotropic methods are more robust against the noise and can produce clearer feature-preserving results. However, in the case of high noise density, these anisotropic methods still fail to preserve geometric features, especially for preserving sharp features (sharp edges and corners). To recover sharp features of the mesh, some methods [14]–[16] apply the bilateral filter on face normals and then update the vertex positions according to the filtered face normals. Although these bilateral normal filtering methods can produce satisfactory results for preserving geometric features, the sharp features are still difficult to recover when the noise level is high. More recently, Yadav et al. [17] proposed a normal filter using a robust error norm, which can efficiently preserve sharp features over the mesh. Although their method preserves sharp feature well, it usually smoothes weak features and fine details of the surface. We should point out that, these methods belong to two-stage mesh denoising methods, i.e., face normal filtering followed by updating vertex positions [3], [14]–[24]. The difference between these two-stage methods is in their normal filtering strategies.

The variational denoising methods formulate the denoising process as an optimization problem and seek for a desired solution that satisfied the optimization goal. The optimization-based methods have became widespread recently. Zheng et al. [14] presented a global normal filtering model using bilateral weight function. Their method works well for preserving fine details and smooth regions, but cannot recover sharp features well. For preserving sharp features over the surface, many researchers used the conception of sparsity to remove the noise from the mesh. He and Schaefer [25] used the ℓ_0 minimization for removing the noise by inducing the sparsity for an discrete edge-based operator. However, the ℓ_0 minimization problem is NP-hard. A critical limitation of the problem is that the CPU cost of solving it is very time consuming. Moreover, for mesh denoising, the ℓ_0 minimization method tends to flatten smooth regions for its high sparsity requirement. In image processing community, the sparsity of the gradient (first order operator) is widely used for preserving image edges. The corresponding ℓ_1 minimization problem is the famous total variation (TV) model [26]. Inspired by the success of TV model in image processing, many researchers [18], [27], [28] extended it to mesh denoising for preserving sharp features. However, due to the using of the first order information over the surface, the TV-based models suffer producing staircase artifacts in smoothly curved regions. In order to overcome the staircase artifacts in smooth regions, Liu et al. [3] presented a ℓ_1-based high order normal filtering model. Yet, the high order model sometimes blurs sharp features, especially for dealing with the large noise. Zhong et al. [29] proposed a ℓ_1-based normal filtering model with three sparsity terms, which can recover both sharp features and smooth regions well. Moreover, their method can deal with outliers. Unfortunately, their method contains too many parameters to manually tune. The process of parameter tuning of their method is tedious.

More recently, researchers proposed some data-driven methods. Wang et al. [30] proposed a method based on the cascades normal regression, which can remove the noise without assumptions about geometric features of the underling surface and noise patterns. Their method first learns nonlinear regression functions mapping the filtered face normal descriptors to the face normals of the ground-truth input, and then applied the learned functions to compute filtered face normals. Wang et al. [31] presented a data-driven method by learning normal variations in two steps. In the first step, they learn mapping from the noisy input to its ground-truth counterpart and use neural networks to remove the noise, which might lost some fine details of the underling surface. Then in the second step, they learn to recover the missing fine details. Although these data-driven methods perform well for real scanned data, the performance of these methods depend on the completeness of the training data set.

As we can see, the above mentioned mesh denoising methods have their own limitations. This paper tries to overcome these limitations by using a novel normal filtering model including only two terms: a high order ℓ_p regularization term for preserving geometric features (sharp features, fine details, and smoothly curved regions) without introducing unnatural artifacts, and an ℓ_1 fidelity term for encouraging the solution be less dependent on the exact value of outliers and noise. After filtering the face normals of the mesh, the vertex positions should be reconstructed to match the filtered face normals. To sum up, the main contributions of the paper are listed as follows:

- We present a novel high order normal filtering model. The ℓ_p regularization of the model enables us to efficiently remove noise and preserve geometric features maximally, while the ℓ_1 fidelity is formulated for dealing with outliers.
- We propose an efficient minimization method based on iterative reweighted ℓ_1 minimization (IRL1) and augmented Lagrangian method (ALM) to solve the problem.
- We demonstrate the performance of our denoising method on synthetic meshes and a variety of scanned data acquired by the laser scanner and Kinect sensors. Experiments illustrate that our method outperforms the compared state-of-the-art methods visually and quantitatively.

The remainder of this paper is organized as follows. In Section II, we present a high order ℓ_p-ℓ_1 normal filtering model. An efficient algorithm based on iteratively reweighted ℓ_1 minimization (IRL1) and augmented Lagrangian method (ALM) is proposed to solve the model. In Section III, vertex positions are reconstructed according to the filtered face normals. In Section IV, we discuss our two-stage mesh denoising method and compare it with typical existing methods both visually and quantitatively. Section V concludes the paper.

II. HIGH ORDER NORMAL FILTERING VIA ℓ_P-ℓ_1 MINIMIZATION

In this section, we introduce the high order ℓ_p-ℓ_1 minimization on face normals followed by some definitions over triangulated meshes. A mesh of arbitrary topology with no degenerate triangles in $\mathbb{R}^3$ is represented as M. The set of vertices, edges, and triangle faces of M are denoted as $\{v_i : i = 1, 2, \cdots, \mathrm{V}\}$, $\{e_i : i = 1, 2, \cdots, \mathrm{E}\}$, and $\{\tau_i : i = 1, 2, \cdots, \mathrm{T}\}$, respectively. Here V, E, and T are the numbers of vertices, edges, and faces of M.

A. HIGH ORDER NORMAL FILTERING MODEL

For a given noisy mesh, we write face normals of the mesh as $\widetilde{\mathrm{N}}$. To remove noise in $\widetilde{\mathrm{N}}$, we treat face normals N as a variable and solve the following ℓ_p-ℓ_1 minimization problem:

$$\min_{\mathrm{N}\in\mathrm{C_N}}\left\{\alpha\sum_{l}\mathrm{len}(l)\|\nabla^2\mathrm{N}|_l\|_p+\sum_{\tau}\mathrm{area}(\tau)\|\mathrm{N}_\tau{-}\widetilde{\mathrm{N}}_\tau\|\right\}, \tag{1}$$

where

$$\mathrm{C_N}=\left\{\mathrm{N}\in\mathbb{R}^{3\times\mathrm{T}}:\|\mathrm{N}_\tau\|=1,\forall\tau\right\},$$
$$\|\nabla^2\mathrm{N}|_l\|_p=\left(\big(\sum_{i=1}^{3}|(\nabla^2\mathrm{N}_i)|_l|^2\big)^{\frac{1}{2}}\right)^p,$$

α is a positive parameter, $\mathrm{len}(l)$ is the length of line l connecting the barycenter with one vertex of triangle τ, and $\mathrm{area}(\tau)$ is the area of triangle τ. The high order operator ∇^2 is defined on each line l of the mesh, which reads as follows

$$(\nabla^2\mathrm{N})|_l=w_{e^+}(\mathrm{N}_\tau-\mathrm{N}_{\tau^+})+w_{e^-}(\mathrm{N}_\tau-\mathrm{N}_{\tau^-}),$$

where e^+ and e^- are two edges sharing the common vertex of l, τ^+ is the triangle sharing e^+ with τ, and τ^- is the triangle sharing e^-. w_{e^+} and w_{e^-} are two positive weights defined as

$$w_e=\exp(-\|\mathrm{N}_{e,1}-\mathrm{N}_{e,2}\|^2/2\sigma^2), \tag{2}$$

where $\mathrm{N}_{e,1}$ and $\mathrm{N}_{e,2}$ are two normals sharing the common edge e. All aforementioned descriptions are indicated in Fig. 1. For more details about the description of the high order operator ∇^2, we refer readers to [3], [32].

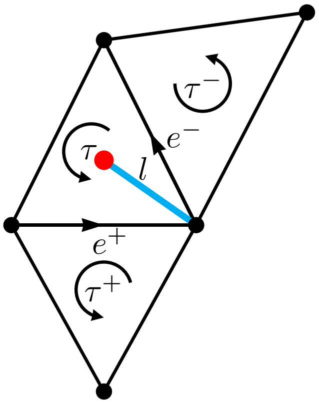


FIGURE 1: The illustration of ∇^2 over the line l plotted in *cyan* in triangle τ with the barycenter plotted in *red*.

The proposed high order ℓ_p-ℓ_1 normal filtering model (1) consists of ℓ_p regularization and ℓ_1 fidelity terms. The ℓ_p regularization makes the solution smooth while preserving geometric features as much as possible. The ℓ_p regularization is nonconvex. It penalizes large variations of the signal less than convex ones (ℓ_1 and ℓ_2), and preserves sharp features better. This means that, using of nonconvex regularization can produce better results than the corresponding convex ones. The view, nonconvex regularization offers richer possibilities to recover high quality images with neat images, has been verified in image processing community widely [33]. The fidelity term of the model makes the solution harmonize well with the input noisy mesh. Yet, if the noisy mesh includes impulsive and mixed noise, the traditional ℓ_2 fidelity cannot produce a satisfactory denoised result. As we know, the ℓ_1 fidelity was first proposed in the work [34] for image restoration with impulsive and mixed noise. Here, we introduce the ℓ_1 fidelity in mesh denoising to help the solution less dependent on the exact value of outliers and noise.

B. NUMERICAL ALGORITHM FOR PROPOSED MODEL

Due to the using of ℓ_p norm, the ℓ_p-ℓ_1 normal filtering model (1) is nonconvex and nonsmooth, which is challenging to directly solve. Fortunately, Candes et al. [35] tackled this challenge by the so-called iteratively reweighted ℓ_1 (IRL1) minimization. Their method approximates the exact solution by solving a sequence of ℓ_1 problems. The IRL1 algorithm was originally introduced to improve the sparsity in ℓ_1 related compressed sensing problems [33], [35].

Here, we use IRL1 algorithm to solve our nonconvex ℓ_p-ℓ_1 model (1). More specifically, (1) can be tackled by solving a sequence of ℓ_1-ℓ_1 problems as followed:

$$\begin{aligned}\mathrm{N}^{k+1}=\arg\min_{\mathrm{N}\in\mathrm{C_N}}&\left\{\alpha\sum_{l}\mathrm{len}(l)\mathrm{W}_l^k\|\nabla^2\mathrm{N}|_l\|\right.\\&\left.+\sum_{\tau}\mathrm{area}(\tau)\|\mathrm{N}_\tau-\widetilde{\mathrm{N}}_\tau\|\right\},\end{aligned} \tag{3}$$

where

$$\mathrm{W}_l^k=\frac{p}{(\|\nabla^2\mathrm{N}_l^k\|+\epsilon)^{1-p}}. \tag{4}$$

ϵ is a small constant to prevent the denominator from being 0 and we set it as 0.001.

Because of the involvement of vectorial seminorms and nonlinear constraints of (3), each ℓ_1-ℓ_1 problem (3) is nondifferentiable and thus still hard to solve. Recently, the variable splitting and augmented Lagrangian method (ALM) have attained intensive attention for their efficiency in solving ℓ_1 related optimization problems [36]. Thus, the minimization problem (3) is first rewritten as a constrained minimization problem by introducing a pair of new variables, which reads

$$\begin{aligned}\min_{\mathrm{N,X,Y}}\quad&\alpha\sum_{l}\mathrm{len}(l)\mathrm{W}_l^k\|\mathrm{X}_l\|+\sum_{\tau}\mathrm{area}(\tau)\|\mathrm{Y}_\tau\|+\Psi(\mathrm{N})\\&\mathrm{s.t.,}\ \ \mathrm{X}=\nabla^2\mathrm{N},\mathrm{Y}=\mathrm{N}-\widetilde{\mathrm{N}},\end{aligned} \tag{5}$$

where

$$\Psi(\mathrm{N})=\begin{cases}0, & \mathrm{N}\in\mathrm{C_N}\\+\infty, & \mathrm{N}\notin\mathrm{C_N}.\end{cases}$$

Then, the augmented Lagrangian function is introduced to handle the above constrained optimization problem

$$\begin{aligned}
&\mathscr{L}(\mathrm{N},\mathrm{X},\mathrm{Y};\Lambda_x,\Lambda_y)\\
&\quad=\alpha\sum_l \mathrm{len}(l)\mathrm{W}_l^k\|\mathrm{X}_l\|+\sum_\tau \mathrm{area}(\tau)\|\mathrm{Y}_\tau\|\\
&\quad+\Psi(\mathrm{N})+\sum_l \mathrm{len}(l)\Lambda_{x,l}\cdot(\mathrm{X}_l-\nabla^2\mathrm{N}|_l)\\
&\quad+\frac{r_x}{2}\sum_l \mathrm{len}(l)\|\mathrm{X}_l-\nabla^2\mathrm{N}|_l\|^2\\
&\quad+\sum_\tau \mathrm{area}(\tau)\Lambda_{y,\tau}\cdot\Big(\mathrm{Y}_\tau-(\mathrm{N}_\tau-\widetilde{\mathrm{N}}_\tau)\Big)\\
&\quad+\frac{r_y}{2}\sum_\tau \mathrm{area}(\tau)\|\mathrm{Y}_\tau-(\mathrm{N}_\tau-\widetilde{\mathrm{N}}_\tau)\|^2, \qquad (6)
\end{aligned}$$

where $\Lambda_x=\{\Lambda_{x,l}\}\in\mathbb{R}^{3\times\mathrm{L}},\Lambda_y=\{\Lambda_{y,\tau}\}\in\mathbb{R}^{3\times\mathrm{T}}$ are two Lagrange multipliers, and r_x and r_y are two positive penalty coefficients. L is the number of lines connecting the barycenter and one vertex of triangle τ. Explicitly, the problem (6) can be divided into three subproblems: the N , X, and Y subproblems.

Sub-minimization problem with respect to N: we can write the N subproblem as

$$\begin{aligned}
&\min_{\mathrm{N}}\ \Psi(\mathrm{N})+\frac{r_x}{2}\sum_l \mathrm{len}(l)\|\nabla^2\mathrm{N}|_l-(\mathrm{X}_l+\frac{\Lambda_{x,l}}{r_x})\|^2\\
&+\frac{r_y}{2}\sum_\tau \mathrm{area}(\tau)\|\mathrm{N}_\tau-\widetilde{\mathrm{N}}_\tau-(\mathrm{Y}_\tau+\frac{\Lambda_{y,\tau}}{r_y})\|^2, \qquad (7)
\end{aligned}$$

which is a quadratic optimization with the unit normal constraints $\Psi(\mathrm{N})$. Here we first ignore the term $\Psi(\mathrm{N})$ and solve a quadratic problem, and then project the solution onto a unit sphere. Specifically, the quadratic problem without the term $\Psi(\mathrm{N})$ is as followed

$$\begin{aligned}
&\min_{\mathrm{N}}\ \frac{r_x}{2}\sum_l \mathrm{len}(l)\|\nabla^2\mathrm{N}|_l-(\mathrm{X}_l+\frac{\Lambda_{x,l}}{r_x})\|^2\\
&+\frac{r_y}{2}\sum_\tau \mathrm{area}(\tau)\|\mathrm{N}_\tau-\widetilde{\mathrm{N}}_\tau-(\mathrm{Y}_\tau+\frac{\Lambda_{y,\tau}}{r_y})\|^2. \qquad (8)
\end{aligned}$$

The solution of the problem (8) is achieved by a sparse linear system, which can be solved by using well developed numerical packages.

Sub-minimization problem with respect to X: we can write X subproblem as

$$\begin{aligned}
&\min_{\mathrm{X}}\ \alpha\sum_l \mathrm{len}(l)\mathrm{W}_l^k\|\mathrm{X}_l\|\\
&+\frac{r_x}{2}\sum_l \mathrm{len}(l)\|\mathrm{X}_l-(\nabla^2\mathrm{N}|_l-\frac{\Lambda_{x,l}}{r_x})\|^2. \qquad (9)
\end{aligned}$$

As the problem (9) is decomposable, we can solve each X_l independently. For each X_l, we need to solve the following problem

$$\min_{\mathrm{X}_l}\ \alpha\mathrm{W}_l^k\|\mathrm{X}_l\|+\frac{r_x}{2}\|\mathrm{X}_l-(\nabla^2\mathrm{N}|_l-\frac{\Lambda_{x,l}}{r_x})\|^2,$$

which has a closed form solution as

$$\mathrm{X}_l=\max(0,\ 1-\frac{\alpha\mathrm{W}_l^k}{r_x\|\zeta_{x,l}\|})\zeta_{x,l}, \qquad (10)$$

where

$$\zeta_x=\nabla^2\mathrm{N}-\frac{\Lambda_x}{r_x}.$$

Sub-minimization problem with respect to Y: we can write Y subproblem as

$$\begin{aligned}
&\min_{\mathrm{Y}}\ \sum_\tau \mathrm{area}(\tau)\|\mathrm{Y}_\tau\|\\
&+\frac{r_y}{2}\sum_\tau \mathrm{area}(\tau)\|\mathrm{Y}_\tau-(\mathrm{N}_\tau-\widetilde{\mathrm{N}}_\tau-\frac{\Lambda_{y,\tau}}{r_y})\|^2. \qquad (11)
\end{aligned}$$

Analogously, the problem (11) also can be decomposed, we can solve each Y_τ solely. For each Y_τ, we need to solve the following problem

$$\min_{\mathrm{Y}_\tau}\ \|\mathrm{Y}_\tau\|+\frac{r_y}{2}\|\mathrm{Y}_\tau-(\mathrm{N}_\tau-\widetilde{\mathrm{N}}_\tau-\frac{\Lambda_{y,\tau}}{r_y})\|^2,$$

which has a closed form solution as

$$\mathrm{Y}_\tau=\max(0,\ 1-\frac{1}{r_y\|\zeta_{y,\tau}\|})\zeta_{y,\tau}, \qquad (12)$$

where

$$\zeta_y=\mathrm{N}-\widetilde{\mathrm{N}}-\frac{\Lambda_y}{r_y}.$$

Assembling the above parts, the overall alternating minimization algorithm for solving the high order ℓ_p-ℓ_1 model (1) is sketched in Algorithm 1. As we can see, $\mathrm{N}^{k,c}$ is the sequence produced by Algorithm 1, where the index k refers to outer iterations and c refers to inner iterations. In each outer iteration, we first solve an ℓ_1-ℓ_1 minimization problem by ALM, and then update the weights W according to the current face normals N. Furthermore, in each inner iteration, the algorithm first solves three subproblems, and then updates the Lagrange multipliers. In summary, it is challenge to find the global minimizer of the nonconvex nonsmooth minimization problem (1) with the nonlinear constraints. We adapt IRL1 and ALM to approximate a well enough solution of the problem (1) numerically. In Section IV, we will show the efficiency of our algorithm, which can produce satisfactory denoising results visually and quantitatively.

III. VERTEX UPDATING SCHEME

After obtaining filtered face normals by solving the high order ℓ_p-ℓ_1 normal filtering model (1), vertex positions of the mesh should be updated to match the filtered face normals. To this end, we use a vertex updating scheme presented by Liu et al. [3]. This scheme updates the vertex positions by minimizing the following problem

$$\begin{aligned}
\min_v \Bigg\{ E(v)=&\sum_{\tau=(v_i,v_j,v_k)}\Big(\mathrm{N}_\tau-\frac{(v_j-v_i)\times(v_k-v_i)}{\|(v_j-v_i)\times(v_k-v_i)\|}\Big)^2\\
&+\frac{\eta}{2}||v-\widetilde{v}||^2\Bigg\}, \qquad (13)
\end{aligned}$$

Algorithm 1: IRL1 and ALM for solving high order ℓ_p-ℓ_1 normal filtering model (1)

```
Initialization: $\mathrm{N}^{-1} = 0, \mathrm{W}^{-1} = 1, k = 0, \varepsilon = 1e-6$;
repeat
    Initialization: $\Lambda_x^{k,0} = \Lambda_y^{k,0} = 0, \mathrm{N}^{k,-1} = \mathrm{X}^{k,-1} = \mathrm{Y}^{k,-1} = 0, c = 0$;
    repeat
        Solve N subproblem
            For fixed $(\Lambda_x^{k,c}, \Lambda_y^{k,c}, \mathrm{X}^{k,c-1}, \mathrm{Y}^{k,c-1})$, compute $\mathrm{N}^{k,c}$ from (8);
            Normalize $\mathrm{N}^{k,c}$;
        Solve X subproblem
            For fixed $(\Lambda_x^{k,c}, \Lambda_y^{k,c}, \mathrm{N}^{k,c}, \mathrm{Y}^{k,c-1})$, compute $\mathrm{X}^k$ from (10) ;
        Solve Y subproblem
            For fixed $(\Lambda_x^{k,c}, \Lambda_y^{k,c}, \mathrm{N}^{k,c}, \mathrm{X}^{k,c})$, compute $\mathrm{Y}^k$ from (12) ;
        Update Lagrange multipliers
            $\Lambda_x^{k,c+1} = \Lambda_x^{k,c} + r_x\left(\mathrm{X}^{k,c} - (\nabla^2\mathrm{N})^{k,c}\right)$ ;
            $\Lambda_y^{k,c+1} = \Lambda_y^{k,c} + r_y(\mathrm{Y}^{k,c} - \mathrm{N}^{k,c} + \widetilde{\mathrm{N}})$ ;
    until $\sum_\tau area(\tau)\|\mathrm{N}^{k,c} - \mathrm{N}^{k,c-1}\| < \varepsilon$ or $c \geq 50$;
    $\mathrm{N}^k = \mathrm{N}^{k,c}$ ;
    Update weight $\mathrm{W}^k$ through (4);
until $\sum_\tau area(\tau)\|\mathrm{N}^k - \mathrm{N}^{k-1}\| < \varepsilon$ or $k \geq 20$;
return $\mathrm{N}^k$.
```

where (v_i, v_j, v_k) are vertices of τ with counterclockwise order, $\widetilde{v}$ is vertex positions of the noisy surface and η is a small positive parameter. This vertex updating model can overcome the trianglewise orientation ambiguity problem produced by the vertex updating model proposed by Sun et al. [37]. Due to this vertex updating model is not our main contribution, we refer interested readers to the work [3] for further information.

We can reformulate partial derivatives of (13) with respect to v_i as follows:

$$\begin{aligned}\nabla_{v_i} E(v) = &\sum_{\tau \in M_1(v_i)} \left(\mathrm{N}_\tau - (\mathrm{N}_\tau \cdot \mathcal{N}_\tau)\mathcal{N}_\tau\right) \times (v_j - v_k) \\ &+ \eta(v_i - \widetilde{v_i}),\end{aligned} \tag{14}$$

where $\mathcal{N}_\tau$ is the updating normal of τ according to updated v (the derivation process of (14) can refer to the work [3]), and $M_1(v_i)$ is the set of triangles containing vertex v_i. With the gradient information calculated from (14) and the initial vertex positions $\widetilde{v}$, we use gradient descent algorithm to minimize the problem (13). We adopt the backtracking line search strategy with our gradient descent algorithm to adaptively choose the step size. In each iteration, the line search strategy uses only the energy and gradient evaluated at current and previous iterations.

IV. EXPERIMENT RESULTS AND COMPARISONS

In this section, numerical experiments are presented on a wide range of triangulated meshes including CAD, non-CAD and raw scanned data (captured by the laser scanner and Kinect sensors). The tested meshes are corrupted by either synthetic or raw scanned noise. The synthetic noise added in random directions is generated by Gaussian, impulsive or mixed Gaussian and impulsive noise. We use two quantities as in the works [3], [38] to verify the robustness of our method to mesh quality. These two quantities are defined as follows:

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

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

The mesh information of the surfaces used in this paper are recorded in Table 1. From Table 1 we can see that, several meshes are very irregular, e.g., David and Child produced by Kinect sensors. Our method can effectively handle all the meshes and produce satisfactory denoising results. The clean meshes tested in this section are listed in Fig. 2.

TABLE 1: Information of meshes tested in this paper

Mesh	#vertices	#triangles	D^{global}	D^{local}
Julius	36,201	71,912	0.004711	0.107538
Part	4,261	8,530	0.207208	0.466109
Fandisk	6,475	12,946	0.020272	0.333103
Joint	20,902	41,808	0.058784	0.286217
Bunny-hi	35,233	70,462	0.001832	0.078386
Child (Fig. 6)	50,002	100,000	0.003699	0.158511
Rabbit	37,394	73,679	0.080543	0.099193
David	51,789	101,937	1.07353e-005	0.022549
Child (Fig. 8)	70,135	138,430	6.37239e-006	0.027263
Dodecahedron	4,610	9,216	0.617998	0.618011
Vase-Lion	50,002	100,000	0.000136	0.025853
Block	8,771	17,550	0.004967	0.006838
Bunny	29,849	59,694	0.140844	0.515311

We have implemented our mesh denoising method and all methods compared in the paper in C++. All of the numerical examples are tested on a standard PC with Intel Core i7 processor and 16GB RAM. For showing faceting effect, all meshes are rendered in flat-shading model. We will first discuss influence of parameters of our method. Then, both visual and quantitative comparisons of state-of-the-art methods are presented.

A. PARAMETER CHOICES

As we know each mesh denosing method has its own parameters to manually tune for producing satisfactory results. Specifically, our method has two parameters, i.e., α and p. α is applied to balance the fidelity and regularization terms of our high order ℓ_p-ℓ_1 model (1), while p is used to control the abilities of feature preserving and noise suppressing.

The parameter α helps the denoising result to harmonize well with the input mesh, which is used to control the degree of denoising. According to our numerical experiments, when α is too small, noise of the mesh cannot be completely

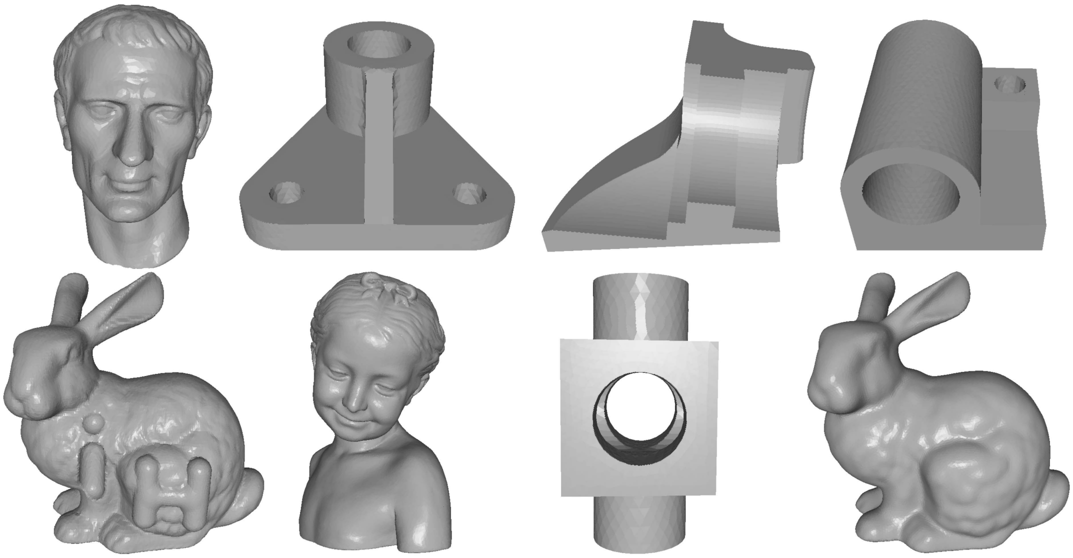


FIGURE 2: The clean meshes tested in Section IV. From top left to bottom right: Julius, Part, Fandisk, Joint, Bunny-Hi, Child, Block, and Bunny.

removed showed in Fig. 3b. During the increasing of α, the noise is removed obviously. However, if α is too large, the result is over-smoothed and a lot of details are flattened; see Fig. 3e. Moreover, for one noisy mesh, these exist a range of values of α to produce promising results. It indicates that our method is insensitive to perturbations of α; see Figs. 3c and 3d for examples.

The parameter p $(0 < p < 1)$ is used to determine the degree of nonconvexity of the ℓ_p regularization of our model (1). Too strong nonconvexity of the regularization will sharpen the result and produce false edges in smooth regions; see Fig. 4b. In contrast, too weak nonconvexity will yield a result with blurred sharp features showed in Fig. 4e. In our experiments, p is suggested in the range of $[0.4, 0.8]$ for producing satisfactory results; see Figs. 4c and 4d.

B. QUALITATIVE COMPARISONS AND EXAMPLES

Our high order ℓ_p-ℓ_1 denoising method (abbreviated as ℓ_p-ℓ_1) is visually compared with five state-of-the-art methods including high order normal filtering [3], total variation normal filtering [18], ℓ_0 minimization [25], robust and high fidelity mesh denoising [17], and bilateral normal filtering [39] (abbreviated as HO, TV, ℓ_0, RHM, and BF, respectively). We should point out that parameters in all the tested methods are elaborately tuned to produce visually well results.

In Fig. 5, we demonstrate and compare denoising results for CAD meshes containing both sharp features (sharp edges and corners) and smooth regions (smoothly curved regions and flat regions). As we can see, except BF, all the other five methods (ℓ_p-ℓ_1, HO, TV, ℓ_0, and RHM) can preserve sharp features in some ways. By using the robust error norm, RHM can recover geometric features, but it usually flattens weak features and curves sharp features; see the zoomed view of Fig. 5f. One important thing can be observed, TV and ℓ_0 produce undesired artifacts in smoothly curved regions, whereas the high order methods (ℓ_p-ℓ_1 and HO) can prevent these unnatural effects. The reasons are as follows. Due to the using of first order information of the mesh, TV suffers from staircase effects in smooth regions; see the zoomed-in view of Fig. 5d. This phenomenon is more serious for ℓ_0. ℓ_0 produces false edges in smooth regions sometimes for its high sparsity requirement; see Fig. 5e. In contrast, due to the using of high order information, both ℓ_p-ℓ_1 and HO do well over smooth regions. In our experiments, when the noise level is low, both ℓ_p-ℓ_1 and HO can eliminate the staircase effects and simultaneously recover sharp features. Yet, when meshes corrupted by considerable noise, HO fails to keep sharp features; see the zoomed-in view of Fig. 5c. In contrast, our ℓ_p-ℓ_1 can enforce its properties and yield more attractive results with sharp features preserving; see Fig. 5b. Thus, qualitative comparisons in Fig. 5 demonstrate that our method outperforms the other compared methods in terms of preserving sharp features and recovering smooth regions.

Fig. 6 shows the comparison results of non-CAD meshes. We can observe that, all the tested methods remove noise effectively. However, TV and ℓ_0 tend to flatten fine details, and ℓ_0 makes this situation even worse; see Figs. 6d and 6e. In some cases, ℓ_0 sharpens geometric features and produces false edges. Although HO and RHM do not sharpen features, they oversmooth features more or less. In addition, the oversmooth effect of RHM is more serious than that of HO; see Figs. 6f and 6c. Compared to these methods, ℓ_p-ℓ_1 and BF produce visually better results; see Figs. 6b and 6g. An important thing can be observed that, with the noncovex ℓ_p regularization, our method can preserve fine details better than BF, leading to higher quality of the results. Furthermore, HO smoothes some weak features, whereas our method ℓ_p-ℓ_1 recovers these weak features as much as possible; see

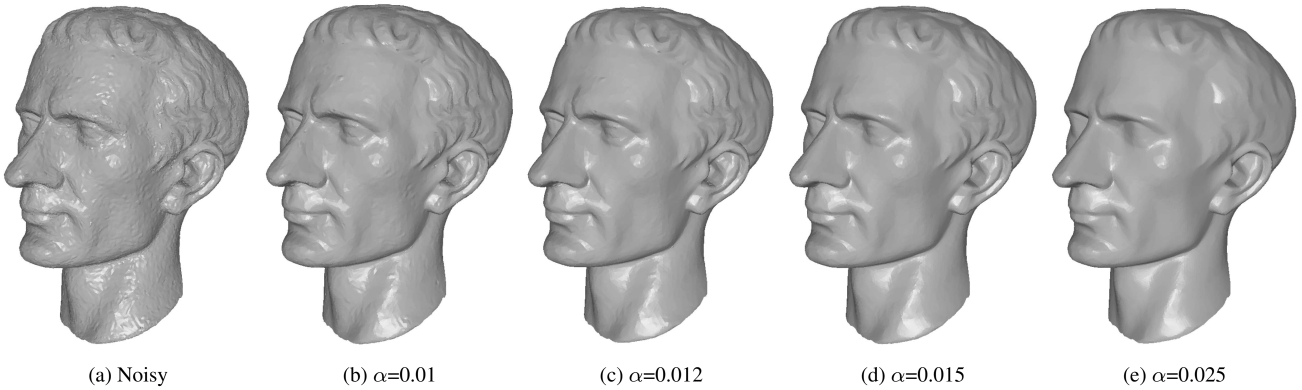


FIGURE 3: Denoising results of Julius for different α. From left to right: input noisy mesh (corrupted by Gaussian noise, standard deviation = 0.1 mean edge length) and results with different α.

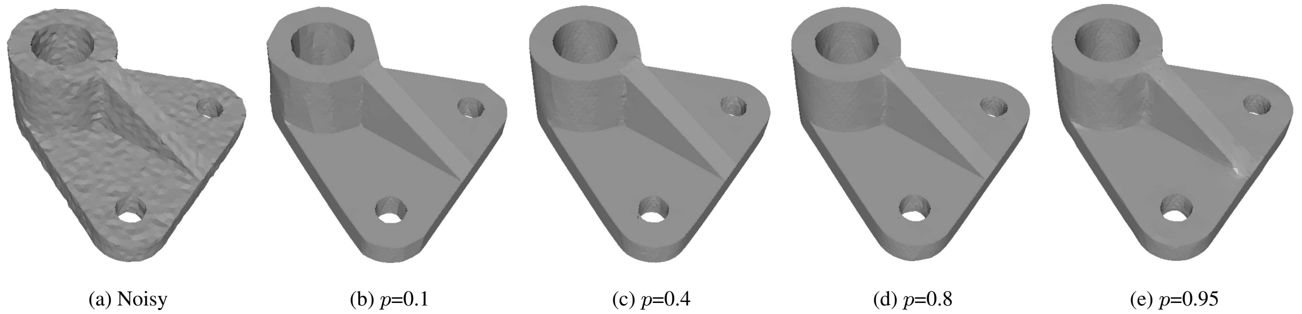


FIGURE 4: Denoising results of Part for different p. From left to right: input noisy mesh (corrupted by Gaussian noise, standard deviation = 0.1 mean edge length) and results with different p.

the zoomed views of Figs. 6b and 6c. In short, the results demonstrate that our method can also do a good job on non-CAD meshes.

In Figs. 7 and 8, we show denoising results on a variety of real scanning data. In Fig. 7, we present the comparison results of a mesh acquired by the laser scanner. As can be seen, our method ℓ_p-ℓ_1 and BF produce high fidelity results; see Figs. 7b and 7g. HO blurs some weak features a little; see Fig. 7c. In contrast, TV and ℓ_0 flatten some fine details. This artifact is more serious for ℓ_0 for its high sparsity requirement, see Figs. 7d and 7e. In addition, RHM oversmoothes small-scale features; see Fig. 7f. We also verify the effectiveness of our method on meshes produced by Kinect sensors. In Fig. 8, we compare the results produced by our method and the data-driven method proposed by Wang et al. [30]. As we can observe, even for the meshes produced by Kinect sensors with high level noise, Our method ℓ_p-ℓ_1 still can produce high quality results preserving geometric features well. In contrast, the method in [30] smoothes some fine details; see Fig. 8. In summary, our method is robust and can recover high quality meshes from the noisy inputs obtained from the real world.

A stress test for our method with increasing noise level is showed in Fig. 9. We can see that, when the noise level is smaller than the feature size, our method can preserve sharp features perfectly. However, if the noise level is too high, our method will fail to remove the noise and preserve sharp features; see the last column of Fig. 9.

It is necessary to explain the differences between our method and the method in [29]. Both methods are variational methods. The method in [29] has five parameters to tune for producing satisfactory results, while our method only has two main parameters. Although both methods use the same high order operator, the norms in regularization terms of these two methods are totally different. Specifically, the method in [29] uses ℓ_1 norm in its high order regularization, while our method uses ℓ_p norm. Thus, our method is more robust for preserving geometric features, see Fig. 10 for example.

Fig. 11 demonstrates comparison results of the meshes corrupted by impulsive or mixed noise. As we can see in Fig. 11b, our method ℓ_p-ℓ_1 can remove impulsive or mixed noise completely and simultaneously preserve geometric features well. In contrast, HO, TV, and ℓ_0 cannot effectively remove impulsive or mixed noise; see Figs. 11c, 11d, and 11e. Although RHM and BF can remove the impulsive or mixed noise, they oversmooth weak features and fine details evidently; see Figs. 11f and 11g. Thus, as the result of using ℓ_1 fidelity, our method has crucial advantages in handling the

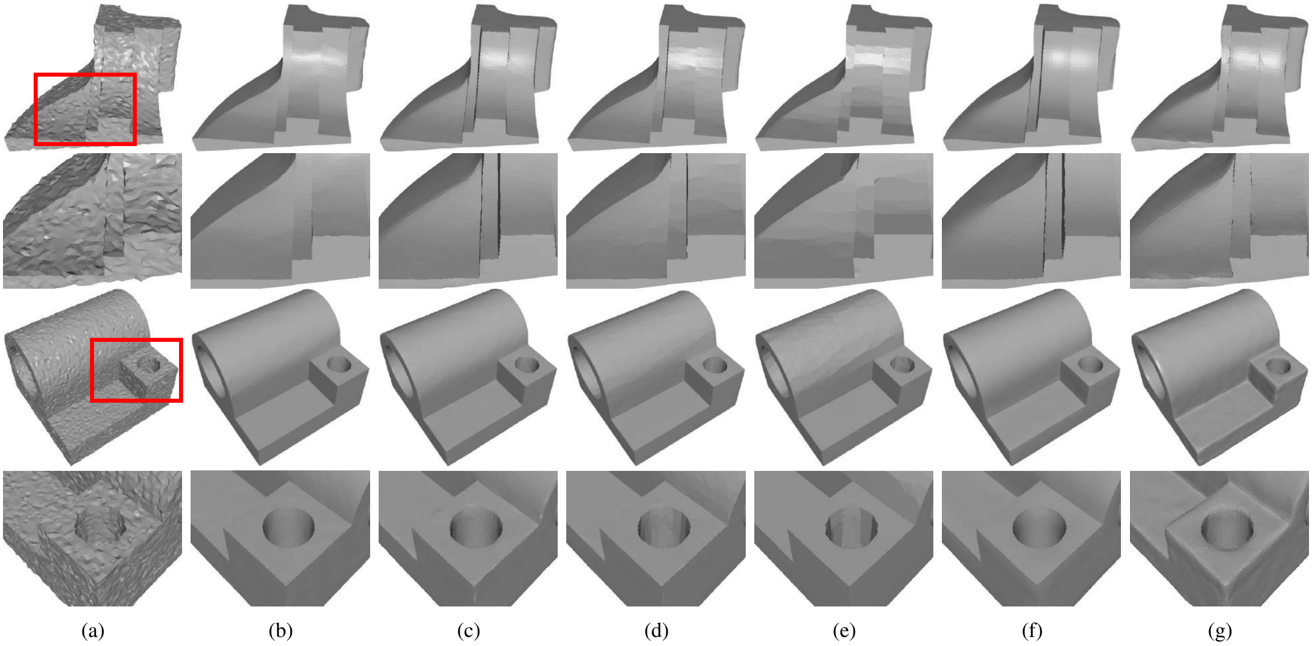


FIGURE 5: Comparisons of denoising results on CAD meshes (corrupted by Gaussian noise, standard deviation = 0.2 mean edge length). From left to right: (a) input noisy meshes, denoising results produced by (b) our method ℓ_p-ℓ_1, (c) HO [3], (d) TV [18], (e) ℓ_0 [25], (f) RHM [17], and (g) BF [39], respectively. The even rows are zoomed-in views.

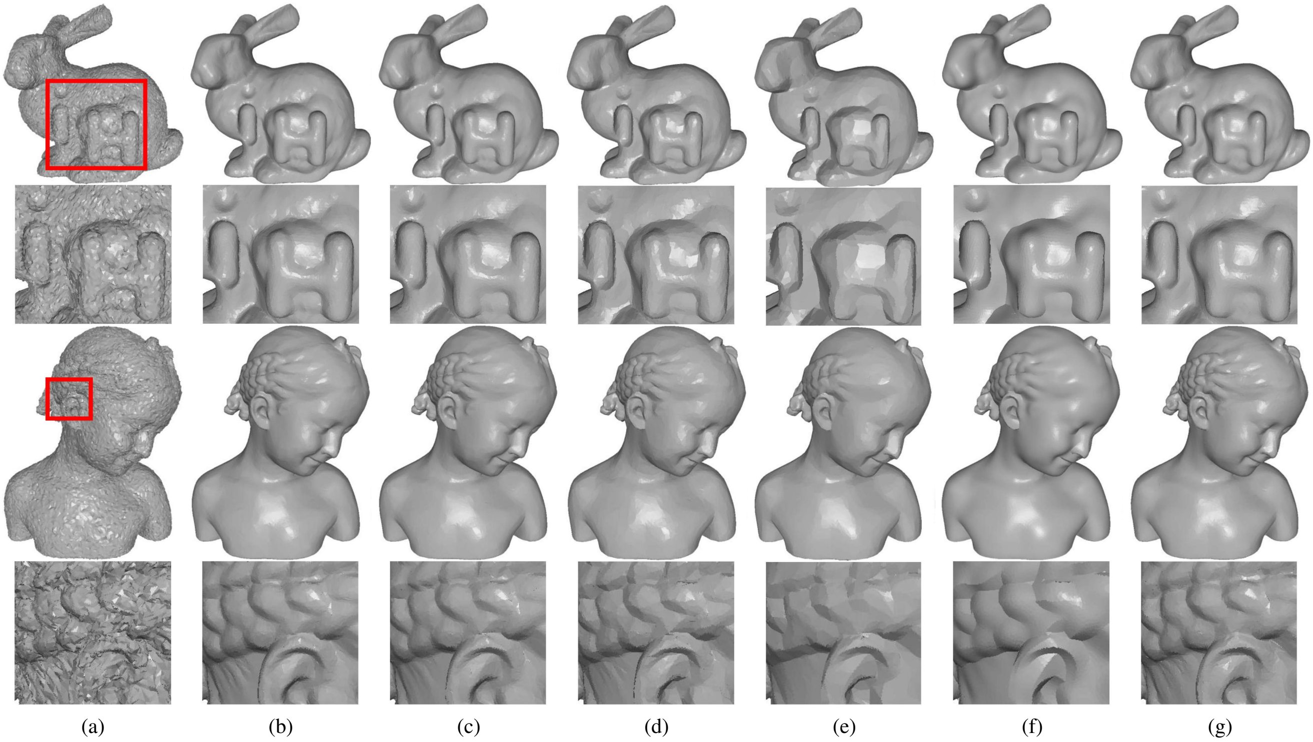


FIGURE 6: Comparisons of denoising results on non-CAD meshes (corrupted by Gaussian noise, standard deviation = 0.3 mean edge length). From left to right: (a) input noisy meshes, denoising results produced by (b) our method ℓ_p-ℓ_1, (c) HO [3], (d) TV [18], (e) ℓ_0 [25], (f) RHM [17], and (g) BF [39], respectively. The even rows are zoomed-in views.

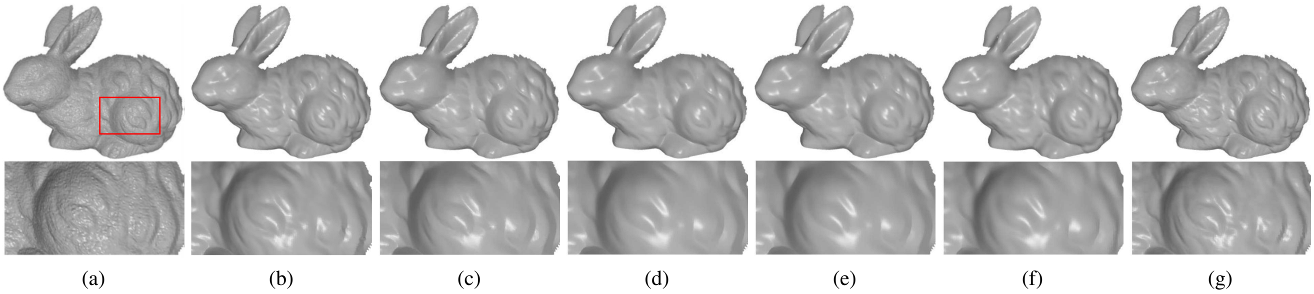


FIGURE 7: Comparisons of denoising results by the laser scanner. From left to right: (a) input noisy mesh, denoising results produced by (b) our method ℓ_p-ℓ_1, (c) HO [3], (d) TV [18], (e) ℓ_0 [25], (f) RHM [17], and (g) BF [39], respectively. The second row is the zoomed-in view.

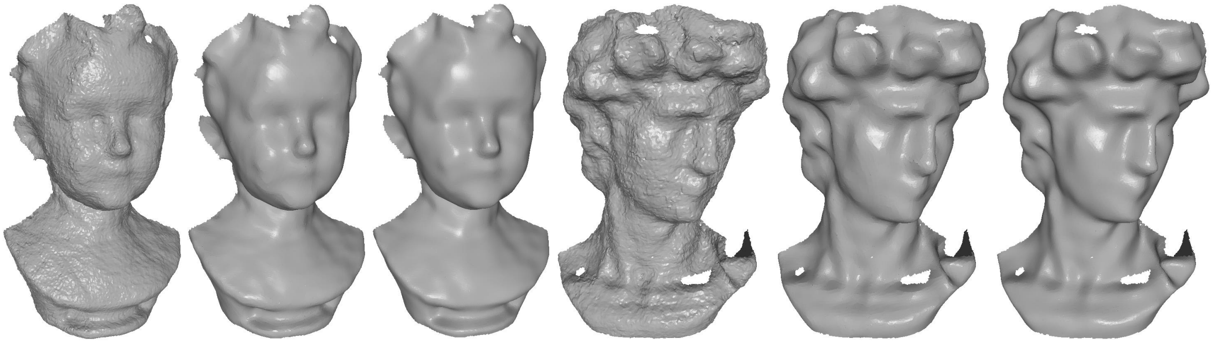


FIGURE 8: Denoising results of scanning data captured by Kinect sensors. From left to right: noisy mesh, denoising results produced by our method ℓ_p-ℓ_1 and the method proposed by Wang et al. [30], noisy mesh, and results produced by ours and the method proposed by Wang et al. [30].

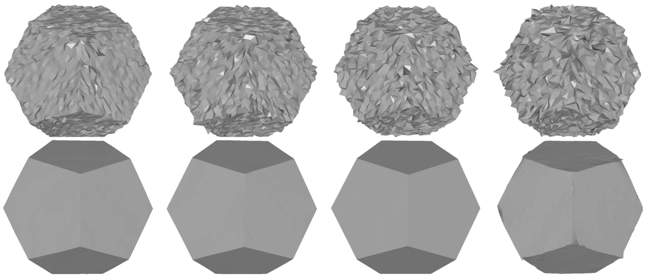


FIGURE 9: Denoising results of Dodecahedron corrupted by different levels of noise. The first row shows noisy meshes (corrupted by Gaussian noise, standard deviation = 0.3, 0.4, 0.5, and 0.6 mean edge length, respectively). The second row shows the corresponding results produced by our method.

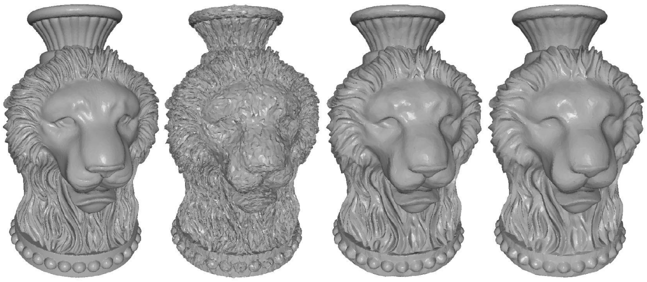


FIGURE 10: Comparison of denoising results on Vase-Lion (corrupted by Gaussian noise, standard deviation = 0.3 mean edge length). From left to right: clean mesh, noisy mesh, and results produced by our method ℓ_p-ℓ_1 and the method proposed by Zhong et al. [29].

meshes corrupted by impulsive or mixed noise.

The above visual comparisons present that our denoisng method has the best denoising result among all the tested methods. In the following paragraphs, we will compare our method to the other five quantitatively.

C. QUANTITATIVE COMPARISONS

In addition to the above qualitative comparisons, to show more objective comparisons, quantitative evaluations are carried out in this subsection. For meshes whose ground truth shapes are known, two widely-used error metrics that, mean square angular error (MSAE) and ℓ_2 vertex-based surface-to-surface error ($\mathrm{E}_{v,2}$), are employed to measure the deviation of the denoising result from the clean mesh [3], [29], [32]. These two error metrics are defined as follows:

- Mean square angular error (MSAE):

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

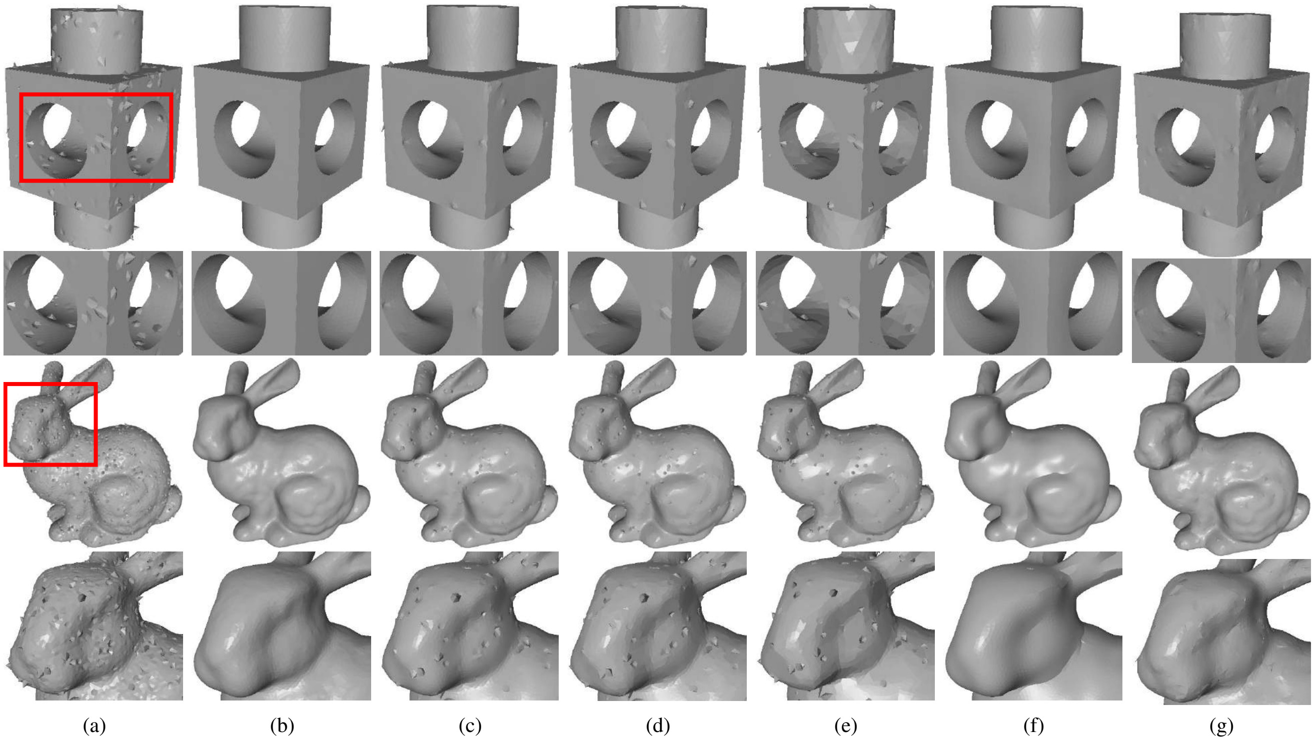


FIGURE 11: Comparisons of denoising results in handling impulsive or mixed noise. From left to right: (a) noisy meshes (Block is corrupted by 10 % of impulsive noise with standard deviation = 1.0 mean edge length, and Bunny is corrupted by 15% of impulsive noise with standard deviation = 1.0 mean edge length and Gaussian noise with standard deviation = 0.1 mean edge lenth), denoising results produced by (b) our method ℓ_p-ℓ_1, (c) HO [3], (d) TV [18], (e) ℓ_0 [25], (f) RHM [17], and (g) BF [39], respectively. The even rows are zoomed-in views.

TABLE 2: Quantitative evaluation results of Figs. 5, 6, and 11 for all the test methods including our method ℓ_p-ℓ_1, HO [3], TV [18], ℓ_0 [25], RHM [17], and BF [39].

	MSAE($\times10^{-3}$), $\mathrm{E}_{v,2}$($\times10^{-3}$); Time(in Seconds)					
Meshes	ℓ_p-ℓ_1	HO	TV	ℓ_0	RHM	BF
Fandisk	**3.51**, **1.76**; 5.44	4.88, 2.39; 1.18	4.13, 1.91; 0.68	5.09, 3.38; 20.1	5.77, 2.87; 8.88	5.31, 3.14; 0.25
Joint	**1.51**, **1.57**; 12.8	2.55, 1.72; 1.07	2.65, 1.61; 0.71	4.85, 2.19; 39.8	2.69, 1.78; 15.3	10.6, 3.41; 1.04
Bunny-Hi	**17.6**, **1.57**; 15.1	19.5, 2.42; 12.2	21.6, 1.77; 9.85	55.9, 2.83; 81.3	34.3, 21.2; 56.3	18.6, 1.62; 5.16
Child	**25.7**, 1.19; 14.5	31.5, 1.23; 11.8	40.5, 1.16; 6.73	99.1, 2.01; 57.4	48.8, 1.27; 34.1	33.9, **0.99**; 2.32
Block	**2.67**, **1.09**; 12.5	9.34, 9.16; 9.14	21.2, 11.6; 3.83	47.6, 17.1; 49.9	5.46, 3.16; 42.1	20.7, 4.88; 1.26
Bunny	**5.59**, **0.87**; 38.4	38.4, 4.15; 28.6	49.1, 4.32; 16.1	51.1, 7.16; 100	24.8, 17.4; 96.4	16.3, 2.02; 8.73

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

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

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

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 $v_i^{'}$, and $M_1(v_i)$ is the set of triangles containing vertex v_i.

The first metric is to estimate the mean square angular error between face normals of the clean mesh and those of the denoising result, and the second one is to measure the mean positional error between the clean mesh and the denoised one. Our method ℓ_p-ℓ_1 is compared to the other state-of-the-art methods by using the two metrics for the examples in Figs. 5, 6, and 11. The evaluation results are listed in Table 2.

As can be seen in Table 2, for both CAD and non-CAD meshes, our method ℓ_p-ℓ_1 outperforms all the other compared methods. The values of MSAE of our method are significantly smaller than the other five methods. From Table 2 we also observe that, in most cases, the results produced by our method ℓ_p-ℓ_1 have least errors of $\mathrm{E}_{v,2}$ showing that the results from our method are more faithful to the corresponding clean meshes. Besides, for the meshes corrupted by impulsive and mixed noise, the values of MSAE and $\mathrm{E}_{v,2}$ of our method are also significantly smaller than the other compared methods. These quantitative evaluations present that our method ℓ_p-ℓ_1 outperforms the other five compared state-of-the-art methods. Accordingly, the quantitative evaluation results are consistent with the qualitative comparisons, which show the superiority of the method proposed by us.

We also record the CPU costs of all the tested methods in Table 2. As we can see, ℓ_0 is the slowest method, while BF is the fastest one. Although RHM is faster than ℓ_0, it is slower than all the other methods. We also can observe that, the first order method (TV) is faster than the second order methods (HO and ℓ_p-ℓ_1) obviously. As expected, our method ℓ_p-ℓ_1 is slower than its convex counterpart (HO). In summary, although our method ℓ_p-ℓ_1 is a little computationally intensive, the CPU costs of it are still acceptable; see the CPU costs of our method in Table 2.

V. CONCLUSION

In this paper, we present a novel high order ℓ_p-ℓ_1 method for feature-preserving mesh denoising. The method applies the ℓ_p-ℓ_1 minimization on face normals, which is a challenge for optimization. We propose an iterative algorithm based on iteratively reweighted ℓ_1 minimization to solve the problem. Each subproblem in the iteration can be solved by augmented Lagrangian method. Then, vertex positions are reconstructed according to the filtered face normals. Various experiment results illustrate the advantages of our mesh denoising method over the state-of-the-art methods on both synthetic meshes and real scanning data.

Our mesh denoising method is somewhat computationally expensive for large meshes. Thus, how to decrease the CPU cost of our method is a future work.

REFERENCES

[1] C.-Y. Chen and K.-Y. Cheng, “A sharpness dependent filter for mesh smoothing,” Comput. Aided Geom. Des., vol. 22, no. 5, pp. 376–391, 2005.

[2] K. Hildebrandt and K. Polthier, “Anisotropic filtering of non-linear surface features,” Comput. Graph. Forum, vol. 23, no. 3, pp. 391–400, 2010.

[3] Z. Liu, R. Lai, H. Zhang, and C. Wu, “Triangulated surface denoising using high order regularization with dynamic weights,” SIAM J. Sci. Comput., vol. 41, no. 1, pp. 1–26, 2019.

[4] D. A. Field, “Laplacian smoothing and delaunay triangulations,” Commun. Appl. Numer. Methods, vol. 4, no. 6, pp. 709–712, 1988.

[5] 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.

[6] 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 Computer Graphics and Interactive Techniques, 1999, pp. 317–324.

[7] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, “Efficiently combining positions and normals for precise 3d geometry,” ACM Trans. Graph., vol. 24, no. 3, pp. 536–543, 2005.

[8] 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, 2000, pp. 145–152.

[9] G. Taubin, “Linear anisotropic mesh filters,” IBM Research Report RC22213(W0110-051), IBM T.J. Watson Research, October 2001.

[10] H. Yagou, Y. Ohtake, and A. Belyaev, “Mesh smoothing via mean and median filtering applied to face normals,” in Geom. Model. and Proc., 2002. Proceedings, 2002, pp. 124–131.

[11] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, “Geometric surface smoothing via anisotropic diffusion of normals,” in Proceedings of the Conference on Visualization, 2002, pp. 125–132.

[12] C. Bajaj and G. Xu, “Anisotropic diffusion of surfaces and functions on surfaces,” ACM Trans. Graph., vol. 22, no. 1, pp. 4–32, 2003.

[13] C. Wang, “Bilateral recovering of sharp edges on feature-insensitive sampled meshes,” IEEE Trans. Vis. Comput. Graph., vol. 12, no. 4, pp. 629–639, 2006.

[14] Y. Zheng, H. Fu, O.-C. Au, and C.-L. Tai, “Bilateral normal filtering for mesh denoising,” IEEE Trans. Vis. Comput. Graph., vol. 17, no. 10, pp. 1521–1530, 2011.

[15] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu, “Guided mesh normal filtering,” Comput. Graph. Forum., vol. 34, no. 7, pp. 23–34, 2015.

[16] M. Wei, J. Yu, W.-M. Pang, J. Wang, J. Qin, L. Liu, and P.-A. Heng, “Bi-normal filtering for mesh denoising,” IEEE Trans. Vis. Comput. Graph., vol. 21, no. 1, pp. 43–55, 2015.

[17] S. Yadav, U. Reitebuch, and K. Polthier, “Robust and high fidelity mesh denoising,” IEEE Trans. Vis. Comput. Graph., vol. 25, no. 6, pp. 2304–2310, 2019.

[18] 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., vol. 21, no. 7, pp. 873–886, 2015.

[19] X. Lu, W. Chen, and S. Schaefer, “Robust mesh denoising via vertex pre-filtering and ℓ_1-median normal filtering,” Comput. Aided Geom. Des., vol. 54, pp. 49–60, 2017.

[20] M. Wei, L. Liang, W. M. Pang, J. Wang, W. Li, and H. Wu, “Tensor voting guided mesh denoising,” IEEE Trans. Autom. Sci. Eng., vol. 14, no. 2, pp. 931–945, 2017.

[21] M. Wei, H. Jin, X. Xie, L. Liu, and Q. Jing, “Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery,” IEEE Trans. Vis. Comput. Graph., pp. 1–1, 2019.

[22] J. Zhang, B. Deng, Y. Hong, Y. Peng, W. Qin, and L. Liu, “Static/dynamic filtering for mesh geometry,” IEEE Trans. Vis. Comput. Graph., vol. pp, no. 14, pp. 1–14, 2018.

[23] S. Yadav, U. Reitebuch, and K. Polthier, “Mesh denoising based on normal voting tensor and binary optimization,” IEEE Trans. Vis. Comput. Graph., vol. 24, no. 8, pp. 2366–2379, 2018.

[24] H. Chen, J. Huang, O. Remil, H. Xie, J. Qin, Y. Guo, M. Wei, and J. Wang, “Structure-guided shape-preserving mesh texture smoothing via joint low-rank matrix recovery,” Comput-Aided Des. (Proceedings of Solid and Physical Modeling), vol. 115, pp. 122–134, 2019.

[25] L. He and S. Schaefer, “Mesh denoising via ℓ_0 minimization,” ACM Trans. Graph., vol. 32, no. 4, pp. 1–8, 2013.

[26] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” Physica D, vol. 60, pp. 259–268, 1992.

[27] X. Wu, J. Zheng, Y. Cai, and C.-W. Fu, “Mesh denoising using extended rof model with ℓ_1 fidelity,” Comput. Graph. Forum, vol. 34, no. 7, pp. 35–45, 2015.

[28] S. Zhong, Z. Xie, W. Wang, Z. Liu, and L. Liu, “Mesh denoising via total variation and weighted laplacian regularizations,” Comput. Animat. Virt. W., vol. 29, no. 3-4, p. e1827, 2018.

[29] S. Zhong, Z. Xie, J. Liu, and Z. Liu, “Robust mesh denoising via triple sparsity,” Sensors, vol. 19, p. 1001, 2019.

[30] P.-S. Wang, Y. Liu, and X. Tong, “Mesh denoising via cascaded normal regression,” ACM Trans. Graph., vol. 35, no. 6, pp. 232:1–232:12, 2016.

[31] J. Wang, J. Huang, F. L. Wang, M. Wei, H. Xie, and J. Qin, “Data-driven geometry-recovering mesh denoising,” Comput-Aided Des. (Proceedings of Solid and Physical Modeling), vol. 114, pp. 133–142, 2019.

[32] Z. Liu, S. Zhong, Z. Xie, and W. Wang, “A novel anisotropic second order regularization for mesh denoising,” Comput. Aided Geom. Des., vol. 71, pp. 190–201, 2019.

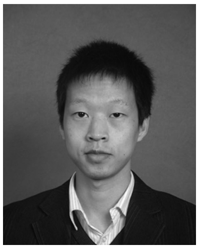
[33] P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock, “On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision,” SIAM J. Imaging Sci., vol. 8, no. 1, pp. 331–372, 2015.

[34] M. Nikolova, “Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers,” SIAM J. Numer. Anal., vol. 40, no. 3, pp. 965–994, 2003.

[35] E. J. Candès, M. B. Wakin, and S. P. Boyd, “Enhancing sparsity by reweighted ℓ_1 minimization,” J. Fourier Anal. Appl., vol. 14, no. 5, pp. 877–905, 2008.

[36] 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., vol. 3, no. 3, pp. 300–339, 2010.
[37] X. Sun, P. Rosin, R. Martin, and F. Langbein, “Fast and effective feature-preserving mesh denoising,” IEEE Trans. Vis. Comput. Graph., vol. 13, no. 5, pp. 925–938, 2007.
[38] Z. Liu, H. Zhang, and C. Wu, “On geodesic curvature flow with level set formulation over triangulated surfaces,” J. Sci. Comput., vol. 70, no. 2, pp. 631–661, 2017.
[39] Y. Zheng, H. Fu, 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, 2011.

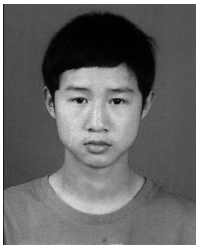
ZHENG LIU received the B.Sc. and M.Sc. degrees in computer science and technology from China University of Geosciences(Wuhan), in 2006 and 2009, respectively, and the Ph.D. degree in instructional technology from Central China Normal University, in 2012. He is currently a Lecturer with National Engineering Research Center of Geographic Information System, China University of Geosciences(Wuhan). From 2013 to 2014, he held a post-doctoral position with School of Mathematical Sciences, University of Science and Technology of China. His main interests include digital geometry processing, image and video processing.



MINGQIANG GUO received the B.Sc. degrees in computer science and technology from China University of Geosciences(Wuhan), in 2007, and the Ph.D. degree in geographic information engineering from China University of Geosciences(Wuhan), in 2013. He is currently an associate professor in School of Geography and Information Engineering, China University of Geosciences(Wuhan). From 2013 to 2015, he held a post-doctoral position with China University of Geosciences(Wuhan). His main interests include vector data rendering and processing.

ZHONG XIE is currently a professor in School of Geography and Information Engineering, China University of Geosciences(Wuhan) . His main research interests include spatial analysis modeling and application, deep learning and so on.

JINQIN LIU received the B.E. degree from Central South University of Forestry and Technology, in 2017, where she is currently pursuing the master’s degree with the Geomatics. She is currently with the School of Geography and Information Engineering, China University of Geosciences(Wuhan), where she is currently involved in digital geometry processing.



BOHONG ZENG received the B.Eng. degree from Huazhong Agricultural University in 2018, and he is currently pursuing the master’s degree with the Department of Software Engineering. He is currently with the National Engineering Research Center for Geographic Information System, China University of Geosciences(Wuhan), where he is currently involved in sparse optimization and mesh filtering.