

Robust and effective mesh denoising using L_0 sparse regularization[☆]

Yong Zhao^{a,b,*}, Hong Qin^{b,**}, Xueying Zeng^a, Junli Xu^c, Junyu Dong^d

^a School of Mathematical Sciences, Ocean University of China, Qingdao 266100, China

^b Department of Computer Science, State University of New York at Stony Brook, NY 11794-4400, USA

^c School of Mathematics and Physics, Qingdao University of Science and Technology, Qingdao 266061, China

^d Department of Computer Science, Ocean University of China, Qingdao 266100, China



ARTICLE INFO

Article history:

Received 2 May 2017

Accepted 1 April 2018

Keywords:

Mesh denoising

L_0 norm

Sparse regularization

Non-convex optimization

Multi-layer approach

ABSTRACT

Mesh denoising is of great practical importance in geometric analysis and processing. In this paper we develop a novel L_0 sparse regularization method to robustly and reliably eliminate noises while preserving features with theoretic guarantee, and our assumption is that, local regions of a noise-free shape should be smooth unless they contain geometric features. Both vertex positions and facet normals are integrated into the L_0 norm to measure the sparsity of geometric features, and are then optimized in a sparsity-controllable fashion. We design an improved alternating optimization strategy to solve the L_0 minimization problem, which is proved to be both convergent and stable. As a result, our sparse regularization exhibits its advantage to distinguish features from noises. To further improve the computational performance, we propose a multi-layer approach based on joint bilateral upsampling to handle large and complicated meshes. Moreover, the aforementioned framework is naturally accommodating the need of denoising time-varying mesh sequences. Both theoretical analysis and various experimental results on synthetic and natural noises have demonstrated that, our method can robustly recover multifarious features and smooth regions of 3D shapes even with severe noise corruption, and outperform the state-of-the-art methods.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

In recent years, 3D shape scanning devices are widely employed to capture digital surface data. However, even with high-precision scanners, the acquired mesh data inevitably contains noises for various reasons. Therefore, the raw data must be denoised to improve their quality for downstream shape analysis and editing.

Mesh denoising has been a very difficult problem in the fields of computer graphics and computer vision. Features and noises are ambiguous because both of them are of high-frequency from the perspective of signal processing. In some cases, even human beings may fail to distinguish features from noises. Removing noises while retaining features is a great challenge. A wealth of mesh denoising algorithms have been proposed in the past two decades [1–10]. Most of them smooth the mesh by averaging the geometry information within local neighborhood. Since the denoising principle is heuristic, these algorithms typically suffer from feature blurring, shape shrinkage, or vertex drifts.

Unlike traditional algorithms relying on local operations, the L_0 minimization can globally maintain salient features. Xu et al. [11] applied the L_0 gradient minimization to deal with edge-preserving image smoothing problem. The L_0 norm can effectively determine the sparsity of image gradient (i.e., the number of non-zero gradients), which reflects the location of prominent edges. Building a new differential operator with respect to vertex positions, He et al. [12] then extended this approach to achieve feature-preserving mesh denoising.

Our algorithm is also based on the L_0 minimization. The key observation is that, local region of a noise-free mesh should be smooth except that it contains geometric features. Actually, vertex positions and facet normals are complementary to each other, and adopting them together can effectively describe 3D meshes. It might overlook some intrinsic surface properties if they were to be adopted separately. This observation is also validated by [13], which integrates positions and normals to obtain precise 3D geometry. In this paper, vertex positions and facet normals are combined together to measure the sparsity of geometric features, thus being capable of decoupling features and noises. The proposed sparse regularization is simple yet powerful to recover the global structures and local details of 3D shapes even in the presence of highly-corrupted noises. Fig. 1 compares our algorithm with that of [12]. It is difficult to handle mixed noises. Sharp edges are blurred in the

[☆] This paper has been recommended for acceptance by Shi-Min Hu.

* Corresponding author at: School of Mathematical Sciences, Ocean University of China, Qingdao 266100, China.

** Corresponding author.

E-mail addresses: yongzhao.ouc@gmail.com (Y. Zhao), qin@cs.stonybrook.edu (H. Qin).

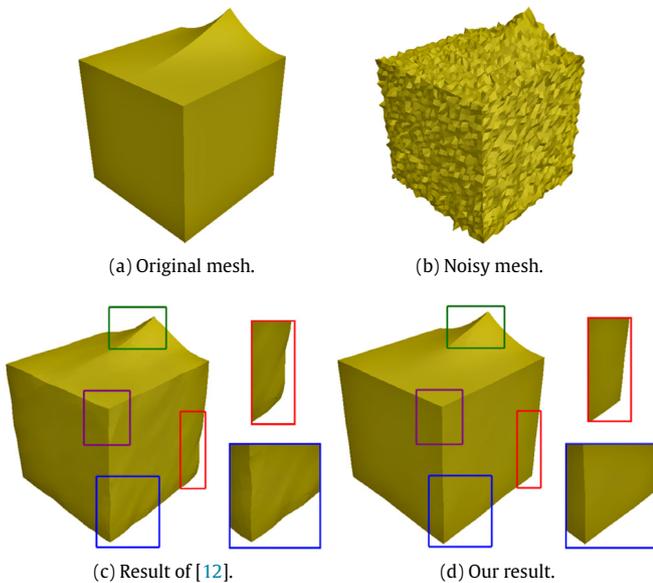


Fig. 1. Comparison with L_0 denoising algorithm [12] on a mesh corrupted by $0.3l_e$ Gaussian noise along normal directions and $0.1l_e$ impulsive noise along random directions. l_e is the average edge length of the original mesh. Please note that, our algorithm recovers sharp edges better.

result of [12], thus causing apparent visual artifacts. Our algorithm preserves these edges well, and generates a satisfactory result.

It is computationally intractable to perform the L_0 minimization due to its high non-convexity. A popular way is to replace the L_0 norm with the L_1 norm. As proved in [14], the L_1 norm is the closest relaxation of the L_0 norm under some conditions. However, this equivalence does not always hold, and the L_0 norm can produce a more sparse solution than the L_1 norm. We improve the alternating optimization strategy [11,12] to address the L_0 norm directly. Through introducing a sequence of auxiliary variables, the non-convex minimization problem is rewritten and separated into two sub-problems, both of which can be easily solved. And then, the optimal solution is sought by optimizing the sub-problems in a two-level iterative framework. Compared with the original solver [11,12], our improved numerical solver achieves guaranteed convergence and stability (see the Appendix for the details of our proof).

Because it is time-consuming to solve the L_0 minimization problem, the denoising can be carried out in a multi-layer way to further improve the performance. A multi-layer representation is first built by conducting clustering-based simplification. And then the denoising can be efficiently performed at the coarsest layer. Finally, we generalize joint bilateral upsampling [15] from images to meshes to obtain the denoising result of the input mesh.

Furthermore, our algorithm can be naturally extended to deal with the denoising of time-varying mesh sequences. Different from static mesh, time-varying mesh sequence has an additional temporal dimension. Usually, it is represented by a group of triangular meshes. It is possible to constrain the temporal coherence during denoising.

In comparison with previous methods, the main contributions of this paper can be summarized as follows:

- A novel sparse regularization is proposed to measure the sparsity of geometric features and distinguish features from noises. Both vertex positions and facet normals are optimized in a L_0 framework to faithfully remove noises and preserve features.

- An improved alternating optimization strategy is developed to address the resulting L_0 minimization problem. Benefiting from the proved convergence and stability, our improved strategy is robust even for meshes of deteriorating quality and with high noise rate.
- A new multi-layer approach is presented to greatly improve the performance, thus facilitating the denoising of large-scale meshes. Joint bilateral upsampling is very efficient and easy to implement.

2. Related work

Towards the goal of obtaining high-fidelity geometry data, mesh denoising has received much attention in computer graphics and computer vision. There exists a large number of denoising algorithms (please refer to the survey [16] for more details).

Feature-preserving Denoising. Early methods intended to perform surface fairing in an isotropic way. Taubin et al. [17] and Vollmer et al. [18] employed the uniformly weighted Laplacian operator. Since the mesh irregularity is neglected, apparent feature distortion and volumetric shrinkage would appear. Liu et al. [19] further offered volume-preserving results. Desbrun et al. [1] extended the method of [17] to irregular meshes, and proposed an implicit integration scheme. Several researchers also introduced global smoothing methods [20–22]. They filtered the Laplacian coordinates and reconstructed the mesh through solving a system of equations. Other methods [23,24,25] employed anisotropic diffusion equations to better maintain shape features. Even if visually-pleasing results could be produced, their apparent weakness is the numerical instability during diffusion.

Bilateral filter is a seminar work in the literature. Inspired by the bilateral filter method of [26] in image processing, Fleishman et al. [3] and Jones et al. [4] extended its concept to denoise 3D meshes, respectively. The technical core of bilateral filter is to take both spatial difference and signal difference into account. Fleishman et al. [3] adjusted mesh vertices along normal directions to prevent vertex-drift. Jones et al. [4] relied on robust estimation of vertex positions. Schall et al. [27] devised a non-local scheme for static and time-varying range data. Solomon et al. [28] generalized bilateral filter to smooth signals on any domain that admits a Laplacian operator.

Considering that facet normals convey higher order surface properties than vertex positions, a two-stage denoising scheme is explored: facet normals are first filtered, and then vertex positions are reconstructed by fitting the filtered facet normals. Since the second stage is straightforward, the first stage plays a significant role in denoising results. Ohtake et al. [29] applied Gaussian filter to facet normals. Shen et al. [30] introduced a fuzzy vector median filter. When calculating local weights, Sun et al. [5] assigned null weight to neighboring facet normals with larger variation to retain sharp edges, and later they used random walks [31]. Zheng et al. [7] employed bilateral filter to better smooth facet normals. Using the average normal in the local patch as the guidance, Zhang et al. [9] adopted joint bilateral filter to improve the above approach. Wang et al. [32] smoothed out small-scale features and kept large-scale features via rolling guidance filter. Yadav et al. [33] utilized normal voting tensor and binary optimization. Wang et al. [34] trained cascaded non-linear regression functions to learn facet normals.

Moreover, a few other methods [35,6,36,37,8,10,38] tried to classify mesh vertices into different categories, and then carried out the denoising in each category independently of others to achieve feature-preservation. Vertex classification is mainly based on volume integral invariant, surface approximation quality, facet normals variation or facet normal tensor voting. Nevertheless, as

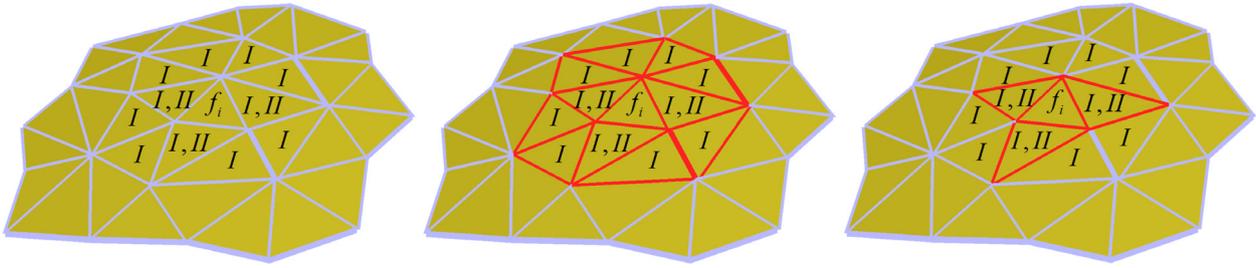


Fig. 2. Demonstration of 1-ring facet neighborhood and corresponding edge sets of f_i . From left to right: two kinds of 1-ring facet neighborhoods represented by I and II , $E_{f_i}^I$ highlighted in red, and $E_{f_i}^{II}$ highlighted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

stated in [10], the presence of high noises might give rise to poor classification results.

Sparsity-driven Denoising. Recently, compressed sensing has become an active subject in signal processing and computer science [39,14,40]. Its profound insight is that, most signals are sparse, i.e., they could be characterized by a small number of features. Sparsity has been adopted in geometry processing to achieve better results [41,42]. Avron [43] reconstructed piecewise smooth point-sampled surfaces under the L_1 norm. Wang et al. [44] extracted a base mesh by a global Laplacian regularization, and then recovered sharp features by the L_1 optimization. Wu et al. [45] utilized the L_1 norm to enhance ROF model.

The sparsity of the L_0 norm is better than that of the L_1 norm. For image smoothing, Xu et al. [11] designed a L_0 gradient minimization framework and an alternating optimization strategy. Later on, He et al. [12] generalized this idea to denoise 3D meshes by developing a differential operator with respect to vertex positions. Sun et al. [46] performed the denoising of point clouds in three stages. However, the optimization strategy of [11] is not guaranteed to be convergent. Cheng et al. [47] further introduced a fused coordinate descent method to solve the L_0 minimization problem. In [46,47], surface normals and positions were reconstructed successively. Different from existing L_0 constraints, our constraint optimizes vertex positions and facet normals together to fully respect the intrinsic surface properties.

3. New robust and effective mesh denoising algorithm

In the followings, the noisy triangular mesh is denoted as $M = \{P, F\}$, where $P = \{\mathbf{p}_i^* \in \mathbb{R}^3 | 1 \leq i \leq n\}$ is the set of n mesh vertices, and $F = \{f_i | 1 \leq i \leq m\}$ is the set of m facets. This paper adopts the L_0 norm to decouple features from noises. For a vector \mathbf{x} , its L_0 norm is defined as the number of non-zero elements: $\|\mathbf{x}\|_0 = \#\{i | x_i \neq 0\}$, where x_i is an element of \mathbf{x} , and $\#\{\}$ is the counting operator.

3.1. L_0 Sparse regularization

It can be observed that a noise-free mesh consists of geometric features and smooth regions. In other words, local region should be smooth unless it contains geometric features. Therefore, for a noisy mesh, our goal is to minimize the surface variation except at geometric features. Vertex positions are the basics of 3D meshes. Facet normals are well-defined, and convey more information. Vertex positions and facet normals are complementary to each other, and collectively reflect the intrinsic properties of the surface. Intrinsic property means underlying shape characteristic that can be used to faithfully recover the surface. It might be likely to overlook some surface properties if vertex positions and facet normals were to be adopted separately. That is, adopting them together can better describe 3D meshes. This observation is also validated by [13], which combines positions and normals to reconstruct precise geometry information. As demonstrated in [13], positions and normals can

improve each other and produce better results than using one of them individually.

We integrate vertex positions and facet normals via the L_0 norm to measure the local smoothness of the mesh and the sparsity of geometric features. Suppose f_i is a facet, and \mathbf{n}_{f_i} is its unit normal. Following [5], there are two kinds of 1-ring facet neighborhood for f_i . The first kind is the set of the facets that share common vertices with f_i , and the second kind is the set of the facets that share common edges with f_i . If local region around f_i is flat, \mathbf{n}_{f_i} should be perpendicular to the neighboring facets of f_i , that is, \mathbf{n}_{f_i} should be perpendicular to the edges of each neighboring facet. If local region around f_i contains geometric features, the perpendicularity does not hold. This constraint can be formulated as

$$\sum_{E_{f_i}} \|\mathbf{n}_{f_i} \cdot \frac{(\mathbf{p}_k - \mathbf{p}_l)}{\|\mathbf{p}_k - \mathbf{p}_l\|_2}\|_0, \quad (1)$$

where E_{f_i} is the edge set of the neighboring facets of f_i , \mathbf{p}_k and \mathbf{p}_l are the two vertices of an edge in E_{f_i} , and \cdot is the inner-product operator. Because of the definition of 1-ring facet neighborhood, E_{f_i} can also be decomposed into two types. Let $E_{f_i}^I, E_{f_i}^{II}$ be the edge set under the two types of 1-ring facet neighborhood, respectively. As shown in Fig. 2, $E_{f_i}^I \supset E_{f_i}^{II}$. Especially, we use $\|\mathbf{p}_k - \mathbf{p}_l\|_2$ to normalize the edge vector to deal with non-uniform sampling of 3D meshes, because all the edges in E_{f_i} contribute to the above constraint equally. Fig. 3 demonstrates the effect of edge normalization on a non-uniform mesh. We obtain a better result with edge normalization.

We set all vertex positions $\{\mathbf{p}_i\}_{i=1}^n$ and facet normals $\{\mathbf{n}_{f_i}\}_{i=1}^m$ as vector \mathbf{p} and vector \mathbf{n} , respectively. Considering all facets in M , the sparse regularization term is written as

$$E_{sr}(\mathbf{p}, \mathbf{n}) = \sum_{i=1}^m w_{f_i} \sum_{E_{f_i}} \|\mathbf{n}_{f_i} \cdot \frac{(\mathbf{p}_k - \mathbf{p}_l)}{\|\mathbf{p}_k - \mathbf{p}_l\|_2}\|_0, \quad (2)$$

where w_{f_i} is the weight for f_i . Because a triangular mesh approximates an underlying surface, each facet represents a patch on this surface. Larger facets should be more important than smaller ones. Therefore, we define w_{f_i} as the area of facet f_i . $E_{sr}(\mathbf{p}, \mathbf{n})$ constrains the smoothness of local regions and the sparsity of geometric features, thus giving rise to reliable noise removal and feature preservation.

In addition, the denoising result should be structurally similar to the original data. Hence, we adopt the following constraint as the data fidelity term

$$E_{ft}(\mathbf{p}, \mathbf{n}) = \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{p}_i^*\|_2^2 + \eta \sum_{i=1}^m \|\mathbf{n}_{f_i} - \mathbf{n}_{f_i}^*\|_2^2, \quad (3)$$

where $\mathbf{n}_{f_i}^*$ is the original unit normal of f_i , and η is a weight. $E_{ft}(\mathbf{p}, \mathbf{n})$ describes the similarity between the noisy mesh and the denoised mesh.

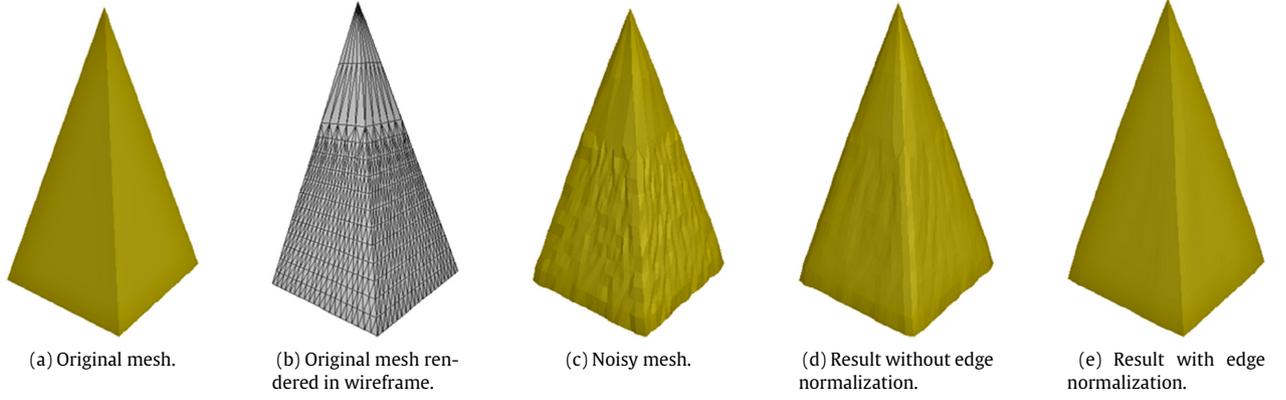


Fig. 3. Effect of edge normalization when handling non-uniform sampling. Please note that, shape features are better retained using edge normalization.

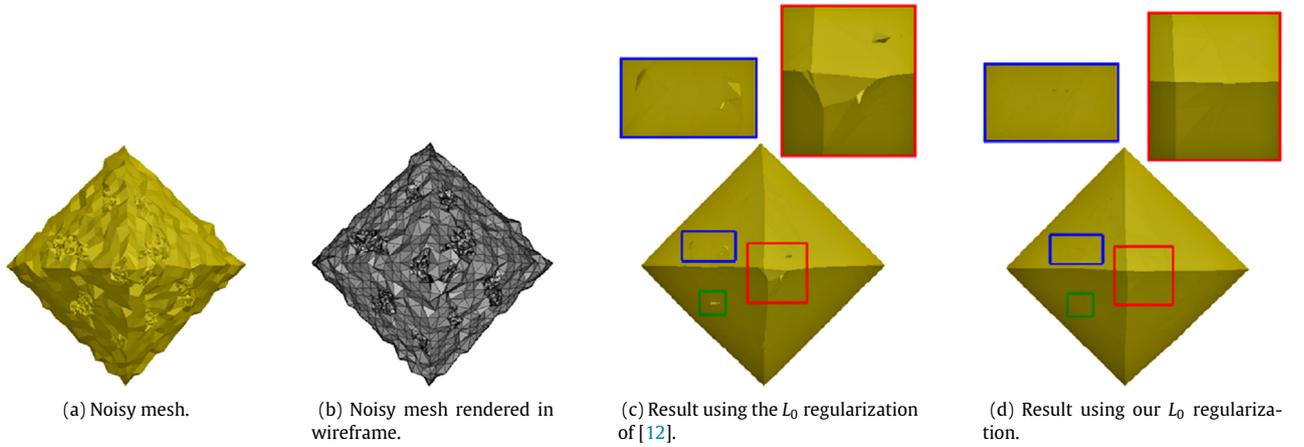


Fig. 4. Comparison between our L_0 regularization and that of [12] on a non-uniform mesh corrupted by $0.5l_e$ Gaussian noise along random directions and $0.3l_e$ impulsive noise along random directions. For a fair comparison, we solve both L_0 minimization problems using the improved optimization strategy in Section 3.2. Please note that, our L_0 regularization is capable of distinguishing features from mixed noises, thus obtaining a better result.

Finally, our energy minimization problem is formulated as

$$\min_{\{\mathbf{p}, \mathbf{n}\}} \{E_{ft}(\mathbf{p}, \mathbf{n}) + \lambda E_{sr}(\mathbf{p}, \mathbf{n})\}, \quad (4)$$

where λ is a weight balancing the two energy terms. In fact, λ is a smoothing parameter, and a larger λ would yield a smoother result.

In Fig. 4, we compare our L_0 regularization with that of [12] on a non-uniform mesh corrupted by mixed noises. Especially, both L_0 minimization problems are solved by our improved alternating optimization strategy in the next subsection. From the close-up view of local regions, our L_0 regularization is able to decouple features and noises and generate a visually-pleasing result. In contrast, the L_0 regularization of [12] does not describe the mesh effectively, thus failing to distinguish features from mixed noises. Even with the help of our improved optimization strategy, visual artifacts still appear in some highly-corrupted regions.

3.2. Improved alternating optimization strategy

It is NP-hard to optimize the L_0 norm due to the problems caused by its non-convexity. In this paper, the alternating optimization strategy [11,12] is improved to directly deal with the L_0 minimization problem.

First of all, a sequence of auxiliary variables are introduced to rewrite the energy minimization problem (4) as

$$\begin{aligned} \min_{\{\mathbf{p}, \mathbf{n}, \delta\}} \{ & \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{p}_i^*\|_2^2 + \eta \sum_{i=1}^m \|\mathbf{n}_{f_i} - \mathbf{n}_{f_i}^*\|_2^2 \\ & + \beta \sum_{i=1}^m w_{f_i} \sum_{E_{f_i}} \|\mathbf{n}_{f_i} \cdot \frac{(\mathbf{p}_k - \mathbf{p}_l)}{\|\mathbf{p}_k - \mathbf{p}_l\|_2} - \delta_{f_i}^{kl}\|_2^2 + \lambda \|\delta\|_0 \}, \end{aligned} \quad (5)$$

where $\delta_{f_i}^{kl}$ is the auxiliary variable for an edge in E_{f_i} , δ is a vector consisting of all auxiliary variables, and β is a weight that controls the similarity between $\delta_{f_i}^{kl}$ and $\mathbf{n}_{f_i} \cdot \frac{(\mathbf{p}_k - \mathbf{p}_l)}{\|\mathbf{p}_k - \mathbf{p}_l\|_2}$. Obviously, when β is large enough, the solution of the minimization problem (5) infinitely approximates that of the minimization problem (4).

The energy minimization problem (5) can be decomposed into two computationally tractable sub-problems. Omitting the terms not involving δ , the first sub-problem is to minimize δ with \mathbf{p} and \mathbf{n} fixed

$$\min_{\{\delta\}} \{ \beta \sum_{i=1}^m w_{f_i} \sum_{E_{f_i}} \|\mathbf{n}_{f_i} \cdot \frac{(\mathbf{p}_k - \mathbf{p}_l)}{\|\mathbf{p}_k - \mathbf{p}_l\|_2} - \delta_{f_i}^{kl}\|_2^2 + \lambda \|\delta\|_0 \}. \quad (6)$$

Fortunately, this apparently sophisticated sub-problem can be spatially decomposed to compute each element of δ independently.

Algorithm 1 Alternating Optimization Strategy of [11,12].

Input: noisy mesh with vertex positions $\mathbf{p}^* = (\mathbf{p}_1^*, \mathbf{p}_2^*, \dots, \mathbf{p}_n^*)$ and facet normals $\mathbf{n}^* = (\mathbf{n}_{f_1}^*, \mathbf{n}_{f_2}^*, \dots, \mathbf{n}_{f_m}^*)$, weights λ, η , parameters β_0, β_{max} , and constant κ

Initialization: $\mathbf{p} \leftarrow \mathbf{p}^*, \mathbf{n} \leftarrow \mathbf{n}^*, \beta \leftarrow \beta_0$

repeat

fix $\mathbf{p}^{(t)}$ and $\mathbf{n}^{(t)}$, solve for $\delta^{(t+1)}$ with (7)

fix $\delta^{(t+1)}$, solve for $\mathbf{p}^{(t+1)}$ and $\mathbf{n}^{(t+1)}$ with (8)

$\beta \leftarrow \kappa \beta$

until $\beta \geq \beta_{max}$

Output: denoising result

According to [11], each element $\delta_{f_i}^{kl}$ has a closed-form solution

$$\delta_{f_i}^{kl} = \begin{cases} 0 & w_{f_i} \|\mathbf{n}_{f_i} \cdot \frac{(\mathbf{p}_k - \mathbf{p}_l)}{\|\mathbf{p}_k - \mathbf{p}_l\|_2}\|_2 \leq \frac{\lambda}{\beta} \\ \mathbf{n}_{f_i} \cdot \frac{(\mathbf{p}_k - \mathbf{p}_l)}{\|\mathbf{p}_k - \mathbf{p}_l\|_2} & otherwise \end{cases} \quad (7)$$

Omitting the term not involving \mathbf{p} and \mathbf{n} , the second sub-problem is to minimize \mathbf{p} and \mathbf{n} with δ fixed

$$\min_{\{\mathbf{p}, \mathbf{n}\}} \left\{ \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{p}_i^*\|_2^2 + \eta \sum_{i=1}^m \|\mathbf{n}_{f_i} - \mathbf{n}_{f_i}^*\|_2^2 + \beta \sum_{i=1}^m w_{f_i} \sum_{E_{f_i}} \|\mathbf{n}_{f_i} \cdot \frac{(\mathbf{p}_k - \mathbf{p}_l)}{\|\mathbf{p}_k - \mathbf{p}_l\|_2} - \delta_{f_i}^{kl}\|_2^2 \right\}. \quad (8)$$

As \mathbf{p} and \mathbf{n} are coupled, we iteratively optimize this sub-problem in two steps: minimizing \mathbf{p} with \mathbf{n} fixed, and minimizing \mathbf{n} with \mathbf{p} fixed. This is algorithmically similar to the Expectation–Maximization (EM) approach. Thanks to many good theoretical properties, the EM-like algorithms have been successfully employed in computer graphics [48,49]. When \mathbf{p} is fixed, Eq. (8) simply becomes a linear quadratic minimization problem with respect to \mathbf{n} , and can be solved easily. When \mathbf{n} is fixed, Eq. (8) is still non-linear with respect to \mathbf{p} due to $\|\mathbf{p}_k - \mathbf{p}_l\|_2$. We also iteratively optimize it in two steps: solving \mathbf{p} with $\|\mathbf{p}_k - \mathbf{p}_l\|_2$ fixed, and updating $\|\mathbf{p}_k - \mathbf{p}_l\|_2$. This process will not stop until the relative error between two successive iterations is below a small user-satisfied threshold. Hence, this non-linear problem can be approximated by a set of linear problems, which is a well-known way to optimize non-linear minimization problems [50,51]. In our experiments, the iteration converges very fast. This is because, the above problem has a complex form, fixing $\|\mathbf{p}_k - \mathbf{p}_l\|_2$ is a practical way to make it computationally tractable. Therefore, no matter \mathbf{p} or \mathbf{n} is fixed in Eq. (8), another variable can be solved effectively. This evidence can guarantee the convergence of Eq. (8). As for the iteration issue for Eq. (8), we also observe fast convergence in the experiments.

In [11,12], the two sub-problems (i.e., Eqs. (6) and (8)) are alternately optimized in a one-level iterative framework shown in Algorithm 1. The weight λ is unchanged during iteration. Starting from a small value β_0 , the weight β is gradually increased by multiplying a constant $\kappa > 1$ at each iteration until its value exceeds β_{max} . And the solution at the current iteration is set as the initial value for the next iteration with updated β . β is to finally force $\mathbf{n}_{f_i} \cdot \frac{(\mathbf{p}_k - \mathbf{p}_l)}{\|\mathbf{p}_k - \mathbf{p}_l\|_2}$ to match with $\delta_{f_i}^{kl}$, thus controlling the sparsity of the regularization term (2) when it becomes large enough.

However, this framework is not guaranteed to be convergent, and leads to visual artifacts. With the weights λ and $\beta^{(r)}$ (i.e., the current value of β) at the r th iteration, the energy minimization problem (5) is non-convex. It is impossible to find the optimal or stable solution if the two sub-problems are solved only once. Hence, the solution at the current iteration is not a good initial

Algorithm 2 Improved Alternating Optimization Strategy.

Input: noisy mesh with vertex positions $\mathbf{p}^* = (\mathbf{p}_1^*, \mathbf{p}_2^*, \dots, \mathbf{p}_n^*)$ and facet normals $\mathbf{n}^* = (\mathbf{n}_{f_1}^*, \mathbf{n}_{f_2}^*, \dots, \mathbf{n}_{f_m}^*)$, weights λ, α, η , parameters $\beta_0, \beta_{max}, t_{max}$, and constant κ

Initialization: $\mathbf{p} \leftarrow \mathbf{p}^*, \mathbf{n} \leftarrow \mathbf{n}^*, \beta \leftarrow \beta_0$

repeat

$t \leftarrow 0$

repeat

fix $\mathbf{p}^{(t)}$ and $\mathbf{n}^{(t)}$, solve for $\delta^{(t+1)}$ with (10)

fix $\delta^{(t+1)}$, solve for $\mathbf{p}^{(t+1)}$ and $\mathbf{n}^{(t+1)}$ with (8)

$t++$

until $t \geq t_{max}$

$\beta \leftarrow \kappa \beta$

until $\beta \geq \beta_{max}$

Output: denoising result

value for the next iteration with updated β . Numerical error will accumulate during iteration, and may give rise to a wrong result with this framework eventually.

From the analysis above, the improvement strategy is intuitive: at each iteration (e.g., with the weights λ and $\beta^{(r)}$), the optimizations of the sub-problems should alternate until convergence. As shown in Algorithm 2, we present a two-level iterative framework for convergence improvement. The outer-level iteration corresponds to the update of β . The inner-level iteration corresponds to the alternating optimization of the two sub-problems, which is controlled by the parameter t .

To achieve guaranteed convergence and stability, a relaxation term should be introduced for the first sub-problem (6) in the inner-level iteration [52,53]. That is, when fixing $\mathbf{p}^{(t)}$ and $\mathbf{n}^{(t)}$ at the t th iteration, $\delta^{(t+1)}$ is obtained by solving the new sub-problem

$$\min_{\{\delta\}} \left\{ \beta \sum_{i=1}^m w_{f_i} \sum_{E_{f_i}} \|\mathbf{n}_{f_i}^{(t)} \cdot \frac{(\mathbf{p}_k^{(t)} - \mathbf{p}_l^{(t)})}{\|\mathbf{p}_k^{(t)} - \mathbf{p}_l^{(t)}\|_2} - \delta_{f_i}^{kl}\|_2^2 + \lambda \|\delta\|_0 + \alpha \|\delta - \delta^{(t)}\|_2^2 \right\}, \quad (9)$$

where the relaxation term $\|\delta - \delta^{(t)}\|_2^2$ is to address the discontinuity of the L_0 norm, and $\alpha > 0$ is its weight. According to [53], the closed-form solution of each element (i.e., Eq. (7)) becomes

$$\delta_{f_i}^{kl(t+1)} = \begin{cases} 0 & \left\| \frac{\beta w_{f_i}}{\beta w_{f_i} + \alpha} \mathbf{n}_{f_i}^{(t)} \cdot \frac{(\mathbf{p}_k^{(t)} - \mathbf{p}_l^{(t)})}{\|\mathbf{p}_k^{(t)} - \mathbf{p}_l^{(t)}\|_2} + \frac{\alpha}{\beta w_{f_i} + \alpha} \delta_{f_i}^{kl(t)} \right\|_2 \leq \frac{\lambda}{\beta w_{f_i} + \alpha} \\ \frac{\beta w_{f_i}}{\beta w_{f_i} + \alpha} \mathbf{n}_{f_i}^{(t)} \cdot \frac{(\mathbf{p}_k^{(t)} - \mathbf{p}_l^{(t)})}{\|\mathbf{p}_k^{(t)} - \mathbf{p}_l^{(t)}\|_2} + \frac{\alpha}{\beta w_{f_i} + \alpha} \delta_{f_i}^{kl(t)} & otherwise \end{cases} \quad (10)$$

Fig. 5 compares our improved alternating optimization strategy with that of [11,12]. We adjust the parameters to optimal values for both methods. For the strategy in [11,12] given in Algorithm 1, $\lambda = 10^{-3}$, $\eta = 10^{-3}$, $\beta_0 = 10^{-3}$, $\beta_{max} = 10^4$, and $\kappa = 1.083$. For our strategy given in Algorithm 2, $\lambda = 10^{-3}$, $\alpha = 10^{-3}$, $\eta = 10^{-3}$, $\beta_0 = 10^{-3}$, $\beta_{max} = 10^3$, $t_{max} = 5$, and $\kappa = 1.414$. From the close-up view of local regions, we observe that our strategy reconstructs sharp features much better, and exhibits a visually-pleasing result (i.e., Fig. 5(c)). Quantitative evidence is shown in Fig. 5(d) by plotting the energy curves of both strategies. Obviously, the energy under our strategy decreases faster, and reaches a smaller minimum.

Remark. Although it generates good results in many cases, the alternating optimization strategy of [11,12] is not guaranteed to be

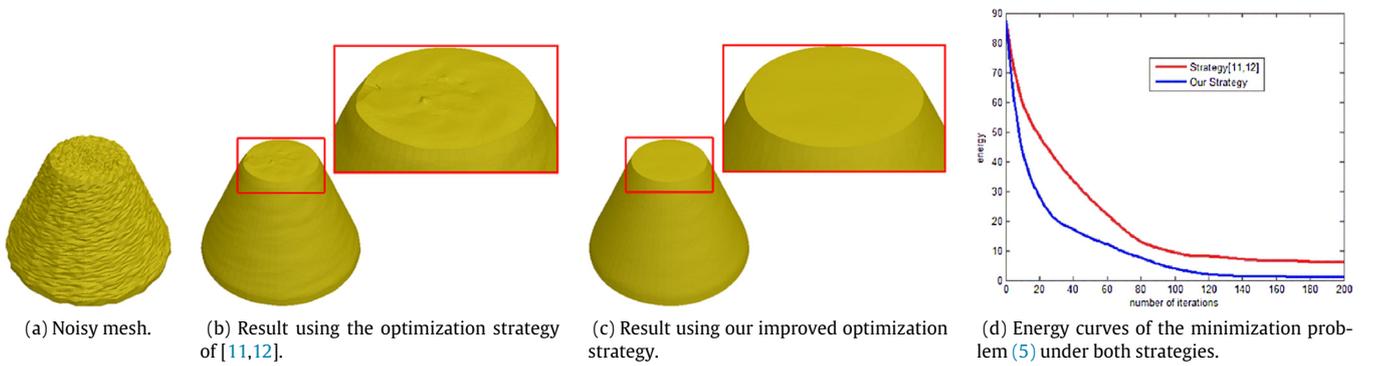


Fig. 5. Comparison between our improved alternating optimization strategy and that of [11,12]. For a fair comparison, we do not adopt the multi-layer approach in Section 4. Please note that, our strategy reaches a smaller minimum, and achieves a better result.

convergent. For the non-convex minimization problem (5) under the current value of β , it solves the two sub-problems only once to get a solution. It is obvious that this solution is not the optimal or stable solution, hence is not a good initial value for the next iteration with updated β . As shown in Fig. 5(b), accumulated numerical errors result in visual artifacts.

With the help of the inner-level iteration and the relaxation term, our improved alternating optimization strategy, whose convergence and stability can be proved, is able to address this problem. In fact, the alternating optimization strategy of [11,12] is a special case of ours. When t_{max} is set to 1 and α is set to 0, the improved alternating optimization strategy degenerates to that of [11,12]. We analyze its convergence and stability in the next subsection.

3.3. Theoretical analysis on convergence and stability

The improved alternating optimization strategy is guaranteed to be convergent and stable. Fixing the outer-level iteration (i.e., fixing β), we first analyze the convergence and stability of the inner-level iteration.

Let $E_\beta(\mathbf{p}, \mathbf{n}, \delta)$ be the objective function of the energy minimization problem (5) with certain fixed β . For δ , the index set of its nonzero elements is denoted as $N(\delta) = \{(f_i, k, l) | \delta_{f_i}^{kl} \neq 0\}$, and the complementary set of $N(\delta)$ is denoted as $\bar{N}(\delta) = \{(f_i, k, l) | \delta_{f_i}^{kl} = 0\}$. We have the following theorem (please refer to Appendix for its proof).

Theorem 1. Let $\{(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)})\}_{t=1,2,\dots}$ be the sequence generated by the inner-level iteration of Algorithm 2, then the following statements hold

- (1) $\{E_\beta(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)})\}_{t=1,2,\dots}$ is strictly monotonic decreasing, and hence converging;
- (2) $\lim_{t \rightarrow +\infty} \|\delta^{(t+1)} - \delta^{(t)}\|_2^2 = 0$, and there exists a positive integer \tilde{t} such that $N(\delta^{(t)})$ remains unchanged for $t \geq \tilde{t}$.

Based on Item (1) of the Theorem, we can obtain a solution with minimal energy value after convergence. Since it minimizes the objective function $E_\beta(\mathbf{p}, \mathbf{n}, \delta)$, this solution can lead to a satisfactory denoising, namely, effective noise elimination and feature preservation. Because δ determines the sparsity of geometric features, Item (2) of the Theorem indicates that we can stably recover geometric features. By the definition of $N(\delta)$, unchanged $N(\delta^{(t)})$ means that the position and the number of the recovered features are stable after the \tilde{t} th iteration. Furthermore, $\delta^{(t+1)}$ is very close to $\delta^{(t)}$ when t is large enough. In fact, from the perspective of numerical computation, $\delta^{(t+1)}$ is equal to $\delta^{(t)}$ when t is large

enough. It implies that the magnitude of the recovered features is also stable.

Compared with the optimization strategy of [11,12], our optimization strategy has two significant improvements. First, we introduce the inner-level iteration to iteratively approximate a stable solution of the energy minimization problem (5) with certain fixed β (i.e., $\min_{(\mathbf{p}, \mathbf{n}, \delta)} \{E_\beta(\mathbf{p}, \mathbf{n}, \delta)\}$). Second, we introduce a relaxation term $\alpha \|\delta - \delta^{(t)}\|_2^2$ for the sub-problem (6), i.e., replacing the sub-problem (6) with the sub-problem (9). Without this relaxation term (i.e., α is set to 0), we could not obtain Theorem 1. It is because the generated objective function sequence $\{E_\beta(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)})\}$ is non-increasing, but may not be strictly monotonic decreasing, hence the variable sequence $\{(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)})\}$ cannot be guaranteed to converge. Adding a relaxation term to the objective function during iteration is a widely-used technique to overcome this drawback in the optimization research community. The solution to the sub-problem (9) is a modification of the solution to the sub-problem (6). According to [53], a smaller α can speed up convergence.

Now we discuss the convergence of the outer-level iteration which contains the update of β and the inner-level iteration. If β is large enough, the solution of the energy minimization problem (5) can infinitely approximate that of the energy minimization problem (4). Nevertheless, directly setting a large value to β leads to poor results. Hence, we also start with a small value β_0 , and then iteratively increase β until it is large enough. It is a well-known strategy in non-convex optimization to avoid getting trapped to local minima [54,53]. In addition, benefiting from the convergence and stability of the inner-level iteration, we can always find a stable solution for the energy minimization problem (5) with current β (i.e., at the current outer-level iteration). Therefore, the solution at the current outer-level iteration can set a good initial value for the next outer-level iteration with updated β . Combining the above two aspects, we can draw a conclusion that the outer-level iteration is convergent.

4. Multi-layer approach

Because it is computationally expensive to optimize the L_0 minimization problem, we introduce a new multi-layer approach to further improve the performance. The overview of our multi-layer approach is shown in Fig. 6. First, a multi-layer representation of the input mesh is built by the clustering-based simplification method [55]. Then, the denoising can be easily carried out on the coarsest layer (i.e., the mesh with the fewest vertices in the above representation). Finally, this solution should be transferred back layer by layer until the denoising result of the input mesh is obtained. We extend joint bilateral upsampling [15] from images to meshes to fulfill this task.

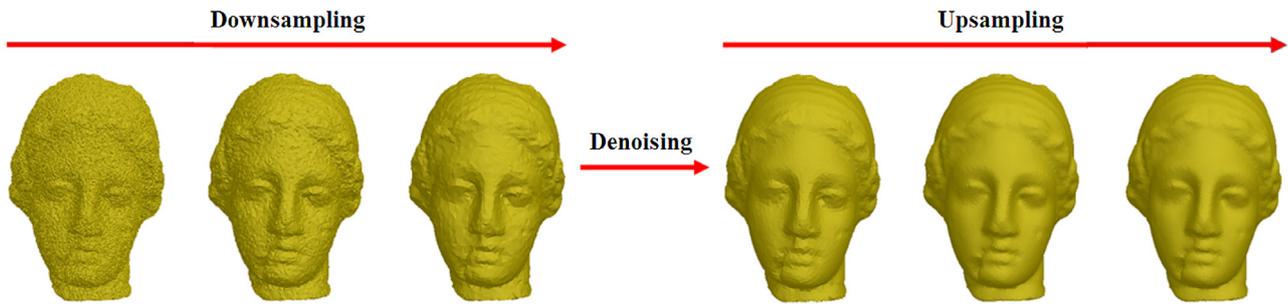


Fig. 6. Overview of the multi-layer approach (e.g., three layers are adopted). The input mesh is first downsampled to construct a multi-layer representation. After the denoising is completed on the coarsest layer, this solution is upsampled to get the denoising result of the input mesh. The number of vertices on the three layers is 100,759, 28,646, and 7,687, respectively.

4.1. Multi-layer representation

Most existing multi-layer methods build the multi-layer representation by the iterative edge contraction simplification [56]. However, as pointed out by [57], clustering-based simplification is faster and more memory-saving than iterative edge contraction simplification, and is well suited for large-scale meshes. In [55], the mesh is recursively split into a set of clusters via binary space partition.

Partition criteria include geometry variation and normal variation. For a region Ω , geometry variation is defined as $\sigma_g = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}$, where $\{\lambda_i\}_{i=1}^3$ are the three eigenvalues of the covariance matrix of Ω , and $\lambda_0 \leq \lambda_1 \leq \lambda_2$. As illustrated in [58], σ_g closely relates to mean curvature. Normal variation is more sensitive to surface anisotropy, and is defined as $\sigma_n = \max_{\mathbf{p}_i \in \Omega} \{\arccos(\frac{\mathbf{n}_{p_i} \cdot \bar{\mathbf{n}}}{\|\bar{\mathbf{n}}\|})\}$, where \mathbf{n}_{p_i} is the unit normal of vertex \mathbf{p}_i , and $\bar{\mathbf{n}}$ is the average of vertex normals on Ω . Partition plane is defined as $(\mathbf{x} - \bar{\mathbf{p}}) \cdot \mathbf{v}_0 = 0$, where $\bar{\mathbf{p}}$ is the centroid of Ω , and \mathbf{v}_0 is the eigenvector of the smallest eigenvalue λ_0 . In other words, Ω is partitioned along the direction of greatest variation.

Binary space partition initially sets $\Omega = M$, and recursively splits M into a set of clusters if geometry variation or normal variation exceeds a threshold [55]. Each cluster is then represented by its centroid to obtain a downsampled mesh for M . Vertices of the downsampled mesh are the centroids of these clusters. Edges of the downsampled mesh are formed according to the adjacency of these clusters. That is, if two clusters are adjacent, there is an edge between their centroids (i.e., two vertices of the downsampled mesh).

To retain shape features during simplification, we empirically require that each cluster contains five vertices at most. Fig. 7 illustrates a clustering result and a downsampled mesh under $0.3l_e$ Gaussian noise along normal directions. Therefore, a multi-layer representation can be constructed by performing the aforementioned simplification procedure repeatedly. In our experiments, three to five layers are sufficient for performance improvement.

4.2. Joint bilateral upsampling

The denoising can be efficiently completed on the coarsest layer. And then, we should transfer the solution on the coarsest layer back to the finest layer (i.e., the input mesh) to obtain the final denoising result. Without loss of generality, we discuss joint bilateral upsampling procedure between two adjacent layers. The central idea is to apply a spatial filter to the coarser layer, meanwhile a range filter is jointly applied to the finer layer. That is, joint bilateral upsampling employs the information of both layers to interpolate the solution on the finer layer.

According to [15], we should establish vertex correspondences between the two layers. Assume \mathbf{v}_i is a vertex on the finer layer



Fig. 7. Clustering-based simplification. From left to right: input mesh (172,962 vertices), clustering result (46,406 clusters), and downsampled mesh (46,406 vertices). Each cluster is rendered with a random color. Please note that, shape features are faithfully retained even under high noise rate.

and belongs to the j th cluster, its corresponding vertex is the j th vertex on the coarser layer $\tilde{\mathbf{v}}_j$ (i.e., the centroid of the j th cluster). Let $\{\tilde{\mathbf{v}}_k\}$ be the 1-ring neighboring vertices of $\tilde{\mathbf{v}}_j$. For each $\tilde{\mathbf{v}}_k$, we compute its nearest vertex \mathbf{v}_i in the k th cluster on the finer layer as the corresponding vertex. The upsampled solution on the finer layer can be expressed as

$$\mathbf{v}_i = \frac{\sum_{\tilde{\mathbf{v}}_k} \tilde{\mathbf{v}}_k f(\|\tilde{\mathbf{v}}_j - \tilde{\mathbf{v}}_k\|) g(\|\mathbf{n}_i - \mathbf{n}_k\|)}{\sum_{\tilde{\mathbf{v}}_k} f(\|\tilde{\mathbf{v}}_j - \tilde{\mathbf{v}}_k\|) g(\|\mathbf{n}_i - \mathbf{n}_k\|)}, \quad (11)$$

where \mathbf{n}_i and \mathbf{n}_k are the unit normal of \mathbf{v}_i and \mathbf{v}_k , respectively, $f(\cdot)$ is a Gaussian function in terms of the position difference between $\tilde{\mathbf{v}}_j$ and $\tilde{\mathbf{v}}_k$, and $g(\cdot)$ is a Gaussian function in terms of the normal difference between \mathbf{v}_i and \mathbf{v}_k . Obviously, Eq. (11) adopts position information of the coarser layer and normal information of the finer layer. As it is efficient and easy to implement, joint bilateral upsampling is a practical scheme to deal with large-scale meshes.

With the multi-layer approach, denoising results depend on three processes: downsampling, denoising on the coarsest layer, and upsampling. In Fig. 8, we evaluate the impact of the multi-layer approach on denoising result. There is little visual difference between Figs. 8(b) and 8(c).

5. Denoising of time-varying mesh sequences

Moreover, this denoising algorithm can be naturally generalized to denoise time-varying mesh sequences. Time-varying mesh sequence is represented by a group of triangular meshes with identical connectivity and different geometries, where the motion is specified by a complete set of new vertex positions for each frame. Suppose $\{M_1, \dots, M_s\}$ is a time-varying mesh sequence, where $M_j (1 \leq j \leq s)$ is a triangular mesh and s is the number of triangular meshes.

Because time-varying mesh sequence has an additional temporal dimension, the temporal coherence between adjacent frames should be maintained during denoising. Besides the regularization

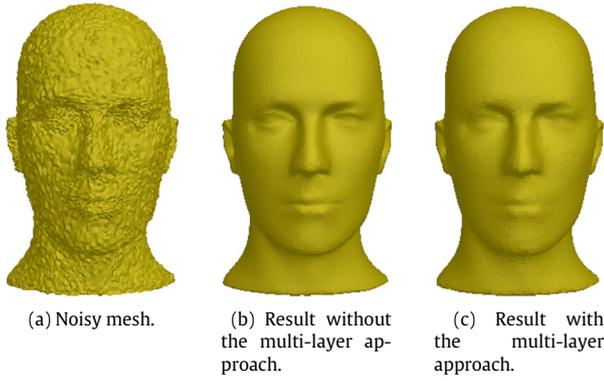


Fig. 8. Impact of the multi-layer approach on denoising result. Please note that, there is little visual difference.

term and the fidelity term for each frame, a coherence term must be introduced

$$\sum_{j=1}^{s-1} \sum_i \|\mathbf{p}_i^j - \mathbf{p}_i^{j+1}\|_2^2,$$

where \mathbf{p}_i^j is the i th vertex on M_j , and \mathbf{p}_i^{j+1} is its corresponding vertex on M_{j+1} .

6. Experimental results and discussion

We implement the proposed algorithm in C++, and test it using a number of meshes with various kinds of features. All the results shown in this paper are run on a laptop with Intel Core i5-3320M CPU. Both synthetic and natural noises are utilized to validate our algorithm. Synthetic noises include Gaussian noise and impulsive noise. Both of them can be added along normal directions or random directions. The intensity of noises is described by standard deviation that is proportional to the average edge length of the original mesh l_e . Real-world raw meshes contain natural noises.

Our objective function consists of a L_0 regularization term and a L_2 fidelity term. The L_0 norm is suitable for relatively flat regions.

The L_2 norm is differentiable, and is suitable for smooth but non-flat regions. Hence, our algorithm can perform well on both CAD mechanical models and non-CAD organic shapes.

Parameters and Examples. Our algorithm involves eight parameters: β_0 , β_{max} , κ , t_{max} , λ , α , η , and E_{f_i} . β_0 , β_{max} and κ determine the number of outer-level iterations together. Following [12], we set $\beta_0 = 10^{-3}$, $\beta_{max} = 10^3$ and $\kappa = \sqrt{2}$ in default. t_{max} is the number of inner-level iterations. In our experiments, we observe that $t_{max} \in [3, 10]$ is enough for the convergence of the inner-level iteration. λ is the weight for the sparse regularization term, thus controlling the smoothness of denoising results. The larger λ is, the smoother the result is. We observe that $\lambda \in [10^{-4}, 10^{-2}]$ well balances feature preservation and noise elimination. α is the weight for the relaxation term. According to [53], a smaller α can speed up convergence. We use 10^{-3} as its default value. η is the weight in the fidelity term, and is set to 10^{-3} in default. E_{f_i} is the neighboring edge set of facet f_i . We find that $E_{f_i}^I$ is more suitable for CAD mechanical meshes and $E_{f_i}^{II}$ is more suited for non-CAD organic shapes.

Fig. 9 illustrates the denoising results with different kinds of synthetic noises. Both sharp features and subtle details are faithfully recovered while noises are cleanly removed. As shown in Fig. 10, the mesh is non-uniformly sampled: the sampling of the left part is denser than that of the right part. Our algorithm is insensitive to non-uniform sampling because of the incorporation of L_0 constraint (1). We also demonstrate the effect of the smoothing parameter λ . When λ is too large, the mesh is overly-smoothed and suffers from feature blurring.

Fig. 11 gives the denoising results on real-world raw meshes. Especially, the last two meshes have open boundaries. These results are very natural, and demonstrate that our algorithm handles real data well. Benefiting from the multi-layer approach, our algorithm can accommodate large meshes with hundreds of thousands of vertices in Fig. 12. These meshes have complex shape and rich details.

As can be seen from Fig. 13, our algorithm is applied to denoise time-varying mesh sequence. Vivid face expression is successfully recovered. Fig. 14 illustrates another example. Swing sequence undergoes a large deformation. Both temporal coherence and subtle details (e.g., wrinkles on the cloth) are reliably maintained.

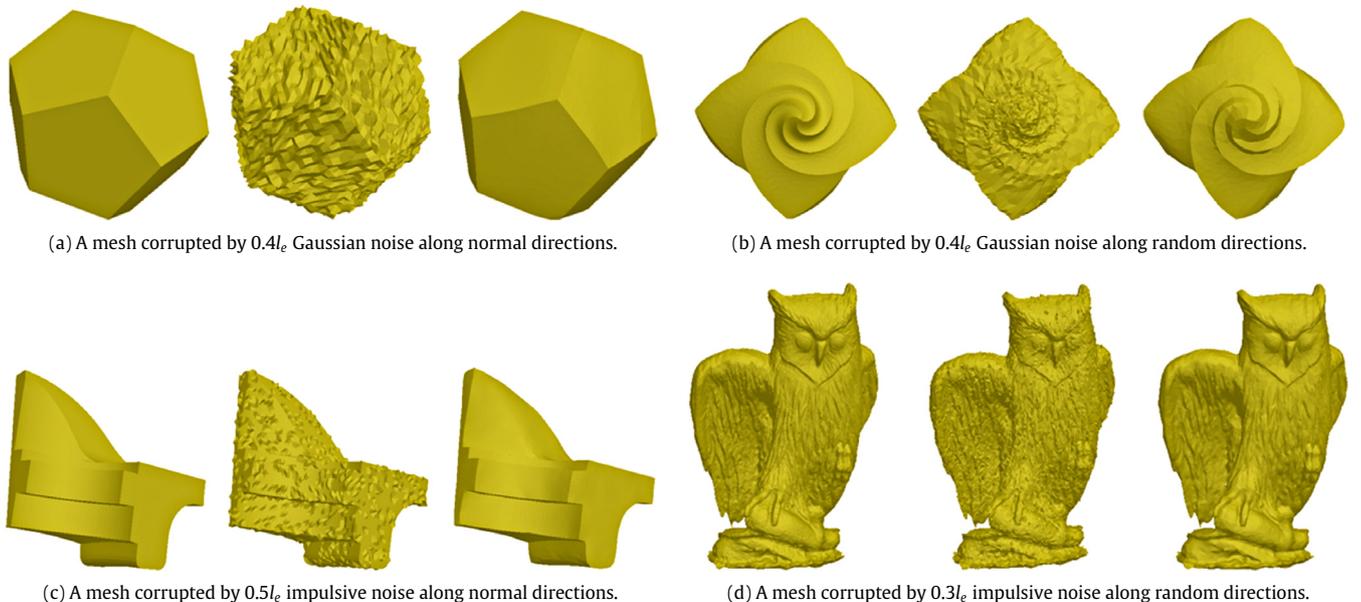


Fig. 9. Denoising results with different kinds of synthetic noises. For each subfigure, from left to right: original mesh, noisy mesh, and our result.

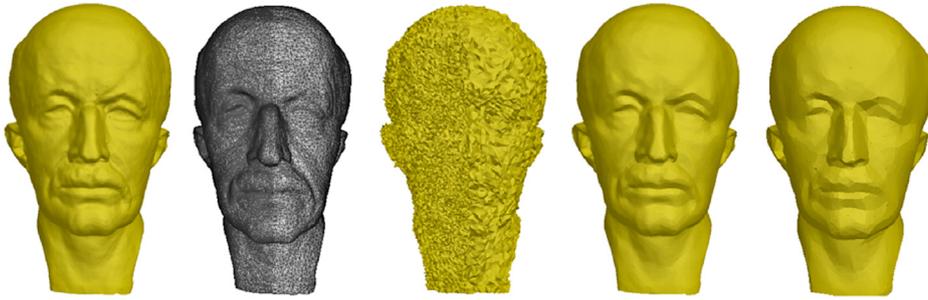


Fig. 10. Denoising results with different smoothing parameters on a non-uniform mesh corrupted by $0.5l_e$ Gaussian noise along normal directions. From left to right: original mesh, original mesh rendered in wireframe, noisy mesh, our result with $\lambda = 4.5 \times 10^{-3}$, and our result with $\lambda = 10^{-2}$.

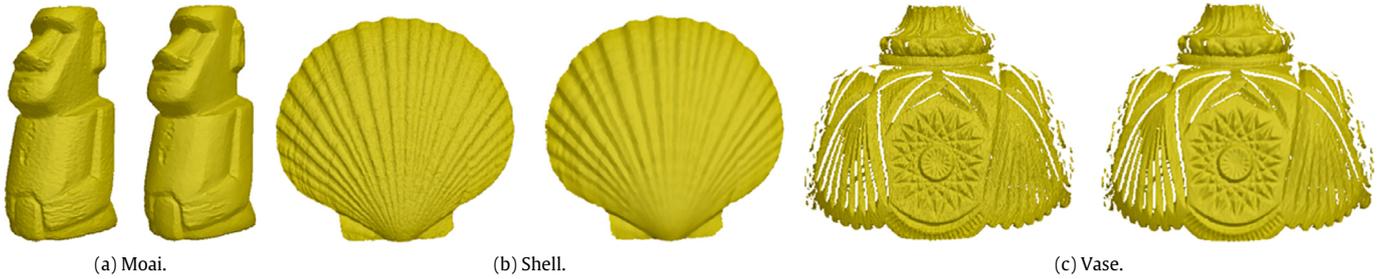


Fig. 11. Denoising results on real-world raw meshes. For each subfigure, from left to right: noisy mesh, and our result.



Fig. 12. Denoising results on large-scale meshes. For each subfigure, from left to right: noisy mesh, and our result.

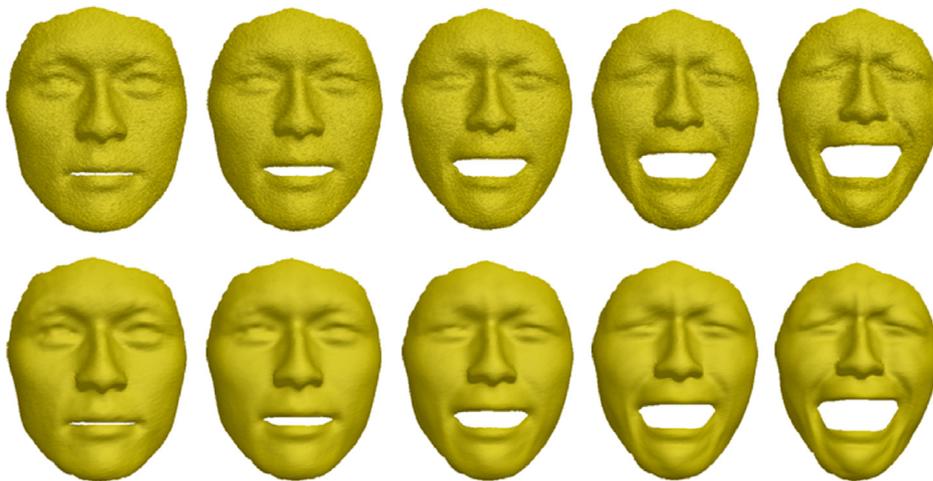


Fig. 13. Denoising result on a face expression sequence. From top to bottom: noisy sequence, and our result. This sequence has 30 frames, and each frame has 29,299 vertices. Here only five key frames are shown (due to space limitation).

Visual Comparisons. Figs. 15–17 compare our algorithm with other state-of-the-art methods [3–5,7,12,9] on meshes corrupted

by Gaussian noise and impulsive noise that are along normal directions and random directions. In particular, both denoising schemes



Fig. 14. Denoising result on a swing sequence. From top to bottom: noisy sequence, and our result. This sequence has 150 frames, and each frame has 9971 vertices. Here only six key frames are shown (due to space limitation).

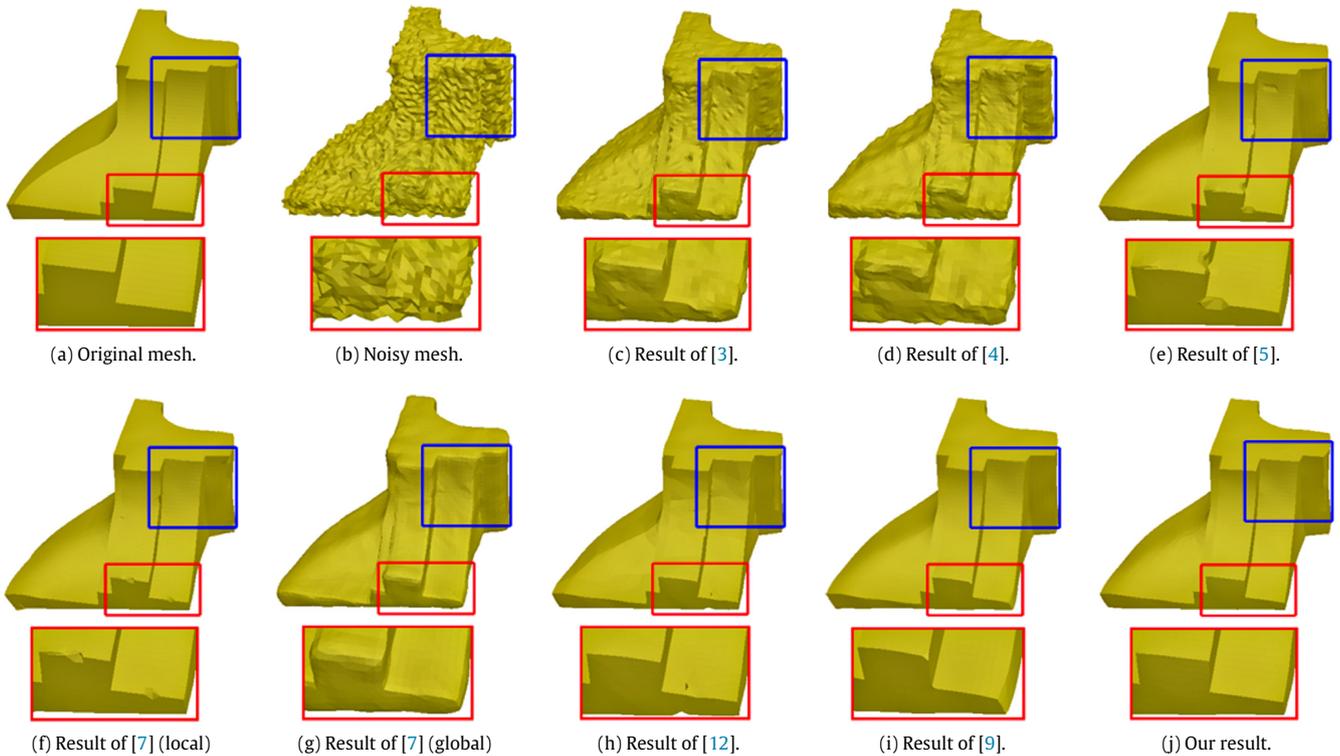


Fig. 15. Comparison between different denoising algorithms on a mesh corrupted by $0.4I_e$ Gaussian noise along normal directions and $0.1I_e$ impulsive noise along normal directions. Please note that, the parameters of each algorithm are tuned carefully to generate their best results.

(i.e., the local scheme and the global scheme) of [7] are employed for comparison. All the methods control the denoising results with several parameters. For different noisy meshes, we carefully tune the parameters of each method to generate visually best results. It is very challenging to deal with mixed noises. These state-of-the-art methods mainly assume that the mesh is corrupted by a single kind of noise (e.g., Gaussian noise or impulsive noise). Hence,

they usually fail under mixed noises. From the close-up view of local regions, these methods suffer from the problems of shape degeneration, feature distortion or facet fold-over. In contrast, our algorithm works well in the presence of mixed noises. Our sparse regularization and improved optimization strategy are powerful enough to effectively preserve sharp features and achieve better results.

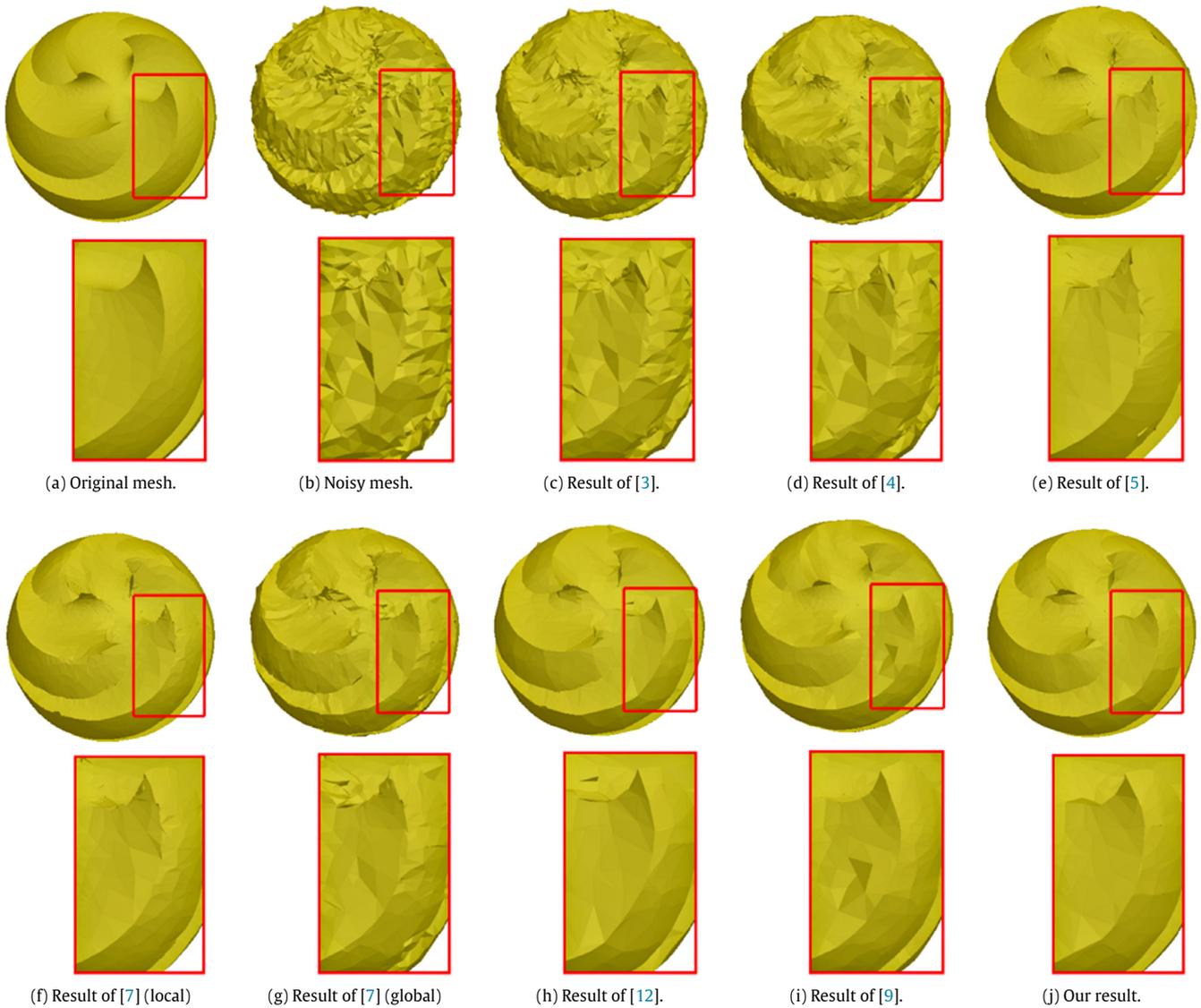


Fig. 16. Comparison between different denoising algorithms on a mesh corrupted by $0.4l_e$ Gaussian noise along random directions and $0.1l_e$ impulsive noise along normal directions. Please note that, the parameters of each algorithm are tuned carefully to generate their best results.

Wu et al. [45] proposed an algorithm to address mixed noises. We compare our algorithm with that of [45] in Fig. 18. Wu et al. [45] assumed that edge lengths were kept during denoising. However, this assumption does not hold under high noise rate, thus causing visual artifacts. In contrast, our algorithm handles mixed noises better. In Fig. 19, our algorithm is further compared with algorithms [12,9]. The mesh is non-uniformly sampled, and is corrupted by extreme noises. Some flat regions and sharp edges are distorted in the result of [12]. Because guided normal field is not constructed properly, apparent shape degeneration occurs in the result of [9]. Our algorithm is insensitive to non-uniform sampling, and produces a visually-pleasing result even under extreme noise corruption. Fig. 20 gives a comparison with classification-based methods [6,10]. As severe noise leads to poor classification results, these methods fail to recover sharp features and some moderate details. Our algorithm is able to robustly reconstruct them to obtain a desirable result.

Quantitative Evaluations. Besides the visual comparisons above, we also analyze the denoising quality of different algorithms through three popular error metrics [5,9].

The first metric is the mean square angular error (MSAE)

$$\text{MSAE} = E[\angle(\mathbf{n}, \mathbf{n}_g)],$$

where E is the expectation operator, and $\angle(\mathbf{n}, \mathbf{n}_g)$ denotes the angles between the denoised facet normals \mathbf{n} and the ground truth facet normals \mathbf{n}_g .

The second metric is the L_2 vertex-based mesh-to-mesh distance

$$E_v = \sqrt{\frac{1}{3A} \sum_{i=1}^n A_i \text{dist}(\mathbf{p}_i, T)^2},$$

where $\text{dist}(\mathbf{p}_i, T)$ is the L_2 distance from the denoised vertex \mathbf{p}_i to the ground truth mesh T , A_i is the area of the 1-ring neighboring facets of \mathbf{p}_i , and A is the area of T .

The third metric is the vertex-based Hausdorff distance from the denoised mesh to the ground truth mesh

$$E_h = \max_i \{\text{dist}(\mathbf{p}_i, T)\}.$$

Using the original mesh as ground truth, we compute the denoising errors for the results shown in Figs. 15–17. As can be seen

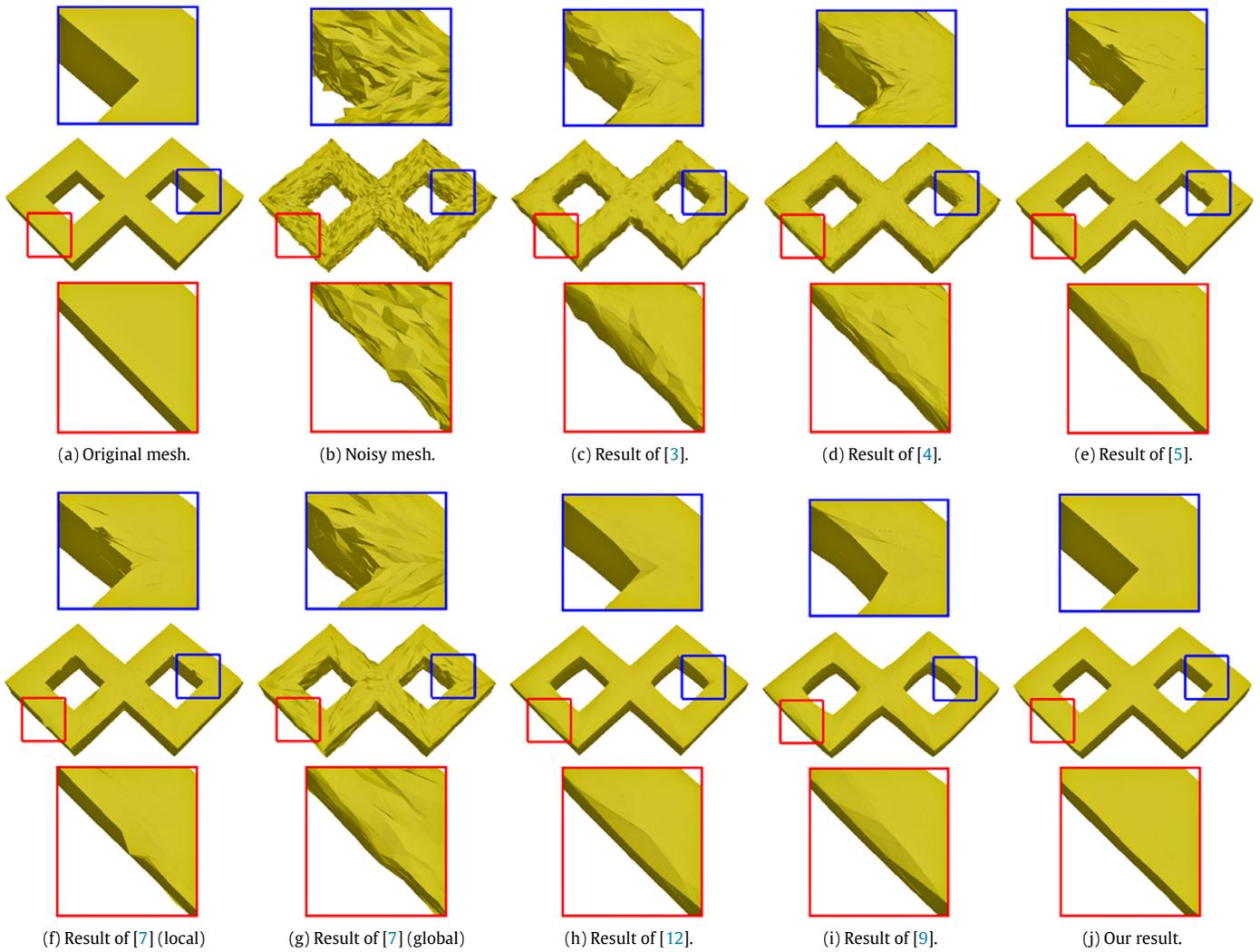


Fig. 17. Comparison between different denoising algorithms on a mesh corrupted by $0.4l_e$ Gaussian noise along random directions and $0.1l_e$ impulsive noise along random directions. Please note that, the parameters of each algorithm are tuned carefully to generate their best results.

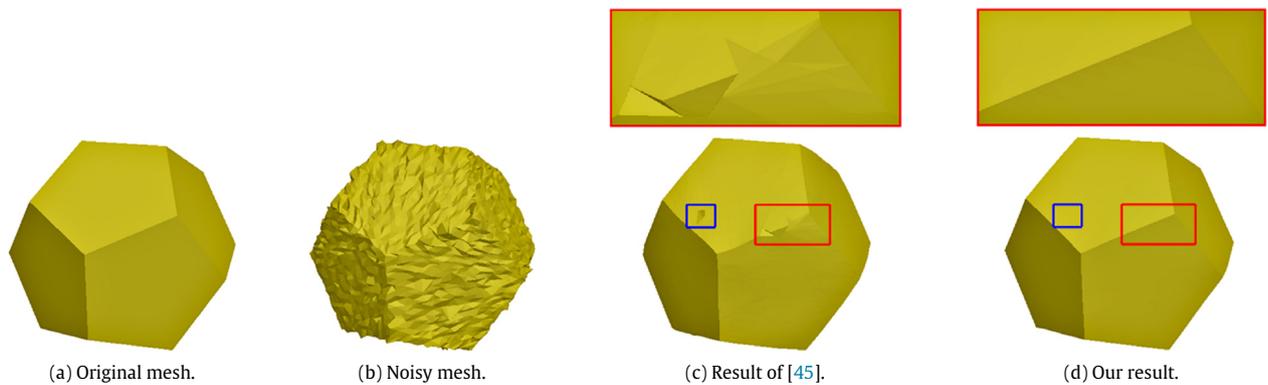


Fig. 18. Comparison with denoising algorithm [45] on a mesh corrupted by $0.3l_e$ Gaussian noise along random directions and $0.15l_e$ impulsive noise along random directions. Please note that, our algorithm preserves shape features better.

from Table 1, our results achieve the smallest denoising errors. In other words, quantitative comparisons also indicate that our algorithm is superior to those state-of-the-art methods.

In our algorithm, vertex positions and facet normals are coupled by the L_0 regularization term, so that they are not totally independent. During optimization, they collectively respect the local

flatness of the mesh, and constrain each other to approach those of a noise-free surface (i.e., the denoised mesh). Our L_0 regularization term is effective in addressing mixed noises or extreme noises, and our improved optimization strategy is guaranteed to be convergent and stable. Finally, vertex positions and facet normals will converge to those of a noise-free surface (i.e., the denoised mesh),

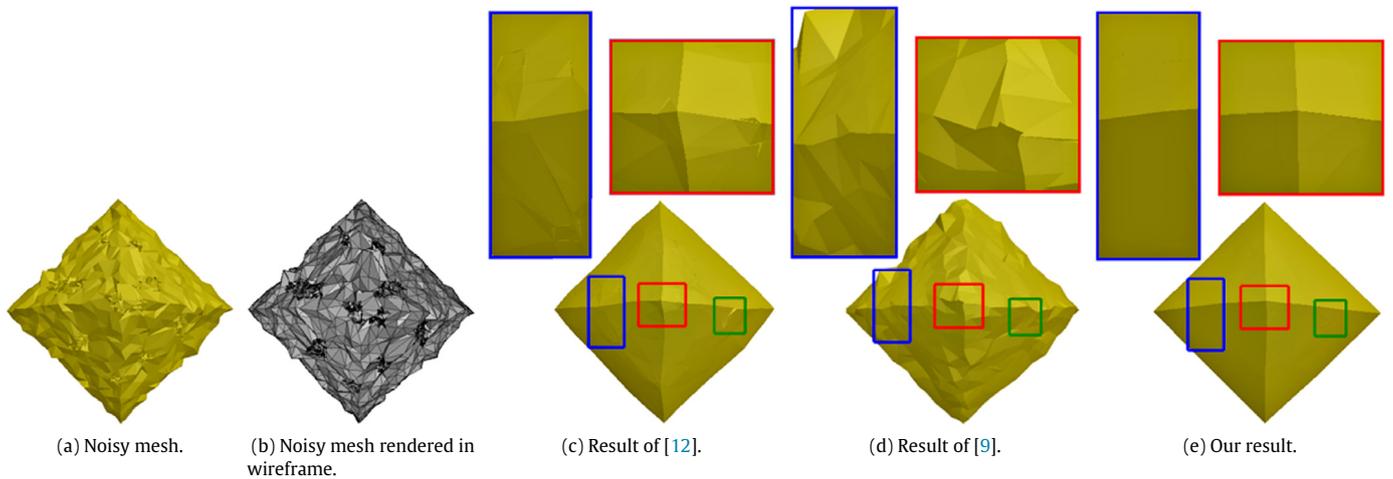


Fig. 19. Comparison with denoising algorithms [12,9] on a non-uniform mesh corrupted by $0.8l_e$ Gaussian noise along random directions. Please note that, our algorithm preserves shape features better.

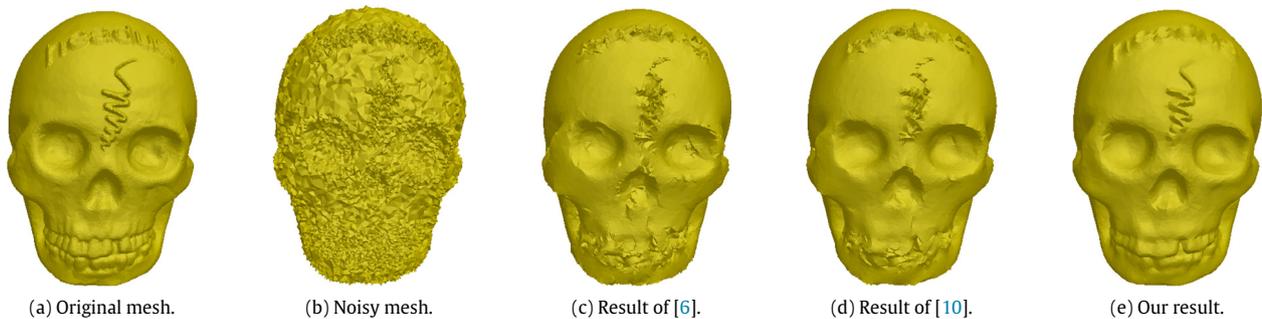


Fig. 20. Comparison with classification-based denoising algorithms [6,10] on a mesh corrupted by $0.5l_e$ Gaussian noise along normal directions.

Table 1
Denoising error comparisons between different algorithms.

Mesh	Error metrics	[3]	[4]	[5]	[7] (local)	[7] (global)	[12]	[9]	Ours
Fandisk (Fig. 15)	MSAE	0.1814	0.1853	0.0722	0.0663	0.1181	0.1012	0.0549	0.0496
	E_v	0.0143	0.0151	0.0067	0.0068	0.0104	0.0074	0.0055	0.0051
	E_h	0.067	0.064	0.061	0.054	0.041	0.058	0.042	0.038
Sharp Sphere (Fig. 16)	MSAE	0.3431	0.3270	0.2293	0.2062	0.2575	0.1846	0.1556	0.1372
	E_v	0.048	0.045	0.034	0.031	0.039	0.026	0.024	0.022
	E_h	0.119	0.105	0.092	0.084	0.097	0.087	0.099	0.079
Double Torus (Fig. 17)	MSAE	0.4862	0.4339	0.3272	0.3150	0.4105	0.2731	0.2848	0.2413
	E_v	0.0194	0.0175	0.0152	0.0168	0.0150	0.0138	0.0142	0.0131
	E_h	0.124	0.115	0.090	0.088	0.097	0.074	0.079	0.063

thus matching each other. To validate this conclusion, we compute the maximal angular error between the obtained facet normals and the facet normals of the obtained vertex positions. For our results in Figs. 15–17, their maximal angular errors are 6.751×10^{-3} , 8.042×10^{-3} , and 5.669×10^{-3} , respectively. Besides, our denoising results are rendered in flat shading mode using the obtained vertex positions and facet normals. Please note that, there are no visual artifacts.

Performance. Table 2 documents smoothing parameter and performance statistics for some denoising examples. The performance relates to geometry information, feature type and noise rate. It is most time-consuming to solve the system of equations. Since the coefficient matrix changes during iteration, we could not precompute a factorization, thus causing longer running times.

The multi-layer approach is introduced to deal with large-scale meshes. For meshes with less than 20,000 vertices, we perform the denoising on the input mesh directly. Overall, our algorithm is a little slower than existing algorithms [12,9]. For the examples in Figs. 15–17, the performance of [12] costs 34.714 s, 60.962 s, and 66.347 s, respectively. The performance of [9] costs 28.093 s, 45.318 s, and 54.271 s, respectively. However, it is very challenging for [12,9] to denoise large meshes shown in Fig. 12. Thanks to the multi-layer approach, our algorithm can accomplish this task naturally.

Limitations. Although it is robust and effective, our algorithm still has some limitations. First, it is difficult to generate a satisfactory result on meshes with extreme triangulation. Fig. 21 presents such an example, where the features on the top part are not

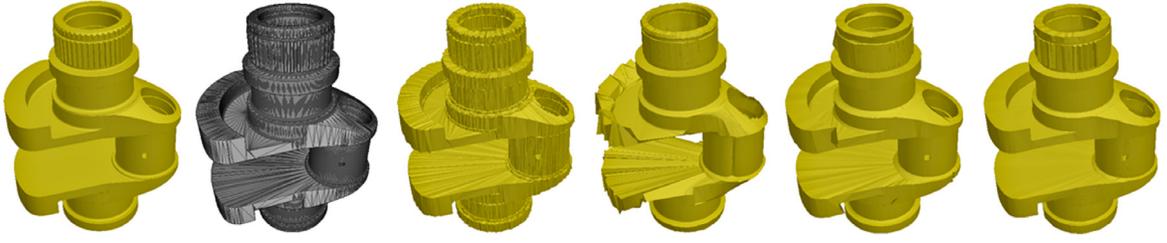


Fig. 21. A failure case on the mesh with extreme triangulation. From left to right: original mesh, original mesh rendered in wireframe, noisy mesh, result of [12], result of [9], and our result.

Table 2
Smoothing parameter and performance data measured in seconds.

Mesh	Number of vertices	Number of facets	Smoothing parameter λ	Time
Dodecahedron (Fig. 9)	4,610	9,216	6×10^{-3}	53.316
Owl (Fig. 9)	39,822	79,652	2.5×10^{-3}	34.575
Max Planck (Fig. 10)	30,942	61,880	4.5×10^{-3}	86.134
Shell (Fig. 11)	43,164	85,425	10^{-3}	30.312
Dragon (Fig. 12)	437,645	869,928	10^{-4}	102.653
Asian Dragon (Fig. 12)	1,164,291	2,322,905	10^{-4}	124.480
Fandisk (Fig. 15)	6,475	12,946	3.5×10^{-3}	45.786
Sharp Sphere (Fig. 16)	10,443	20,882	6×10^{-3}	73.183
Double Torus (Fig. 17)	8,702	17,408	10^{-2}	79.051
Skull (Fig. 20)	20,002	40,000	4×10^{-3}	61.279

retained as expected. As a matter of fact, it is also a challenging case for the state-of-the-art methods [12,9]. Second, we should tune the parameters to obtain visually-pleasing results. The default value or the range of these parameters are discussed for the meshes shown in this paper. However, they may not be suitable for all meshes with different features or noise types.

7. Conclusion and future work

Instead of local filtering, this paper has proposed a novel sparse regularization for mesh denoising with feature-preservation. Both vertex positions and facet normals are integrated into a L_0 minimization framework to decouple features and noises with distinguishing power. Then, an improved alternating optimization strategy is presented to effectively address the non-convex minimization problem, which guarantees the convergence and stability. During the denoising process, our algorithm can faithfully reconstruct the global structures and local details of the mesh. Using joint bilateral upsampling, a multi-layer approach is introduced to achieve higher performance. Furthermore, the aforementioned algorithm is generalized to handle time-varying mesh sequences, which has demonstrated the robustness, versatility, and flexibility of our algorithm.

As for the immediate future work, the advantage of the L_0 norm promises to expand its application scope to other applications in geometry processing and 3D graphics, such as completion, registration, segmentation, and editing. Because of the non-convex nature of the L_0 norm, GPU will be a good necessity for real-time performing enhancement. These topics deserve our serious exploration in the near future.

Acknowledgments

This research is supported in part by National Natural Science Foundation of China (61532002, 61672149, 11701538, 41606006, U1706218, and 61303145), National Science Foundation of USA (IIS-1715985, IIS-0949467, IIS-1047715, and IIS-1049448), Natural Science Foundation of Shandong Province (ZR2018MF006), and Open Project of State Key Lab of CAD&CG, Zhejiang University (A1809).

Appendix

We first present a technical lemma, and then prove [Theorem 1](#).
A.1. Lemma

Assume w is the maximal value of the facet weights, i.e., $w = \max_{\{f_i\}} \{w_{f_i}\}$.

Lemma 1. Let $\{\delta^{(t)}\}_{t=1,2,\dots}$ be the sequence generated by Eq. (10), then the following statements hold

- (1) $|\delta_{f_i}^{kl(t)}| > \sqrt{\frac{\lambda}{\beta w + \alpha}}$ for all $(f_i, k, l) \in N(\delta^{(t)})$;
- (2) $\|\delta^{(t+1)} - \delta^{(t)}\|_2^2 > \frac{\lambda}{\beta w + \alpha}$ if $N(\delta^{(t+1)}) \neq N(\delta^{(t)})$.

Proof. According to the closed-form solution (10), $|\delta_{f_i}^{kl(t)}| > \sqrt{\frac{\lambda}{\beta w_{f_i} + \alpha}} \geq \sqrt{\frac{\lambda}{\beta w + \alpha}}$ for all $(f_i, k, l) \in N(\delta^{(t)})$, thus immediately obtaining Item (1) of the lemma. It remains to prove Item (2) of the lemma.

Suppose that $N(\delta^{(t+1)}) \neq N(\delta^{(t)})$, it follows that there exists at least a pair $(\tilde{f}_i, \tilde{k}, \tilde{l})$ such that $(\tilde{f}_i, \tilde{k}, \tilde{l}) \in N(\delta^{(t)}) \cap \bar{N}(\delta^{(t+1)})$ or $(\tilde{f}_i, \tilde{k}, \tilde{l}) \in \bar{N}(\delta^{(t)}) \cap N(\delta^{(t+1)})$. By Item (1) of the lemma, $|\delta_{\tilde{f}_i}^{kl(t)}| > \sqrt{\frac{\lambda}{\beta w + \alpha}}$ and $\delta_{\tilde{f}_i}^{kl(t+1)} = 0$ for the first case while $\delta_{\tilde{f}_i}^{kl(t)} = 0$ and $|\delta_{\tilde{f}_i}^{kl(t+1)}| > \sqrt{\frac{\lambda}{\beta w + \alpha}}$ for the second case. For both cases, we have $(\delta_{\tilde{f}_i}^{kl(t+1)} - \delta_{\tilde{f}_i}^{kl(t)})^2 > \frac{\lambda}{\beta w + \alpha}$. This, together with $\|\delta^{(t+1)} - \delta^{(t)}\|_2^2 \geq (\delta_{\tilde{f}_i}^{kl(t+1)} - \delta_{\tilde{f}_i}^{kl(t)})^2$, yields Item (2) of the lemma. \square

A.2. Proof of Theorem 1

Proof. As $E_\beta(\mathbf{p}, \mathbf{n}, \delta)$ is the objective function of the energy minimization problem (5) with certain fixed β , the objective function of the first sub-problem (9) is equal to $E_\beta(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta) - \sum_{i=1}^n \|\mathbf{p}_i^{(t)} - \mathbf{p}_i^*\|_2^2 - \eta \sum_{i=1}^m \|\mathbf{n}_i^{(t)} - \mathbf{n}_i^*\|_2^2 + \alpha \|\delta - \delta^{(t)}\|_2^2$.

Since $\delta^{(t+1)}$ is the solution to the first sub-problem (9), the energy value at $\delta^{(t+1)}$ should be not larger than that at $\delta^{(t)}$. Substituting $\delta^{(t+1)}$ and $\delta^{(t)}$ into the objective function of the first sub-problem (9) respectively, we have $E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t+1)}) - \sum_{i=1}^n \|\mathbf{p}_i^{(t)} - \mathbf{p}_i^*\|_2^2 - \eta \sum_{i=1}^m \|\mathbf{n}_{f_i}^{(t)} - \mathbf{n}_{f_i}^*\|_2^2 + \alpha \|\delta^{(t+1)} - \delta^{(t)}\|_2^2 \leq E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)}) - \sum_{i=1}^n \|\mathbf{p}_i^{(t)} - \mathbf{p}_i^*\|_2^2 - \eta \sum_{i=1}^m \|\mathbf{n}_{f_i}^{(t)} - \mathbf{n}_{f_i}^*\|_2^2 + \alpha \|\delta^{(t)} - \delta^{(t)}\|_2^2$. That is, $E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t+1)}) + \alpha \|\delta^{(t+1)} - \delta^{(t)}\|_2^2 \leq E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)})$.

Since $(\mathbf{p}^{(t+1)}, \mathbf{n}^{(t+1)})$ is the solution to the second sub-problem (8), we have $E_{\beta}(\mathbf{p}^{(t+1)}, \mathbf{n}^{(t+1)}, \delta^{(t+1)}) \leq E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t+1)})$. Combining the above two inequalities, we obtain

$$E_{\beta}(\mathbf{p}^{(t+1)}, \mathbf{n}^{(t+1)}, \delta^{(t+1)}) \leq E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)}) - \alpha \|\delta^{(t+1)} - \delta^{(t)}\|_2^2. \quad (12)$$

That is, $E_{\beta}(\mathbf{p}^{(t+1)}, \mathbf{n}^{(t+1)}, \delta^{(t+1)}) < E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)})$.

Because $E_{\beta}(\mathbf{p}, \mathbf{n}, \delta)$ is bounded below by zero (i.e., may not be smaller than zero), we can conclude $\{E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)})\}_{t=1,2,\dots}$ is strictly monotonic decreasing and bounded below by zero. Therefore, $\{E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)})\}_{t=1,2,\dots}$ converges. Item (1) of Theorem 1 is then proved.

Assume T is a positive integer, summing inequality (12) from $t = 1$ to T gives rise to $\sum_{t=1}^T \|\delta^{(t+1)} - \delta^{(t)}\|_2^2 \leq \frac{1}{\alpha} \sum_{t=1}^T (E_{\beta}(\mathbf{p}^{(t)}, \mathbf{n}^{(t)}, \delta^{(t)}) - E_{\beta}(\mathbf{p}^{(t+1)}, \mathbf{n}^{(t+1)}, \delta^{(t+1)})) = \frac{1}{\alpha} (E_{\beta}(\mathbf{p}^{(1)}, \mathbf{n}^{(1)}, \delta^{(1)}) - E_{\beta}(\mathbf{p}^{(T+1)}, \mathbf{n}^{(T+1)}, \delta^{(T+1)}))$. Taking the limit as $T \rightarrow +\infty$ on both sides, $\frac{1}{\alpha} (E_{\beta}(\mathbf{p}^{(1)}, \mathbf{n}^{(1)}, \delta^{(1)}) - E_{\beta}(\mathbf{p}^{(T+1)}, \mathbf{n}^{(T+1)}, \delta^{(T+1)}))$ becomes a constant because of Item (1) of this theorem. Therefore, $\sum_{t=1}^{+\infty} \|\delta^{(t+1)} - \delta^{(t)}\|_2^2$ converges and $\lim_{t \rightarrow +\infty} \|\delta^{(t+1)} - \delta^{(t)}\|_2^2 = 0$. This limit implies that there exists a positive integer \tilde{t} such that $\|\delta^{(t+1)} - \delta^{(t)}\|_2^2 \leq \frac{\lambda}{\beta w + \alpha}$ for $t \geq \tilde{t}$. According to Item (2) of Lemma 1, $N(\delta^{(t+1)}) = N(\delta^{(t)})$ for $t \geq \tilde{t}$. In other words, $N(\delta^{(t)})$ remains unchanged for $t \geq \tilde{t}$. Item (2) of Theorem 1 is then proved. \square

References

- [1] Desbrun M, Meyer M, Schröder P, Barr AH. Implicit fairing of irregular meshes using diffusion and curvature flow. In: SIGGRAPH '99: Proceedings of the 26th annual conference on computer graphics and interactive techniques. 1999. p. 351–8.
- [2] Bajaj CL, Xu G. Anisotropic diffusion of surfaces and functions on surfaces. *ACM Trans Graph* 2003;22(1):4–32.
- [3] Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising. *ACM Trans Graph* 2003;22(3):950–3.
- [4] Jones TR, Durand F, Desbrun M. Non-iterative, feature-preserving mesh smoothing. *ACM Trans Graph* 2003;22(3):943–9.
- [5] Sun X, Rosin PL, Martin RR, Langbein FC. Fast and effective feature-preserving mesh denoising. *IEEE Trans Vis Comput Graphics* 2007;13(5):925–38.
- [6] Fan H, Yu Y, Peng Q. Robust feature-preserving mesh denoising based on consistent subneighborhoods. *IEEE Trans Vis Comput Graphics* 2011;16(2): 312–24.
- [7] Zheng Y, Fu H, Au OK-C, Tai C-L. Bilateral normal filtering for mesh denoising. *IEEE Trans Vis Comput Graphics* 2011;17(10):1521–30.
- [8] Wei M, Yu J, Pang W-M, Wang J, Qin J, Liu L, Heng P-A. Bi-normal filtering for mesh denoising. *IEEE Trans Vis Comput Graphics* 2015;21(1):43–55.
- [9] Zhang W, Deng B, Zhang J, Bouaziz S, Liu L. Guided mesh normal filtering. *Comput Graph Forum* 2015;34(7):23–34.
- [10] Lu X, Deng Z, Chen W. A robust scheme for feature-preserving mesh denoising. *IEEE Trans Vis Comput Graphics* 2016;22(3):1181–94.
- [11] Xu L, Lu C, Xu Y, Jia J. Image smoothing via L_0 gradient minimization. *ACM Trans Graph* 2011;30(6):174 11 pages.
- [12] He L, Schaefer S. Mesh denoising via L_0 minimization. *ACM Trans Graph* 2013;32(4):64 7 pages.
- [13] Nehab D, Rusinkiewicz S, Davis J, Ramamoorthi R. Efficiently combining positions and normals for precise 3D geometry. *ACM Trans Graph* 2005;24(3): 536–43.
- [14] Candès E, Romberg J, Tao T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans Inform Theory* 2006;52(2):489–509.
- [15] Kopf J, Cohen MF, Lischinski D, Uyttendaele M. Joint bilateral upsampling. *ACM Trans Graph* 2007;26(3):96 5 pages.
- [16] Botsch M, Pauly M, Kobbelt L, Alliez P, Lévy B, Bischoff S, Rössl C. Geometric modeling based on polygonal meshes. In: Siggraph '07: ACM SIGGRAPH 2007 courses. 2007.
- [17] Taubin G. A signal processing approach to fair surface design. In: SIGGRAPH '95: Proceedings of the 22nd annual conference on computer graphics and interactive techniques. 1995. p. 351–8.
- [18] Vollmer J, Mencl R, Müller H. Improved Laplacian smoothing of noisy surface meshes. *Comput Graph Forum* 1999;18(3):131–8.
- [19] Liu X, Bao H, Shum H-Y, Peng Q. A novel volume constrained smoothing method for meshes. *Graph. Models* 2002;64(3–4):169–82.
- [20] Nealen A, Igarashi T, Sorkine O, Alexa M. Laplacian mesh optimization. In: GRAPHITE '06: Proceedings of the 4th international conference on computer graphics and interactive techniques in Australasia and Southeast Asia. 2006. p. 381–9.
- [21] Liu L, Tai C-L, Ji Z, Wang G. Non-iterative approach for global mesh optimization. *Comput Aided Des* 2007;39(9):772–82.
- [22] Su Z, Wang H, Cao J. Mesh denoising based on differential coordinates. In: SMI '09: Proceedings of the IEEE international conference on shape modeling and applications. 2009. p. 1–6.
- [23] Clarenz U, Diewald U, Rumpf M. Anisotropic geometric diffusion in surface processing. In: VIS '00: Proceedings of the IEEE visualization. 2000. p. 397–405.
- [24] Tasdizen T, Whitaker RT, Burchard P, Osher S. Geometric surface smoothing via anisotropic diffusion of normals. In: VIS '02: Proceedings of the IEEE visualization. 2002. p. 125–32.
- [25] Hildebrandt K, Polthier K. Anisotropic filtering of non-linear surface features. *Comput Graph Forum* 2004;23(3):391–400.
- [26] Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: ICCV '98: Proceedings of the 6th international conference on computer vision. 1998. p. 839–46.
- [27] Schall O, Belyaev A, Seidel H-P. Adaptive feature-preserving non-local denoising of static and time-varying range data. *Comput Aided Des* 2008;40(6): 701–7.
- [28] Solomon J, Crane K, Butscher A, Wojtan C. A general framework for bilateral and mean shift filtering. *CoRR abs/1405.4734*. 2014.
- [29] Ohtake Y, Belyaev AG, Seidel H-P. Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normals. In: VMV '02: Proceedings of the vision, modeling, and visualization conference. 2002. p. 203–10.
- [30] Shen Y, Barner KE. Fuzzy vector median-based surface smoothing. *IEEE Trans Vis Comput Graphics* 2004;10(3):252–65.
- [31] Sun X, Rosin PL, Martin RR, Langbein FC. Random walks for feature-preserving mesh denoising. *Comput Aided Geom Design* 2008;25(7):437–56.
- [32] Wang P-S, Fu X-M, Liu Y, Tong X, Liu S-L, Guo B. Rolling guidance normal filter for geometric processing. *ACM Trans Graph* 2015;34(6):173 9 pages.
- [33] Yadav SK, Reitebuch U, Polthier K. Mesh denoising based on normal voting tensor and binary optimization. *CoRR abs/1607.07427*. 2016.
- [34] Wang P-S, Liu Y, Tong X. Mesh denoising via cascaded normal regression. *ACM Trans Graph* 2016;35(6):232 12 pages.
- [35] Bian Z, Tong R. Feature-preserving mesh denoising based on vertices classification. *Comput Aided Geom Design* 2011;28(1):50–64.
- [36] Wang J, Zhang X, Yu Z. A cascaded approach for feature-preserving surface mesh denoising. *Comput Aided Des* 2012;44(7):597–610.
- [37] Zhu L, Wei M, Yu J, Wang W, Qin J, Heng P-A. Coarse-to-fine normal filtering for feature-preserving mesh denoising based on isotropic subneighborhoods. *Comput Graph Forum* 2013;32(7):371–80.
- [38] Wei M, Liang L, Pang W-M, Wang J, Li W, Wu H. Tensor voting guided mesh denoising. *IEEE Trans Autom Sci Eng* 2017;14(2):931–45.
- [39] Candès E, Tao T. Decoding by linear programming. *IEEE Trans Inform Theory* 2005;51(12):4203–15.
- [40] Donoho DL. Compressed sensing. *IEEE Trans Inform Theory* 2006;52(4): 1289–306.
- [41] Huang J, Jiang T, Shi Z, Tong Y, Bao H, Desbrun M. L_1 -based construction of polycube maps from complex shapes. *ACM Trans Graph* 2014;33(3):25 11 pages.
- [42] Xu L, Wang R, Zhang J, Yang Z, Deng J, Chen F, Liu L. Survey on sparsity in geometric modeling and processing. *Graph. Models* 2015;82:160–80.
- [43] Avron H, Sharf A, Greif C, Cohen-Or D. L_1 -sparse reconstruction of sharp point set surfaces. *ACM Trans Graph* 2010;29(5):135 20 pages.
- [44] Wang R, Yang Z, Liu L, Deng J, Chen F. Decoupling noise and features via weighted L_1 -analysis compressed sensing. *ACM Trans Graph* 2014;33(2):18 12 pages.
- [45] Wu X, Zheng J, Cai Y, Fu C-W. Mesh denoising using extended ROF model with L_1 fidelity. *Comput Graph Forum* 2015;34(7):35–45.
- [46] Sun Y, Schaefer S, Wang W. Denoising point sets via L_0 minimization. *Comput Aided Geom Design* 2015;35–36:2–15.
- [47] Cheng X, Zeng M, Liu X. Feature-preserving filtering with L_0 gradient minimization. *Comput. Graph.* 2014;38:150–7.

- [48] McLachlan GJ, Krishnan T. The EM algorithm and extensions. In: Wiley series in probability and statistics. John Wiley & Sons; 1997.
- [49] Kwatra V, Essa I, Bobick A, Kwatra N. Texture optimization for example-based synthesis. *ACM Trans Graph* 2005;24(3):795–802.
- [50] Madsen K, Nielsen H, Tingleff O. Optimization with constraints. Technical University of Denmark; 2004.
- [51] Huang J, Shi X, Liu X, Zhou K, Wei L-Y, Teng S-H, Bao H, Guo B, Shum H-Y. Subspace gradient domain mesh deformation. *ACM Trans Graph* 2006;25(3):1126–34.
- [52] Botte J, Sabach S, Teboulle M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math Program* 2014;146(1–2):459–94.
- [53] Shen L, Xu Y, Zeng X. Wavelet inpainting with the L_0 sparse regularization. *Appl. Comput. Harmonic Anal.* 2016;41(1):26–53.
- [54] Portilla J, Tristán-Vega A, Selesnick IW. Efficient and robust image restoration using multiple-feature L_2 -relaxed sparse analysis priors. *IEEE Trans Image Process* 2015;24(12):5046–59.
- [55] Zhao Y, Qian H, Lu S. A deformation-aware hierarchical framework for shape-preserving editing of static and time-varying mesh data. *Comput. Graph.* 2015;46:80–8.
- [56] Garland M, Heckbert PS. Surface simplification using quadric error metrics. In: SIGGRAPH '97: Proceedings of the 22nd annual conference on computer graphics and interactive techniques. 1997. p. 209–216.
- [57] Garland M, Shaffer E. A multiphase approach to efficient surface simplification. In: VIS '02: Proceedings of the IEEE visualization. 2002. p. 117–124.
- [58] Pauly M, Gross M, Kobbelt L. Efficient simplification of point-sampled surfaces. In: VIS '02: Proceedings of the IEEE visualization. 2002. p. 163–70.