Geometric and Learning-Based Mesh Denoising: A Comprehensive Survey

HONGHUA CHEN, Nanjing University of Aeronautics and Astronautics, China
ZHIQI LI, Bournemouth University, UK
MINGQING WEI and JUN WANG, Nanjing University of Aeronautics and Astronautics, China

Mesh denoising is a fundamental problem in digital geometry processing. It seeks to remove surface noise while preserving surface intrinsic signals as accurately as possible. While traditional wisdom has been built upon specialized priors to smooth surfaces, learning-based approaches are making their debut with great success in generalization and automation. In this work, we provide a comprehensive review of the advances in mesh denoising, containing both traditional geometric approaches and recent learning-based methods. First, to familiarize readers with the denoising tasks, we summarize four common issues in mesh denoising. We then provide two categorizations of the existing denoising methods. Furthermore, three important categories, including optimization-, filter-, and data-driven-based techniques, are introduced and analyzed in detail, respectively. Both qualitative and quantitative comparisons are illustrated, to demonstrate the effectiveness of the state-of-the-art denoising methods. Finally, potential directions of future work are pointed out to solve the common problems of these approaches. A mesh denoising benchmark is also built in this work, and future researchers will easily and conveniently evaluate their methods with state-of-the-art approaches. To aid reproducibility, we release our datasets and used results at https://github.com/chenhonghua/Mesh-Denoiser.

CCS Concepts: • Computing methodologies → Computer graphics; Shape modeling; Mesh models;

ACM Reference format:
Honghua Chen, Zhiqi Li, Mingqing Wei, and Jun Wang. 2023. Geometric and Learning-Based Mesh Denoising: A Comprehensive Survey. ACM Trans. Multimedia Comput. Commun. Appl. 20, 3, Article 85 (November 2023), 28 pages.
https://doi.org/10.1145/3625098

1 INTRODUCTION

Recent advances in scanning devices and improvements in techniques that generate and synthesize 3D shapes have made 3D mesh surfaces widespread in various fields, including computer graphics, virtual reality, and reverse engineering. However, during the acquisition of a 3D mesh surface, various noises from uncertain sources inevitably awkward the data quality. This hinders many
downstream applications, like model reconstruction [1], digital human [94], and virtual reality [5]. Consequently, it is desirable to be able to acquire a high-fidelity 3D model by mesh denoising, when considering the above applications.

First, we wish to clarify several synonymous concepts in the literature of mesh denoising. Mesh smoothing contains two main aspects: denoising and fairing [9]. The goal of denoising is to remove noise or unwanted artifacts from the mesh while preserving the original geometric features and details. Mesh denoising is often applied to 3D models obtained from noisy sources like 3D scanning, reconstruction from point clouds, or models generated by simulation. The proposed method should be robust, efficient, and applicable to a wide range of mesh types, noise levels, and geometries. In fairing, it aims to smooth out or regularize the mesh while minimizing any distortions or changes to the overall shape of the model, by using constrained energy minimization. Mesh fairing is often applied to improve the aesthetic appearance of 3D models or to prepare them for further processing, such as mesh parameterization, remeshing, or subdivision. It is worth noting that while the goals and methods of the two processes may overlap, they are distinct processes that address different challenges in 3D mesh processing. In this work, we focus on mesh denoising.

The problem of mesh denoising has been extensively researched, and there has been a huge volume of mesh denoising approaches in recent decades. Among them, the traditional geometric methods aim to reduce noise and irregularities in the geometric representation (vertices, edges, normals, and faces) while preserving important features, based on certain prior assumptions. Aside from filtering in the spatial (geometry) domain, there are alternative methods that involve generating a collection of local Fourier spectra. These spectra can be explicitly analyzed and manipulated, enabling more advanced filtering techniques, such as least-squares optimal, inverse, or feature enhancement [10]. Recently, learning-based techniques have focused more on an automatic and overall smoothing effect, regardless of the surface properties or noise distribution, with the aid of a large training dataset.

Historically, mesh denoising has evolved in conjunction with the work of image filtering. For instance, bilateral mesh/normal filter evolved from gray and color image filtering [69], sparsity priors were first developed to deal with the edge-preserving image smoothing problem [88], non-local low-rank techniques also have exhibited impressive results in image processing [23] before their extended applications in the 3D domain. Last but not least, advanced deep learning techniques in the image domain, also undoubtedly inspire many 3D mesh surface denoising works.

Generally, we can place all the mesh denoising methods into a general framework, which can be split into several typical sub-steps. These sub-steps include coarse mesh denoising, vertex classification, normal estimation, and final vertex position updating. Any one of the existing methods covers one, two, or full of these sub-stages. For example, some one-stage methods [18, 21, 26, 30, 31, 67, 68, 71, 72, 107] directly move the mesh vertices to their suitable positions. Although this kind of method is simple, it may either over-smooth sharp features or cannot fully eliminate all noise. To alleviate the feature blurring problem, many researchers [12, 33, 35, 39, 59, 63, 90, 91, 93, 101, 108, 109] propose to first filter facet normals and then synchronize the vertex position with corresponding newly computed facet normals. The basic observation of these methods is that the normal variation is more sensitive than the vertex position variation in reflecting the surface geometry. Moreover, several recent learning-based mesh denoising methods [38, 73, 78, 81, 106] also consist of the two sub-steps above: facet normal learning and vertex position adjustment. Note that some multi-stage denoising schemes, containing extra operations like vertex classification [7, 20, 28, 46, 86, 89], or pre-filtering [4, 7, 34, 45, 47, 70, 76, 82, 86, 89, 112], usually demonstrate more promising results,
but with more time required. Besides, the intermediate result of each sub-step may also greatly affects the final result. In summary, every method contains one or more of these sub-steps. The key differences lie in how to solve these sub-steps effectively, efficiently, and robustly, by exploiting various kinds of priors, assumptions, and mathematical models.

This article surveys the advances in mesh denoising including both geometric methods and learning-based methods during the past decades. A more detailed classification will be discussed in this work. We hope the survey provides readers with some insights into the state-of-the-art denoising methods and potential directions for future work. It may motivate researchers to develop new ideas for high-quality and robust mesh denoising. A benchmark is also built in this work, which includes detailed qualitative and quantitative evaluations of 19 advanced denoising algorithms. We will release all of these results on our specially designed website.

The survey is organized as follows. In Section 2, we first introduce the problem statement and then discuss several main challenges existing in mesh denoising. Section 3 provides two categorizations to overview the existing mesh denoising methods. From Sections 4 to 6, three mainstream categories of mesh denoising methods are discussed in detail. Section 7 concludes several mesh vertex position adjustment methods. Furthermore, a general comparison is demonstrated in Section 8 to give readers a basic evaluation of several state-of-the-art approaches. Conclusions and potential directions for future research are presented in Section 9.

2 PROBLEM STATEMENT AND MAIN CHALLENGES IN MESH DENOISING

Problem statement: Feature-preserving mesh denoising aims to recover the underlying surface signal (e.g., position, normal) from a noisy measurement, denoted as \( M' \). Essentially, the goal is to remove the noise \( \epsilon = M' - M \) while preserving the underlying features. This problem is inherently ill-posed, as both \( M \) and \( \epsilon \) are unknown. To make the problem tractable, priors and assumptions about \( M \) and \( \epsilon \) are often employed. For example, the noise pattern \( \epsilon \) might be assumed to follow a Gaussian distribution or to be independent and identically distributed. Meanwhile, recent learning-based mesh denoising methods adopt certain noise or shape priors in a data-driven manner. Note that when designing any denoising approach, several fundamental problems must be taken into consideration. Whether these denoising methods can handle these problems well is essential to users. We summarize these problems, in order to give the reader some basic knowledge.

Feature degradation: We consider the feature degradation problem from two aspects: feature blurring and feature over-sharpening. On one hand, as observed from the third sub-figure of Figure 1, since noise and certain-scale features are both regarded as high-frequency data, it is difficult to distinguish features from noise. Consequently, both of them may be regarded as noise and smoothed out. On the other hand, for preserving sharp features, some methods tend to over-sharpen the fine details in the models, as shown in the second sub-figure of Figure 1.

Volume shrinkage and collapse: Take a biomedical modeling case as an example, due to the parameter setting during CT/MRI scanning and organ segmentation, stepping noise and segmentation error often occur, which need to be eliminated. For this type of noise, isotropic filters achieve smoother results. However, it is very likely to cause serious deformation of the organ shape, as shown in Figure 2, which may affect the doctor’s diagnosis.

Mesh facet quality degradation: Generally speaking, the final step of denoising is always to move the vertex positions of the mesh, which is likely to cause triangle structure flipping, intersection, and overlapping.

3 CLASSIFICATION OF MESH DENOISING ALGORITHMS

In this section, we suggest two different criteria for classifying the mesh denoising methods.
3.1 Classification Based on the Ability of Surface Feature Preservation

An important principle of mesh denoising is the preservation of geometric features. Most early works focused on isotropic algorithms that ignore sharp features (i.e., independent of surface geometry), while recent methods are anisotropic and attempt to preserve geometric features in the data. Therefore, from the perspective of feature preservation, we can classify existing methods into the following two categories.

**Isotropic methods:** Isotropic methods do not take the geometric features of the surface into consideration, and treat the features and noise equally as the high-frequency information. For example, Taubin et al. [67] proposed a two-stage fairing method of surface signal processing, by alternately shrinking and expanding. Because of the linear complexity in both time and memory, this method can smooth large meshes very quickly. Later, Vollmer et al. [71] presented an improved Laplacian smoothing algorithm that filters noise efficiently but does not preserve features and shrinks the surface. Desbrun et al. [18] introduced an implicit fairing method, by using implicit integration of a diffusion process that allows for efficiency, quality, and stability. Their method can also guarantee volume preservation during smoothing. Some other smoothing methods [32, 40, 49, 62] are also considered as isotropic approaches, since they all lack awareness of the surface features. Generally, isotropic denoising methods are effective at removing noise; however, this comes at the cost of feature blurring or even complete loss of surface features. As a result, these methods may inadvertently destroy desirable surface characteristics while attempting to reduce noise. This tradeoff between noise removal and feature preservation is a critical aspect to consider when employing isotropic denoising methods.

**Anisotropic methods:** Given that isotropic methods hardly preserve geometric features in the object, the focus of many recent works has moved to anisotropic techniques [6, 16, 21, 22, 27, 31, 37, 45–48, 51, 63, 66, 73–76, 83, 84, 86, 95, 101, 112]. Most of these methods are derived from the
image processing domain, such as the anisotropic diffusion and classical bilateral filtering [69] in image processing.

Anisotropic diffusion based methods [6, 16, 27, 51, 53, 66, 102] take sharp features into account, such as the curvature tensor and the normal information of the mesh. For example, Ohtake et al. [51] proposed an adaptive smoothing method, which incorporates the local curvature information and allows the reduction of possible over-smoothing. Tasdizen et al. [66] extended the anisotropic diffusion, a PDE-based, edge-preserving, image-smoothing technique, to surface processing. Their 3D smoothing version can remove complex, noisy surfaces while preserving (and enhancing) sharp, geometric features. Hildebrandt et al. [27] defined an anisotropic mean curvature vector, which was used for feature-preserving noise removal. Based on the observation that $L_0$ minimization will lead to unexpected feature degradation and over-smoothness on sharp features, Liu et al. [44] proposed a novel second-order regularization approach for mesh denoising. Facet normals of the noisy mesh are optimized through variable splitting and the augmented Lagrangian method.

Another kind of anisotropic denoising method generalizes the bilateral filter from the image domain to 3D geometry. Fleishman et al. [21] and Jones et al. [31] pioneered designing mesh bilateral filtering methods, by directly adjusting vertex positions. Others [33, 61, 63, 77, 101, 108] applied the bilateral filter to the surface normals instead, using various normal similarity functions.

### 3.2 Classification Based on Mathematical Models

According to the mathematical model used for denoising, the existing mesh denoising methods can be divided into three main classes: optimization-based methods, filtering-based methods, and data-driven methods. Since we will introduce these three types of methods in detail in three later sections, we will not elaborate on them in this subsection.

**Optimization-based method:** For this set of techniques, the triangular mesh denoising problem is transformed into a geometric optimization problem. That is to say, based on the input mesh information and a set of constraints defined by the real geometric information or the noise distribution prior, an energy equation can be formulated. Solving the optimization function is equivalent to the geometry smoothing process.

**Filter-based method:** Filter-based mesh denoising is a class of mesh denoising techniques that use filtering operations to remove noise and improve the quality of a 3D mesh. These methods are inspired by image processing techniques, where filters are commonly applied to remove noise and enhance images. In the context of mesh denoising, filters are applied to the vertex positions, normals, or other attributes of the mesh to smooth out noise while trying to preserve important geometric features.

**Data-driven method:** The data-driven method builds a universal regression framework, by learning the mapping from noisy inputs to real surfaces. No user parameter tuning is required during the inference stage. It has achieved very satisfactory results in many testing models.

### 4 OPTIMIZATION-BASED METHODS

#### 4.1 Sparsity Optimization

Sparsity-based optimization and sparse signal representation have proven to be extremely powerful tools for processing various types of signals, such as audio, images, and videos. Figure 3 shows an illustration of smoothing a 1D signal by different error norms [64]. In particular, in the image smoothing domain, some detailed components in the input image can be largely smoothed out with the sparsity regularization, while keeping large structures. Similarly, for the 3D mesh model, we can build many sparsity assumptions to remove noise.
He et al. [26] generalized the $L_0$ norm, which directly measures sparsity, from the image to the triangular mesh, to preserve prominent sharp features and smooth the remainder of the surface. In the $L_0$ minimization framework, the general optimization function can be formulated as follows:

$$\min_s \| s - s^* \|_2^2 + \lambda \| \nabla s \|_0.$$  \hfill (1)

The first term in the optimization problem ensures signal fidelity, while the second term aims to create a piecewise smooth signal, where $\nabla s$ represents the vector of gradients of the signals. As can be observed from Figure 3, minimizing the $L_0$ norm of the signal gradients effectively smooths small signal fluctuations while preserving the most significant signal variations. Consequently, it is intuitive to extend the $L_0$ norm to the 3D domain. For 3D mesh surfaces, a discrete differential operator is required to serve as the function of the gradient and describe the smoothness of local structures. During the test, He et al. [26] found that using the discrete vertex-based Laplacian operator may fail to recover sharp features and shrink the surface away from the features. Therefore, they proposed to generalize the vertex-based Cotan operator to an area-based edge operator. Notably, the key observation of $L_0$ minimization is that a noise-free mesh should be smooth, except for its geometric feature regions. Hence, it achieves better results for computer-aided design (CAD)-like mesh models with piece-wise smooth surfaces. However, when facing mixed noises or texture-rich models, this method may yield apparent visual artifacts, like sharp feature distortion or over-sharpening. In addition, the objective function of the $L_0$ smoothing algorithm is non-convex and nonlinear. Each iteration of the algorithm must solve a different large linear system, leading to relatively large time consumption.

Since $L_0$ minimization is indeed a non-convex problem, He et al. [26] employed the same strategy in [88], by introducing auxiliary variables and alternating $L_0$ and $L_2$ minimization. However, the $L_2$ term is sensitive to outliers and may propagate the errors to the $L_0$ term. Thus, the results calculated by their method are not sparse enough. To deal with this problem, Cheng et al. [15] proposed a new approximation algorithm for the $L_0$ gradient minimization problem. This algorithm is based on a fused coordinate descent framework and is able to obtain a solution with good gradient sparsity and sufficiently close to the original input. Cheng et al. [15] applied this scheme to smooth mesh facet normals and their method achieves a better sharp feature preservation effect.

Later, in the same framework of $L_0$ minimization, Zhao et al. [107] presented a novel sparse regularization term to measure the sparsity of geometric features and distinguish features from noises. Both vertex positions and facet normals are optimized in the $L_0$ framework to faithfully remove noises and preserve features. Furthermore, an improved alternating optimization strategy was also developed in [107] to address the $L_0$ minimization problem with guaranteed convergence and stability.
In view of the difficulty and long time required to solve the $L_0$ norm, \cite{45, 79, 87} replaced the $L_0$ norm with the $L_1$ norm and turned to solve a convex optimization problem. In particular, based on the compressed sensing theory, Wang et al. \cite{79} presented a $L_1$-analysis compressed sensing optimization to recover sharp features from the residual between the base mesh and the input mesh. The base mesh is first computed by a global Laplacian regularization denoising scheme. The main insights in this work lie in two aspects: (1) it has been proven that the unknown reasonable signals can be recovered if they are sparse in the standard coordinate basis or sparse with respect to some orthogonal basis; (2) the $C^0$ sharp features on the shape can be sparsely represented in a coherent dictionary. Both theoretical and experimental results have shown that this method can faithfully decouple noise and sharp features.

Recently, under the observation that the Total Variation \cite{99} and $L_0$ \cite{26} based minimization methods can preserve sharp features well, but generate the side-effect of staircase artifacts, Liu et al. \cite{42} presented a high order normal filtering model with dynamic weights for preserving sharp features and removing the staircase effect in smooth regions simultaneously. The dynamic weights are applied in the proposed model to significantly improve the effectiveness of preserving sharp features. As stated in this work, the dynamic weights can penalize smooth regions more than sharp features, which can be applied to achieve the lower-than-$L_1$-sparsity effect. During our test, this method achieves better smooth results than $L_0$ [26], especially for the regions with stepping noise (see from evaluations section). TGV \cite{43} presents a Total Generalized Variation-based denoising method for 3D mesh surfaces. TGV incorporates both a first-order term and a second-order term, which together enable the preservation of sharp features and the recovery of smoothly curved regions. This method demonstrates favorable performance in comparison to DNF-Net \cite{36} on various types of meshes, including real-scanned models with sharp features and smooth regions, as well as CAD models.

4.2 Low-Rank Optimization

In the last decade, the concept of non-local filtering has been extensively developed in various image processing applications, by combining the non-local self-similarity prior and the low-rank prior. It assumes that similar patches exist in the whole field of an image. If these similar patches are reshaped as vectors, they are highly linearly correlated. Thus, a patch matrix $G$ constructed by stacking these patch vectors is low-rank. The general formulation as follows:

$$\min_X \text{rank}(X) + \lambda \|X - G\|_F^2, \quad (2)$$

where the first term is the rank of $X$, the second term represents the data fidelity measured by the square of the matrix $F$-norm, and $\lambda$ is a tradeoff parameter between the loss function and the low-rank regularization.

Intuitively, this kind of method can also be extended to the 3D domain, due to the fact that the surface patches with similar intrinsic properties often exist on the underlying surface of a noisy mesh (see Figure 4). However, the methods developed in the image domain assume a 2D signal defined in a rectangular image domain with uniform grid sampling, so they cannot be used directly to handle 3D meshes. The main challenges are three-fold \cite{82}:

- Q1: How do you handle irregular connectivities and samplings of 3D surfaces?
- Q2: How do you define the similarity between any two 3D surface patches?
- Q3: How do you recover the low rank from a noisy matrix effectively?

Motivated by the above problems, some researchers have tried to solve them \cite{13, 37, 48, 82}. They all follow a similar pipeline: local patch definition, similarity metric definition, similar patch identification, low-rank matrix construction, low-rank matrix recovery, and final surface calculation.
Fig. 4. An illustration of non-local similarity. The red patch is the reference patch and the blue patches are similar patches. Note that this figure is taken from [13].

Table 1. Comparison of the Processing Schemes for Three Challenges of the Four Low-Rank Mesh Smoothing Methods

| Methods          | Q1                          | Q2                          | Q3                          |
|------------------|-----------------------------|-----------------------------|-----------------------------|
| [82]             | 2D parameterization         | normal voting tensor        | kernel low-rank recovery     |
| [37]             | ring-based ordering scheme  | guided normal patch covariance| truncated \( y \) norm      |
| [13]             | corresponding nearest points| iterative closest point     | nuclear norm minimization   |
| [48]             | local isotropic structure   | tensor voting               | improved weighted nuclear norm minimization |

Meanwhile, they all do not directly apply the low-rank recovery on the vertex position domain of the mesh, but on the normal field. We describe the main differences between the four methods, as shown in Table 1.

**Wei et al.’s method:** For each reference facet, Wei et al. [82] first collected a set of facets with the most similar local surface properties from the whole mesh and constructed a local patch for each of these facets. To measure the similarity between any two local patches, they employed the robust normal tensor voting:

$$T_f = \sum_{f_i \in P(f)} \mu_i n_i n_i^T,$$

where \( P(f) \) denotes the facet \( f \)'s associated patch, \( \mu_i \) is certain weight, and \( n_i \) is the normal of \( f_i \). The similarity metric is defined as

$$\rho_{i,j} = \|\lambda_{1,i} - \lambda_{1,j}\|^2_2 + \|\lambda_{2,i} - \lambda_{2,j}\|^2_2 + \|\lambda_{3,i} - \lambda_{3,j}\|^2_2,$$

where \( \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0 \) are the eigenvalues of \( T_f \).

To address the issue of patch irregularity, the authors employed a 2D parameterization scheme, as depicted in Figure 5. They reshaped the normals of each regular patch into a patch vector and used these vectors to construct a patch matrix \( G \). To handle the nonlinear structure contained in the data, they pursued low-rank matrix recovery in the kernel space by utilizing half-quadratic minimization and the specifics of a proximal-based coordinate descent method. The authors ultimately obtained reliable normals from each recovered matrix, which they then provided as guidance to the bilateral normal filter for further mesh denoising. By leveraging this approach, the researchers were able to overcome the challenges posed by patch irregularity and develop a more effective denoising method for 3D mesh surfaces.

**Li et al.’s method:** Different from Wei et al. [82] searching similar facets, Li et al. [37] searched for each vertex \( v_i \) on the input mesh a set of similar vertices. To determine the similarity between two vertices, the authors introduced the **Normal Patch Covariance (NPC)** descriptor, which is
designed to describe the geometry of the local neighborhood surrounding a given vertex:

$$C(v_i) = \frac{1}{N_p} \sum_{l=1}^{N_p} (n_l - \bar{n}_i)(n_l - \bar{n}_i)^T.$$  (5)

Actually, the above equation is a normal-based PCA. To effectively calculate the distance between two covariance matrices, this approach defined the NPC distance between vertices $v_i$ and $v_j$ as

$$d_{NPC}(v_i, v_j) = \sqrt{\frac{1}{N_p} \sum_{l=1}^{N_p} (n_l - \bar{n}_j)(C(v_i) + C(v_j))^{-1}(n_l - \bar{n}_j)^T}.$$  (6)

Due to the noise impacts, the above metric computed from the noisy input is unstable. Hence, to further improve the selection of similar vertices, Li et al. took a pre-filtered mesh as a guidance signal to compute the guided normal patch covariance (G-NPC) distance:

$$d_{G-NPC}(v_i, v_j) = d_{NPC}(v_i, v_j) \cdot d_G(v_i, v_j),$$  (7)

where $d_G(v_i, v_j)$ is computed from the pre-filtered mesh.

This method then used the ring-based NPC ordering scheme to achieve regular normal patch structures, without using the 2D parameterization. Finally, an improved truncated $\gamma$ norm is utilized to recover the low-rank matrix.

Following the above two representative works, Chen et al. [13] designed a joint low-rank recovery model to smooth out rich geometric features on a clean surface. At the same time, it is worth noting that Lu et al. [48] proposed a new idea to utilize the low-rank prior for 3D geometry filtering. The interesting idea is that they filled the low-rank matrix with the normals from all similar local isotropic structures. That is to say, all the normals in their patch matrix are nearly the same. Figure 6 shows an illustration of a normal patch matrix built by [13, 37, 82] (left) and [48] (right).

### 4.3 Spectral Optimization

Spectral processing encompasses computation on the eigenvalues, eigenvectors, or eigenspace projections obtained from suitably defined mesh operators [98]. Arvanitis et al. [3] proposed a coarse-to-fine strategy. This approach includes a distinct coarse denoising step that identifies sharp and small-scaled geometric features, followed by a feature-aware guided normal filtering step to enhance the quality of the denoised outcomes.

### 5 FILTER-BASED METHODS

Filter-based mesh methods usually assume that the surface noise is high-frequency, and denoise the noisy surface by a vertex-based or normal-based filter. The earlier methods belong to the...
vertex-based filtering method, for example, Laplace smoothing or its improved versions [18, 54, 67]. However, they will smooth out the surface features.

Later, more methods tend to perform filtering in the mesh normal field, since first-order normal variations better capture the local surface variations. Although there are various kinds of normal-based filters, all of them can be concluded in a general form:

$$\mathbf{n}_{t+1}^i = \Lambda \left( \sum_{j \in N(i)} w_j \mathbf{n}_j^i \right),$$  \hfill (8)

where $\Lambda(\cdot)$ is a normalization operation, and $w_j$ is the weight that the neighboring normal $\mathbf{n}_j^i$ contributes to the updating of $\mathbf{n}_{t+1}^i$. The above function means the weighted average of the neighboring facet normals of the current facet. Actually, the key contribution of most normal-based filters is to compute the most reliable weight $w_j$. Specifically, the mean filter algorithm [91] defines the weight by the area of its neighboring facets. Following it, the alpha-trimming filtering approach proposed by [92] replaces the area-based weight with the normal differences between the current facet and its neighbors. The fuzzy vector median filtering [59] first uses vector median filtering, and then uses the Gaussian function to compute the weight, in order to obtain more accurate normals.

In addition to the above lateral (namely single weight) normal filters, another well-known filter is the bilateral (namely two weights) filter, which was first proposed by Tomasi et al. [69]. This kind of filter is able to preserve edges by means of a nonlinear combination of nearby image values. The bilateral filtering for image $I(u)$, at coordinate $u = (x, y)$, is defined as

$$\hat{I}(u) = \frac{\sum_{p \in N(u)} W_c(||p - u||) W_s(||I(u) - I(p)||) I(p)}{\sum_{p \in N(u)} W_c(||p - u||) W_s(||I(u) - I(p)||)},$$  \hfill (9)

where $N(u)$ is the neighborhood of $u$. This filter contains two monotonically decreasing Gaussian functions $W_c(\cdot)$ and $W_s(\cdot)$, which are used to measure the spatial weight and intensity weight.

When extending this filter to the 3D domain, there are two main categories: filtering the vertex positions or facet normals. In the following, we will give the details on how to use the bilateral filter on mesh surfaces, as well as its improved variants.

### 5.1 Bilateral Filter on Mesh Vertex Positions

Fleishman et al. [21] proposed to modify the vertex positions in the normal direction via a weighted average. The two weights are (i) the spatial weight that measures the closeness from the neighbor vertex to the target vertex, and (ii) the offset weight that measures the distance between the neighbor vertex to the tangent plane defined by the target vertex and its normal. Jones et al. [31] also followed a bilateral filtering framework. Besides the similar spatial weight, they used a new influence weight that depends on the distance between the prediction and the original position of the target vertex.
5.2 Bilateral Filter on Mesh Normals

5.2.1 Bilateral Facet Normal Filter. Zheng et al. [108] developed two versions of bilateral filter in the mesh facet normal field. Both the two denoising schemes regard the facet normals as a surface signal parameterized on an input mesh and formulate the influence of both spatial difference and signal difference into bilateral weighting. Specifically, the signal weight is defined as

\[ W_s(\|\mathbf{n}_i - \mathbf{n}_j\|) = \exp\left(-\frac{\|\mathbf{n}_i - \mathbf{n}_j\|^2}{2\sigma_s^2}\right). \]  

(10)

The spatial weight is defined as

\[ W_c(\|\mathbf{c}_i - \mathbf{c}_j\|) = \exp\left(-\frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2}{2\sigma_c^2}\right), \]  

(11)

where \(\sigma_s\) and \(\sigma_c\) are both corresponding standard deviations, with which we can adjust the denoising and feature preservation power. Spatial difference focuses on smoothing the surface based on the proximity of vertices, emphasizing the preservation of the overall structure and shapes. Signal difference emphasizes the preservation of edges and fine details by taking into account the intensity variations between vertices and avoiding blending across sharp boundaries. By combining these two weightings, bilateral filtering effectively reduces noise in surfaces while maintaining important features and edges. Typically, \(\sigma_s\) lies in the range of \([0.2, 0.6]\), and \(\sigma_c\) is set as the average distance of all adjacent facets in an input mesh.

The local version of mesh normal filtering is formulated as

\[ \mathbf{n}_{i}^{t+1} = K_i \sum_{j \in N(i)} \xi_{ij} W_c(\|\mathbf{c}_i - \mathbf{c}_j\|) W_s(\|\mathbf{n}'_i - \mathbf{n}'_j\|) \mathbf{n}'_j, \]  

(12)

where \(K_i = K(\mathbf{c}_i) = 1/\sum_{j \in N(i)} \xi_{ij} W_c(\|\mathbf{c}_i - \mathbf{c}_j\|) W_s(\|\mathbf{n}'_i - \mathbf{n}'_j\|)\) is the normalization factor, \(N(i)\) is the 1-ring facet neighborhood of a facet \(f_i\), and \(\xi_{ij}\) is the weight to account for the influence from surface sampling rate. Notably, during the test, the authors found that the 1-ring facet neighborhood works well for non-CAD models and the 2-ring is more suitable for CAD-like models. This is caused by more facets involved in the 2-ring neighborhood, and thus it is able to better characterize sharp features, especially sharp edges.

Applying a local bilateral filter iteratively is able to propagate the filtering effect throughout the whole surface. Apart from it, Zheng et al. [108] also proposed an alternative solution to update all the new normals in a single pass by minimizing:

\[ E = \sum_i A_i \left(\|\mathbf{n}'_i - K_i \sum_{j \in N(i)} \omega_{ij} \mathbf{n}'_j\|^2 + \lambda \sum_i A_i \|\mathbf{n}'_i - \mathbf{n}_i\|^2\right), \]  

(13)

where \(\mathbf{n}'_i\) are the unknown normals for the denoised mesh, and \(\omega_{ij} = \xi_{ij} W_c W_s\) is the averaging weight in Equations (10) and (11) measured on the input mesh, \(A_i\) is a weight of face area, and \(\lambda\) is a balance parameter. The first term is the Laplacian normal smoothness term, and the latter one constrains the produced normals as similar to the original normals. By tuning \(\lambda\), users can control the degree of denoising. In contrast, the local, iterative scheme is faster and has lower memory consumption than the global, non-iterative scheme. However, when the input is polluted by high-level noise, the local version produces a more pleasing result (see from Figure 7). Dai et al. [17] proposed a new avenue to address the mesh denoising problem. To avoid the disturbance of anisotropic information, Dai et al. [17] first cluster the mesh model via an edge metric for mesh segmentation. This framework can be easily embedded into a common mesh denoising framework [101, 108].
5.2.2 Guided Normal Filter. Although the bilateral filter has achieved impressive performance, one obvious drawback is that the intensity difference weight is less reliable in the feature/structure regions. This is because the noisy input provides less reliable intensity (or the normal field in the 3D mesh surface) information. To resolve this problem, the joint bilateral filter was further proposed in the image processing domain, for processing flash/no-flash image pairs [55] and [19]. The key idea is that the intensity weight is computed from another image, called the guided image or guidance, instead of the input image. Theoretically speaking, if we can provide more reliable information about the image structure, joint bilateral filtering can produce better results.

Inspired by the success of the joint bilateral filter in image smoothing, Zhang et al. [101] proposed a new triangular mesh normal filtering framework, called guided normal filter (GNF). Its formulation is written as

$$\mathbf{n}_i^{t+1} = K_i \sum_{j \in \mathcal{N}(i)} \zeta_{ij} W_c(||\mathbf{c}_i - \mathbf{c}_j||) W_s (||\mathbf{g}_n^i - \mathbf{g}_n^j||) \mathbf{n}_j^t,$$

where $\mathbf{n}_i^{t+1}$ is the filtered normal for facet $f_i$, and the $\mathbf{g}_n^j$ is the so-called guidance. The main challenge lies in the construction of the guidance normal field, which needs to be defined in the same domain as the input while providing enough information about the desired output. To realize it, Zhang et al. [101] defined an isotropic neighboring patch for each facet. In this patch, the normals of all facets are nearly in the same direction (see Figure 8). Then, the average normal of the chosen patch is used as the guidance normal for this facet. As we observed from Figure 8, the neighboring patch stays on the one side of a sharp feature and will not cross the edge, so the produced guided normal is feature-aware and thus leads to better feature preservation results.

A drawback of this approach is the strict requirement for correct guidance normal. This limitation was addressed in the work of [80, 104, 105], but it is still hard to handle complex structures, such as narrow edges and corners.

Guided normal [101] has provided a robust estimation for filtering the noisy mesh. However, the ability of sharp feature preserving was alleviated on meshes that contain narrow structures, multiscale features, and fine details. To address this problem, Guo et al. [24] proposed a new consistency
measurement. This method selects consistent facets around a central facet in a patch-shift manner. Then, different adaptive neighborhoods on the mesh are constructed through iterative graph-cut to match the corresponding local shapes. This method improves the results of the guided normal filtering with the help of consistent neighborhoods. Corner features are often ignored or over-smoothed in previous guided mesh filtering methods. Based on this observation, Liu et al. [41] proposed a facet normal filtering method considering corner features. A further classification step on feature points is executed via the normal voting tensor (NVT) method, where the non-feature points are updated with isotropic neighborhood normals, which effectively suppressed the sharp edges from being smoothed. The feature points are updated based on local geometric constraints. Updating points according to different categories enables mesh feature preservation and avoids sharp pseudo features.

5.2.3 Rolling Guidance Normal Filter. Instead of removing noise, smoothing geometric details (or geometry texture) like bumps, ridges, creases, and repeated patterns, is also essential for many applications [77]. Wang et al. [77] proposed a simple and effective scale-aware mesh smoothing filter, called the rolling guidance normal filter (RGNF), to process different scales of geometry features of triangular meshes. The key idea is to iteratively smooth out small geometric variations while preserving large-scale features, by applying an iterative joint bilateral filter to face normals. The iterative process can be summarized as

$$f_{n+1} = \sum_{j \in N(i)} \xi_{ij} W_c(\|c_i - c_j\|) W_s(\|f_{n+1} - f_n\|) n_j,$$

where $f_{n+1}$ is the filtered normal for facet $f_i$. The main differences between RGNF and GNF lie in three aspects: (1) the first iteration of RGNF is the Gaussian filter in nature, since $f_n^0 = 0$; (2) the filtered normal result $f_{n+1}$ is a weighted average of the original normal $n_j$, which is constant during all iterations; (3) the guidance is the normals from last iteration. The principle of the rolling guidance filter can be understood as follows. In the first iteration, the filter is actually a Gaussian filter, thus the small geometric features whose scale is smaller than $\sigma_s$ can be smoothed effectively. Unfortunately, large structures are also smoothed to some degree. In the later iterations, $f_n$ is no longer zero. For the removed small features, their guidance normals are almost equal. So, the weight term $W_s$ approximately equals 1, and the filter is still a Gaussian filter, resulting in small structures still being removed. For large-scale structures, they are recovered gradually, since their guidance normals can provide effective weights, and thus joint bilateral filter sharpens the smoothed edges of large structures. Note that the task of geometry feature removal is different from denoising. As demonstrated in this work, it is clear that this method cannot recover the sharp feature well.

5.2.4 Static/Dynamic Filter. Recently, Zhang et al. [100] proposed a new approach for filtering signals defined on mesh surfaces, by utilizing both static and dynamic guidance. Static guidance means a guidance signal is obtained beforehand, and keeps unchanged during the filtering processing, while the dynamic guidance is just like the $f_{n+1}$ in RGNF, which is iteratively updated. Both of the two guidance methods have their own advantages and disadvantages: static guidance enables direct and intuitive control over the filtering process, but may be inconsistent with the input and lead to unsatisfactory results; dynamic guidance is adjusted according to the current signal values, but sensitive to noises and outliers. For taking benefits from both kinds of guidance, Zhang et al. [100] formulated the target function as

$$E_{SD} = \sum_i A_i ||n_i - \hat{n}_i||^2 + \lambda \sum_{j \in N(i)} \sum_{f_j \in N(i)} [A_j \cdot \phi_{\mu}(\|c_i - c_j\|) \cdot \phi_{\nu}(\|g_i - g_j\|) \cdot \psi_{\mu}(\|n_i - n_j\|)\].$$

ACM Trans. Multimedia Comput. Commun. Appl., Vol. 20, No. 3, Article 85. Pub. date: November 2023.
The first term in the target function is a fidelity term that requires the output normal field to be close to the original normals, while the second term is a regularization for the output normal field. Specifically, \((g_i, g_j)\) are from static guidance and \((n_i, n_j)\) are dynamic current normals. Extensive experimental cases show that this method is able to effectively smooth surface noise or small-scale geometric features.

5.2.5 Bi-Normal Filter. As we observed, the above-mentioned filters only use the facet normal field. Nevertheless, there are two common normal fields of a mesh surface, i.e., the facet normal field and the vertex normal field. As known, each vertex’s normal can be directly computed, by averaging its neighboring facet normals. As a result, there may exist redundant geometry information of the same mesh surface in the two fields. Hence, most of the normal based denoising methods can achieve quality results, by using only a single normal field [86]. However, Wei et al. [86] believed that these two types of normal fields also have some differences. Specifically, the facet normal field tends to reflect the global geometric variations of a mesh surface, while the vertex normal field is more illustrative of the local details of mesh vertices.

Based on this observation, Wei et al. [86] designed a novel strategy, called Bi-normal filtering, to more accurately compute the two normal fields and employ them to collaboratively guide the optimization of the vertex position for preserving features and dealing with irregular surface sampling. First, all vertices in the input mesh are labeled as feature points and non-feature points, such that the mesh surface can be processed piecewise. The local bilateral normal filter is then used to initially estimate the facet normal field, for avoiding the influences of noise on the following steps. In order to more accurately estimate the vertex normal field and refine the facet normal field at the feature regions, a neighboring facets clustering scheme is proposed, which can provide an isotropic neighborhood for the normal estimation of feature vertices. Finally, all the mesh vertices are moved to new positions to simultaneously match the two normal fields.

6 DATA-DRIVEN METHODS

Data-driven methods have gained widespread popularity in image denoising, and developing data-driven mesh denoising techniques for the 3D domain has become a burgeoning area of research. However, the direct application of learning-based techniques to irregular 3D mesh structures is challenging, unlike the regular grid structure found in 2D images. To this end, two intuitive approaches are proposed: (1) extracting meaningful handcrafted feature descriptors or representing the mesh surface in a regular format, enabling the usage of established networks; and (2) designing a specialized mesh denoising network architecture, similar to PointNet [56]. In the following sections, we will first introduce several datasets available for learning-based mesh denoising, and then detailedly discuss these approaches in greater detail.

6.1 Dataset for Learning Based Mesh Denoising

6.1.1 SynCNR. Three types of models were collected in CNR [76], including CAD-like models, smooth models, and models with rich fine-scale features. For the training set, CNR selected 21 models and added three different levels of Gaussian noise, yielding 63 meshes in total. In the testing set, there are 29 models selected which contain 87 meshes in total. It is important to note that facet normals or vertex normals are portable, as they are relatively easy to compute.

6.1.2 Kinect Repository. Wang et al. [76] proposed a real-scan dataset for learning-based mesh denoising. Seven models are scanned using high-precision Artec Spider™ (accuracy 0.5 mm) scanner where the results are regarded as ground-truth models. CNR [76] employs two generations of Microsoft Kinect to create datasets. There are three variants of the Kinect dataset in CNR, including Kinect v1, Kinect v2, and Kinect Fusion. The primary distinction between them is that Kinect
Fusion is reconstructed using the KinectFusion technique [29]. The Kinect repository contains the David model (24 meshes), the Girl model (25 Kinect v1 and 24 Kinect v2 meshes), the Cone model (12 meshes), and the Pyramid model (12 meshes). Each real-scanned model is paired with a ground-truth model and registered by the ICP technique.

6.1.3 PrintData. Recently, Shen et al. [60] introduced a new real-scan dataset: PrintData, comprising 20 3D models from a 3D model repository available online at 3dmag.org. Subsequently, physical models were created by 3D-printing these digital models utilizing a high-end 3D printer (Stratasys Eden260v), and then scanned through high-resolution scanners (Artec SpiderTM and SHINNING 3D EinScan Pro 2x).

6.2 Handcrafted Descriptor Based Method

6.2.1 Cascaded Normal Regression for Mesh Denoising. Specifically, Wang et al. [78] developed a filtered facet normal descriptor (FND) for expressing the geometry features around each facet on the noisy mesh and modeled a regression function with neural networks for mapping the FNDs to the facet normals of the denoised mesh. According to the filter used in constructing FNDs, two kinds of FND are designed:

1. **Bilateral filtered facet normal descriptor (B-FND):**
   \[
   S_i := \left( n_i^1(\sigma_{s1}, \sigma_{r1}), \ldots, n_i^K(\sigma_{sL}, \sigma_{rL}) \right)
   \]
   (17)

2. **Guided filtered facet normal descriptor (G-FND):**
   \[
   S_i^g := \left( gn_i^1(\sigma_{s1}, \sigma_{r1}), \ldots, gn_i^K(\sigma_{sL}, \sigma_{rL}) \right)
   \]
   (18)

where each B-FND consists of a series of bilateral normal filtering [108] results with different parameter pairs, including Gaussian parameters \((\sigma_{s1}, \sigma_{r1})\) and the max iteration number \(K\), while each G-FND is computed by guided normal filtering [101] with the same parameters. The key idea of this method lies in the design of FND. In the past, when we used bilateral normal filter to remove noise, it was hard to find the most accurate parameters when the input mesh had features at different scales. Therefore, the denoised mesh is not always optimal. In this work, the designed FND models multi-scale features well and is robust to noise. Besides, with the help of the neural network, we do not need to do the tedious parameters tuning, because the network directly learns an optimal mapping from FND to the ground-truth normal results. Actually, we can intuitively and simply consider that the network learns a weighted combination of many bilateral normal filtering results, but best fitting the real accurate normal. Visual results and quantitative error analysis both demonstrated that this method outperforms the previous state-of-the-art mesh denoising methods and successfully removes different kinds of noise for meshes with various geometry features.

6.2.2 Data-Driven Geometry-Recovering Mesh Denoising. As known, the magnitudes of geometric details are usually similar to and smaller than the noiseâ€”i.e., and thus noise and geometric details are commonly removed simultaneously by existing structure-preserving mesh filters. Moreover, existing denoising techniques have no mechanism to recover mesh geometry once it is lost.

Motivated by this problem, and following the learning-based framework of [78], Wang et al. [73] proposed to remove noise, while recovering compatible surface geometry, by a two-step learning
scheme. First, they learn the mapping function from the noisy model set to its ground-truth counterpart set using neural networks for removing noise, and then learn the reverse procedure of mesh filtering and recover geometry from denoised meshes using the learned regression function sequences. More simply put, there are two sub-networks: one is for removing noise, and another is for recovering lost geometry.

**Noise removal module:** In the first-step learning for noise removal, the B-FND (see Equation (17)) is directly used as the feature descriptor.

**Geometry recovery module:** In the second-step learning for geometry recovery, a new reverse B-FND ($r$B-FND) is proposed. The reverse normal filtering is formulated as

$$n^{k+1} = \Lambda(n^k + n^* - f(n^k)),$$

(19)

where $k$ denotes the iteration number, $n^*$ is the initial filtered facet normal which is unchanged during the reverse filtering, and $f(\cdot)$ is the bilateral normal filtering operator. Then, the rB-FND is defined as

$$S_{ij} := \Lambda (2n_i - n_i (\sigma_{s1}, \sigma_{r1})) \ldots \Lambda (2n_i - n_i (\sigma_{sL}, \sigma_{rL}))).$$

(20)

In the training stage, Wang et al. [73] mapped two large sets of B-FNDS and rB-FNDS to the corresponding facet normals of the ground-truth mesh, respectively. The extreme learning machine is employed as the regression method. Detailed quantitative and qualitative results on various data demonstrate that this two-step learning algorithm competes favorably with the previous state-of-the-art methods, especially in terms of mesh geometry preservation.

Recently, Zhu et al. [111] introduced a geometry-aware cascaded guided bilateral normal filtering method. This approach constructs a multi-scale guided filtered normal descriptor (M-GFND) and employs height maps to update facet normals. By doing so, M-GFND effectively enhances the edge-preserving capabilities of normal filtering algorithms. Nousias et al. [50] proposed a normal filtering method that takes a patch of neighboring faces as input, which describes each local patch with a scale, translation, and rotation invariant representation. This method contributes to the field of geometric deep learning where the sampling of the latent space is nonuniform. It uses predefined parameters, which saves the cost of searching for optimal parameters for each model. However, it still requires separate training for different noise types.

**6.2.3 Voxel-Based NormalNet.** The voxelization strategy is another useful and intuitive scheme, by which we can convert the irregular local structure into a regular volumetric representation. Based on this representation, Zhao et al. [106] designed a learning-based normal filtering scheme for mesh denoising, called NormalNet, to mimic the framework of GNF. Specifically, this scheme follows the iterative framework of filtering-based mesh denoising. During each iteration, first, the voxelization strategy is applied on each face in a mesh to transform the irregular local structure into the regular volumetric representation (see Figure 9). Therefore, both the structure and face normal information are preserved and the convolution operations in CNN can be easily performed. Secondly, instead of the RTV-based guidance normal generation and the guided filter in GNF [101], a deep CNN is utilized, which takes the ground-truth normal as the guidance, to simulate the filtering process of GNF. Finally, the vertex positions are updated based on the filtered normals. It is important to note that the voxel-like structure demands more memory and increased training time when using high voxel resolutions. However, low voxel resolutions struggle to accurately represent shape information. Consequently, this tradeoff can be considered a limitation of this method.

**6.2.4 NormalF-Net: Normal Filtering Neural Network.** To circumvent complex voxelization operations for regularization, Li et al. [38] proposed an innovative mesh normal filtering network that
emulates low-rank recovery. This method integrates non-local similarity (which the authors consider as geometry domain knowledge) with existing CNNs. Specifically, the approach follows the general pipeline of low-rank recovery-based mesh denoising work, with the primary distinction being the use of CNNs to mimic the function of low-rank recovery. Both objective and subjective evaluations attest to the method’s superiority in terms of noise removal and feature preservation. However, the search for non-local similar patches renders this method time-consuming, as evidenced by the timing comparisons in [38].

6.2.5 Mesh Defiltering via Cascaded Geometry Recovery. In this section, we introduce another non-traditional but practically meaningful work, mesh defilter [81]. This reverse-filtering approach (termed as DeFilter) seeks to recover the geometry of a set of filtered meshes to their artifact-free status. This work is motivated by these observations: (1) there is a huge volume of mesh denoising methods, thus resulting in many denoised results; and (2) despite significant advancements, mesh denoising remains an ill-posed problem, and no existing algorithm can serve as a universal solution for diverse applications. These algorithms inevitably produce denoised results that involve a tradeoff between noise removal and geometry preservation.

Firstly, Wei et al. [81] smooth a set of ground-truth models to obtain over-smoothed results. They also employ a set of mesh filters to generate a variety of over-smoothed models, which lose surface geometry to different extents. Subsequently, they propose the generalized reverse filtered facet normal descriptor (grFND) as the feature descriptor and formulate each ground-truth facet normal as a function of the over-smoothed facet’s grFND. They then learn the function between the grFNDs and the ground-truth facet normals using the N-EML. Multiple iterations are necessary, as the initial regression function only coarsely establishes the correspondence between the grFNDs and the ground-truth normals. The performance of DeFilter has been thoroughly validated through various results, demonstrating its ability to recover the geometry of most denoised meshes without requiring knowledge of the specific filter used previously.

6.3 Direct Deep Learning Based Method

6.3.1 DNF-Net. DNF-Net [36] encodes vertex positions and adjacency matrices using a neural network. This pioneering work denoises meshes by filtering raw facet normals. DNF-Net processes facet normals of local patches extracted from the original mesh model. By training patches grouped on noisy meshes and ground-truth meshes, the network learns to predict clean facet normals through cascaded residual learning units. However, as with most data-driven methods, limitations persist. Differences in noise between well-characterized mesh data and an unknown test mesh can lead to unsatisfactory results. Further exploration in data transfer and domain adaptation is needed to address these challenges.

6.3.2 IMD-Net. IMD-Net [8] employs a deep neural network with gauge equivariant convolutional layers, enabling the network to denoise irregular noisy meshes through global mesh transformations as well as local feature constellations and orientations. IMD-Net’s computational complexity is comparable to that of traditional conv2D kernels. The algorithm is designed to preserve
natural object features while eliminating noise from the mesh. It has been demonstrated to be competitive with existing state-of-the-art techniques in terms of both metric evaluations and visual inspection.

6.3.3 Mesh Denoising Based on Graph Convolutional Network. Graph Convolutional Networks (GCNs) were first applied in [2] to learn a graph composed of noisy normals and noisy points. This work also follows a multi-scale feature extraction strategy on the graph of facets; however, it does not account for surface geometry information. A subsequent work, GCN-Denoiser [60], utilized a novel graph-based representation for local surface patches. GCN-Denoiser employed several static and dynamic graph convolutions to progressively regress noise-free facet normals. Another variant of the mesh denoising framework based on the graph structure, GeoBiGNN [103], uses two graph neural networks operating in both normal and spatial domains to achieve better results in feature preservation. To enhance the robustness of the algorithm, GeoBiGNN introduced a cascaded weight estimation module in conjunction with a graph pooling layer. The weakness of GeoBi-GNN is its requirement for iterative vertex updating. In contrast, DDMP [25] proposes a paired graph-based approach to perform end-to-end denoising without any post-processing (e.g., iterative vertex updating). DDMP synchronously trains two GCNs, PosNet for positions and NormNet for facet normals. During the collaborative training of the two graph neural networks, vertex positions and facet normals are repaired, respectively. However, this method requires an input prior to estimating vertex displacements and generating clean normals. A new approach DOGNET [65] performs localized graph neural network and propagates information across the mesh. DOGNET aggregates features from neighborhoods by iteratively updating node representations based on their local connectivity patterns. DOGNET outperforms other methods in terms of preserving fine-scale geometric details and maintaining the structural integrity of the mesh.

7 VERTEX UPDATING
As we can see from Sections 4 to 6, we introduce many normal filtering techniques. However, we need to keep in mind that mesh smoothing seeks to adjust mesh vertex positions to the best estimates of their true positions. Hence, in these works, the next step after normal filtering is the vertex position updating. Moreover, in many cases, a single iteration cannot yield satisfactory results, and they often require multiple iterations. In the following, we give a brief introduction of several typical vertex position updating approaches.

7.1 Locally Iterative Updating
Based on the observation that the filtered normal should be orthogonal to the three edges of each triangular facet in the mesh, Taubin [68] formulated the vertex updating problem in a least-squares sense as follows:

$$E(V) = \sum_{f \in F} \sum_{(i,j) \in \partial f} (n_f \cdot (v_i - v_j))^2,$$

(21)

where \(\partial f\) is the set of edges that constitute the boundary of facet \(f\). The classical gradient descent method can be directly used to minimize \(E(V)\), so vertex position updating is implemented as

$$v_i' = v_i + \lambda \sum_{j \in N_v(i)} \sum_{(i,j) \in \partial f_k} n_k' \cdot (n_k' \cdot (v_j - v_i)),$$

(22)

where \(N_v(i)\) is the 1-ring vertex neighborhood of a vertex \(v_i\) and \(\lambda > 0\) is the learning rate.
Since the choice of $\lambda$ is vital to the final results, to avoid carefully setting the value of $\lambda$, Ohtake et al. [52] explicitly fixed $\lambda = \frac{1}{6 \sum_{k \in F_V(i)} A_k}$. So, the vertex updating is reformulated as

$$v'_i = v_i + \frac{1}{3} \sum_{k \in F_V(i)} A_k \sum_{j \in N_V(i) \cap \partial f_k} A_k n'_k \left( n'_k \cdot (v_j - v_i) \right),$$

(23)

where $A_k$ is the area of the triangle $f_k$. Unlike Taubin’s algorithm, this method does not have the problem of choosing a learning rate, but it is computationally more expensive since it needs to compute triangle areas.

Later, to speed up the vertex updating scheme of [52], Sun et al. [63] revised the area weight:

$$v'_i = v_i + \frac{1}{3 |F_V(i)|} \sum_{j \in N_V(i) \cap \partial f_k} A_k n'_k \left( n'_k \cdot (v_j - v_i) \right).$$

(24)

This simple modification gives the same weight for each adjacent facet, regardless of its area. This is due to the fact that if a facet has a large area, there is generally a large distance between its vertices, and vertices with larger distances from the target vertex should have a weaker influence on the position updating.

### 7.2 Poisson Deformation

For some models with large-scale geometric features, the filtered triangle mesh facet normal may change greatly. If we directly employ the above-mentioned updating algorithm, large vertex offsets will occur, thus causing triangle flipping.

To circumvent this problem, Wang et al. [77] regarded this normal-guided vertex updating problem as a mesh deformation problem and adopted the Poisson Mesh Deformation [97] as their solver. The optimization function is formulated as

$$E_{\text{Poisson}} = \sum_{f_i} A_i \sum_{j=1}^{3} \left\| \nabla g_{i,j} - \nabla \hat{f}_{i,j} \right\|^2_2,$$

(25)

where $\nabla g_{i,j}, j = 1, 2, 3$ are the gradient vectors on the face $f_i$ with unknown vertex positions, and $\nabla \hat{f}_{i,j}, j = 1, 2, 3$ are gradients of the piecewise-linear basis nodal function defined on the triangle $\bar{f}_i$. $\bar{f}_i$ is obtained by rotating the original triangle $f_i$, according to the rotation transformation defined by the original normal and filtered normal. Minimizing this energy function concerning vertex positions leads to the well-known Poisson equation, which can be solved efficiently.

However, in some cases, the geometry texture is so complicated that a rapid change of face normals will still cause triangle flipping in minimizing $E_{\text{Poisson}}$. To ease this problem, a gradient smoothness regularization term is added:

$$E_{\text{smooth}} = \sum_{e_{ij}} l_{e_{ij}} \sum_{k=1}^{3} \left\| \nabla g_{i,k} - R_{ij} \nabla g_{j,k} \right\|^2_2,$$

(26)

where $e_{ij}$ represents an edge adjacent to faces $f_i$ and $f_j$, $l_{e_{ij}}$ is the edge length, and $R_{ij}$ is the rotation matrix that rotates triangle $f_j$ along $e_{ij}$ to be in the same plane of $f_i$. The added smoothness term penalizes the quick changes, thus reducing triangle flips.

The final energy function can be written as

$$E_{\text{update}} = \frac{E_{\text{Poisson}}}{A} + \lambda \frac{E_{\text{smooth}}}{l_e},$$

(27)

ACM Trans. Multimedia Comput. Commun. Appl., Vol. 20, No. 3, Article 85. Pub. date: November 2023.
where average face area $\bar{A}$ and average edge length $\bar{l}_e$ are used to normalize $E_{\text{Poisson}}$ and $E_{\text{smooth}}$ under an uniform scale, and $\lambda$ balances two energy terms. A larger $\lambda$ value reduces more flipped elements while increasing the approximation error of the target normals. It is set as 0.5 by default.

### 7.3 Bi-Normal Updating

There is also another kind of updating scheme, namely, incorporating both vertex normal and facet normal into the optimization function. The joint optimization problem proposed by Wei et al. [86] is formulated as follows:

$$\arg\min_{\mathbf{v}'} [E_1(\mathbf{v}') + \lambda E_2(\mathbf{v}')] .$$  \hfill (28)

The first term is defined as

$$E_1(\mathbf{v}') = \sum_{f \in \mathcal{N}_v(f)} [\mathbf{n}_f \cdot (\mathbf{v}' - \mathbf{c}_f)]^2,$$  \hfill (29)

where the first term encodes the sum of squared distances from vertex $\mathbf{v}'$ to its one-ring facets, $\mathbf{n}_f$ is the filtered normal of facet $f$, and $\mathbf{c}_f$ is its triangle barycenter.

The latter term is written as

$$E_2(\mathbf{v}') = (\mathbf{v}' - \mathbf{v}) \cdot (\mathbf{v}' - \mathbf{v}) - [\mathbf{n}_v \cdot (\mathbf{v}' - \mathbf{v})] \cdot [\mathbf{n}_v \cdot (\mathbf{v}' - \mathbf{v})],$$  \hfill (30)

where $\mathbf{n}_v$ represents the unit normal of vertex $\mathbf{v}$. The term encourages the vertex to move along its normal direction in the optimization. Only minimizing the first term derived from filtered facet normals, inevitably leads to feature blurring and vertex drifts to some extent. Thanks to the second quadric energy term, using the estimated vertex normals can produce better results.

### 8 EVALUATIONS

Given the wide diversity in mesh denoising methods, in this section, we first give a comprehensive comparison that describes the general abilities of the previous state-of-the-art methods to deal with different denoising challenges. Both visual and quantitative denoising results by different methods are then reported. To achieve a fair comparison, all of the results are provided by the authors, or produced by the open-source codes with our careful parameter tuning. We summarize the parameters of these methods in Table 1 in the supplemental material. For some results provided by the authors, without providing corresponding parameters, we directly leave them empty. The selected methods for the comparison cover all the common denoising categories.

The methods involved are as follows: Fleishman et al. [21] (BMF), Zheng et al. [108] (BNF), Zhang et al. [101] (GNF), Zhao et al. [104] (GGNF), Lu et al. [46], He et al. [26] ($L_0$), Li et al. [37] (NLLR), Liu et al. [42] (HOF), Yadav et al. [90] (ROFI), Pan et al. [54] (HLO), Zhang et al. [100] (SDF), Wang et al. [78] (CNR), Zhao et al. [106] (NormalNet), Wei et al. [82] (PCF), Wang et al. [73] (DGRMD), Li et al. [36] (DNF-Net), Li et al. [38] (NormalF-Net), Zhang et al. [103] (GeoBi-CNN), Hattori et al. [25] (DDMP), and Shen et al. [60] (GCN-Denoiser). Note that for some testing models, we only show limited denoised results, since we cannot collect all the denoised results of these methods mentioned above. We will execute all of these methods and upload the results to our GitHub repository for further reference and analysis.

All of the compared denoising results and the corresponding quantitative statistics will be included in our benchmark and released in our GitHub repository. Future researchers can directly download them for a fast and fair comparison. We sincerely hope that other researchers could upload their corresponding results to this website, to further facilitate subsequent researchers.
8.1 Abilities for Different Challenges

The main abilities of all denoising methods are concluded as the following four aspects:

— **A1**: Does it possess the capability of shallow/fine feature preservation? Note that for effectively removing noise, several existing methods may over-smooth the input surface, which causes the shallow/fine feature to disappear.

— **A2**: Does it possess the capability of avoiding feature over-sharpening? Note that for preserving sharp features, several existing methods may introduce a common extra artifact, namely feature over-sharpening.

— **A3**: Does it possess the capability of avoiding volume shrinkage?

— **A4**: Does it possess the capability of avoiding mesh quality degradation (triangle flipping, intersection, and overlapping)?

From these above four aspects, Table 2 in the supplemental material reports an overall comparison of different kinds of methods, including the optimization-based approach (I), Filter-based one-stage approach (II), Filter-based two-stage approach (III), Filter-based multi-stage approach (IV), and Data-driven-based approach (V). We summarize these capabilities through experimental observation and theoretical analysis. To avoid ambiguity in the evaluation, the symbol of $\times$ represents that a certain method has a poor performance in this aspect, or even does not have this capacity, while the symbol of $\checkmark$ on the contrary. The most important characteristic of each method is also summarized in the last column of this table. From this table, we can conclude that: (1) Filter-based two/multi-stage methods are able to preserve features well since they often first compute a reliable mesh normal field; (2) Optimized-based methods do well in removing noise, but with the cost of smoothing fine details and high time expense; (3) the recent popular learning-based approaches achieve the most impressive performance, based on their capability in all the four aspects.

8.2 Visual Comparison

The subjective comparison of five single models and three scenes with multiple objects is visualized in Figures 1–8 in the supplemental material. The testing models contain one CAD model with sharp features, one model with rich details, one model containing both sharp and shallow features, two raw scans acquired by Kinect from [78], and three scenes scanned by a portable laser scanner. Each scene contains several CAD models with different levels of sharp features. We also add extra Gaussian noise to the latter two scenes.

Taking the Block model as an example (see Figure 1 in the supplemental material), we find that BMF [21], BNF [108], NLLR [37], and HOF [42] achieve less pleasing results in some irregularly-sampled regions, due to lack of the mechanism for handling the problem of the sampling irregularity.

For the model with rich details (see Figure 2 in the supplemental material), we can observe that both $L_0$ [26] and SDF [100] over-smooth small-scale geometric features (the hair region), since $L_0$ pursues the feature sparsity in the whole surface, and SDF is designed especially for the mesh feature removal.

As observed in Figure 3 in the supplemental material, the shallow features can be recovered by most filter-based methods and all the learning-based approaches. The result produced by DGRMD [73] is more satisfactory than that by CNR [78], because there is an elaborately designed mechanism in DGRMD that is designed to recover lost geometric features when smoothing the input surface.

It is rather interesting to find that, from Figures 4 and 5 in the supplemental material, both the results produced by filter-based methods and those by learning-based methods are less satisfactory.
One main reason is that the inputs acquired by Kinect are corrupted by heavy stepping noise, whose structure is similar to certain-scale sharp features. Therefore, these methods inspired by the bilateral filter may retain these fake features. For the learning-based methods, like CNR [78] and DGRMD [73], their input descriptor is designed by the bilateral filter, so their denoising performance is significantly limited by the effect of the bilateral filter. NormalNet [106] mimics the process of GNF [101], which still is a variant of bilateral filter. We can observe that NormalNet can produce a pleasing result better than GNF, but with residual noise. We also observe $L_0$ [26] cannot handle this kind of noise because it also regards certain noise as geometric features. Among all of the denoising results in Figure 5 in the supplemental material, it is obvious that the surfaces of the results produced by HLO [54] and SDF [100] are very smooth, due to the strong filtering capability of the two methods. However, we can find the expense of strong filtering is the surface deviation, as shown in Table 3 in the supplemental material (see from the metric value of $d$ of the Cone model).

We often measure a complex scene with multiple objects. To evaluate existing denoising methods on this type of data, we scan three scenes, each of which contains several CAD-like objects. We briefly denote the three scenes as Scene-1, Scene-2, and Scene-3. The detailed results are shown in the supplemental material. In the case of Scene-1, the noise level is slight. We observe that most existing methods can produce satisfactory results. Especially, SDF [100] and $L_0$ [26] generate the flattest surfaces while keeping sharp features well preserved. For the latter two scenes (Scene-2 and Scene-3), we add extra Gaussian noise. Generally, GNF [101] produces better results than other filter-based methods, like BMF [21], BNF [108], and ROFI [90]. SDF [100] preserves large sharp features rather well but may over-smooth shallow geometric features. Among the learning-based approaches, the denoising result of NormalF-Net [38] still contains slight noise, while the latter two are more pleasing. We think the slight noise is generated, due to the fact that it is hard to keep the searched local patches sufficiently similar.

### 8.3 Quantitative Comparison

Apart from the subjective comparison, quantitative statistics about denoising errors are also demonstrated. To evaluate the quality of the denoised surface, we introduce three classical metrics:

1. The average angular difference $\theta$ is a normal based error measurement. It is the mean angular difference of ground-truth facet normals and facet normals from the denoised mesh. This metric measures the smoothness of a local surface, therefore it is widely used to support the numerical evaluation of a mesh denoising algorithm. A lower $\theta$ indicates a better result.
2. The root mean square of the distance between the denoised mesh and the ground-truth mesh (denoted as $d$). This metric is a vertex-based error measurement. It represents the imperfection of the fit of the estimator to the data.
3. The average Hausdorff distance between the denoised mesh and the known ground-truth mesh (denoted as $hd$). It is normalized by the diagonal length of the bounding box of the mesh. It represents the distance between two subsets of a metric space, which measures the geometric consistency of the denoised mesh to the ground-truth mesh.

Note that we do not use the metric $hd$, since some scanned ground-truth models often contain extra parts, which will bring error to the Hausdorff distance. We compare the objective performance on all of the testing models or scenes. The quantitative results of $\theta$ and $d$ are shown in Table 3 in the supplemental material. The best results are in bold. As shown in this table, the results produced by the learning-based techniques generally are more pleasing than traditional geometric methods. It is worth noting that the results yielded by PCF [82] have lower errors than many other methods. This is because PCF [82] pre-filters the mesh facet normal field first, via a non-local
low-rank recovery scheme, and then the generated reliable normal field is fed to the GNF as accurate guidance.

It is important to note that we do not report detailed running time statistics in this work, as many results are directly provided by the authors. Instead, we only introduce a simple time complexity for traditional denoising approaches and floating-point operations per second (FLOPs) for learning-based methods. Please be aware that the actual time complexity may vary depending on the specific implementation, mesh topology, and the number of stages or iterations. Note also that several methods do not provide well-trained models, so we are unable to provide their FLOPs.

8.4 Differences with Point Cloud Denoising
Mesh denoising and point cloud denoising are two related but distinct processes used to improve the quality of 3D data. Both processes aim to remove noise and artifacts introduced during data acquisition or processing. While point cloud denoising focuses on removing noise from a set of unordered points in 3D space, mesh denoising specifically targets noise removal in 3D meshes, which have an additional layer of connectivity information in the form of faces and edges. The necessity of mesh denoising as opposed to just point cloud denoising arises from several factors:

1. Connectivity and topology information: Meshes provide a structured representation of 3D objects, with vertices, edges, and faces defining the connectivity and topology of the object. This additional information can be exploited to develop more effective and feature-preserving denoising algorithms, taking into account the relationships between neighboring vertices and faces.

2. Applications: Many applications, such as CAD, virtual reality, video games, and medical imaging, rely on mesh representations for efficient rendering, simulation, and analysis. In these cases, mesh denoising is necessary to ensure that the mesh data is clean and suitable for further processing and usage.

3. Mesh-specific artifacts: During the process of converting a point cloud to a mesh, certain mesh-specific artifacts can be introduced, such as small isolated components, poorly connected vertices, or degenerate faces. Mesh denoising techniques can be designed to handle these artifacts more effectively, ensuring a clean and noise-free mesh representation.

While mesh denoising is a necessary and important process, it should be noted that point cloud denoising also has its merits and is applicable in different contexts [110]. In some cases, it might be preferable to perform denoising at the point cloud level, especially when dealing with raw sensor data or when a mesh representation is not yet available or needed. The choice between mesh denoising and point cloud denoising ultimately depends on the specific requirements and constraints of the application and data at hand. For more comprehensive details about point cloud denoising, I recommend referring to the recent survey work [110].

9 CONCLUSION
The field of mesh denoising has evolved from methods that design various geometric priors to approaches that seek general and automatic solutions. Our survey offers insights into this diverse range of methods, emphasizing the strengths and limitations currently present in the field. A brief discussion of future research directions is presented below:

1. In practice, we often need to scan large objects, resulting in real-scanned models with extremely large data sizes. In our testing, many existing methods either cannot process such meshes or may require significant time. This is particularly true when the input is corrupted by heavy noise, necessitating more iterations or neighborhood scales.
While deep learning techniques have proven useful in the image domain, their extension to 3D mesh processing still requires improvement. Considering the complexity of 3D structures, it is challenging to directly apply mature 2D learning network architectures to mesh surfaces. The recently popular graph-learning technology may be a suitable network for mesh processing since the mesh surface can be regarded as a graph structure. Besides, as we discovered, very few learning methods exist for mesh denoising, while it is a prominent topic for point cloud processing [11, 14, 57, 58, 96].

Lastly, there is currently a limited number of real-scanned mesh datasets containing both ground-truth and noisy parts available for learning-based mesh denoising. This highlights the need for collective efforts to create a dataset that includes various types of models. In addition, investigating how to harness the strengths of abundant synthetic data and limited real-scanned data to improve denoising performance presents an interesting research direction. Furthermore, researchers can develop more robust and generalizable denoising algorithms that better handle real-world scenarios and enhance overall performance, drawing from recent advancements in popular domain adaptation techniques.

REFERENCES

[1] Hassan Afzal, Djamila Aouada, Bruno Mirbach, and Björn Ottersten. 2018. Full 3D reconstruction of non-rigidly deforming objects. ACM Transactions on Multimedia Computing, Communications, and Applications 14, 1s (2018), 1–23.
[2] Matthieu Armando, Jean-Sébastien Franco, and Edmond Boyer. 2020. Mesh denoising with facet graph convolutions. IEEE Transactions on Visualization and Computer Graphics 28, 8 (2020), 2999–3012.
[3] Gerasimos Arvanitis, Aris S. Lalos, Konstantinos Moustakas, and Nikos Fakotakis. 2019. Feature preserving mesh denoising based on graph spectral processing. IEEE Transactions on Visualization and Computer Graphics 25, 3 (2019), 1513–1527.
[4] Haim Avron, Andrei Sharf, Chen Greif, and Daniel Cohen-Or. 2010. L1-sparse reconstruction of sharp point set surfaces. ACM Transactions on Graphics 29, 5 (2010), 135.
[5] Kanchan Bahirat, Chengyuan Lai, Ryan P. McMahan, and Balakrishnan Prabhakaran. 2018. Designing and evaluating a mesh simplification algorithm for virtual reality. ACM Transactions on Multimedia Computing, Communications, and Applications 14, 3s (2018), 1–26.
[6] Chandrakant L. Bajaj and Guoliang Xu. 2003. Anisotropic diffusion of surfaces and functions on surfaces. ACM Transactions on Graphics 22, 1 (2003), 4–32.
[7] Zhe Bian and Ruofeng Tong. 2011. Feature-preserving mesh denoising based on vertices classification. ACM Transactions on Graphics 22, 1 (2003), 4–32.
[8] Jan Botsch, Hardik Jain, and Olaf Hellwich. 2022. IMD-Net: A deep learning-based icosahedral mesh denoising network. IEEE Access 10 (2022), 38635–38649.
[9] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. 2010. Polygon Mesh Processing. AK Peters/CRC Press.
[10] Thomas Bulow. 2004. Spherical diffusion for 3D surface smoothing. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 12 (2004), 1650–1654.
[11] Pedro Hermosilla Casajus, Tobias Ritschel, and Timo Ropinski. 2019. Total denoising: Unsupervised learning of 3D point cloud cleaning. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. 52–60.
[12] Marco Centin and Alberto Signoroni. 2018. Mesh denoising with (geo)metric fidelity. IEEE Transactions on Visualization and Computer Graphics 24, 8 (2018), 2380–2396.
[13] Honghua Chen, Jin Huang, Oussama Remil, Haoran Xie, Jing Qin, Yanwen Guo, Mingqiang Wei, and Jun Wang. 2019. Structure-guided shape-preserving mesh texture smoothing via joint low-rank matrix recovery. Computer-Aided Design 115 (2019), 122–134.
[14] Honghua Chen, Zeyong Wei, Xianzhi Li, Yabin Xu, Mingqiang Wei, and Jun Wang. 2022. RePCD-Net: Feature-aware recurrent point cloud denoising network. International Journal of Computer Vision 130, 3 (2022), 615–629.
[15] Xuan Cheng, Ming Zeng, and Xinguo Liu. 2014. Feature-preserving filtering with L0 gradient minimization. Computers & Graphics 38 (2014), 150–157.
[16] Ulrich Clarenz, Udo Diewald, and Martin Rumpf. 2000. Anisotropic geometric diffusion in surface processing. In Proceedings of the IEEE Visualization 2000 Conference. 397–405.
Geometric and Learning-Based Mesh Denoising: A Comprehensive Survey

17. Chaofan Dai, Wei Pan, and Xuequan Lu. 2020. Segmentation based mesh denoising. arXiv:2008.01358. Retrieved from https://arxiv.org/abs/2008.01358.

18. Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. Citeseer, 317–324.

19. Elmar Eisemann and Frédéric Durand. 2004. Flash photography enhancement via intrinsic relighting. ACM Transactions on Graphics 23, 3 (2004), 673–678.

20. Hanqi Fan, Yizhou Yu, and Qunsheng Peng. 2009. Robust feature-preserving mesh denoising based on consistent subneighborhoods. IEEE Transactions on Visualization and Computer Graphics 16, 2 (2009), 312–324.

21. Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. 2003. Bilateral mesh denoising. ACM Transactions on Graphics 22, 3 (2003), 950–953.

22. Zhanheng Gao, Zeyun Yu, and Michael J. Holst. 2013. Feature-preserving surface mesh smoothing via suboptimal Delaunay triangulation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2862–2869.

23. Mingqianguo, Zhenzhong Song, Chengde Han, Saishang Zhong, Ruina Lv, and Zheng Liu. 2021. Mesh denoising via adaptive consistent neighborhood. Sensors 21, 2 (2021), 412.

24. Shota Hattori, Tatsuya Yatagawa, Yutaka Ohtake, and Hiromasa Suzuki. 2022. Learning self-prior for mesh denoising using dual graph convolutional networks. In Proceedings of the 17th European Conference on Computer Vision. 363–379.

25. Lei He and Scott Schaefer. 2013. Mesh denoising via L0 minimization. ACM Transactions on Graphics 32, 4 (2013), 64.

26. Klaus Hildebrandt and Konrad Polthier. 2004. Anisotropic filtering of non-linear surface features. Computer Graphics Forum 23, 3 (2004), 391–400.

27. Hui Huang and Uri M. Ascher. 2008. Surface mesh smoothing, regularization, and feature detection. SIAM Journal on Scientific Computing 31, 1 (2008), 74–93.

28. Shahram Izadi, David Kim, Ottmar Hilliges, David Molyneaux, Richard A. Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew J. Davison, and Andrew W. Fitzgibbon. 2011. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. 559–568.

29. Zhiqi Li, Yiningkui Zhang, Yidan Feng, Xingyu Xie, Qiong Wang, Mingqiang Wei, and Pheng-Ann Heng. 2020. NormalF-Net: Normal filtering neural network for feature-preserving mesh denoising. Computer-Aided Design 127 (2020), 102861.

30. Bingxuan Guo, Xiaobingwang, Ningxia Gao, and Wei Qiu. 2021. 3D mesh pre-processing method based on feature point classification and anisotropic vertex denoising considering scene structure characteristics. Remote Sensing 13, 11 (2021), 2145.
[42] Zheng Liu, Rongjie Lai, Huayuan Zhang, and Chunlin Wu. 2019. Triangulated surface denoising using high order regularization with dynamic weights. SIAM Journal on Scientific Computing 41, 1 (2019), B1–B26.

[43] Zheng Liu, Yanlei Li, Weina Wang, Ligang Liu, and Renjie Chen. 2021. Mesh total generalized variation for denoising. IEEE Transactions on Visualization and Computer Graphics 28, 12 (2021), 4418–4433.

[44] Zheng Liu, Saishang Zhong, Zhong Xie, and Weina Wang. 2019. A novel anisotropic second order regularization for mesh denoising. Computer Aided Geometric Design 71 (2019), 190–201.

[45] Xuequan Lu, Wenzhi Chen, and Scott Schaefer. 2017. Robust mesh denoising via vertex pre-filtering and L1-median normal filtering. Computer Aided Geometric Design 54 (2017), 49–60.

[46] Xuequan Lu, Zhigang Deng, and Wenzhi Chen. 2015. A robust scheme for feature-preserving mesh denoising. IEEE Transactions on Visualization and Computer Graphics 22, 3 (2015), 1181–1194.

[47] Xuequan Lu, Xiaohong Liu, Zhigang Deng, and Wenzhi Chen. 2017. An efficient approach for feature-preserving mesh denoising. Optics and Lasers in Engineering 90 (2017), 186–195.

[48] Xuequan Lu, Scott Schaefer, Jun Luo, Lizhuang Ma, and Ying He. 2020. Low rank matrix approximation for 3D geometry filtering. IEEE Transactions on Visualization and Computer Graphics 28, 4 (2020), 1835–1847.

[49] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. 2005. Efficiently combining positions and normals for precise 3D geometry. ACM Transactions on Graphics 24, 3 (2005), 536–543.

[50] Stavros Nousias, Gerasimos Arvanitis, Aris S. LaLos, and Konstantinos Moustakas. 2020. Fast mesh denoising with data driven normal filtering using deep variational autoencoders. IEEE Transactions on Industrial Informatics 17, 2 (2020), 980–990.

[51] Yutaka Ohtake, Alexander G. Belyaev, and Ilia A. Bogaevski. 2000. Polyhedral surface smoothing with simultaneous mesh regularization. In Proceedings of the Geometric Modeling and Processing 2000. Theory and Applications. IEEE, 229–237.

[52] Yutaka Ohtake, Alexander G. Belyaev, and Ilia A. Bogaevski. 2001. Mesh regularization and adaptive smoothing. Computer-Aided Design 33, 11 (2001), 789–800.

[53] Ahmed Foud El Ouafdi, Djemel Ziou, and H. Krim. 2008. A smart stochastic approach for manifold smoothing. Computer Graphics Forum 27, 5 (2008), 1357–1364.

[54] Wei Pan, Xuequan Lu, Yuanhao Gong, Wenming Tang, Jun Liu, Ying He, and Guoping Qiu. 2020. HLO: Half-kernel Laplacian operator for surface smoothing. Computer-Aided Design 121, C (2020), 102807.

[55] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. 2004. Digital photography with flash and no-flash image pairs. ACM Transactions on Graphics 23, 3 (2004), 664–672.

[56] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep learning on point sets for 3D data. IEEE Transactions on Computer Vision and Pattern Recognition 28, 1 (2018), 1–14.

[57] Riccardo Roveri, A. Cengiz Öztireli, Ioana Pandele, and Markus H. Gross. 2018. PointProNets: Consolidation of point clouds with convolutional neural networks. Computer Graphics Forum 37, 2 (2018), 87–99.

[58] Yuzhong Shen and Kenneth E. Barner. 2004. Fuzzy vector median-based surface smoothing. IEEE Transactions on Visualization and Computer Graphics 10, 3 (2004), 252–265.

[59] Yuefan Shen, Hongbo Fu, Zhongshuo Du, Xiang Chen, Evgeny Burnaev, Denis Zorin, Kun Zhou, and Youyi Zheng. 2022. GCN-denoiser: Mesh denoising with graph convolutional networks. ACM Transactions on Graphics 41, 1 (2022), 1–14.

[60] Justin Solomon, Keenan Crane, Adrian Butscher, and Chris Wojtan. 2014. A general framework for bilateral and mean shift filtering. arXiv:1405.4734. Retrieved from https://arxiv.org/abs/1405.4734

[61] Zhixun Su, Hui Wang, and Junjie Cao. 2009. Mesh denoising based on differential coordinates. In Proceedings of the IEEE International Conference on Shape Modeling and Applications. 1–6.

[62] Xianfang Sun, Paul L. Rosin, Ralph Martin, and Frank Langbein. 2007. Fast and effective feature-preserving mesh denoising. IEEE Transactions on Visualization and Computer Graphics 13, 5 (2007), 925–938.

[63] Yuefan Shen, Scott Schaefer, and Wenping Wang. 2015. Denoising point sets via vertex pre-filtering and L1-median normal filtering. Computer Aided Geometric Design 35–36 (2015), 2–15.

[64] Wenming Tang, Yuanhao Gong, and Guoping Qiu. 2023. Feature preserving 3D mesh denoising with a dense local graph neural network. Computer Vision and Image Understanding 233 (2023), 103710.

[65] Tolga Tasdizen, Ross Whitaker, Paul Burchard, and Stanley Osher. 2002. Geometric surface smoothing via anisotropic diffusion of normals. In Proceedings of the Conference on Visualization’02. IEEE Computer Society, 125–132.

[66] Gabriel Taubin. 1995. A signal processing approach to fair surface design. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. ACM, 351–358.
[68] Gabriel Taubin. 2001. Linear Anisotropic Mesh Filtering. Research Report RC2213 IBM 1, 4.
[69] Carlo Tomasi and Roberto Manduchi. 1998. Bilateral filtering for gray and color images. In Proceedings of the International Conference on Computer Vision. 839–846.
[70] Shoichi Tsuchie and Masatake Higashi. 2012. Surface mesh denoising with normal tensor framework. Graphical Models 74, 4 (2012), 130–139.
[71] Jörg Vollmer, Robert Mencl, and Heinrich Mueller. 1999. Improved Laplacian smoothing of noisy surface meshes. Computer Graphics Forum 18, 3 (1999), 131–138.
[72] Charlie C. L. Wang. 2006. Bilateral recovery of sharp edges on feature-insensitive sampled meshes. IEEE Transactions on Visualization and Computer Graphics 12, 4 (2006), 629–639.
[73] Jun Wang, Jin Huang, Fu Lee Wang, Mingqiang Wei, Haoran Xie, and Jing Qin. 2019. Data-driven geometry-recovering mesh denoising. Computer-Aided Design 114 (2019), 133–142.
[74] Jun Wang and Zeyun Yu. 2011. Quality mesh smoothing via local surface fitting and optimum projection. Graphical Models 73, 4 (2011), 127–139.
[75] Jun Wang and Zeyun Yu. 2012. Feature-preserving mesh denoising via anisotropic surface fitting. Journal of Computer Science and Technology 27, 1 (2012), 163–173.
[76] Jun Wang, Xi Zhang, and Zeyun Yu. 2012. A cascaded approach for feature-preserving surface mesh denoising. Computer-Aided Design 44, 7 (2012), 597–610.
[77] Peng-Shuai Wang, Xiao-Ming Fu, Yang Liu, Xin Tong, Shi-Lin Liu, and Baining Guo. 2015. Rolling guidance normal filter for geometric processing. ACM Transactions on Graphics 34, 6 (2015), 173.
[78] Peng-Shuai Wang, Yang Liu, and Xin Tong. 2016. Mesh denoising via cascaded normal regression. ACM Transactions on Graphics 35, 6 (2016), 232:1–232:12.
[79] Ruimin Wang, Zhouwang Yang, Ligang Liu, Jiansong Deng, and Falai Chen. 2014. Decoupling noise and features via weighted L1-analysis compressed sensing. ACM Transactions on Graphics 33, 2 (2014), 18:1–18:12.
[80] Renjie Wang, Wenbo Zhao, Shaohui Liu, Debin Zhao, and Chun Liu. 2017. Feature-preserving mesh denoising based on guided normal filtering. In Proceedings of the 18th Pacific-Rim Conference on Multimedia. 920–927.
[81] M. Wei, X. Guo, J. Huang, H. Xie, H. Zong, R. Kwan, F. L. Wang, and Jing Qin. 2019. Mesh defiltering via cascaded geometry recovery. Computer Graphics Forum 38, 7 (2019), 591–605.
[82] Mingqiang Wei, Jin Huang, Xingyu Xie, Ligang Liu, Jun Wang, and Jing Qin. 2019. Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery. IEEE Transactions on Visualization and Computer Graphics 25, 10 (2019), 2910–2926.
[83] Mingqiang Wei, Luming Liang, Wai-Man Pang, Jun Wang, Weishi Li, and Huisi Wu. 2017. Tensor voting guided mesh denoising. IEEE Transactions on Automation Science and Engineering 14, 2 (2017), 931–945.
[84] Mingqiang Wei, Wuyao Shen, Jing Qin, Jianhuang Wu, Tien-Tsin Wong, and Pheng-Ann Heng. 2013. Feature-preserving optimization for noisy mesh using joint bilateral filter and constrained Laplacian smoothing. Optics and Lasers in Engineering 51, 11 (2013), 1223–1234.
[85] Mingqiang Wei, Jun Wang, Xianglin Guo, Huisi Wu, Haoran Xie, Fu Lee Wang, and Jing Qin. 2018. Learning-based 3D surface optimization from medical image reconstruction. Optics and Lasers in Engineering 103 (2018), 110–118.
[86] Mingqiang Wei, Jinze Yu, Wai-Man Pang, Jun Wang, Jing Qin, Ligang Liu, and Pheng-Ann Heng. 2014. Bi-normal filtering for mesh denoising. IEEE Transactions on Visualization and Computer Graphics 21, 1 (2014), 43–55.
[87] Xiaojun Wang, Jiannin Zheng, Yiyu Cai, and Chi-Wing Fu. 2015. Mesh denoising using extended ROF model with L1 fidelity. Computer Graphics Forum 34, 7 (2015), 35–45.
[88] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. 2011. Image smoothing via L0 gradient minimization. ACM Transactions on Graphics 30, 6 (2011), 174.
[89] Sunil Kumar Yadav, Ulrich Reitebuch, and Konrad Polthier. 2018. Mesh denoising based on normal voting tensor and binary optimization. IEEE Transactions on Visualization and Computer Graphics 24, 8 (2018), 2366–2379.
[90] Sunil Kumar Yadav, Ulrich Reitebuch, and Konrad Polthier. 2019. Robust and high fidelity mesh denoising. IEEE Transactions on Visualization and Computer Graphics 25, 6 (2019), 2304–2310.
[91] Hirokazu Yagou, Yutaka Ohtake, and Alexander Belyaev. 2002. Mesh smoothing via mean and median filtering applied to face normals. In Proceedings of the Geometric Modeling and Processing. Theory and Applications. IEEE, 124–131.
[92] Hirokazu Yagou, Yutaka Ohtake, and Alexander G. Belyaev. 2003. Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding. In Proceedings of the 2003 Computer Graphics International Conference. 28–33.
[93] Shin Yoshizawa, Alexander Belyaev, and H.-P. Seidel. 2006. Smoothing by example: Mesh denoising by averaging with similarity-based weights. In Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006. IEEE, 9–9.
[94] Hang Yu, Chilam Cheang, Yanwei Fu, and Xiangyang Xue. 2022. Multi-view shape generation for 3D human-like body. ACM Transactions on Multimedia Computing, Communications, and Applications 19, 1, Article No. 11 (2022), 1–22.

ACM Trans. Multimedia Comput. Commun. Appl., Vol. 20, No. 3, Article 85. Pub. date: November 2023.
[95] Jinze Yu, Mingqiang Wei, Jing Qin, Jianhuang Wu, and Pheng-Ann Heng. 2014. Feature-preserving mesh denoising via normal guided quadric error metrics. Optics and Lasers in Engineering 62 (2014), 57–68.

[96] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. 2018. PU-Net: Point cloud upsampling network. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition. 2790–2799.

[97] Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. 2004. Mesh editing with poisson-based gradient field manipulation. ACM Transactions on Graphics 23, 3 (2004), 644–651.

[98] Hao Zhang, Oliver van Kaick, and Ramsay Dyer. 2010. Spectral mesh processing. Comput. Graph. Forum 29, 6 (2010), 1865–1894.

[99] Huayan Zhang, Chunlin Wu, Juyong Zhang, and Jiansong Deng. 2015. Variational mesh denoising using total variation and piecewise constant function space. IEEE IEEE Transactions on Visualization and Computer Graphics 21, 7 (2015), 873–886.

[100] Juyong Zhang, Bailin Deng, Yang Hong, Yue Peng, Wenjie Qin, and Ligang Liu. 2019. Static/dynamic filtering for mesh geometry. IEEE Transactions on Visualization and Computer Graphics 25, 4 (2019), 1774–1787.

[101] Wangyu Zhang, Bailin Deng, Juyong Zhang, Sofien Bouaziz, and Ligang Liu. 2015. Guided mesh normal filtering. Computer Graphics Forum 34, 7 (2015), 23–34.

[102] Ying Zhang and A. Ben Hamza. 2007. Vertex-based diffusion for 3-D mesh denoising. IEEE Transactions on Image Processing 16, 4 (2007), 1036–1045.

[103] Yingkui Zhang, Guibao Shen, Qiong Wang, Yinling Qian, Mingqiang Wei, and Jing Qin. 2022. GeoBi-GNN: Geometry-aware bi-domain mesh denoising via graph neural networks. Computer-Aided Design 144 (2022), 103154.

[104] Wenbo Zhao, Xianning Liu, Shiqi Wang, Xiaopeng Fan, and Debin Zhao. 2019. Graph-based feature-preserving mesh normal filtering. IEEE Transactions on Visualization and Computer Graphics 27, 3 (2019), 1937–1952.

[105] Wenbo Zhao, Xianning Liu, Shiqi Wang, and Debin Zhao. 2017. Multi-scale similarity enhanced guided normal filtering. In Proceedings of the 18th Pacific-Rim Conference on Multimedia. 645–653.

[106] Wenbo Zhao, Xianning Liu, Yongsen Zhao, Xiaopeng Fan, and Debin Zhao. 2019. NormalNet: Learning based guided normal filtering for mesh denoising. arXiv:1903.04015. Retrieved from https://arxiv.org/abs/1903.04015

[107] Yong Zhao, Hong Qin, Xueying Zeng, Junli Xu, and Junyu Dong. 2018. Robust and effective mesh denoising using L0 sparse regularization. Computer-Aided Design 101 (2018), 82–97.

[108] Youyi Zheng, Hongbo Fu, Oscar Kin-Chung Au, and Chiew-Lan Tai. 2011. Bilateral normal filtering for mesh denoising. IEEE Transactions on Visualization and Computer Graphics 17, 10 (2011), 1521–1530.

[109] Saishang Zhong, Zhong Xie, Weina Wang, Zheng Liu, and Ligang Liu. 2018. Mesh denoising via total variation and weighted Laplacian regularizations. Journal of Visualization and Computer Animation 29, 3–4 (2018).

[110] Lang Zhou, Guoxing Sun, Yong Li, Weiqing Li, and Zhongyang Su. 2022. Point cloud denoising review: From classical to deep learning-based approaches. Graphical Models 121 (2022), 101140.

[111] Dingkun Zhu, Yingkui Zhang, Zhiqi Li, Weiming Wang, Haoran Xie, Mingqiang Wei, Gary Cheng, and FuLee Wang. 2021. Cascaded normal filtering neural network for geometry-aware mesh denoising of measurement surfaces. IEEE Transactions on Instrumentation and Measurement 70 (2021), 1–13.

[112] Lei Zhu, Mingqiang Wei, Jinze Yu, Weiming Wang, Jing Qin, and Pheng-Ann Heng. 2013. Coarse-to-fine normal filtering for feature-preserving mesh denoising based on isotropic subneighborhoods. Computer Graphics Forum 32, 7 (2013), 371–380.

Received 26 December 2022; revised 29 July 2023; accepted 4 September 2023