Surface Smoothing based on Discrete Orthogonal Polynomial

Lei Lu[1], Ning Li[1,2], Wei Pan[2], Wenming Tang[3]

[1]Henan University of Technology, lulei@haut.edu.cn
[1]Henan University of Technology, ln_1013@163.com
[2]OPT Machine Vision Tech Co.,Ltd., vpan@foxmail.com
[3]Shenzhen Institute of Information Technology, tangwenming@sziit.edu.cn

Corresponding author: Wei Pan, vpan@foxmail.com

Abstract. Significant progress has been made in mesh denoising techniques for recovering high-quality models from three-dimensional meshes disturbed by noise in recent years. However, even state-of-the-art methods still struggle to handle various noisy 3D models stably. The primary challenge of robust mesh denoising is to remove noise while preserving geometric features as much as possible. This paper proposes a novel robust and feature-preserving mesh denoising method aimed at effectively eliminating noise while maintaining the sharp features of the mesh. Initially, the normal directions are estimated using non-local similar structures. Subsequently, the normal directions are filtered using orthogonal polynomial fitting on line matrices. Finally, effective filtering and denoising of triangle meshes are achieved by updating vertex positions. Experimental results demonstrate that our method produces results comparable to or even higher quality than state-of-the-art techniques.

Keywords: Mesh Processing, Surface Denoising, Dimensional Accuracy, Process Optimization
DOI: https://doi.org/10.14733/cadaps.2025.1027-1039

1 INTRODUCTION

The triangular mesh surface model serves as a versatile representation method for three-dimensional objects, playing a significant role in various aspects of computer-aided engineering, including reverse design, rapid prototyping, 3D printing, and virtual simulation. Consequently, the reconstruction and processing of triangular mesh surface models constitute a significant research focus within the field. A large amount of 3D grid data in industry is obtained through 3D measurement, where the resulting grid model often incorporates noise introduced during scanning and digitization. This noise can significantly impede the usability of the grid model, necessitating its removal during the pre-processing stage.

Most early mesh smoothing methods could not preserve geometric features because they were isotropic (i.e., independent of surface geometry). Taubin [18] proposed a non-shrinkage two-step smoothing method

implemented through signal processing. Vollmer et al. [19]introduced a simple, fast Laplacian smoothing algorithm. However, it results in surface contraction and cannot preserve sharp features. Researchers later developed various isotropic denoising methods based on volume preservation, frequency, or differential properties [10, 8]. Due to the difficulty in preserving geometric features with isotropic methods, anisotropic denoising methods gained widespread attention. Various methods based on diffusion or differential information have been proposed, such as [13, 5] To address this issue, anisotropic methods have emerged. An early work by Hildebrandt and Polthier [5] uses mean curvature flow to preserve features while denoising mesh shapes. Subsequently, two-step methods, such as bilateral filtering techniques [4, 9] and others [17, 10, 11, 2], have been proposed to better preserve features. These methods typically involve normal smoothing and vertex updating, showing promising results for robust, feature-preserving mesh denoising. In recent years, researchers have explored classification techniques for distinguishing features during mesh denoising [3, 1, 23, 20, 22]. However, these strategies often focus on local neighborhoods and are susceptible to noise. To mitigate this, Lu et al. [11] introduced a pre-filtering technique before denoising to reduce the impact of excessive noise

Another approach in anisotropic mesh denoising focuses on a sparse perspective, where feature vertices are computed by solving linear sparse systems. For instance, He and Schaefer [4] proposed an L0-minimisation framework, which preserves mesh features but suffers from non-convex and slow minimization processes. To address this, Zhao et al. [25] introduced an alternating optimization strategy for L0-minimisation. Similarly, Lu et al. [11] presented an L1-minimisation method to preserve mesh features. Recently, Pan et al. [15] proposed a half-kernel Laplacian operator, which reduces feature damages while removing noise by constructing a half window of the local neighborhood for each vertex. Additionally, Wang et al. proposed a method using mesh segmentation to preserve features.

In this paper, we adopt a method that first estimates the normal direction using non-local similar structures [12] and then filters the normal direction using orthogonal polynomial fitting line matrices to achieve filtering and denoising effects on the triangular mesh [14]. The matrix constructed with non-local homologous similar structures has been proven to be more representative and robust. Additionally, orthogonal polynomials of different orders do not interfere with each other, reducing both time costs and computational complexity in the denoising process. Experiments have shown that our method can produce results comparable to or of higher quality than state-of-the-art methods. The specific algorithm flow chart is shown in the following Figure 1.

2 THE CONSTRUCTION OF NON-LOCAL SIMILAR STRUCTURES

To better understand the proposed non-local similar structures, we first briefly introduce local structures and local co-oriented structures, and finally explain how non-local similar structures are obtained through screening.

2.1 The Tensor of Local Structure

Given a surface mesh $M = (V, E, F)$ with N vertices, we have the set of vertices V, the set of edges E, and the set of faces F. The i-th vertex $v_i \in V$ is represented by the coordinates $v_i = (x_i, y_i, z_i)$ Each face f_i. has a local structure S_i, which consists of the r -ring structure of f_i. Due to the surface irregularities in the noisy triangular mesh model, estimating the normal direction using a single face may not be accurate enough. Therefore, we use the local structure to calculate a normal tensor, which replaces the initial normal direction. The specific definition is as follows:

$$T_{ij} = \eta(\|c_i - c_j\|)\phi(\theta_{ij})\mathbf{n}_j^T\mathbf{n}_j \tag{1}$$

Where c_i refers to the centroid of the current face f_i, and n_i refers to the normal of f_i. η and ϕ are the weights induced by spatial distances and intersection angles θ_{ij} of two neighboring normals, which are given

by [7] the specific definition is as follows:

$$\eta(x) = e^{-\left(\frac{p}{\sigma x}\right)^2} \tag{2}$$

$$\phi(\theta) = e^{-\left(\frac{1-\cos(\sigma\theta)}{1-\cos(\theta)}\right)^2} \tag{3}$$

σ_p and σ_θ are the parameters, Based on experiments, σ_p is set to twice the maximum distance between two points in the set of faces of the r-ring neighborhood of f_i. σ_θ is set to be $30°$.

For each local structure S_i, we can derive the accumulated tensor by aggregating all the induced tensor votes (4), $i, j \in S_i$. This final tensor encodes the local structure, which provides a reliable, representative orientation.

$$T_i = \sum_{j \in S_i} T_{ij} \tag{4}$$

When obtaining the local principal components, we use the matrix decomposition technique to decompose the T_i(3*3) matrix. Consequently, three eigenvalues and their corresponding eigenvectors are obtained. We select the eigenvector λ_{iso} associated with the maximum eigenvalue V_{iso} as the principal direction of this local structure. This is also referred to as the local tensor.

2.2 Local Isotropic Structure

After obtaining the representative direction V_{iso} for each local structure, we use V_{iso} as a reference to filter out faces in the face set that are approximately aligned with this local direction. This approach eliminates faces with large errors in the local structure, preparing for the next step of filtering out non-locally aligned similar structures. The filtering criterion involves comparing the angle θ_{th} between the normal of each face f_i in set S_i and the direction V_{iso}. If the noise level in the model is high, the threshold angle θ_{th} is set larger; otherwise, it is set smaller.

2.3 Non-local Similar Structures

Expanding the neighborhood range, searching for the R-ring neighborhood of face f_i (with R>r), for each face f_i within the neighborhood, identifying its local co-aligned structure S_j^{iso}, comparing S_j^{iso} with S_i^{iso}, and determining the angle θ_{th} between them. For cases with high noise, the angle threshold θ_{th} is set to $40°$. while for cases with low noise, it is set to$20°$. Identifying all local co-aligned structures that meet the criteria and adding them to S_i^{nls}, ultimately obtaining the non-locally co-aligned similar structures for each face in the model.

3 ORTHOGONAL POLYNOMIAL FITTING ESTIMATION FOR NORMAL

Orthogonal polynomials are distinguished by their property of orthogonality concerning a specific weight function, implying that their inner product is zero under this weight function. Orthogonal polynomial fitting thus provides an effective approach to address the challenges associated with higher-order polynomial fitting.

3.1 The Definition of Orthogonal Polynomials

The definition of orthogonal polynomials:

A polynomial sequence $p_n(x)$ for $n = 0, 1, 2, \ldots$ with degree $[p_n(x)] = n$ is said to be orthogonal on the open interval (a, b) with respect to the weight function $w(x)$ if for each n, the following condition holds:

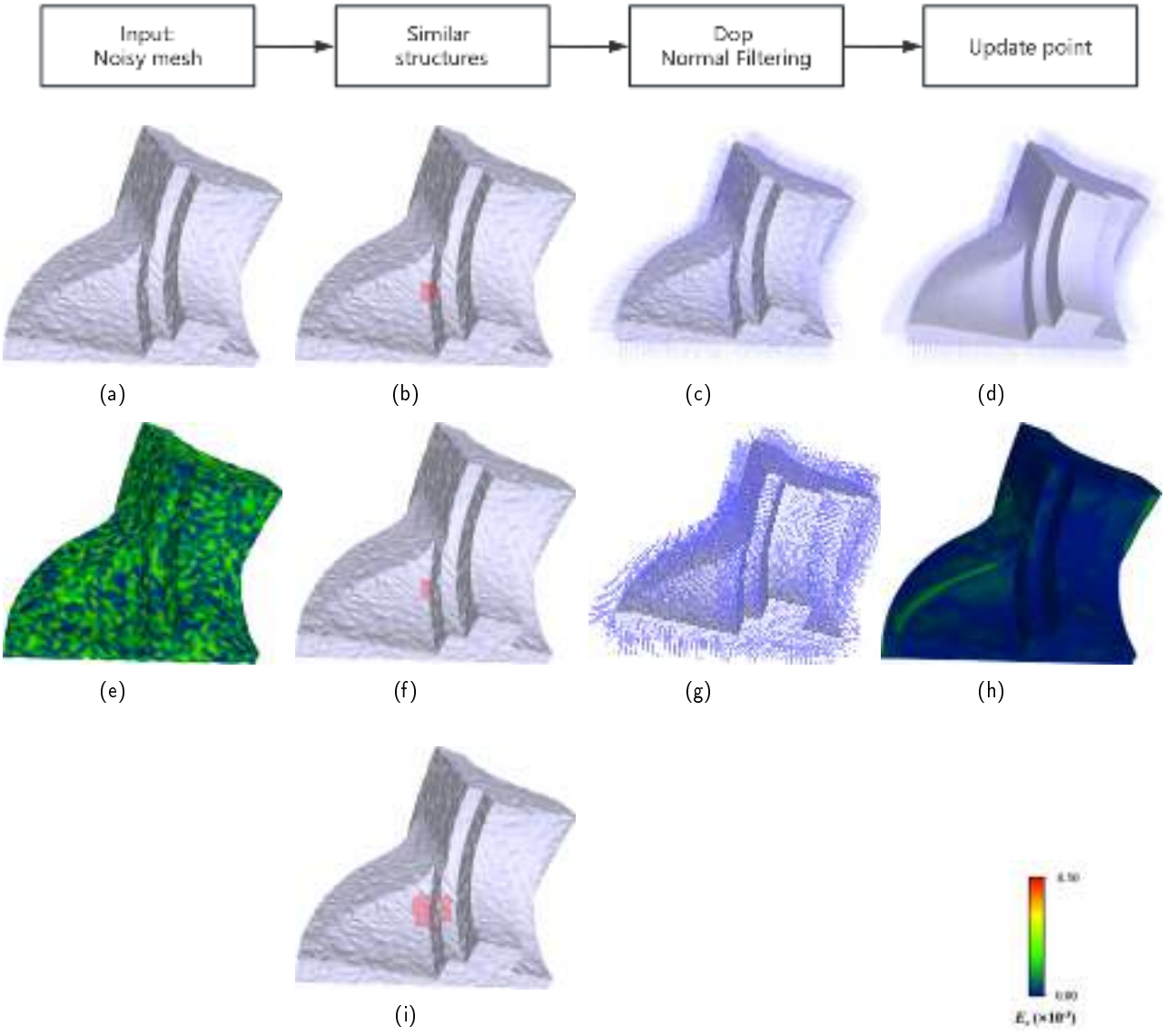

Figure 1: A general overview of the algorithm

$$\int_a^b w(x)p_m(x)p_n(x)\,dx = h_n\delta_{mn} \tag{5}$$

Where δ_{mn} is the Kronecker delta function, and h_n is a constant. Here, the weight function $w(x)$ on the interval (a,b) is assumed to be continuous and positive, such that the following integrals exist:

$$\mu_n = \int_a^b w(x)x^n\,dx, \quad n = 0,1,2,\ldots \tag{6}$$

The inner product of two polynomials $f(x)$ and $g(x)$ is defined as:

$$\langle f,g\rangle := \int_a^b w(x)f(x)g(x)\,dx \tag{7}$$

The interval (a,b) is called the orthogonal interval, which may or may not be finite.

3.2 Legendre Polynomials

Legendre polynomials are polynomials defined on the interval $(-1,1)$ that are orthogonal concerning the weight function $w(x)=1$. Legendre polynomials are a special case of Jacobi polynomials when $\alpha=\beta=0$. The expression of Legendre polynomials is

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left[(x^2-1)^n\right] \tag{8}$$

The recurrence formula for Legendre polynomials is:

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \tag{9}$$

The specific expansion calculation of Legendre polynomials is as follows. The plot of the first six orders of polynomials is shown in the following Figure 2.

$$\begin{cases} \varphi_0(x) = 1 \\ \varphi_1(x) = x - \frac{(x,\varphi_0(x))}{(\varphi_0(x),\varphi_0(x))}\varphi_0(x) = x - \frac{\int_1^1 x\,dx}{\int_{-1}^1 dx}\cdot 1 = x \\ \varphi_2(x) = x^2 - \frac{\left(x^2,\varphi_0(x)\right)}{(\varphi_0(x),\varphi_0(x))}\varphi_0(x) - \frac{\left(x^2,\varphi_1(x)\right)}{(\varphi_1(x),\varphi_1(x))}\varphi_1(x) = x^2 - \frac{1}{3} \\ \varphi_3(x) = x^3 - \frac{\left(x^3,\varphi_0(x)\right)}{(\varphi_0(x),\varphi_0(x))}\varphi_0(x) - \frac{\left(x^3,\varphi_1(x)\right)}{(\varphi_1(x),\varphi_1(x))}\varphi_1(x) - \frac{\left(x^3,\varphi_2(x)\right)}{(\varphi_2(x),\varphi_2(x))}\varphi_2(x) = x^3 - \frac{3}{5}x \end{cases} \tag{10}$$

For each non-local similar structure S_i^{nls} for the isotropic structure S_i^{iso} associated with the face f_i, we append the face normals of S_i^{nls} as rows to a matrix M. Note that the dimensions of this matrix are $r\times 3$. This matrix already has a maximal rank of 3 and is a low-rank matrix. To make the low-rank matrix approximation meaningful, we reshape the matrix M to be close to a square matrix R. first, we need to construct the orthogonal bases for the X and Y directions of matrix R. Assuming the fitting order in the X-direction is a and in the Y-direction is b, the expressions for these two orthogonal bases can be represented as:$B_x = \{\varphi_0(x),\varphi_1(x),\varphi_2(x).....\varphi_a(x)\}$,$B_y = \{\varphi_0(y),\varphi_1(y),\varphi_2(y).....\varphi_a(y)\}$.

The expression of the fitting matrix is obtained by discretizing B_x at uniform intervals within the range [-1, 1], resulting in a total of n nodes. From this discretization, the basis matrix $B_x(n\times a)$ and its transpose matrix $B_x^T(a\times n)$ are obtained. Similarly, the basis matrix in the Y direction $B_y(m\times b)$ and its transpose matrix

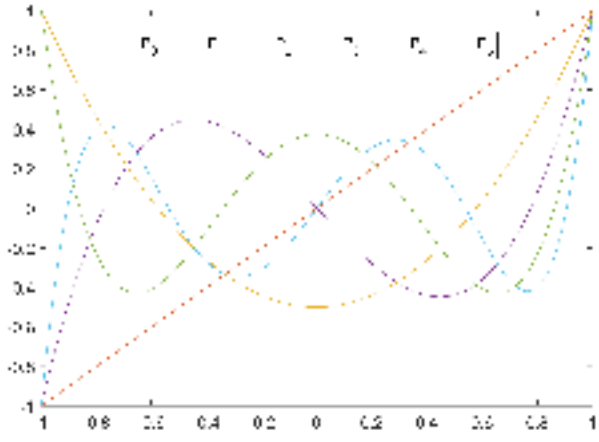

Figure 2: The first six Legendre polynomials

$B_y^T(b \times m)$ are also obtained through m uniform intervals. Here, Z represents the matrix after orthogonal polynomial filtering.

$$Z_{m\times n} = B_{y(m\times b)} * B^T_{y(m\times b)} * R_{m\times n} * B_{x(n\times a)} * B^T_{x(a\times n)} \tag{11}$$

When constructing similar structures for each face, adjacent faces form a set of similar structures $S = \{f_1, f_2, \ldots, f_n\}$, where overlapping regions may occur. In such cases, a weight is assigned to each face f_i to track its frequency of use. The final matrix is then obtained by accumulating contributions from each face, which are divided by the corresponding weight coefficient. This process results in the updated normal direction for each face.

4 POSITION UPDATE

After obtaining the updated normals corresponding to each triangle, we update the position of each vertex in the triangle according to the new normals. We calculate the new position of the point using the following formula:

$$x_i' = \mathbf{x}_i + \frac{1}{|F_V(i)|} \sum_{k\in F_V(i)} \mathbf{n}_k' \left(\mathbf{n}_k' \cdot (\mathbf{c}_k - \mathbf{x}_i)\right) \tag{12}$$

$F_V(i)$ refers to the set of torus adjacent to a triangle and refers to the center of the triangle where the current point is located.

5 EXPERIMENTAL RESULTS AND ANALYSIS

In our experiments, we tested our method on numerous mesh models corrupted by synthesized or original scan noise. Additionally, we evaluated several state-of-the-art mesh denoising methods on the same test set for comparison. In this section, The proposed method is denoted as Method I. The selected state-of-the-art mesh denoising methods include L0-minimisation (L0) [4], the Bilateral Normal Filter (BNF)[9], the Unilateral Normal Filter (UNF)[17], the Guided Normal Filter (GNF) [24]. the L1-median Filter (L1) [11], the Half-kernel Laplacian Operator (HLO) [15] and Segmentation-Driven Feature-Preserving(SNF) [21].

The summary of parameters for the methods used in our comparative experiments is presented in Table 1. For all methods, we employ the recommended parameters and carefully adjust them to achieve the desired outputs.

Our method varies in screening non-local similar structures based on the parameter $\cos\theta_{th}$ and the increment of o at each iteration according to different noise levels. For models with lower noise levels, we typically set the initial $\cos\theta_{th}$ within the range of 0.7 to 0.9, and the increment is generally set to 0.5. For models with higher noise levels, we usually set the initial value within the range of 0.5 to 0.65, and the increment is set to 0.8 to 1.0. These parameter settings are close to the optimal solutions for our method. For n_{iter} and v_{iter}, we set them to 5 and 10, respectively.

5.1 Visual Results

Synthetic models. We compare our method with the selected state-of-the-art methods on various models corrupted with synthetic noise. Following state-of-the-art mesh smoothing techniques, we generate synthetic models by adding zero-mean Gaussian noise with standard deviation σ_n to the corresponding ground truth. σ_n is proportional to the mean edge length l_e of the input mesh.Figure 3 to Figure 6 depicts the experimental results on synthetic models. Based on the data in Table 2, we can observe that our method outperforms other methods for these models, or achieves comparable results to the best-performing method. Moreover, as shown in the local details in Figure 5 and Figure 6, our method is effective in preserving local features.

Raw Scanned Models In addition to models corrupted by synthesized noise, we also tested all methods on real 3D scanned data. Figure 7 and Figure 8 show that when all methods are applied to the original scan of the angle model, although our method slightly lags behind the method of SNF, it still outperforms other methods in denoising and preserving features. Additionally, we conducted a set of denoising experiments on the same bunny scanned model with different levels of noise as Figure 9 It can be seen that for noise levels of 0.1 and 0.3, our method performs quite well. However, when the noise level is too high, reaching 0.5, our method has certain limitations. This issue is discussed in the conclusion.

5.2 Quantitative Evaluations

We compare the state-of-the-art techniques with our approach from a quantitative perspective. Specifically, we employ E_v and MSAE (mean square angular error) to respectively evaluate the positional error and normal error, as suggested by previous works [17, 9, 16]. These two metrics are calculated between the smoothing results and their corresponding ground truth.

According to [17, 6], E_v is the L^2 vertex-based mesh-to-mesh error metric, and MSAE measures the mean square angular error between the face normals of the denoised mesh and those of the ground truth.

$$E_v = \sqrt{\frac{1}{3\sum_{k\in F} A_k} \sum_{i\in V} \sum_{j\in F_V(i)} A_j dist(x_i^{'}, T)^2}, \tag{13}$$

Where A_k is the area of face k, and $dist(x_i^{'}, T)$ is the L^2 distance between the updated vertex $x_i^{'}$ and a triangle of the reference mesh T which is closest to $x_i^{'}$.

$$\text{MSAE} = \frac{\sum_{k\in F} \theta_k^2}{N_F} \tag{14}$$

Where θ_k is the angle between the k-th face normal of the denoised model and its corresponding normal in the ground-truth model, and N_F is the number of faces in the 3D shape.

Table 2 summarizes the statistical numbers of E_v and MSAE over most models for all the compared methods.

Table 1: Parameters of the state-of-the-art mesh smoothing methods.

Methods	Number of Parameters	Parameters Description
L0	3	βmax: maximum value of beta $\alpha 0$: initial value for alpha λ: weight for the L0 term in the target function
BNF	3	σ_s: Variance parameter for the spatial kernel n_{iter}: Number of iterations for normal update v_{iter}: Number of iterations for vertex update
UNF	3	T: Threshold for controlling the averaging weights n_{iter}: Number of iterations for normal update v_{iter}: Number of iterations for vertex update
GNF	5	r: Radius for finding a geometrical neighborhood σ_s: Variance parameter for the spatial kernel σ_r: Variance parameter for the range kernel n_{iter}: Number of iterations for normal update v_{iter}: Number of iterations for vertex update
L1	3	σ: variance parameter for the spatial kernel n_{iter}: Number of iterations for normal update v_{iter}: Number of iterations for vertex update
HLO	1	$iter$: number of filtering iterations
SNF	1+X	D_{thr}: edge-based segmentation threshold X:the parameters of other methods
Ours	4	$\cos\theta_{th}$: Screening non-local similar structures δd: The increment of o at each iteration n_{iter}: Number of iterations for normal update v_{iter}: Number of iterations for vertex update

Table 2: Quantitative comparisons with representative mesh smoothing methods.

Models	Methods	MSAE($\times 10^{-2}$)	$E_v(\times 10^{-3})$	Parameters
	L0	0.960	8.960	(1000,0.0033,0.01)
Cube	BNF	0.889	4.758	(0.45,80,40)
Figure 3	UNF	0.135	1.229	(0.35,20,50)
($\sigma_n = 0.2l_e$)	GNF	0.425	1.267	(2,1,0.35,20,10)
$\|V\|$:12288	L1	0.153	1.303	(80,40,45)
$\|F\|$:6146	HLO	0.153	1.303	(3)
	SNF(L1)	0.153	1.303	(80,40,45)
	OURS	**0.075**	**1.224**	(0.7,0.05,6,20)
	L0	**5.2099**	15.1554	(1000, 0.0027, 0.01)
Fandisk	BNF	12.182	8.536	(0.4,25,20)
Figure 4	UNF	11.324	10.437	(0.3,30,20)
($\sigma_n = 0.4l_e$)	GNF	10.739	6.976	(2,1,0.25,25,20)
$\|V\|$:7229	L1	20.817	8.147	(30,100,100)
$\|F\|$:14454	HLO	8.545	18.208	(3)
	SNF(UNF)	9.379	8.750	(0.3,30,20)
	OURS	10.349	**5.983**	(0.65,0.08,5,10)
	L0	2.861	202.264	(1000,0.0013,1)
Cad	BNF	2.973	207.254	(0.35,25,20)
Figure 5	UNF	3.654	239.814	(0.55,20,40)
($\sigma_n = 0.3l_e$)	GNF	3.624	252.715	(2,1,0.25,25,20)
$\|V\|$:19398	L1	3.408	186.496	(80,40,45)
$\|F\|$:38792	HLO	8.713	793.123	(5)
	SNF(BNF)	2.529	**181.763**	(0.35,25,20)
	OURS	**0.640**	218.55	(0.75,0.05,5,10)
	L0	17.043	6.138	(1000, 0.0021, 0.01)
Octaflower	BNF	5.938	2.520	(0.35,25,20)
Figure 6	UNF	9.726	3.769	(0.55,20,40)
($\sigma_n = 0.1l_e$)	GNF	8.6778	2.354	(2,1,0.25,25,20)
$\|V\|$:7919	L1	2.209	**0.968**	(20,10,30)
$\|F\|$:15834	HLO	7.371	2.152	(3)
	SNF(L1)	**2.158**	**0.968**	(20,10,30)
	OURS	3.74359	1.78147	(0.85,0.05,5,10)

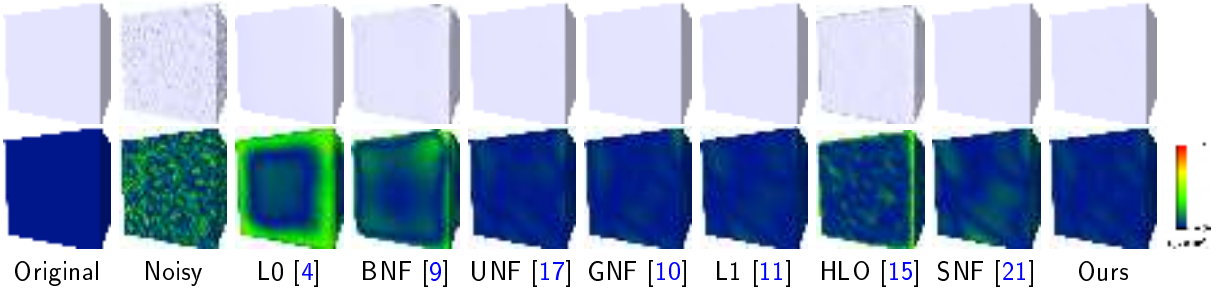

Figure 3: Coloured Ev for Cube with noise $\sigma_n = 0.2l_e$

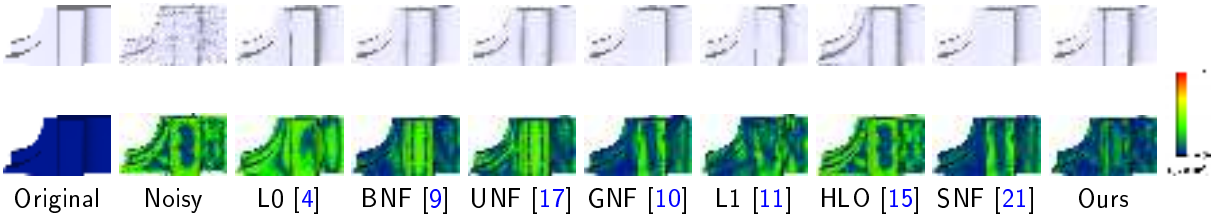

Figure 4: Coloured Ev for fandisk with noise $\sigma_n = 0.4l_e$

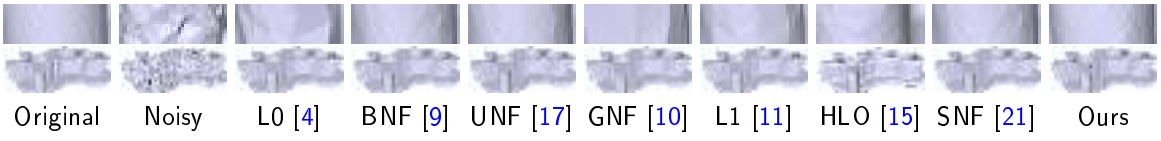

Figure 5: Visual comparison for cad with noise $\sigma_n = 0.3l_e$

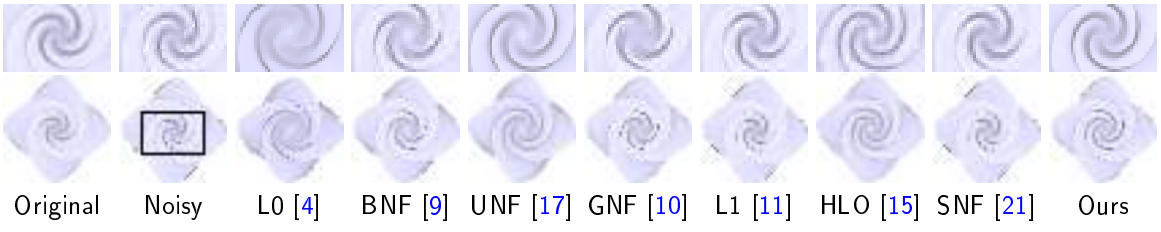

Figure 6: Visual comparison for Octaflower with noise $\sigma_n = 0.1l_e$

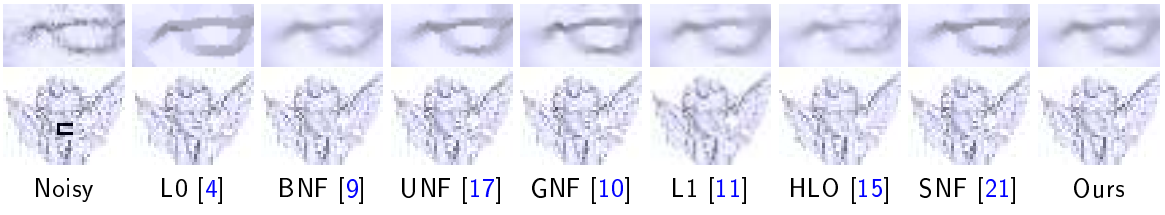

Figure 7: Visual comparison for a raw scanned Angel mesh,

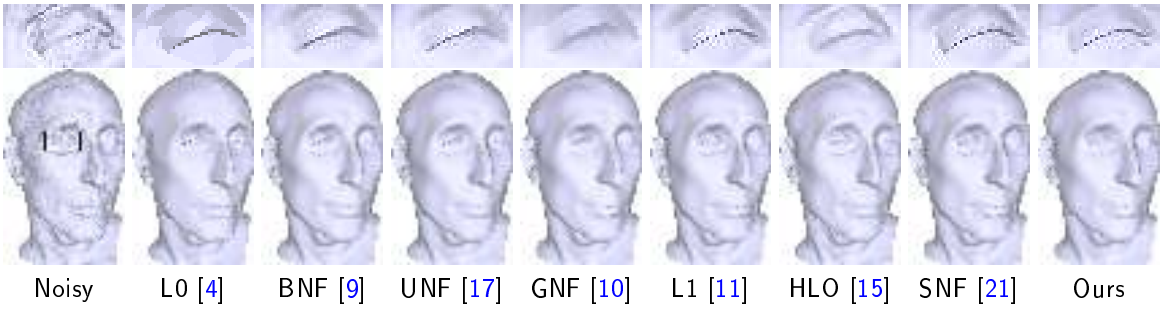

Figure 8: Visual comparison for Nicolo with noise $\sigma_n = 0.1l_e$

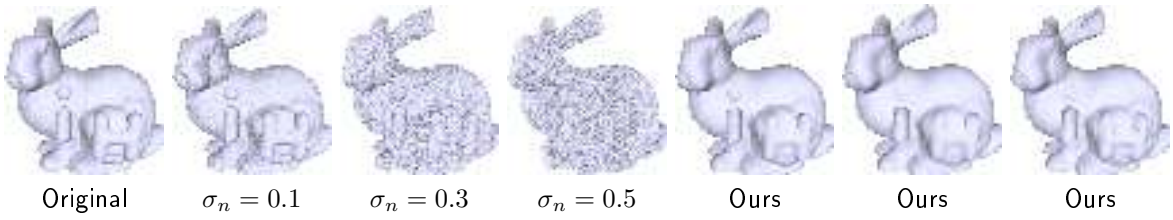

Figure 9: Results of denoising different noise levels on the same irregular model

6 CONCLUSIONS

In this paper, the construction of non-local similar structures based on normal matrices is more representative and robust. Orthogonal polynomials are easier to implement for preserving features in mesh denoising compared to regular polynomials. Similar to other methods, our approach has limited robustness against excessive noise and irregular triangulation. As for future work, we can consider fitting orthogonal polynomials separately to the normals after partitioning the mesh. For instance, setting the mesh's corner and edge parts as feature parts and the rest as non-feature parts and applying thresholding separately to these segments. We hope that through this method, we can design more robust algorithms.

ACKNOWLEDGEMENTS

Lu Lei, https://orcid.org/0000-0002-3050-6542
Ning Li, https://orcid.org/0009-0006-9149-8087
Wei Pan, https://orcid.org/0000-0002-0933-2453
Wenming Tang, https://orcid.org/0000-0002-1427-3216

REFERENCES

[1] Bian, Z.; Tong, R.: Feature-preserving mesh denoising based on vertices classification. Computer Aided Geometric Design, 28(1), 50–64, 2011. http://doi.org/10.1016/j.cagd.2010.10.001.

[2] Chen, S.; Wang, J.; Pan, W.; Gao, S.; Wang, M.; Lu, X.: Towards uniform point distribution in feature-preserving point cloud filtering. Computational Visual Media, 9(2), 249–263, 2023. http://doi.org/10.1007/s41095-022-0278-4.

[3] Fan, H.; Yu, Y.; Peng, Q.: Robust feature-preserving mesh denoising based on consistent subneighborhoods. IEEE Transactions on Visualization and Computer Graphics, 16(2), 312–324, 2009. http://doi.org/10.1109/tvcg.2009.70.

[4] He, L.; Schaefer, S.: Mesh denoising via l 0 minimization. ACM Transactions on Graphics (TOG), 32(4), 1–8, 2013. http://doi.org/10.1145/2461912.2461965.

[5] Hildebrandt, K.; Polthier, K.: Anisotropic filtering of non-linear surface features. In Computer Graphics Forum, vol. 23, 391–400. Wiley Online Library, 2004. http://doi.org/10.1111/j.1467-8659.2004.00770.x.

[6] Hu, K.; Zhang, Y.J.; Li, X.; Xu, G.: Feature-aligned surface parameterization using secondary laplace operator and loop subdivision. Procedia engineering, 163, 186–198, 2016. http://doi.org/10.1016/j.proeng.2016.11.047.

[7] Huang, H.; Wu, S.; Gong, M.; Cohen-Or, D.; Ascher, U.; Zhang, H.: Edge-aware point set resampling. ACM transactions on graphics (TOG), 32(1), 1–12, 2013. http://doi.org/10.1145/2421636.2421645.

[8] Kim, B.; Rossignac, J.: Geofilter: Geometric selection of mesh filter parameters. In Computer Graphics Forum, vol. 24, 295–302. Citeseer, 2005. http://doi.org/10.1111/j.1467-8659.2005.00854.x.

[9] Lee, K.W.; Wang, W.P.: Feature-preserving mesh denoising via bilateral normal filtering. In Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05), 6–pp. IEEE, 2005. http://doi.org/10.1145/1925059.1925086.

[10] Liu, X.; Bao, H.; Shum, H.Y.; Peng, Q.: A novel volume constrained smoothing method for meshes. Graphical Models, 64(3-4), 169–182, 2002. http://doi.org/10.1006/gmod.2002.0576.

[11] Lu, X.; Chen, W.; Schaefer, S.: Robust mesh denoising via vertex pre-filtering and l1-median normal filtering. Computer Aided Geometric Design, 54, 49–60, 2017. http://doi.org/10.1016/j.cagd.2017.02.011.

[12] Lu, X.; Schaefer, S.; Luo, J.; Ma, L.; He, Y.: Low rank matrix approximation for 3d geometry filtering. IEEE transactions on visualization and computer graphics, 28(4), 1835–1847, 2020. http://doi.org/10.1109/tvcg.2020.3026785.

[13] Ohtake, Y.; Belyaev, A.G.; Bogaevski, I.A.: Polyhedral surface smoothing with simultaneous mesh regularization. In Proceedings Geometric Modeling and Processing 2000. Theory and Applications, 229–237. IEEE, 2000. http://doi.org/10.1109/gmap.2000.838255.

[14] O'Leary, P.; Harker, M.: Discrete polynomial moments and savitzky-golay smoothing. International Journal of Computer and Information Engineering, 4(12), 1993–1997, 2010. https://publications.waset.org/vol/48.

[15] Pan, W.; Lu, X.; Gong, Y.; Tang, W.; Liu, J.; He, Y.; Qiu, G.: Hlo: Half-kernel laplacian operator for surface smoothing. Computer-Aided Design, 121, 102807, 2020. http://doi.org/10.1016/j.cad.2019.102807.

[16] Sorkine, O.: Differential representations for mesh processing. In Computer Graphics Forum, vol. 25, 789–807. Wiley Online Library, 2006. http://doi.org/10.1111/j.1467-8659.2006.00999.x.

[17] Sun, X.; Rosin, P.L.; Martin, R.; Langbein, F.: Fast and effective feature-preserving mesh denoising. IEEE transactions on visualization and computer graphics, 13(5), 925–938, 2007. http://doi.org/10.1109/tvcg.2007.1065.

[18] Taubin, G.: A signal processing approach to fair surface design. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, 351–358, 1995. http://doi.org/10.1145/218380.218386.

[19] Vollmer, J.; Mencl, R.; Mueller, H.: Improved laplacian smoothing of noisy surface meshes. In Computer graphics forum, vol. 18, 131–138. Wiley Online Library, 1999. http://doi.org/10.1111/1467-8659.00334.

[20] Wang, J.; Zhang, X.; Yu, Z.: A cascaded approach for feature-preserving surface mesh denoising. Computer-Aided Design, 44(7), 597–610, 2012. http://doi.org/10.1016/j.cad.2012.03.001.

[21] Wang, W.; Pan, W.; Dai, C.; Dazeley, R.; Wei, L.; Rolfe, B.; Lu, X.: Segmentation-driven feature-preserving mesh denoising. The Visual Computer, 1–17, 2023. http://doi.org/10.1007/s00371-023-03161-w.

[22] Wei, M.; Liang, L.; Pang, W.M.; Wang, J.; Li, W.; Wu, H.: Tensor voting guided mesh denoising. IEEE transactions on automation science and engineering, 14(2), 931–945, 2016. http://doi.org/10.1109/tase.2016.2553449.

[23] Wei, M.; Yu, J.; Pang, W.M.; Wang, J.; Qin, J.; Liu, L.; Heng, P.A.: Bi-normal filtering for mesh denoising. IEEE transactions on visualization and computer graphics, 21(1), 43–55, 2014. http://doi.org/10.1109/tvcg.2014.2326872.

[24] Zhang, W.; Deng, B.; Zhang, J.; Bouaziz, S.; Liu, L.: Guided mesh normal filtering. In Computer Graphics Forum, vol. 34, 23–34. Wiley Online Library, 2015. http://doi.org/10.1111/cgf.12742.

[25] Zhao, Y.; Qin, H.; Zeng, X.; Xu, J.; Dong, J.: Robust and effective mesh denoising using l0 sparse regularization. Computer-Aided Design, 101, 82–97, 2018. https://api.semanticscholar.org/CorpusID:44138666.