

# Robust and High Fidelity Mesh Denoising

Sunil Kumar Yadav, Ulrich Reitebuch, and Konrad Polthier

**Abstract**—This paper presents a simple and effective two-stage mesh denoising algorithm, where in the first stage, the face normal filtering is done by using the bilateral normal filtering in the robust statistics framework. *Tukey's bi-weight function* is used as similarity function in the bilateral weighting, which is a robust estimator and stops the diffusion at sharp edges to retain features and removes noise from flat regions effectively. In the second stage, an edge weighted Laplace operator is introduced to compute a differential coordinate. This differential coordinate helps the algorithm to produce a high-quality mesh without any face normal flips and makes the method robust against high-intensity noise.

**Index Terms**—Mesh Denoising, Robust Statistics, Face normal processing, Tukey's bi-weight function, High Fidelity Mesh, Differential Coordinate.



## 1 INTRODUCTION

MESH denoising is one of the most active and fascinating research areas in geometry processing as digital scanning devices become widespread to capture the 3D points of a surface. In the process of data acquisition, noise is inevitable due to various internal, and external sources. Mesh denoising algorithms are focused to remove this undesired noise and compute a high-quality smooth function on a triangular mesh for further use in rendering, scientific analysis, reverse engineering and medical visualization. In general, both sharp features and noise are high-frequency components on the surface and are ambiguous in nature. During the surface denoising, it is a challenging task to decouple these components, which is essential to compute a smooth surface with sharp features.

### 1.1 Related Work

In last two decades, a wide variety of smoothing algorithms have been introduced to remove undesired noise, while preserving sharp features in the geometry. For a comprehensive review on mesh denoising, readers are referred to [1], [2] and [3]. Here, an overview of current state-of-the-art methods is given, which are based on differential coordinates and bilateral filtering.

The concept of differential coordinates was initially introduced by Alexa [4] as a local shape descriptor of a geometry. Differential coordinates are able to preserve the fine details on a triangular mesh, which leads to their further applications in mesh processing, for example, Lipman et al. [5] used the same concept for mesh editing. Mesh quantization [6] and shape approximation [7] are also done by using the concept of differential coordinates. Sorkine et al. [8] introduced a general differential coordinate-based framework for mesh processing. Later, Su et al. [9] utilized differential coordinates to remove noise and compute a smooth surface.

The bilateral filtering was initially proposed by Tomasi et al. [10] for image smoothing. The relation between

anisotropic diffusion [11] and bilateral filtering is explained by Barash [12]. In continuation, Black et al. [13] and Durand et al. [14] expressed anisotropic diffusion and bilateral filtering in the robust statistics framework [15]. The concept of bilateral smoothing was extended for mesh denoising by Fleish et al. [16] and later, Jones et al. [17] explained the same concept in the robust statistics framework. Bilateral filtering was applied to the two-stage surface denoising method by Zheng et al. [18], where the weight functions were computed based on the normal differences (similarity measurement) and spatial distances between neighbouring faces. Later, researchers proposed a general framework for bilateral and mean shift filtering in any arbitrary domain [19]. Recently, several bilateral normal-based denoising algorithms have been published. For example, Wei et al. [20] utilized both face and vertex normals to produce a smooth surface and Yhang et al. [21] proposed a guided mesh normal filtering based on the joint bilateral filtering [22]. A tensor voting guided mesh denoising was also introduced by Wei et al. [23], where feature classification was done using the normal voting tensor and then MLS fitting was applied to remove noise.

The compressed sensing framework is also utilized to produce feature preserving mesh denoising algorithms. He et al. [24] extended the  $L^0$  image smoothing concept to a mesh denoising algorithm where the algorithm maximizes the piecewise flat regions on noisy surfaces to remove noise. The  $L^1$  optimization based algorithms were introduced by Wang et al. [25] and Wu et al. [26] where features were preserved from the residual data by the extended ROF (Rudin, Osher and Fatemi) model. In continuation, Lu et al. [27] detected features on the noisy surface using quadratic optimization and then removed noise using  $L^1$  optimization while preserving features. Later, researchers utilized the  $L^1$ -median normal filtering along with vertex preprocessing to produce a noise free surface [28]. Recently, Yadav et al. [3] proposed a binary optimization-based mesh denoising algorithm, where noise was removed by assigning a binary values to a face normal-based covariance matrix. Centin et al. [2] proposed a normal-diffusion-based mesh denoising algorithm, which removes noise components without tempering the metric quality of a surface. In this paper, a mesh

- Sunil Kumar Yadav, Ulrich Reitebuch and Konrad Polthier are with the Department of Mathematics and Computer Science, Freie Universität Berlin.  
E-mail: sunil.yadav@fu-berlin.de

denoising algorithm is introduced to provide a high fidelity mesh structure against different levels of noise without creating any normal flips.

## 1.2 Contribution

We propose a two-stage mesh denoising method, where the face normal processing is mainly motivated by the robust statistics framework in anisotropic diffusion and bilateral filtering. Basically, the robust statistics framework is focused on developing different estimators which are robust to outliers. In the mesh denoising and robust statistic framework, sharp features are considered as *outliers* and a proper estimator will avoid these *outliers* to preserve sharp features and remove undesired noise components from a noisy surface.

We apply *Tukey's bi-weight function* for the similarity function as a robust estimator which stops the diffusion at sharp features and produces smooth umbilical regions. In the vertex update step, we use a differential coordinate-based Laplace operator along with edge-face normal orthogonality constraint. This Laplace operator is computed using the edge length as the weight function to produce a high-quality mesh without face normal flips and it also makes the algorithm more robust against high-intensity noise.

## 2 METHOD

The proposed method is implemented in two different stages. In the first stage, noisy face normals are processed using the bilateral filtering in the robust statistics framework and in the second stage, a high fidelity mesh is reconstructed using a edge-face normal orthogonality constraint and a edge-weighted differential coordinate.

### 2.1 Robust Bilateral Face Normal Processing

In the first step of the proposed algorithm, we smooth noisy face normals using the bilateral filter, which is a non-linear filter and it computes output using the weighted average of the input:

$$\tilde{\mathbf{n}}_i = \frac{1}{K_i} \sum_{j \in \Omega_i} a_j f(|\mathbf{c}_i - \mathbf{c}_j|, \sigma_c) g(|\mathbf{n}_i - \mathbf{n}_j|, \sigma_s) \mathbf{n}_j, \quad (1)$$

where  $\mathbf{n}_j$  represents a noisy face normal and belongs to the geometric neighbour disk  $\Omega_i$  of the central face normal  $\mathbf{n}_i$ . The geometric neighborhood is more robust against non-uniform meshes compared to the combinatorial neighborhood [3]. The normalization factor  $K_i$  is defined as:

$$K_i = \sum_{j \in \Omega_i} a_j f(|\mathbf{c}_i - \mathbf{c}_j|, \sigma_c) g(|\mathbf{n}_i - \mathbf{n}_j|, \sigma_s),$$

where  $a_j$  is the area of the corresponding face. The term  $f(|\mathbf{c}_i - \mathbf{c}_j|, \sigma_c)$  and  $g(|\mathbf{n}_i - \mathbf{n}_j|, \sigma_s)$  represent the closeness and the similarity functions whose kernel widths are controlled by  $\sigma_c$  and  $\sigma_s$  respectively and  $\mathbf{c}_i$  and  $\mathbf{c}_i$  are the centroids of face  $i$  and  $j$ .

### Robust Estimation

The problem of the estimation of a smooth surface from a noisy surface can be written by using the tools of robust statistics. As shown by Black et al. [13], Durand et al. [14] and Jones et al. [17], to compute a smooth surface, Equation (1) can be written as the following minimization problem:

$$\min_{\mathbf{n}_i} \sum_{j \in \Omega_i} a_j f(|\mathbf{c}_i - \mathbf{c}_j|, \sigma_c) \rho(|\mathbf{n}_i - \mathbf{n}_j|, \sigma_s), \quad (2)$$

where  $\rho$  is a robust error norm. The minimization of the above function will lead to smooth face normals. To solve Equation (2), the first derivative of the  $\rho$ -function should be zero which is a so-called *influence function*  $\Psi(x, \sigma_s) = \rho'(x, \sigma_s)$ , where  $x$  can be any variable (in our algorithm  $x = |\mathbf{n}_i - \mathbf{n}_j|$ ). The influence function shows the behaviour of a  $\rho$ -function.

For anisotropic diffusion and bilateral filtering, the influence function  $\Psi(x, \sigma_s)$  and  $\rho(x, \sigma_s)$ -function can be written in terms of the similarity function  $g(x, \sigma_s)$  [14]:

$$\Psi(x, \sigma_s) = xg(x, \sigma_s), \quad \rho(x, \sigma_s) = \int xg(x, \sigma_s) dx.$$

The selection of a  $\rho$ -function will be crucial to avoid outliers. In our framework, the outliers will be sharp features and an appropriate  $\rho$  function will help the algorithm to retain sharp features and remove noise components.

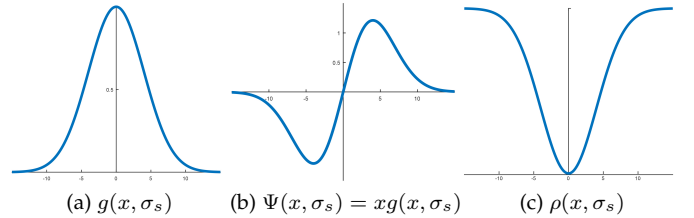


Figure 1: A Gaussian-based similarity function and corresponding influence function and  $\rho$ -function [13].

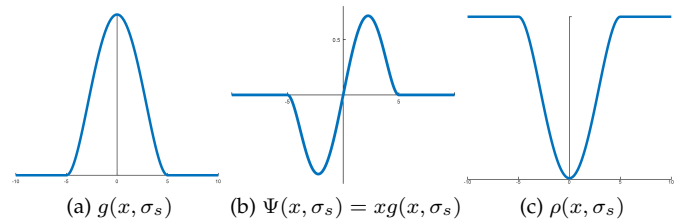


Figure 2: Tukey bi-weight similarity function and corresponding influence function and  $\rho$ -function [13].

For example, the least square estimator (a quadratic  $\rho$ -function) will be the most naive estimator because it will have a linear influence function without any bounds and will be very sensitive to outliers. Figure 1 shows the Gaussian-based  $\rho$ -function which has a bounded influence function  $\Psi(x)$  and gives smaller weight to outliers compared to the least square estimators. Figure 2 shows that *Tukey's bi-weight function* produces more robust results compared to the least square and Gaussian-based estimators

because it completely cuts off the diffusion at sharp features (outliers) and removes noise from non-feature areas. *Tukey's bi-weight function* is defined as [13]:

$$g(x, \sigma_s) = \begin{cases} \frac{1}{2} \{1 - (x/\sigma_s)^2\}^2 & \text{if } |x| \leq \sigma_s, \\ 0 & \text{if } |x| > \sigma_s. \end{cases} \quad (3)$$

In terms of preserving features, *Tukey's bi-weight function* is more effective compared to the Gaussian function because it does not allow the diffusion across sharp features and removes noise components along these sharp features effectively. In our algorithm, we use *Tukey's bi-weight function* as the similarity function to produce feature preserving smooth face normals. For the closeness function  $f(\mathbf{c}_i - \mathbf{c}_j, \sigma_c)$ , a Gaussian function is used.

## 2.2 High Fidelity Mesh Reconstruction

In the second step of the denoising process, we need to impose the effect of processed face normals to the corresponding vertices. To reconstruct a high fidelity mesh, we minimize the following quadratic energy function:

$$\min_{\tilde{\mathbf{v}}_i} \left\{ \|\mathbf{v}_i - \tilde{\mathbf{v}}_i\|^2 + \sum_{j=0}^{N_v(i)-1} \sum_{(i,j) \in \partial F_k} \|\tilde{\mathbf{n}}_k \cdot (\mathbf{v}_i - \mathbf{v}_j)\|^2 + \|\mathbf{R}_i\|^2 \right\}, \quad (4)$$

where  $\tilde{\mathbf{v}}_i$  and  $\mathbf{v}_i$  are the newly computed and the noisy vertex positions. The terms  $N_v(i)$  and  $\partial F_k$  represent the number of vertices and the boundary edge of the vertex star of  $\mathbf{v}_i$  respectively. The second term of the energy function synchronizes a vertex position to the corresponding face normal by using the concept of orthogonality between edge vector and face normal [29]. The term  $\mathbf{R}_i$  helps algorithm to produce a high fidelity mesh and makes the proposed algorithm robust against high-intensity noise. This term is computed using a differential coordinate and can be written as:

$$\mathbf{R}_i = \mathbf{D}_i + \mathbf{D}_{t_i}, \quad (5)$$

where  $\mathbf{D}_i$  and  $\mathbf{D}_{t_i}$  represent the differential coordinate

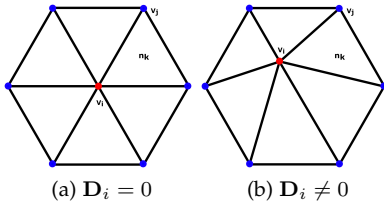


Figure 3: An example to show the effect of differential coordinate  $\mathbf{D}_i$  at the second stage of the proposed algorithm.

and its tangential component respectively. The differential coordinate at the vertex  $\mathbf{v}_i$  and is defined as:

$$\mathbf{D}_i = \Delta \mathbf{v}_i - \mathbf{v}_i, \quad (6)$$

where

$$\Delta \mathbf{v}_i = \frac{1}{\sum_{j \in N(i)} w_{ij}} \sum_{j \in N(i)} w_{ij} \mathbf{v}_j. \quad (7)$$

The weighting term is defined as  $w_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\|^2$ , which is basically based on the edge length of connected vertices. As shown in Figure 3, the differential coordinate  $\mathbf{D}_i$  computes the deviation vector from the centroid of the 1-ring neighbour.

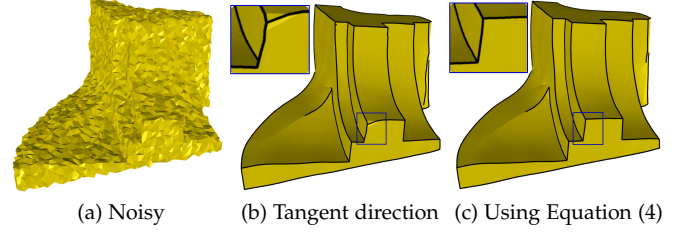


Figure 4: The Fandisk model corrupted by a uniform noise ( $\sigma_n = 0.65l_e$ ) in random direction where  $l_e$  is the average edge length. Figure (b) and (c) show effect of the diffusion of  $\mathbf{D}_i$  only in the tangent direction and using Equation (4) respectively. The sharp feature curve (black curve) is computed using the dihedral angle ( $\theta = 35^\circ$ ). The magnified view shows that the diffusion only in tangent direction is not able to reconstruct the sharp corner.

The term  $\mathbf{D}_{t_i}$  is computed w.r.t the corresponding vertex normal and the vertex normal at the vertex  $\mathbf{v}_i$  is approximated using the processed face normals:

$$\tilde{\mathbf{n}}_{v_i} = \frac{1}{\sum_{j \in F_v(i)} a_j} \sum_{j \in F_v(i)} a_j \tilde{\mathbf{n}}_j, \quad (8)$$

where  $\tilde{\mathbf{n}}_j$  and  $a_j$  represents the processed face normal and the area of the corresponding face, similar to Equation (1). The term  $F_v(i)$  represents the set of faces, which are connected to the vertex  $\mathbf{v}_i$ . Now by using this 1-ring vertex normal, the differential coordinate component in the tangent direction is computed as:

$$\mathbf{D}_{t_i} = \mathbf{D}_i - \langle \mathbf{D}_i, \tilde{\mathbf{n}}_{v_i} \rangle \tilde{\mathbf{n}}_{v_i}, \quad (9)$$

where the term  $\langle \mathbf{D}_i, \tilde{\mathbf{n}}_{v_i} \rangle \tilde{\mathbf{n}}_{v_i}$  represents the component of the differential coordinate in the vertex normal direction. The normal component of the differential coordinate will be more effective against high-intensity noise because mostly noise affects in surface normal direction. Diffusion only in normal direction leads to more shrinkage because the shape information of a geometry is embedded in the surface normal. Similarly, diffusion of tangential components leads to a better mesh quality, less shrinkage, and less shape deformation. However, it is less effective against high-intensity noise. Figure 4 shows the effect of diffusion only in tangent direction and another one by using Equation (4). As it can be seen, the diffusion along the tangent and the normal directions produces a better result compared to the diffusion only in the tangent direction.

Minimization of energy function mentioned in equation (4) can be done by the gradient descent method and the updated vertex position will be:

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i + \frac{1}{3|F_v(i)|} \sum_{j=0}^{N_v(i)-1} \sum_{(i,j) \in \partial F_k} (\tilde{\mathbf{n}}_k \cdot (\mathbf{v}_i - \mathbf{v}_j)) + \lambda_I \mathbf{R}_i, \quad (10)$$

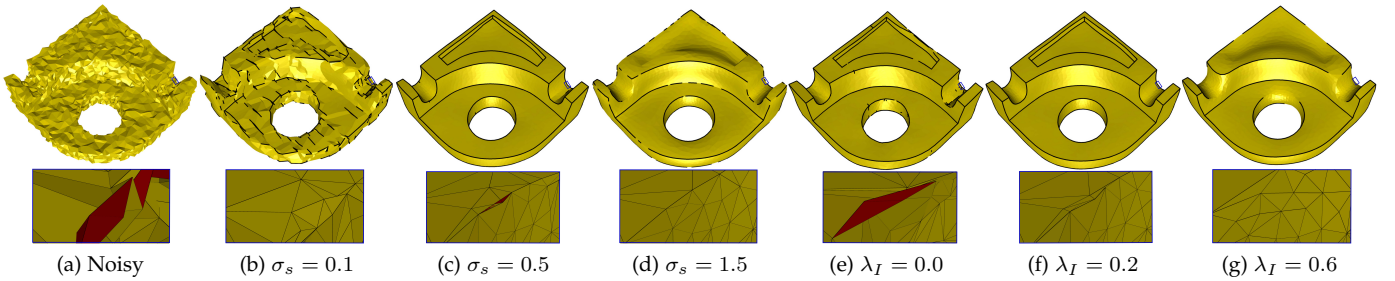


Figure 5: The Bearing model is corrupted with an uniform noise ( $\sigma_n = 0.5l_e$ ) in random direction (a). For Figure (b)-(d),  $\lambda_I = 0.02$  and for Figure (e)-(g),  $\sigma_s = 0.8$ . The black curve shows the sharp edge information in the smooth geometry and is detected using the dihedral angle  $\theta = 65^\circ$ . The second row shows the magnified view mesh structure at a sharp edge. The red color faces are with wrong orientation (flipped normals).

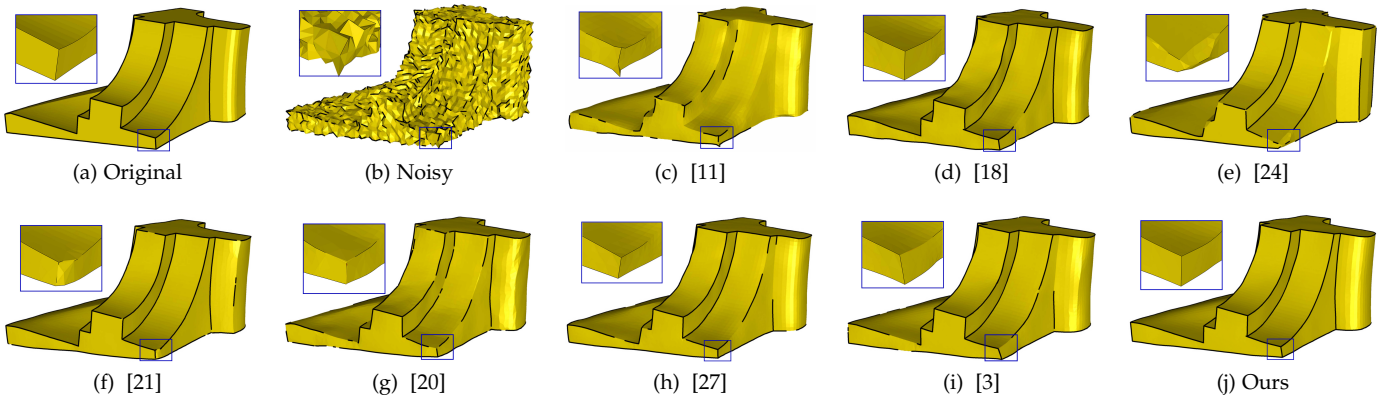


Figure 6: Fandisk model corrupted with a Gaussian noise ( $\sigma_n = 0.3l_e$ ) in random direction. The first row shows the results produced by state-of-the-art methods and our proposed method. The black curve shows the sharp edge information in the smooth geometry and is detected using the dihedral angle  $\theta = 70^\circ$ .

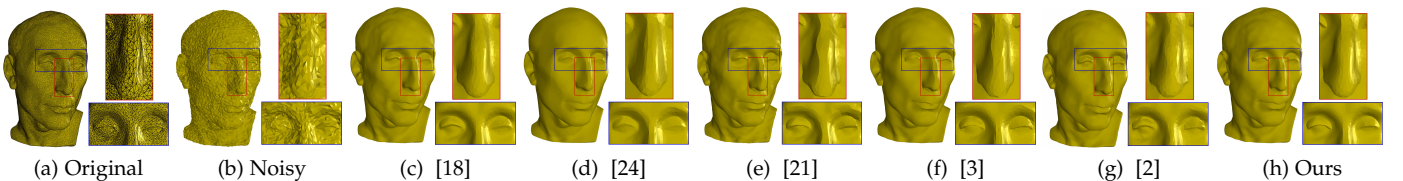


Figure 7: The Nicola model, a non-uniform triangulated mesh surface corrupted by Gaussian noise in random direction. The magnified view of the eyes and the nose shows that the proposed method preserves better sharp features without creating piecewise flat areas compared to state-of-the-art methods.

where  $\lambda_I$  is the isotropic smoothing or mesh fidelity factor. We iterate the whole procedure mentioned in Equation (10) several times to get the desired result. In each iteration, using the same value of  $\lambda_I$  leads to feature blurring, shape deformation and volume shrinkage in the smoothed surface as noise is getting lower with the iterations. To overcome this problem, the effect of isotropic smoothing should get reduced after each iteration. We reduce  $\lambda_I$  in every iteration by  $\lambda_I^{p+1} = 0.6\lambda_I^p$ , where  $p$  represents the number iterations.

### 3 EXPERIMENTS, RESULTS AND DISCUSSION

We have performed our denoising scheme on various kinds of CAD (Figure 4, 5, 6, 8 and 9) and CAGD (Figure 7 and 9)

models. These models consist of different levels of features and are corrupted with different kind of synthetic noise (Gaussian and uniform) in different random directions. We also included real scanned data (Figure 10, 11 and 12) models in our experiments. We provide the visual and quantitative comparison of our method to several state-of-the-art denoising methods in which we implemented [3], [16], [18], [11] and [24] based on their published article and several results of [2], [20], [21], [30] and [27] are provided by their authors.

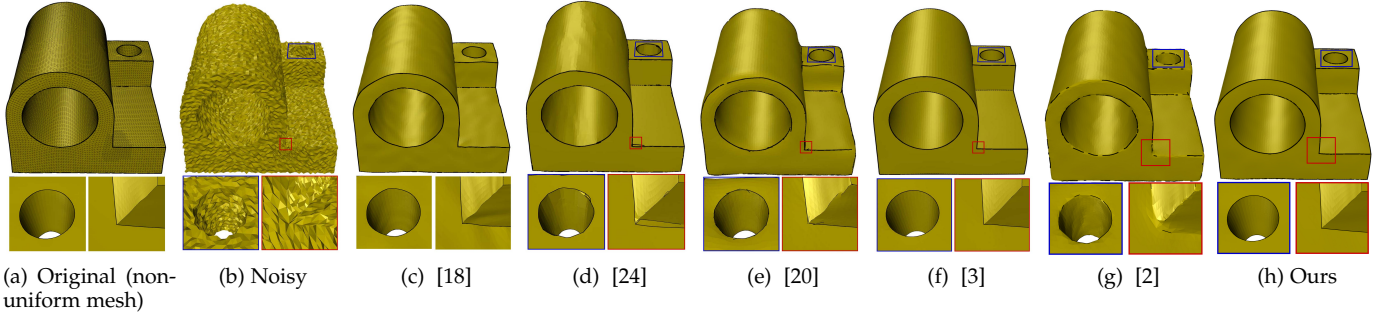


Figure 8: Non-uniform triangulated mesh surface corrupted by Gaussian noise ( $\sigma_n = 0.35l_e$ ) in normal direction. The first row shows the results obtained by state-of-the-art methods and the proposed method. The second row shows the magnified view of the corner and the cylindrical hole of the corresponding geometry.

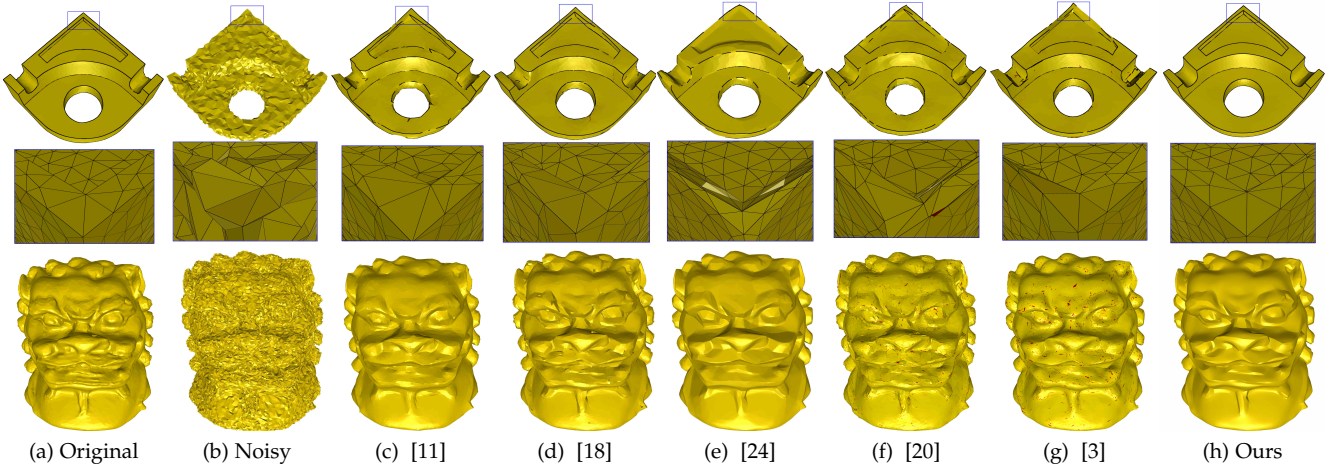


Figure 9: Figure (a) shows the noisy models, which are corrupted by a uniform noise ( $\sigma_n = 0.5l_e$ ) in random directions. Figure (b)-(h) show the results produced by state-of-the-art methods and the proposed method. The second row shows the magnified view of the mesh quality at the sharp corners. The sharp feature curves are computed at  $\theta = 65^\circ$  (for the Bearing model) and the red color faces are with wrong orientation (flipped normals).

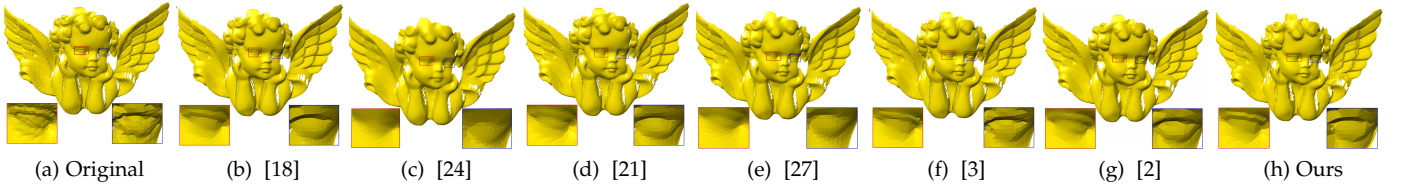


Figure 10: The Angel model, a triangulated mesh surface (real data) obtained by a 3D scanner. The figures shows the result produced by the proposed method, better preserving sharp features (both eyes) compared to state-of-the-art methods.

### 3.1 Parameters

In the proposed method, we discussed about five different parameters: kernel widths of the closeness and similarity functions ( $\sigma_c$  and  $\sigma_s$ ), geometric neighbour disk ( $\Omega$ ), isotropic/mesh fidelity factor  $\lambda_I$  and the number of iterations  $p$ . Throughout the algorithm, the radius of the neighbour disk is defined as  $r = 2c_a$ , where  $c_a$  represents the average distance between the centroid of faces and the kernel width of the closeness function is half of the radius of the neighbour disk ( $\sigma_c = c_a$ ). The average centroid distance-based radius will make the algorithm robust

against irregular sampling as the denser region will have more faces compared to the rare region during the face normal processing. Effectively, the user has to tune only 3 parameters ( $\sigma_s$ ,  $\lambda_I$  and  $p$ ) to get the desired results. In the quantitative comparison table, the proposed method parameters are mentioned in the following format ( $\sigma_s, \lambda_I, p$ ). Similarly, ( $\tau, r, p$ ) for [3], where  $\tau$  is the feature threshold,  $r$  is the radius of the geometric neighborhood disk and  $p$  is the number of iterations. The parameters ( $\sigma_c, \sigma_s, p$ ) for [16], ( $\sigma_s, p$ ) for [18], ( $\lambda, s, p$ ) for [11] and  $\alpha$  for [24], where  $\sigma_c$  and  $\sigma_s$  are the standard deviation of the Gaussian function in

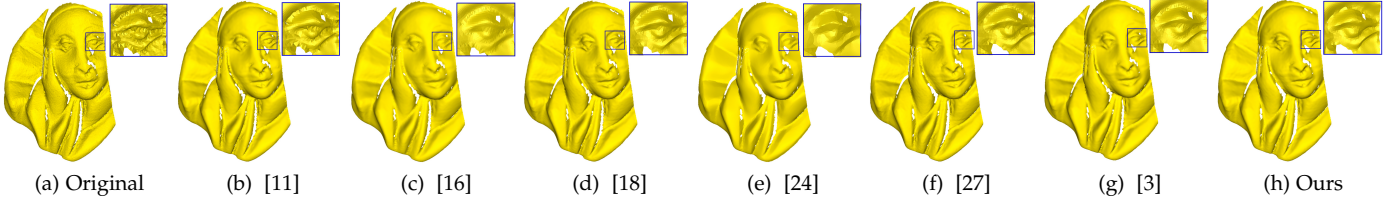


Figure 11: The Pierrot model (real data) is corrupted by a 3D scanner noise. The figures shows the results obtained by state-of-the-art methods and the proposed method.

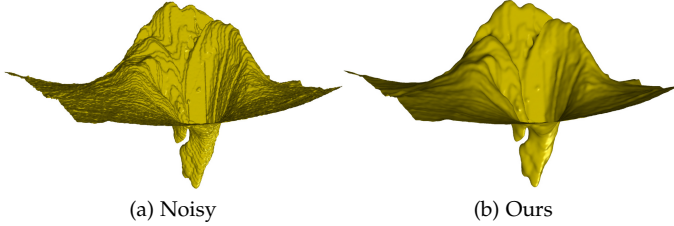


Figure 12: Figure (a) shows the noisy model of a human retina with marching cube artifacts and the proposed method removes these artefacts by choosing the appropriate value of  $\lambda_I$ .

the bilateral weighting. The variables  $s$  and  $\lambda$  represent the step size and the smoothing threshold. The term  $\alpha$  controls the amount of smoothing in [24].

Figure 5 shows the effect of the kernel width parameter  $\sigma_s$  of the similarity function and isotropic factor  $\lambda_I$ . The black curve represents sharp features and is computed using the dihedral angle,  $\theta = 65^\circ$  between neighboring faces. From Figure 5 (b-d), the isotropic factor is fixed,  $\lambda_I = 0.02$ . The small value of the  $\sigma_s = 0.1$  produces a piecewise flat region and is not able produce a noise free surface with sharp features (Figure 5 (b)). For  $\sigma_s = 1.5$ , the sharp features are smoothed out and for  $\sigma_s = 0.5$ , the algorithm produces an optimal result. Similarly, from Figure 5 (e-g), the parameter  $\sigma_s$  is fixed,  $\sigma_s = 0.8$ . For  $\lambda_I = 0$ , we can see that the reconstructed surface has several edge flips (faces with wrongly oriented normals) and the algorithm is not able to preserve all desired features (Figure 5(e)). However, the algorithm produces a high fidelity mesh with sharp features when we use  $\lambda_I = 0.2$  (Figure 5(f)) and blurs the sharp features with bigger isotropic factor (Figure 5 (g)). In general, the value of  $\lambda_I$  depends on noise intensity on surfaces. The higher the intensity of noise the more normal flips will be produced during the denoising process, therefore, it is better to use a bigger value of  $\lambda_I$  to avoid these normal flips. At the same time, higher values for  $\lambda_I$  have to be carefully as these could produce edge blurring and volume shrinkage.

### 3.2 Visual Comparison

Figure 6-11 represent a visual comparison of the proposed method with several state-of-the-art methods. Figure 6 shows the Fandisk model, which is corrupted with a moderate intensity Gaussian noise in random direction. The sharp

feature curve (black curve) shows that the proposed method preserves sharp features (especially at corners) effectively and does not produce false features at the cylindrical region compared to state-of-the-art methods. Method [3] and [27] manage to produce features at sharp corners but do not recover the sharp features on the whole surface. Method [24] creates false features at the cylindrical region and is not able to preserve the corner features [2]. The Nicola model (Figure 7) has a non-uniform mesh and is corrupted by a Gaussian noise in random direction. The magnified view shows that our method preserves sharp features around the eye regions better than state-of-the-art methods and does not produce any false features around the nose area (piecewise flat areas). Figure 8 shows the Joint sharp model with non-uniform mesh corrupted with a Gaussian noise ( $\sigma_n = 0.35l_e$ ) in normal direction, where  $l_e$  is the average edge length. As it shown, the proposed method reconstructs the cylindrical region and corners without creating any false features. Method [3] produces a quite similar result to ours while Method [21] and [24] produce false features around the cylindrical region. The other state-of-the-art methods are not able to preserve sharp features. Figure 9 provides a visual comparison between state-of-the-art methods and the proposed method where models are corrupted with high-intensity of noise ( $\sigma_n = 0.5l_e$ ). As shown in Figure 9, our method is able to remove noise components properly and retains the sharp features without creating any edge flips. Method [18] is able to preserve sharp features but produces edge flips and the mesh quality is not optimal. Method [24] is using an isotropic regularization factor which helps the algorithm to produce a good quality mesh on the planar areas but fails to produce similar triangles near the sharp edges (second row Figure 9(e)).

In general, real data is captured using the 3D scanners and have quite low intensity of noise. Thus, it is difficult to see a considerable difference between the results of the proposed method and state-of-the-art methods. Figure 10 shows that our algorithm is capable of better preserving features around both eyes compared to state-of-the-art methods. Methods [3] and [18] are able to retain the feature in the right eye but fail for the left eye. The other state-of-the-art methods are not able to preserve the fine level of features in both eyes. In case of the Pierrot model (Figure 11), our method removes noise effectively around the eyes and preserves features. Methods [27] and [3] retain similar scale of the feature but are not able to remove noise properly. Figure 12 shows the applicability of the proposed method to medical data. As shown in Figure 5, the level of the feature can be controlled by using the parameter  $\lambda_I$ , which is effective in real data smoothing. For example, Figure 12(a) shows a noisy model of human retina obtained by the optical coherence tomography [31]. The surface has the

marching cube artefacts which is removed by the proposed method along with noise components effectively.

Table 1: Quantitative Comparison

Models	Methods	MSAE	$E_v \times 10^{-3}$	$Q$	Parameters
Nicola $ F  = 29437$ $ V  = 14846$	[16]	7.45	172.7	1.18	(0.3, 0.3, 30)
	[11]	6.269	76.87	1.00	(0.05, 0.05, 50)
	[18]	<b>5.25</b>	<b>66.07</b>	1.02	(0.3, 60)
	[24]	6.50	99.58	0.91	(1.4)
	[21]	5.82	69.63	1.05	(Default)
	[3]	5.900	72.8	1.02	(0.3, 0.15, 50)
	<b>Ours</b>	5.5	66.14	<b>0.9</b>	(1.6, 0.06, 25)
Bearing $ F  = 6950$ $ V  = 3475$	[18]	3.17	0.590	1.45	(0.3, 50)
	[24]	4.70	1.24	1.58	(4.0)
	[20]	3.95	0.783	2.62	(Default)
	[3]	4.43	0.698	1.66	(0.3, 0.08, 80)
	<b>Ours</b>	<b>2.32</b>	<b>0.588</b>	<b>0.74</b>	(0.8, 0.2, 100)
Joint $ F  = 52226$ $ V  = 26111$	[16]	3.777	0.530	0.83	(0.3, 0.3, 40)
	[11]	2.630	0.766	0.78	(0.009, 0.05, 50)
	[18]	1.808	0.263	0.77	(0.4, 100)
	[24]	1.768	0.500	0.75	(1.4)
	[21]	0.956	0.179	0.83	(Default)
	[20]	2.874	0.366	1.67	(Default)
	[27]	1.16	1.49	<b>0.71</b>	(Default)
	[3]	<b>0.829</b>	<b>0.171</b>	0.91	(0.3, 0.05, 60)
	<b>Ours</b>	1.032	0.198	0.73	(1.0, 0.2, 150)
Fandisk $ F  = 12946$ $ V  = 6475$	[16]	8.567	4.422	0.87	(0.4, 0.4, 40)
	[11]	5.856	4.910	0.79	(0.07, 0.05, 30)
	[18]	2.727	1.877	0.76	(0.4, 70)
	[24]	4.788	5.415	0.766	(1.4)
	[21]	2.221	<b>1.702</b>	0.97	(Default)
	[27]	3.1	4.42	0.7	(Default)
	[3]	2.692	1.964	0.89	(0.3, 0.2, 50)
	<b>Ours</b>	<b>2.201</b>	<b>1.844</b>	<b>0.7</b>	(0.55, 0.2, 100)
Chinese Lion $ F  = 118276$ $ V  = 59140$	[18]	14.88	19.64	2.72	(1.0, 100)
	[24]	8.93	25.34	1.51	(1.4)
	[20]	19.22	21.5	2.38	(Default)
	[3]	21.39	21.89	2.16	(0.3, 1.0, 100)
	<b>Ours</b>	<b>8.795</b>	<b>18.35</b>	<b>0.85</b>	(0.4, 0.3, 100)

### 3.3 Quantitative Comparison

For the quantitative comparison, we are using three different parameters. Two of them show differences between the ground truth and smooth model and the so-called  $L^2$  vertex-based positional error and the orientation error respectively. The third parameter computes the mesh quality of a smoothed surface. The  $L^2$  vertex-based positional error from the original ground truth model is represented by  $E_v$  and defined as [32]:

$$E_v = \sqrt{\frac{1}{3 \sum_{k \in F} a_k} \sum_{i \in V} \sum_{j \in F_v(i)} a_j \text{dist}(\tilde{\mathbf{v}}_i, T)^2},$$

where  $F$  is the triangular element set and  $V$  represents the set of vertices. The terms  $a_k$  and  $a_j$  are the corresponding face areas. The distance  $\text{dist}(\tilde{\mathbf{v}}_i, T)$  is the closest  $L^2$ -distance between the newly computed vertex  $\tilde{\mathbf{v}}_i$  and the triangle  $T$  of the reference model.

The MSAE computes the orientation error between the original model and the smooth model and is defined as:

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

where  $\tilde{\mathbf{n}}$  is the newly computed face normal and  $\mathbf{n}$  represents the face normal of the reference model. The term  $E$  stands for the expectation value.

Mesh quality of the surface mainly refers to the shape of the faces in the mesh. For a better mesh quality, the size and angles of the triangular face should not be too small or too large. To measure the quality of faces, we use the ratio of the

circumradius-to-minimum edge length of the corresponding face [33]:

$$Q = \frac{r}{e_{\min}},$$

where  $r$  and  $e_{\min}$  are circumradius and minimum edge length of the corresponding triangle. Ideally, every face belonging to the mesh should be an equilateral triangle with a quality index of  $Q = 1/\sqrt{3}$ .

Table 1 shows the quantitative comparison of our method with state-of-the-art methods. For the models which are corrupted with high-intensity noise (Bearing, Chinese Lion and Sharp Sphere), the proposed method has lower values of  $E_v$ , MSAE and  $Q$  compared to Methods [18] and [24]. Method [18] produces a quite similar value of  $E_v$  but bigger values for MSAE and  $Q$  because of the several edge flips. He et al. [24] produces a good quality mesh in planar regions but does not produce a similar quality of triangles near sharp features, which leads to a higher value mesh quality index. For the Nicola model, our method produces a quite similar values for the position and orientation error (slightly bigger) to Method [18], because our algorithm also follows the bilateral filtering technique and in addition, we use an isotropic factor which introduces a small amount of volume shrinkage. At the same time, it produces a good quality mesh.

Table 2: Running Time (in seconds)

Models	Fandisk	Bearing	Nicola	Joint	Lion
Time (s)	5.8	4.2	4.3	45.3	92

### 3.4 Running Time

As it is explained by [3], the two-stage denoising methods have quite similar running time complexity. The proposed algorithm also follows the two-stage denoising scheme and has the complexity of  $O(n_c \cdot n_f \cdot p)$ , where  $n_c$  is the number of faces within the neighbourhood,  $n_f$  and  $p$  are the numbers of faces and iterations respectively. The implementation of the algorithm is quite straightforward and simple. First of all, we compute the smooth face normals using Equation (1) then rearrange vertices by following the Equation (10). Our implementation is done in a single computation thread using Java. Table 2 shows the running time of our algorithm for different models according to the parameter values mentioned in Table 1.

## 4 CONCLUSION

In this paper, we presented a robust mesh denoising algorithm which produces a high-quality mesh along with sharp features. In the first step of the proposed method, a robust statistics framework is applied for noisy face normal smoothing where *Tukey's bi-weight function* is used as the robust estimator. *Tukey's bi-weight function* completely stops the diffusion across sharp edges and removes noise along the sharp features and from flat areas. In the second step, a vertex position is updated using edge-face normal orthogonality constraints along with differential coordinates. Differential coordinates helps the algorithm to produce a high-quality mesh, makes it robust against high-intensity of noise and avoids normal flips during the denoising process. In section 3, we have shown the robustness of the proposed

algorithm not only against different kinds and levels of noise but also against the face normal flipping. The quantitative comparison table shows that our method produces better mesh quality compared to state-of-the-art methods. Similarly, for the positional and the orientation error, the proposed method is either better or similar to state-of-the-art methods.

## REFERENCES

- [1] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. uno Levy, *Polygon Mesh Processing*. AK Peters, 2010.
- [2] M. Centin and A. Signoroni, "Mesh denoising with (geo)metric fidelity," *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2017.
- [3] S. K. Yadav, U. Reitebuch, and K. Polthier, "Mesh denoising based on normal voting tensor and binary optimization," *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2017.
- [4] M. Alexa, "Differential coordinates for local mesh morphing and deformation," *The Visual Computer*, vol. 19, no. 2-3, pp. 105–114, 2003.
- [5] Y. Lipman, O. Sorkine, M. Alexa, D. Cohen-Or, D. Levin, C. Rössl, and H.-P. Seidel, "Laplacian framework for interactive mesh editing," *International Journal of Shape Modeling (IJSM)*, vol. 11, no. 1, pp. 43–61, 2005.
- [6] O. Sorkine, D. Cohen-Or, and S. Toledo, "High-pass quantization for mesh encoding," in *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, ser. SGP '03, 2003, pp. 42–51.
- [7] O. Sorkine, D. Cohen-Or, D. Irony, and S. Toledo, "Geometry-aware bases for shape approximation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 171–180, Mar. 2005.
- [8] O. Sorkine, "Differential representations for mesh processing," *Computer Graphics Forum*, vol. 25, no. 4, pp. 789–807, 2006.
- [9] Z.-X. Su, H. Wang, and J.-J. Cao, "Mesh denoising based on differential coordinates," in *Shape Modeling and Applications, 2009. SMI 2009. IEEE International Conference on*, June 2009, pp. 1–6.
- [10] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–846.
- [11] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Computer Graphics Forum*, vol. 23, pp. 391–400, 2004.
- [12] D. Barash, "Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 6, pp. 844–847, Jun 2002.
- [13] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 421–432, Mar 1998.
- [14] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '02. New York, NY, USA: ACM, 2002, pp. 257–266.
- [15] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics - The Approach Based on Influence Functions*. Wiley, 1986, missing.
- [16] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03. New York, NY, USA: ACM, 2003, pp. 950–953.
- [17] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 943–949, Jul. 2003.
- [18] Y. Zheng, H. Fu, O.-C. Au, and C.-L. Tai, "Bilateral normal filtering for mesh denoising," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 10, pp. 1521–1530, Oct 2011.
- [19] J. Solomon, K. Crane, A. Butscher, and C. Wojtan, "A general framework for bilateral and mean shift filtering," *ArXiv e-prints*, vol. 32, April 2014.
- [20] M. Wei, J. Yu, W. M. Pang, J. Wang, J. Qin, L. Liu, and P. A. Heng, "Bi-normal filtering for mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 1, pp. 43–55, Jan 2015.
- [21] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu, "Guided mesh normal filtering," *Computer Graphics Forum (Special Issue of Pacific Graphics 2015)*, vol. 34, pp. 23–34, 2015.
- [22] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.
- [23] M. Wei, L. Liang, W. M. Pang, J. Wang, W. Li, and H. Wu, "Tensor voting guided mesh denoising," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 931–945, April 2017.
- [24] L. He and S. Schaefer, "Mesh denoising via  $L^0$  minimization," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 64:1–64:8, Jul. 2013.
- [25] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, "Decoupling noise and features via weighted  $L^1$ -analysis compressed sensing," *ACM Trans. Graph.*, vol. 33, no. 2, pp. 18:1–18:12, Apr. 2014.
- [26] X. Wu, J. Zheng, Y. Cai, and C.-W. Fu, "Mesh Denoising using Extended ROF Model with L1 Fidelity," *Computer Graphics Forum*, 2015.
- [27] X. Lu, Z. Deng, and W. Chen, "A robust scheme for feature-preserving mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 3, pp. 1181–1194, March 2016.
- [28] X. Lu, W. Chen, and S. Schaefer, "Robust mesh denoising via vertex pre-filtering and l1-median normal filtering," *Computer Aided Geometric Design*, vol. 54, pp. 49 – 60, 2017.
- [29] G. Taubin, "Linear anisotropic mesh filtering," in *IBM Research Report RC22213(W0110-051)*, IBM T.J. Watson Research, 2001.
- [30] A. Adams, N. Gelfand, J. Dolson, and M. Levoy, "Gaussian kd-trees for fast high-dimensional filtering," in *ACM SIGGRAPH 2009 Papers*, ser. SIGGRAPH '09. New York, NY, USA: ACM, 2009, pp. 21:1–21:12.
- [31] S. K. Yadav, S. Motamedi, T. Oberwahrenbrock, F. C. Oertel, K. Polthier, F. Paul, E. M. Kadas, and A. U. Brandt, "Cube: parametric modeling of 3d foveal shape using cubic b ezier," *Biomed. Opt. Express*, vol. 8, no. 9, pp. 4181–4199, Sep 2017.
- [32] X. Sun, P. Rosin, R. Martin, and F. Langbein, "Fast and effective feature-preserving mesh denoising," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 5, pp. 925–938, Sept 2007.
- [33] J. R. Shewchuk, "Delaunay Refinement Mesh Generation," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1997, available as Technical Report CMU-CS-97-137.