Surface Denoising Based on Normal Filtering in a Robust Statistics Framework

Yadav, Sunil Kumar; Skrodzki, M.; Zimmermann, Eric; Polthier, Konrad

DOI
10.1007/978-981-16-5576-0_6

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the Forum "Math-for-Industry" 2018

Citation (APA)
Yadav, S. K., Skrodzki, M., Zimmermann, E., & Polthier, K. (2022). Surface Denoising Based on Normal Filtering in a Robust Statistics Framework. In *Proceedings of the Forum "Math-for-Industry" 2018* (pp. 103-132). Springer. https://doi.org/10.1007/978-981-16-5576-0_6

Surface Denoising Based on Normal Filtering in a Robust Statistics Framework

Sunil Kumar Yadav, Martin Skrodzki, Eric Zimmermann, and Konrad Polthier

1 Introduction

Surface denoising—generally being part of the preprocessing stage in the geometry processing pipeline—is designed to remove high-frequency noise corrupting a geometry. The noise generally arises from scanning or other acquisition processes. In contrast to smoothing, we are interested in preserving attributes and features of the geometry like edges and corners. Here, the difficulty lies in distinguishing these from noise, depending on the intensity of noise and the level of the attributes' details.

Denoising can therefore be considered as being part of the area of smoothing. It is used in all applications asking for a cleaned, i.e., noise-free, surface with the additional property of keeping features. But more importantly, it is recognized as being a major tool in the preprocessing stage of geometry processing. The reason is that—besides computer-designed models—the acquisition of real world models via 3D scanning processes unfortunately adds noise and outliers to the data due to mechanical limitations and sub-optimal surrounding conditions. These artifacts influence meshes and point sets alike and have to be removed to obtain a clean model for further use in different industry applications, e.g., scientific analysis, automotive, medical diagnosis, rendering, and other geometry processing algorithms like surface reconstruction, feature detection, computer-aided design, or 3D printing, see (Yadav et al. 2018b) for applications in medical diagnoses and (Botsch et al. 2010) for a variety of application scenarios.

S. K. Yadav · M. Skrodzki · E. Zimmermann · K. Polthier
Department of Computer Science and Mathematics, Freie Universität Berlin, Berlin, Germany

S. K. Yadav (✉)
Technical Development, Nocturne GmbH, Berlin, Germany
e-mail: sunil.yadav@fu-berlin.de

M. Skrodzki
Computer Graphics and Visualization, TU Delft, Delft, The Netherlands

J. Cheng et al. (eds.), *Proceedings of the Forum "Math-for-Industry" 2018*,
Mathematics for Industry 35, https://doi.org/10.1007/978-981-16-5576-0_6

A typical challenge arising in the denoising process is the decoupling of noise and features of a geometry. This is, because both are high-frequency components of the geometry in terms of the spectral setting. Other problems arise as noisy geometries include outliers, which are far away from the underlying ground truth. Furthermore, the amplitude of noise can be significant when compared to the feature size. To solve these problems, in both cases—for meshes and point sets—a variety of surface denoising algorithms have been published. These state-of-the-art methods can be categorized into:

1. One-stage methods, where noise components are removed by adjusting the vertex positions based on curvature information;
2. Two-stage methods, wherein the first stage, surface normals are filtered and then in the second stage, vertex positions are adjusted according to the filtered normals.

Two-stage methods are more effective in terms of feature preservation as well as noise removal and obtain minimum volume shrinkage compared to one-stage methods, see (Centin and Signoroni 2018; Yadav et al. 2018c, 2019). In the two-stage methods, surface normal filtering is the key part as it is responsible for both noise removal and feature preservation. Therefore, several procedures have been published for normal filtering. Each of these algorithms is effective in different aspects (like robustness against noise, feature preservation, or detection of outliers). However, there is no unified theoretical framework available in which we can discuss the benefits and drawbacks of the normal filtering algorithms and in which we can derive relations between these methods.

In this paper, we focus on this issue and introduce such a unified framework making use of robust statistics to derive relations between (both linear and nonlinear) state-of-the-art surface normal filtering methods. On the basis of these relations, we discuss the robustness of each algorithm against noise and its respective feature preservation capability. The presented framework can be used to provide pros and cons of published methods for the development of new algorithms. Furthermore, it can serve as a comparison possibility for such new procedures to state-of-the-art methods on a theoretically sound basis.

1.1 Notation

Throughout the whole paper, we will use the following notation. Let I, J, K denote index sets as subsets of $\mathbb{N}$. We consider a mesh $\mathcal{M} = (P, E, F)$ consisting of a set of points or vertices $P = \{p_i\}_{i \in I} \subset \mathbb{R}^3$ (which will be used in the point set setting as well), (undirected) edges E, and faces F. In general, we will assume that the mesh $\mathcal{M}$ or the point set P is corrupted by noise. The set of normals is given as $N = \{n_j\}_{j \in J} \subset \mathbb{S}^2$, with $\mathbb{S}^2$ the two-dimensional unit sphere in $\mathbb{R}^3$ and neighborhoods are labeled Ω_k for $k \in K$. Sometimes we only refer to the neighborhood by Ω and to its representatives by $p, q \in \Omega$ without further labels, to simplify the notation where it is unambiguous. The used type of neighborhood will get specified when necessary and receive a dedicated index set, as it further depends on the context, i.e., to which object (points, faces, ...) we are going to relate it. Consequently, normals and neighborhoods apply for faces and points depending on whether we discuss the mesh or point set setting. Let $|X|$ denotes the size of a set X and let $\|v\|$ as well as v^T be the Euclidean norm and the transpose of a vector $v \in \mathbb{R}^3$, respectively. A surface area or a vertex, both of high curvature (in comparison with the other elements of the geometry) will be referred to as a *feature* of the mesh or the point set, respectively.

1.2 Related Work

In the last two decades, many surface smoothing algorithms have been developed. Due to the large number of available methods, for a comprehensive overview we refer to (Botsch et al. 2010; Centin and Signoroni 2018). Here, we give a short overview of methods highly related to the robust statistics setting and of the most important state-of-the-art methods. As stated above, the removal of noise components is equivalent to the removal of high-frequency components. Here, the Fourier transform is a common tool, allowing efficient implementations of low-pass filters to cut off high frequencies. It has been generalized to manifold harmonics to be applicable to 2-manifold surfaces via the eigenfunctions of the Laplace–Beltrami operator of these surfaces. Its matrix representation encodes the *natural vibrations* of a triangle mesh in its eigenvectors and the *natural frequencies* in its eigenvalues, see (Taubin 1999, 2001a). One drawback is its cost for many applications as the eigenvector decomposition of the Laplace matrix is numerically challenging to compute; see (Vallet and Levy 2008).

A similar removal of high-frequency components can be achieved by utilizing the diffusion flow, which dampens high frequencies (instead of cutting them off) by a multiplication with a Gaussian kernel. It can be computed directly on the mesh, making it cheaper and hence more practical than the Fourier transform. Let $f(\mathbf{p}, t) : \mathbb{R}^{3|P|+1} \to \mathbb{R}$ be a given signal with $\mathbf{p} = (p_1, \dots, p_{|P|})^T$. The diffusion equation:

$$\frac{\partial f(\mathbf{p}, t)}{\partial t} = \lambda \Delta f(\mathbf{p}, t) \tag{1}$$

describes the change of f over time by a scalar diffusion coefficient $\lambda \in \mathbb{R}$ multiplied with its spatial Laplacian Δf, which can be replaced by the Laplace–Beltrami operator on manifolds. As the discretization asks for small time steps to be numerically robust in the integration, the authors of (Desbrun et al. 2001) proposed an implicit time integration providing unconditional robustness even for large time steps. A smoothing procedure can be derived from this as update of the vertex positions p_i by a point-wise update scheme

$$
\begin{aligned}
p_i &\leftarrow p_i + h\lambda \Delta p_i, \\
\text{with} \;\; \Delta p_i &= -2\,H n_i,
\end{aligned}
\tag{2}
$$

because the Laplace–Beltrami operator on vertices corresponds to the mean curvature. Hence, all vertices p_i move in the corresponding normal direction n_i by a magnitude regulated by the mean curvature H. This is known as the *mean curvature flow*, see (Desbrun et al. 2001).

The isotropic Laplacian has been extended by a data-dependent diffusion tensor yielding the anisotropic flow equation:

$$
\frac{\partial f}{\partial t} = \mathrm{div}[g_\sigma(\|\nabla f\|)\nabla f], \tag{3}
$$

where f is a signal as in Eq. (1) and $g_\sigma(\cdot)$ is an edge stopping function (anisotropic weighting function), which is responsible for feature preservation with a user input parameter σ during denoising operations, see (Perona and Malik 1990; Clarenz et al. 2000). Further examples for the usage of the anisotropic diffusion equation can be found in (Bajaj and Xu 2003; Hildebrandt and Polthier 2004). The same concept is extended to the context of point set smoothing by Lange and Polthier (Lange and Polthier 2005) and to face normal filtering by Tasdizen et al. (Tasdizen et al. 2002).

Another set of denoising techniques consists of two-stage mesh denoising algorithms. Here, at the first stage, face normals are filtered and in the second stage vertex positions are updated according to the newly computed face normals, see (Taubin 2001b). Face normal filtering is performed by using several linear and nonlinear filters in order to preserve sharp features (Centin and Signoroni 2018; Yadav et al. 2018c; Yagou et al. 2002, 2003; Ohtake et al. 2002; Belyaev and Ohtake 2001) and vertex updates are performed by using the edge-face orthogonality (Sun et al. 2007).

Finally, there are several denoising methods utilizing bilateral filtering. It arose from image processing (Tomasi and Manduchi 1998) and uses a combination of two different weighting functions: a spatial kernel and a range kernel to preserve features and remove noise components. It got adapted to surface denoising for instance in (Fleishman et al. 2003), where the information of spatial distances and the local variation of vertex normal vectors are combined for denoising. Bilateral filters are extended for face normal filtering, where a range kernel (Gaussian function) is defined based on the normal differences in the neighborhood (Yadav et al. 2019; Zheng et al. 2011). A variation of bilateral filtering is also used extensively in mesh denoising in order to remove noise and retain sharp features (Jones et al. 2003; Zhang et al. 2015).

1.3 Face Normal Filtering Versus Vertex Position Filtering

Broadly, surface smoothing algorithms can be divided into two categories, direct vertex position filtering, which is also known as one-stage smoothing and two-stage filtering, which includes (face) normal filtering and vertex position updates as described above.

Most of the one-stage denoising algorithms (vertex position filtering) follow the concept of mean curvature flow, which is related to the Laplace–Beltrami operator and the mean curvature on the surface as shown in Eq. (2) and as discussed above. Basically, noise components are removed by minimizing the mean curvature on the surface, where the mean curvature is computed using the area gradient on the surface. Therefore, minimizing the curvature will result in minimizing the area, which will lead to volume shrinkage. This applies to most of the anisotropic and isotropic diffusion-based surface smoothing algorithms. These methods use vertex position filtering in their minimization. To illustrate this problem, Fig. 1a shows a noisy model and Fig. 1b shows the result obtained by using the mean curvature flow-based method of (Hildebrandt and Polthier 2004). More precisely, Fig. 1b shows two different surfaces, the original surface (green) and the denoised one (yellow). The difference between these two surfaces is visible due to volume shrinkage during the minimization.

On the other hand, in two-stage surface denoising, noise removal is performed based on the face normals. Basically, face normals are treated as signals on the vertices of the dual graph of the mesh with values in the unit sphere. The face normal denoising is generally performed by rotating the face normals on the unit sphere according to the weighted average of the corresponding neighbor face normals (see Eq. (5) for a formalization). In other words, for noise removal, we operate in the dual space of the mesh and minimize the variation of face normals. This operation does not involve the curvature minimization on the vertex positions. Therefore, in two-stage surface denoising algorithms, volume shrinkage is minimal, as shown in Fig. 1c, d.

Furthermore, in two-stage surface denoising, noise removal can be performed also on vertex normals (Fleishman et al. 2003) instead of face normals. However, in terms of sharp feature preservation, vertex normal filtering will not be as effective as face normal filtering because of the following reasons:

1. The vertex normals of a mesh are usually derived from face normals. Therefore, processing face normals will avoid the ill-posedness and increase the robustness of the algorithm.
2. At a sharp feature, the angle between vertex normals is smaller than the angle between the face normals. Therefore, face normals are more robust in feature preservation compared to vertex normals.

As shown in Fig. 1c, d, face normal filtering better preserves sharp features compared to vertex normal filtering methods. However, in the context of point set surfaces, face normals are not available and denoising has to be performed using vertex normals.

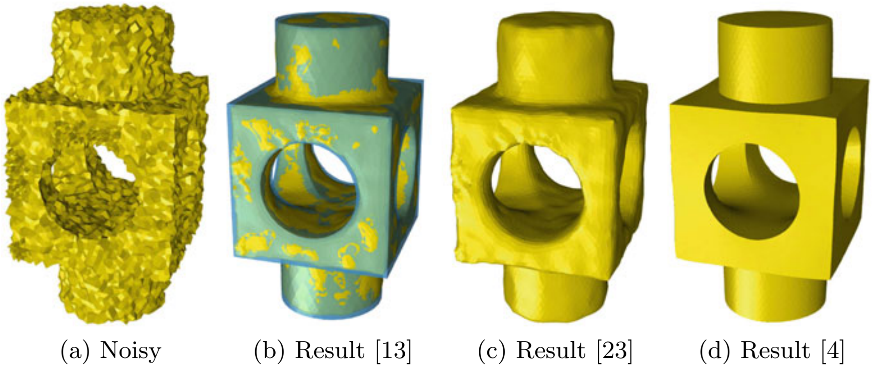

(a) Noisy (b) Result [13] (c) Result [23] (d) Result [4]

Fig. 1 A visual comparison between vertex position, vertex normal, and face normal filtering methods. **a** shows the noisy block model and **b** shows the denoised result of the method presented in Hildebrandt and Polthier (2004), based on mean curvature flow. More precisely, it shows two different surfaces, the original surface (green) and the denoised one (yellow). The difference between these two surfaces is visible due to volume shrinkage during the minimization. In contrast, **c**, **d** show the result of the face normal filtering methods (Fleishman et al. 2003) and (Yadav et al. 2018c), respectively, which do not suffer from volume shrinkage

1.4 Scope

From our discussion in the last section, it is clear that the two-stage surface denoising algorithms are robust and efficient in terms of noise removal and feature preservation. Therefore, in this paper, we will cover surface normal filtering (face normal in the context of mesh surfaces and vertex normals in the context of point set surfaces) in a robust statistics framework.

In the context of surface denoising, the most challenging task is to decouple sharp features from noise to treat them appropriately. Robust statistics is an efficient tool to identify the deviating substructures (*outliers*) from the bulk data. Here, we will treat features on the geometry as *outliers* because we want to deal with features differently compared to the non-feature areas. Based on this assumption, we derive relationships between different state-of-the-art methods for surface normal filtering using the concept of the robust error norm and its corresponding influence functions; see Sect. 2. We also discuss the robustness of these algorithms within the presented framework, see Sects. 3 and 4.

2 Robust Statistical Estimation

This paper is concerned with robust statistics handling outliers during statistical data modeling. The field of robust statistics has developed methods to handle outliers in the data modeling process, see (Mrázek et al. 2006). These methods describe the structure of best fitting the bulk of the data and identifying deviating substructures (outliers), see (Black and Rangarajan 1996). In this section, we translate the robust statistics framework to the setting of surface denoising. As explained above, surface denoising is a preprocessing operation in many geometry processing algorithms, which removes noise components and retains sharp features. In the robust statistics framework, surface features can be seen as outliers and methods from robust statics can identify these, which in turn can be treated differently for feature-preserving surface denoising, see (Yadav et al. 2019). As stated in the notation, we consider both a face and a vertex of the surface mesh to be a *feature*, respectively, if the corresponding normals of its neighbors have a high variation. Note that this is also the case for noisy faces and vertices, but not for outliers as they will not have a close neighborhood.

As reasoned in Sect. 1.4, we focus on two-stage mesh denoising algorithms. Recall that—as it is mentioned in Sect. 1.1—the surface $\mathcal{M}$ is corrupted by noise. Therefore, the vertices P and face normals N contain noise components, too. Let us first assume that the noise-free surface is represented by $\hat{\mathcal{M}}$ with $\hat{P}$ and $\hat{N}$ its vertices and face normals, respectively. The noisy and noise-free face normals can be related by:

$$n = \hat{n} + \eta, \tag{4}$$

where η is a random variable representing the noise corrupting the surface. If η is a zero-mean Gaussian random variable and the surface is flat, then the denoised face normals can be computed by minimizing the following L_2 error to compute the mean:

$$E(\hat{n}) = \sum_{n\in\Omega} \left\|\hat{n} - n\right\|^2, \qquad \hat{n} = \frac{1}{|\Omega|}\sum_{n\in\Omega} n. \tag{5}$$

However, in real-life scenarios, the noise η is not always normally distributed and surfaces have sharp features, which can be seen as outliers. Therefore, in the following we will aim at computing an approximation $\tilde{n}$ of $\hat{n}$. To deal with this complicated situation, we use robust error norms, which lead to the theory of M-estimators, see Sect. 2.1 for details. An M-estimator of a face normal from noisy normals can be obtained as the minimum of the following error functional:

$$E_\sigma(\tilde{n}) = \sum_{n\in\Omega} \rho_\sigma\left(\|\tilde{n} - n\|\right), \tag{6}$$

where $\rho_\sigma(\cdot) : \mathbb{R} \to \mathbb{R}$ is a loss function and commonly called ρ-function or error norm (Black and Rangarajan 1996; Black et al. 1998; Durand and Dorsey 2002) and the quantity σ is a user input. See Table 1 for different choices for ρ_σ. To minimize the effect of outliers, the loss function should not grow rapidly. To see the growing speed of the robust error norm $\rho_\sigma(\cdot)$, its derivative is computed, which is referred to as influence function ($\psi_\sigma(\cdot)$) in robust statics (Winkler et al. 1998). Thus, the loss function and influence function are related as follows

$$\rho'_\sigma(x) =: \psi_\sigma(x), \tag{7}$$

where for convenience, let us put $x := \|\tilde{n} - n\|$.

During mesh denoising, at sharp features, the effect of the influence function should be minimal. The input parameter x will be related to features, i.e., to the variation of normals. Therefore, when $x \to \infty$, the influence function should be zero, that is

$$\lim_{x\to\infty} \psi_\sigma(x) = 0.$$

In our setting, feature values (x) are basically defined by the variation of normals, which is measured by the differences between the neighboring normals n_j and the central normal n_i. However, these differences cannot approach infinity practically as $n_i, n_j \in \mathbb{S}^2$ for all $i, j \in I$. Therefore, the above equation indicates that for bigger values of x the influence function should be diminished.

Equation (6) can be extended to take into account spatial weights in local neighborhoods using the following formulation:

$$E_{\sigma,\sigma_d}(\tilde{n}) = \sum_{n\in\Omega} \rho_\sigma\left(\|\tilde{n} - n\|\right) f_{\sigma_d}(d), \tag{8}$$

where the function $f_{\sigma_d}(d)\colon \mathbb{R} \to \mathbb{R}$ is an isotropic weighting factor, which takes the spatial distance d between the considered geometry elements as the input argument and is responsible for smoothing out high-frequency components of the geometry. The term σ_d controls the width of the spatial kernel and generally depends on the resolution (sampling density) of the given geometry. In case of mesh denoising, the distance is computed between the centroid of neighboring faces and the processed central face. For point set denoising, the term d is computed between neighboring vertices and the processed central vertex.

Throughout the whole paper, concerning the error functionals, we are going to ignore constant factors in the arguments for both the isotropic (σ_d) and the anisotropic (σ) case. This is to focus on the qualitative differences between the presented methods rather than on smaller variations.

2.1 M-estimators

M-estimators are collections of different robust error norms to handle outliers. Any estimator defined by Eq. (6) is called an “M-estimator.” The name comes from the generalized maximum likelihood concept, which can be deduced from Eq. (6), when $-\rho_\sigma(x)$ is the likelihood function. Then, minimizing the energy $E_\sigma(\cdot)$ of Eq. (6) will be equivalent to the maximum likelihood estimate (Chu et al. 1998; Hampel et al. 2005). As motivated above, in general, the robust estimators should have the following two properties:

1. The error norm $\rho_\sigma(x)$ should not grow rapidly.
2. The influence function $\psi_\sigma(x) = \rho'_\sigma(x)$ should be bounded.

For an efficient mesh denoising procedure, the influence function should be a *re-descending function*, i.e., $\psi_\sigma(x) \to 0$ when $x \to \infty$. In this case, the corresponding error norm $\rho_\sigma(x)$ is called *re-descending influence error norm* (Hampel et al. 2005).

In general, surface normal (i.e., face and vertex normal) filtering is performed by computing weighted averages of neighboring normals; see Eq. (11). The weighting functions are vital for feature-preserving normal filtering, and they can be either linear or nonlinear. Here, we will formulate the relationship between weighting function, robust error norm, and the corresponding influence function.

From Eq. (3), we know that the anisotropic diffusion is controlled by an edge stopping function, which is represented by $g_\sigma(x)$. In this paper, we termed it as anisotropic weighting function. Equation (6) can be minimized using gradient descent to update the surface normal:

$$n^{t+1} = n^t + \lambda \nabla E_\sigma(x) = n^t + \lambda \sum_{n \in \Omega} \nabla \rho_\sigma(\|\tilde{n} - n\|), \tag{9}$$

where t is the iteration number and λ represents the step size. Here, ρ_σ is interpreted as a concatenation, taking the norm of a vector as argument, while the norm receives $(\tilde{n}) \in \mathbb{R}^3$ as argument. The complete function then maps from $\mathbb{R}^3$ to $\mathbb{R}$. The differentiation let us consider the gradient of ρ_σ as a natural generalization of the derivative in the one-dimensional case. Following the reasoning of Jones et al. (2003), also adapted by Zheng et al. (2011), we adapt the procedure introduced in Tomasi and Manduchi (1998) for signal processing to the context of mesh processing by feeding the normal distance x—as defined above—into the error norm ρ_σ and a spatial distance into the spatial weighting function f_σ. This analogy motivates us to analyze the following well-established relation from signal processing [consider for a specific derivation (Black and Rangarajan 1996, Sects. 4.1 and 5.3) and more generally (Hampel et al. 2005; Huber 1981),

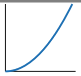
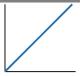
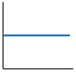
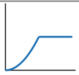
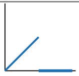
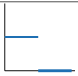
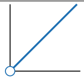
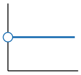
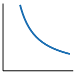
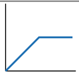
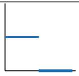
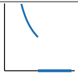
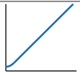
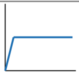
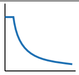
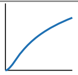
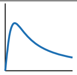
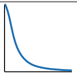
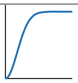
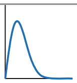
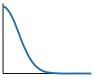
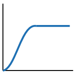
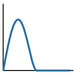
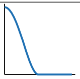
$$g_\sigma(x) = \frac{\rho'_\sigma(x)}{x} =: \frac{\psi_\sigma(x)}{x}. \tag{10}$$

Applications of this relation in image and geometry processing can be found in Jones et al. (2003), Black et al. (1998), Durand and Dorsey (2002).

The weighting function $g_\sigma(x)$ should capture the anisotropic behavior of the mesh or the point set, respectively, and should be chosen based on the above relations in the robust statistics framework. Table 1 consists of several well-known M-estimators with their robust error norms, their influence functions, and their corresponding anisotropic weighting functions.

Equation (5) shows an example of an estimator with a quadratic error norm ($\rho_\sigma(x) = x^2$). This norm grows rapidly, and its influence function ($\psi_\sigma(x) = 2x$) is unbounded (non-re-descending) as shown in Table 1. Therefore, the quadratic estimator is very sensitive to outliers and not useful in feature-preserving mesh denoising.

Table 1 M-estimators

Error norm $\rho_\sigma(x)$	Error norm $\rho_\sigma(x)$	Influence function $\psi_\sigma(x) = \rho'_\sigma(x)$	Weighting function $g_\sigma(x) = \frac{\psi_\sigma(x)}{x}$
◇ L_2-norm (Black et al. 1998), independent of σ, $\rho_\sigma(x) = x^2$			
◇ Truncated L_2-norm (Black and Rangarajan 1996) $\rho_\sigma(x) = \begin{cases} x^2 & \lvert x\rvert < \sqrt{\sigma} \\ \sigma & otrw. \end{cases}$			
◇ L_1-norm (Hampel et al. 2005), independent of σ, $\rho_\sigma(x) = \lvert x\rvert$			
◇ Truncated L_1-norm (Hampel et al. 2005) $\rho_\sigma(x) = \begin{cases} \lvert x\rvert & \lvert x\rvert < \sigma \\ \sigma & otrw. \end{cases}$			
◇ Huber's minimax (Huber 1981) $\rho_\sigma(x) = \begin{cases} \frac{x^2}{2\sigma} + \frac{\sigma}{2} & \lvert x\rvert < \sigma \\ \lvert x\rvert & otrw. \end{cases}$			
◇ Lorentzian-norm (Black et al. 1998) $\rho_\sigma(x) = \log\left[1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right]$			
◇ Gaussian norm (Black and Rangarajan 1996) $\rho_\sigma(x) = 1 - e^{\left(-\frac{x^2}{\sigma^2}\right)}$			
◇ Tukey's norm (Beaton and Tukey 1974) $\rho_\sigma(x) =$ $\begin{cases} \frac{x^2}{\sigma^2} - \frac{x^4}{\sigma^4} + \frac{x^6}{3\sigma^6} & \lvert x\rvert < \sigma \\ \frac{1}{3} & otrw. \end{cases}$			

The quadratic error norm can be truncated in order to convert it into a re-descending influence error norm. The second row of Table 1 shows the truncated quadratic error norm that has a re-descending influence function $\psi_\sigma(x)$ with a bounded error norm $\rho_\sigma(x)$. However, the behavior of $\psi_\sigma(x)$ is linearly increasing within the range of the user input σ, which is not desired for feature preservation.

As shown in Table 1, the L_1 error norm ($\rho_\sigma(x) = \lvert x\rvert$, third row) and Huber's minimax error norm (fifth row) do not have re-descending influence functions even though they are bounded by a nonzero constant value. These two perform better in terms of separating outliers compared to the (truncated) quadratic error norm.

The other error norms listed in Table 1, which include the truncated L_1 error norm as well as the Lorentzian, Gaussian, and Tukey's norms have re-descending influence functions. Among all re-descending influence error norms, the truncated L_1

and Tukey's error norm cut off the influence function's response strictly while the other norms have a nonzero influence function on a larger interval.

3 Face Normal Filtering in the Robust Statistics Framework

In this section, we will discuss state-of-the-art methods for face normal filtering utilizing the robust statistics framework and M-estimators as described above. Based on the relationship between the robust error norm, the influence function, and the weighting function as established in Eq. (10), we will discuss the robustness and effectiveness of state-of-the-art methods for removing noise and preserving features.

The face normals N of a triangulated mesh $\mathcal{M}$ can be seen as graph signals on the graph induced by the dual mesh of $\mathcal{M}$ with values in the unit sphere. The centroid of each face f_i is denoted by c_i, which can be treated as the vertex position on the dual mesh. In general, the filtered face normal $\tilde{n}_i$ corresponding to a noisy face normal n_i can be computed using the following equation:

$$\tilde{n}_i = \frac{1}{\omega} \sum_{j \in \Omega_i} g_\sigma \left(\left\| n_i - n_j \right\|^2 \right) f_{\sigma_d}(\left\| c_i - c_j \right\|^2) n_j, \tag{11}$$

where $\omega = \left\| \sum_{j \in \Omega_i} g_\sigma(\left\| n_i - n_j \right\|^2) f_{\sigma_d}(\left\| c_i - c_j \right\|^2) n_j \right\|$ ensures $\tilde{n}_i$ to be of unit length. The term Ω_i represents the mesh neighborhood around the ith triangle, which can be combinatorial or a geometrical disk of some (user-defined) radius. The above equation represents a general formula for face normal filtering and follows the error functional presented in Eq. (8). The efficiency of this approach heavily depends on the choice of the weighting functions $g_\sigma(\cdot)$ and $f_{\sigma_d}(\cdot)$.

In the following, we will present several state-of-the-art approaches for these choices. The listed algorithms use different input arguments for the robust error functionals. Common choices are the Euclidean distance of normals $\left\| n_i - n_j \right\|$, the angle between two normals $\angle(n_i, n_j)$, or the quantity $\arccos(n_i \cdot n_j)$. We will stick to the notation used in the respective original paper in the following discussion. However, note that these input arguments are related. In particular, we obtain

$$\cos(\angle(n_i, n_j)) = \frac{n_i \cdot n_j}{\|n_i\| \left\| n_j \right\|} = n_i \cdot n_j \Rightarrow \angle(n_i, n_j) = \arccos(n_i \cdot n_j).$$

by the Euclidean scalar product because all normals considered are of unit length. Furthermore, (by the law of cosines) it is

$$\begin{aligned}\left\|n_i - n_j\right\|^2 &= \|n_i\|^2 + \left\|n_j\right\|^2 - 2 \cdot \|n_i\| \cdot \left\|n_j\right\| \cdot \cos(\angle(n_i, n_j))\\ &= 2 - 2\cos(\angle(n_i, n_j))\\ \Rightarrow \angle(n_i, n_j) &= \arccos\left(1 - \frac{\left\|n_i - n_j\right\|^2}{2}\right).\end{aligned}$$

3.1 Unilateral Normal Filtering

Unilateral normal filtering performs noise removal from noisy normals using a single anisotropic kernel function. From our setup in Eq. (8), it is clear that the unilateral normal filtering algorithms are using $g_\sigma(x)$ as anisotropic weighting function while the spatial filter will be equal to one, i.e., $f_{\sigma_d}(d) \equiv 1$. These methods are effective against low intensity of noise and enhance sharp features. However, they are not robust against moderate or high levels of noise because of the unavailability of the spatial filter $f_{\sigma_d}(d)$.

(a) Belyaev and Ohtake (2001) introduce nonlinear diffusion of face normals to enhance the features of the geometry. Their algorithm uses the following weighting function:

$$g_\sigma(x) = \exp\left(-\frac{x^2}{\sigma^2}\right). \tag{12}$$

This weight is a nonlinear function, and the input argument is encoding the directional curvature. It is given as

$$x = \frac{\angle(n_i, n_j)}{d},$$

where $\angle(n_i, n_j)$ denotes the angle between n_i and n_j, the term $d = \left\|c_i - c_j\right\|$ represents the distance between the centroids (as presented above) of the central face and its neighboring face, and $n_i, n_j \in N$ are faced normals of the central face and its neighboring face, respectively. The term σ is a user input to better adapt the algorithm to the given geometry. It is chosen based on the amount of noise, curvature, and the resolution of the geometry. The directional curvature x measures the similarity between neighboring normals. In the robust statistics framework, by using Eq. (10), we can deduce the used error norm as

$$\rho_\sigma(x) = \int_0^x x' g_\sigma(x') dx' = \frac{\sigma^2}{2}\left(1 - \exp\left(-\frac{x^2}{\sigma^2}\right)\right). \tag{13}$$

Similarly, the influence function can be derived as

$$\psi_\sigma(x) = x g_\sigma(x) = x \exp\left(-\frac{x^2}{\sigma^2}\right), \qquad \lim_{x\to\infty} \psi_\sigma(x) = 0. \tag{14}$$

The above two equations indicate that this algorithm applies the Gaussian error norm (second last row of Table 1), which has a re-descending influence function and makes the algorithm robust against outliers. However, the spatial smoothing function $f_{\sigma_d}(\cdot)$ is not used in this algorithm, which reduces the robustness of the algorithm against significant noise.

(b) Yagou et al. (2002) apply mean and median filtering to face normals. Mean filtering of normals is performed by simply uniformly averaging neighboring normals. Therefore, the anisotropic weighting function $g_\sigma(x) \equiv 1$ leads to an error norm and influence function of

$$\rho_\sigma(x) = \int_0^x x' g_\sigma(x') dx' = x^2 \qquad \text{and} \qquad \psi_\sigma(x) = x g_\sigma(x) = x, \tag{15}$$

respectively. From the equation above, it is clear that mean filtering follows the quadratic error norm ($\rho_\sigma(x) = x^2$, $g_\sigma(x) = 1$) (the first row in Table 1) and it has an unbounded influence function ($\lim_{x\to\infty} \psi_\sigma(x) = \infty$), which makes the algorithm sensitive to outliers and produces feature blurring. This method uses the triangle area as a weighting function, i.e., in the notation of Eq. (8), it computes $f_{\sigma_d}(d)$ for a given face f_i as area(f_i). However, this makes the algorithm only insensitive to irregular sampling.

On the other hand, median filtering is estimated using the L_1 error norm (Hampel et al. 2005). Therefore, the corresponding error norm and influence function can be derived as

$$\rho_\sigma(x) = |x| \qquad \text{and} \qquad \psi_\sigma(x) = \rho'_\sigma(x) = \begin{cases} 1 & |x| \neq 0 \\ \text{undefined} & x = 0. \end{cases} \tag{16}$$

By using the relation from Eq. (10), the anisotropic weighting function can be written as

$$g_\sigma(x) = \frac{\psi_\sigma(x)}{x} = \begin{cases} \frac{1}{|x|} & |x| \neq 0 \\ \text{undefined} & x = 0. \end{cases} \tag{17}$$

In this algorithm, the input x is given by the Euclidean distance of the neighboring normal $n_j \in N$ to the central normal n_i, i.e., $x = \left\| n_i - n_j \right\|$. The L_1-norm is better compared to the quadratic error norm in terms of robustness to outliers. However, the corresponding influence function is not re-descending (see Table 1) and produces a constant value for outliers.

Weighted median filtering is applying a spatial weighting function to provide higher weights to closer points compared to distant points; see (Yagou et al. 2002). This weighting function is truncating the effect of local neighboring faces. Therefore, the weighted median follows a truncated L_1-norm and its corresponding influence function can be derived as

$$\psi_\sigma(x) = \rho'_\sigma(x) = \begin{cases} 0 & |x| < \sigma \\ \text{sign}(x) & 0 < |x| \leq \sigma \\ \text{undefined} & x = 0 \end{cases}. \tag{18}$$

By using the relation from Eq. (10), the anisotropic weighting function can be written as

$$g_\sigma(x) = \frac{\psi_\sigma(x)}{x} = \begin{cases} 0 & |x| < \sigma \\ \frac{\text{sign}(x)}{x} & 0 < |x| \leq \sigma \\ \text{undefined} & x = 0 \end{cases}. \tag{19}$$

The truncated L_1-norm has a re-descending influence function, which enhances the feature preservation capability of the algorithm compared to mean and median filtering.

From the influence functions of the L_1-norm and the truncated L_1-norm, it is clear that these norms are capable of feature preservation during the process of face normal filtering. However, these influence functions and their corresponding anisotropic weighting functions are not well-defined at $x = 0$, which is not desirable.

(c) Huber (1981) proposes a slight modification of the weighting function before mentioned to overcome the issue of not being well-defined at $x = 0$. He suggests

$$\rho_\sigma(x) = \begin{cases} \frac{x^2}{2\sigma} + \frac{\sigma}{2} & |x| < \sigma \\ |x| & \text{otrw.} \end{cases}. \tag{20}$$

This modified error norm is commonly known as Huber's minimax norm (see fifth row in Table 1). The corresponding influence and anisotropic weighting functions can be derived as

$$\psi_\sigma(x) = \begin{cases} \frac{x}{\sigma} & |x| < \sigma \\ \text{sign}(x) & \text{otrw.} \end{cases}, \qquad g_\sigma(x) = \begin{cases} \frac{1}{\sigma} & |x| < \sigma \\ \frac{\text{sign}(x)}{x} & \text{otrw.} \end{cases}. \tag{21}$$

The above equation indicates that Huber's minimax norm has a re-descending influence function and has a well-defined anisotropic weighting function. This norm is widely used in image processing applications but has—to the best of our knowledge—not been used for face normal filtering yet and is therefore not included in Table 2.

(d) Yadav et al. (2018c) introduced a face normal filtering technique using a box filter as the anisotropic weighting function

$$g_\sigma(x) = \begin{cases} 1 & |x| < \sigma \\ 0.1 & \text{otrw.} \end{cases}, \qquad \text{with} \qquad x = \angle(n_i, n_j), \tag{22}$$

where $\angle(n_i, n_j)$ denotes the angle between the central normal n_i and it neighboring normal n_j. The corresponding error norm and influence function can be derived as

$$\rho_\sigma(x) = \int_0^x x' g_\sigma(x') dx' = \begin{cases} x^2 & |x| < \sigma \\ 0.1(x^2 + 9\sigma^2) & \text{otrw.} \end{cases}, \tag{23}$$

$$\psi_\sigma(x) = x g_\sigma(x) = \begin{cases} x & |x| < \sigma \\ 0.1x & \text{otrw.} \end{cases}. \tag{24}$$

From the above error norm and influence function, we can see that this filtering is using an error norm quite similar to the truncated quadratic error norm (see second row in Table 1) for the computation of the element-based normal voting tensor. The corresponding influence function is neither bounded nor re-descending, but the outlier effect will be quite minimal. This is because of the downscaling of the argument in the influence function for bigger x. Therefore, the algorithm is able to preserve sharp features. However, it is less robust against high noise intensities because of the non-re-descending and unbounded influence function.

(e) Shen et al. (2004) introduced the fuzzy vector median-based surface smoothing algorithm, which is quite similar to the algorithm of (Belyaev and Ohtake 2001) (explained in paragraph **a)** in the beginning of this section). The anisotropic weighting function $g_\sigma(x)$ is a Gaussian function as given in Eq. (12) and the input x is given as

$$x = \left\| n_j - n_{vd} \right\|,$$

where n_j represents neighboring normals to the processed central face f_i and the term n_{vd} performs *vector directional median filtering* on the normal vectors including the central normal n_i. Vector directional median filtering is an extension of *median filtering* for multivariate data, see (Trahanias and Venetsanopoulos 1993), and can be computed as

$$n_{vd} = \underset{n}{\operatorname{argmin}} \sum_{j \in \Omega_{vd}} \angle(n, n_j), \tag{25}$$

where $\angle(n, n_j)$ denotes the angle between n and n_j and the set $\Omega_{vd} = \Omega_i \cup \{i\}$ consists of indices of the neighbor normals n_j together with the index i of the central normal n_i.

The corresponding influence function will be re-descending as shown in Eqs. (13) and (14). The input argument of $g_\sigma(x)$ is the Euclidean difference between the neighboring normals and their median. This method performs well in terms of feature preservation but is not robust during noise removal because of the unavailability of the spatial filter. As it is clear from Eqs. (3), the anisotropic weighting function $g_\sigma(x)$ is similar to the edge stopping function in the diffusion process.

(f) Tasdizen et al. (2002) apply—based on the relationship between bilateral filtering and nonlinear diffusion (Barash 2002)—the diffusion of face normals for filtering by using the Gaussian function as anisotropic weighting function. Curvature information is used as input x in this algorithm. Similar to the method of (Belyaev and Ohtake 2001), from Eqs. (12), (13), and (14), it can be derived that this method also follows the Gaussian error norm and has a bounded, re-descending influence function, which helps preserving sharp features. However, due to the unavailability of the spatial filter, this algorithm is not robust against significant noise.

(g) Centin et al. (2018) also introduce a face normal diffusion method using the following anisotropic weighting function

$$g_\sigma(x) = \begin{cases} 1 & |x| < \sigma \\ \frac{\sigma^2}{(\sigma - x)^2 + \sigma^2} & \text{otrw.} \end{cases}, \qquad \text{where} \qquad x = \kappa \cdot \ell_{avg}. \tag{26}$$

The term κ represents curvature information computed at each face by averaging the curvature at the corresponding vertices and ℓ_{avg} represents the average edge length computed over the entire geometry. The corresponding influence function can be derived as

$$\psi_\sigma(x) = x g_\sigma(x) = \begin{cases} x & |x| < \sigma \\ \frac{x\sigma^2}{(\sigma - x)^2 + \sigma^2} & \text{otrw.} \end{cases}. \tag{27}$$

The above influence function is bounded and re-descending, which makes this algorithm effective in terms of feature preservation. This method falls somewhere between the Lorentzian error norm (decaying of $g_\sigma(x)$ for $x \geq \sigma$) and Huber's minimax error norm (constant $g_\sigma(x)$ for $x < \sigma$). Due to the absence of a spatial filter, this algorithm is not robust against high intensities of noise.

3.2 Bilateral Normal Filtering

Bilateral normal filtering is one of the most effective and robust approaches for denoising of normals. In contrast to unilateral normal filtering, the weighting function in bilateral normal filtering consists of two different Gaussian kernels. As above, one kernel carries the anisotropic nature and is commonly known as range filter (we termed it anisotropic weighting function $g_\sigma(x)$) while the other kernel is known as spatial kernel (given as $f_{\sigma_d}(d)$ in Eq. (8)) and is isotropic in nature.

(a) Zheng et al. (2011) define these kernels as:

$$g_\sigma(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \qquad \text{and} \qquad f_{\sigma_d}(d) = \exp\left(-\frac{d^2}{2\sigma_d^2}\right), \tag{28}$$

where σ_d is the average distance between neighboring faces and the central face. The input arguments x and d are defined as:

$$x = \left\| n_i - n_j \right\| \qquad \text{and} \qquad d = \left\| c_i - c_j \right\|,$$

where c_i and c_j are the centroids of the central face f_i and the neighboring face f_j, respectively.

In the robust statistics framework, our main focus is the anisotropic weighting function $g_\sigma(x)$, its corresponding error norm, and the corresponding influence function because $g_\sigma(x)$ is responsible for feature preservation. From Eqs. (12), (13), and (14), it is clear that the method of (Zheng et al. 2011) has a re-descending influence function (second last row of Table 1). Thereby, this algorithm is capable of preserving sharp features effectively and removes noise better compared to the algorithms mentioned above because of the utilized spatial filter $f_{\sigma_d}(d)$.

(b) Zhang et al. (2015) describe a procedure of guided mesh normal filtering following the Gaussian error norm and uses the same spatial filter as the method of (Zheng et al. 2011) presented above. The guided mesh normal is based on a joint bilateral filter, where an anisotropic weighting function (range kernel) works on the guidance signal. That is, the input variable x is defined as:

$$x = \left\| G_i - G_j \right\|, \tag{29}$$

where G_i and G_j are the guidance normals, which are computed by averaging similar normals in the respective neighborhood.

(c) Yadav et al. (2019) introduce a bilateral normal filtering using the following anisotropic weighting function:

$$g_\sigma(x) = \begin{cases} \frac{1}{2}\left[1 - \left(\frac{x}{\sigma}\right)^2\right]^2 & |x| \leq \sigma \\ 0 & \text{otrw.} \end{cases}, \qquad \text{where} \qquad x = \left\| n_i - n_j \right\|. \tag{30}$$

The above function is known as Tukey's biweight function (Beaton and Tukey 1974). The spatial filter $f_{\sigma_d}(d)$ is a Gaussian function similar to that used in the method of (Zheng et al. 2011) as described above. In the robust statistics framework, the corresponding influence function and error norm can be derived as

$$\psi_\sigma(x) = x g_\sigma(x) = \begin{cases} \frac{x}{2}\left[1 - \left(\frac{x}{\sigma}\right)^2\right]^2 & |x| < \sigma \\ 0 & \text{otrw.} \end{cases}, \tag{31}$$

$$\rho_\sigma(x) = \int_0^x x' g_\sigma(x') dx' = \begin{cases} \frac{x^2}{\sigma^2} - \frac{x^4}{\sigma^4} + \frac{x^6}{3\sigma^6} & |x| < \sigma \\ \frac{1}{3} & \text{otrw.} \end{cases}. \tag{32}$$

From the influence function and error norm, it is clear that Tukey's biweight function is more robust compared to the Gaussian function in terms of feature preservation because it strictly cuts off outliers with respect to the user-chosen parameter σ. Also, the Gaussian spatial filter helps to remove noise components effectively.

4 Point Set Surface Denoising in the Robust Statistics Framework

In this section, we will shift our focus slightly. Instead of an input mesh $\mathcal{M}$, we will now consider a point set sample (PSS) of a surface as input. Thus, we are only given vertices $P = \{p_i\}_{i \in I} \subseteq \mathbb{R}^3$ with corresponding normals $N = \{n_i\}_{i \in I}$, i.e., compared to the above we cannot use edges to induce connectivity between the vertices nor can we use the area of faces as weighting terms in the filtering process.

Despite these challenges, a multitude of procedures and algorithms has been proposed for the denoising of PSS. This is mostly due to two advantages of PSS over meshes. First, point sets are often the raw output of 3D acquisition devices and processes. Thus, if an algorithm is available to work on a PSS, it can be directly—possibly even on site—applied to the acquired data. Second, as there is no connectivity information in the point set, no such data has to be stored, which amounts to significantly lower storage costs compared to meshes. Furthermore, no topological problems—like non-manifold edges or fold-overs—and no numerical problems—like slivers—are introduced as the PSS only gives an implicit handle on the underlying surface geometry.

In the following, we will focus on adaptations of face normal filtering algorithms from meshes to point sets as well as on original methods proposed directly in the PSS setting. Note that any method on point sets can easily be applied to the meshed setting by simply disregarding the edge and face connectivity information.

4.1 Unilateral Normal Filtering

As for meshes, we will first focus on unilateral normal filtering procedures. These do not use a specific spatial filter, i.e., $f_{\sigma_d}(d) \equiv 1$. This makes them less robust against moderate or high levels of noise.

(a) Öztireli et al. (2009) introduced a modification of the moving least squares (MLS) procedure (Alexa et al. 2003) aiming at the integration of feature preservation into the MLS pipeline. Their core objective is an iterative minimization and can be understood as iterative trilateral filtering, as it makes use of three types of weights. The first one is isotropic in nature and appears as $\mathcal{C}^3$ continuous polynomial approximation of the Gaussian, i.e.,

$$f_i(p) = \left(1 - \frac{\|p - p_i\|^2}{h_i^2}\right)^4 \tag{33}$$

where the argument p is some point (not necessarily from P), as the objective is an implicit, signed distance function. The value h_i is a weight adapting the local density, chosen within a range from 1.4 to 4 as experimentally evaluated by the authors (Öztireli et al. 2009). For the second weighting term—using the height over an estimated hyperplane at p and thus capturing both isotropic and anisotropic quantities—the authors discuss M-estimators and include the Gaussian error norm and its respective Gaussian error weight, see Eq. (12), into their optimization problem. The arguments are

$$d = y_i - \tilde{\eta}^{k-1}(p_i) \qquad \text{and} \qquad \sigma_d = \frac{h_i}{2},$$

with y_i the heights of the samples p_i taken over the local least-squared best fitting hyperplane, and $\tilde{\eta}^{k-1}$ the corresponding local approximation. The value for σ_d is set fix throughout the whole paper by the authors. The third and final weighting terms are anisotropic and make use of a Gaussian function with arguments

$$x = \left\| \nabla \eta^k(p) - n_i \right\| \qquad \text{and} \qquad \sigma \in \mathbb{R},$$

where η is an implicit, signed distance function as main objective, p some point at which we want to evaluate the function η, n_i the normal at sample point p_i, and σ a parameter that regulates the sharpness where typical choices range from 0.5 up to 1.5. This last weighting term penalizes the deviation of normals when we reach sharp features. The influence function and error norm are of Gaussian nature and are derived in Eqs. (14) and (13). The assembled combination yields a robust implicit surface definition via MLS, which can represent both smooth surface patches and sharp features and was coined robust implicit MLS (RIMLS). Similar to Method (Belyaev and Ohtake 2001), this algorithm is capable of retaining and enhancing sharp features. However, the unavailability of a spatial filter $f_{\sigma_d}(d)$ makes the algorithm less effective against moderate and high levels of noise.

(b) Mattei and Castrodad (2016) start their paper with the assertion that the principal component analysis (PCA) operation for the estimation of local reference planes is not robust. They proceed to construct a moving robust PCA (MRPCA). Their main ingredient of interest in the given context is a minimization problem, which makes use of anisotropic weights determined via the Gaussian weight function as given in Eq. (12) with arguments

$$x = \arccos(n_i \cdot n_j) \qquad \text{and} \qquad \sigma \in \mathbb{R},$$

where n_i, n_j are the unit normals at the considered point p_i and at one of its neighbors p_j (with a k-nearest neighborhood utilized). Furthermore, σ is a bandwidth parameter affecting the reconstruction of sharp features. The authors propose values of $\sigma \in (\pi/12, \pi/6)$. Using this anisotropic weight function yields the Gaussian error norm along with its re-descending influence function as given in Eqs. (14) and (13). Similar to (Belyaev and Ohtake 2001), this algorithm is capable of retaining and enhancing sharp features. However, the unavailability of a spatial filter $f_{\sigma_d}(d)$ makes the algorithm less effective against moderate and high levels of noise.

4.2 Bilateral Normal Filtering

We will now turn to bilateral normal filtering procedures for PSS. These use two different weighting kernels. As for meshes, one kernel carries the anisotropic nature while the other one of isotropic behavior.

(a) Li et al. (2009) presented one of the first approaches applying bilateral filtering to PSS. The authors first estimate the likelihood ℓ_i that a given sample point $p_i \in P$ is close to the underlying surface geometry. They propose to compute ℓ_i based on the MLS technique of (Alexa et al. 2003). The normal denoising utilizes the bilateral filtering scheme, which includes a Gaussian weighting (following Eq. (12)) as a spatial filter $f_{\sigma_d}(d)$ with the following input arguments in the isotropic setting

$$d = \left\| p_i - p_j \right\| \qquad \text{and} \qquad \sigma_d = \frac{r}{2},$$

and another Gaussian weighting function $g_\sigma(x)$ in the anisotropic setting with following input arguments

$$x = \arccos(n_i \cdot n_j) \qquad \text{and} \qquad \sigma \in \mathbb{R},$$

the latter chosen to be the standard deviation of the normal variation given in x. Here, r is the radius of the enclosing sphere of the geometric neighborhood Ω_i. Observe that the values presented here differ from those given in (Li 2009), because we adjust them to fit the Gaussian given in Eq. (12). Lastly, the closeness of the point p_i to the underlying surface, measured by ℓ_i, the feature intensity, and the bilateral filtering for normals are used in a final sample point filtering step to remove noise from the PSS. The mentioned method follows the Gaussian error norm similar to the bilateral normal filtering of (Zheng et al. 2011). As shown in Eq. (14), the applied anisotropic weighting function $g_\sigma(x)$ has a re-descending and bounded influence function, which makes the algorithm robust in terms of feature preservation and also the availability of the spatial filter $f_{\sigma_d}(d)$ ensures the effectiveness toward different levels of noise.

(b) Zheng et al. (2017) proposed a four-stage method for point set denoising. It consists of sharp feature detection, multiple normals computation, guided normal filtering, and point updating. Concerning the feature detection, the authors provide a two-step procedure: feature candidate detection and feature point selection. The former is to find the global feature structure and utilizes the framework of robust statistics. Namely, after a first computation of normals using PCA, the normal similarity is evaluated via the Gaussian weight function, see Eq. (12), with arguments

$$x = \left\| n_i - n_j \right\| \qquad \text{and} \qquad \sigma \in \mathbb{R},$$

with a user-given angle-threshold σ, which ranges from 0.05 to 0.3 in the experiments of the authors, n_i the normal at the considered point and n_j the normal at one of its neighbors, while using the k-nearest neighbors as neighborhood notion. In contrast to the single normal used in the normal similarity described above, the authors of (Zheng et al. 2017) attach bundles—a multitude of normals—to every point. A comparable approach is then chosen to estimate averaged normals utilizing spatial weights evaluated once more via the Gaussian weight function (12) with arguments

$$d = \|p_i - p_j\| \qquad \text{and} \qquad \sigma_d \in \mathbb{R},$$

with σ_d ranging from 0.1 to 0.5 in the authors' experiments. Finally, both weightings are combined in the actual bilateral normal filtering. This method is an extension of guided mesh normal filtering (Zhang et al. 2015), which we have mentioned in Eq. (29). From the explanation for guided mesh normal filtering in Sect. 3.2, it is clear that this method also follows the Gaussian error norm along with a bounded and re-descending influence function and has similar robustness in terms of feature preservation and noise removal. The computation of guided normals makes this algorithm slightly better compared to bilateral normal filtering.

(c) Park et al. (2013) proposed a three-staged point set filtering approach including feature detection, normal re-calculation, and a point position update. Their feature detection tensor, adaptive sub-neighborhood, and point update all use the Gaussian weighting function given in Eq. (12), where for the first two, the arguments are of anisotropic nature given as

$$x = \sqrt{s^2 + c\kappa^2} \qquad \text{and} \qquad \sigma \in \mathbb{R},$$

with a prescribed constant c, σ set by the authors to the neighborhood range, which is 4δ with δ the arithmetic mean of all distances of the points to their closest neighbors respectively. The value s represents the arc-length on the tangent plane and κ the curvature obtained by the circle, which goes through both the center point p_i and its considered neighbor p_j and which is also tangent to the attached normals n_i and n_j. These normals are calculated via an initial normal estimation following (Hoppe et al. 1992). To compute the feature detection tensor, the method uses a Gaussian function as the anisotropic weighting, which has a re-descending influence function ψ_σ and a derived Gaussian error norm ρ_σ as given in Eqs. (14) and (13), respectively. In terms of feature sensitivity, it will be as effective as MRPCA. However, this algorithm is not robust against moderate and high levels of noise.

(d) Digne and de Francis (2017) proposed an extension of the bilateral filtering on meshes to points via a parallel implementation of (Fleishman et al. 2003) using points. The whole procedure consists of a point update using non-oriented normals and utilizes Gaussian weights, Equation (12), twice, with isotropic

$$d = \|p_i - p_j\| \qquad \text{and} \qquad \sigma_d = \frac{1}{3}r,$$

and anisotropic arguments

$$x = \left|n_i \cdot (p_j - p_i)\right| \qquad \text{and} \qquad \sigma = \frac{1}{3}r',$$

with user-given radii r and r'. If these are not given, the authors use a heuristic and set $r = \ell\sqrt{20/|P|}$, where ℓ denotes the size of the bounding box and $|P|$ the number of vertices. The values σ_d and σ are set to be equal in this case. The point p_i is the one considered to be updated and p_j represents one of its neighbors within a geometrical neighborhood Ω_i. The weights determined by f_{σ_d} measure the spatial distance, and those by g_σ evaluate the distance of neighbors to the plane spanned by the point p_i and its normal. As the weights are of Gaussian nature, we can derive the influence function and Gaussian error norm given in Eqs. (14) and (13). In terms of feature preservation and noise removal, this algorithm will be as effective as bilateral normal filtering (Zheng et al. 2011) as both of them are using same robust error norm with a slightly different input argument.

(e) Zheng et al. (2018) propose an iterative two-staged denoising algorithm which—in contrast to most methods—smooths out smaller features while preserving larger ones. The iterative normal filtering (with initial normals obtained via (Hoppe et al. 1992) and the following point position update (solved iteratively via gradient descent) make use of the Gaussian weighting, Equation (12), with the isotropic arguments

$$d = \left\|p_i - p_j\right\| \qquad \text{and} \qquad \sigma_d \in \mathbb{R}$$

and the anisotropic arguments

$$x = \left\|n_i - n_j\right\| \qquad \text{and} \qquad \sigma \in \mathbb{R},$$

where $\sigma_d \in [0.01, 0.5]$ and $\sigma \in [0.1, 0.5]$ given in the authors' experiments, p_i the considered point, p_j representing its neighbor (k-nearest neighbors are used), and n_i, n_j the respective normals. Consequently, the evaluation is similar and on the one hand uses spatial distances of points while on the other hand using closeness of normals. The used Gaussian weights yield the influence function and Gaussian error norm given in Eqs. (14) and (13), which make this algorithm robust in terms of feature preservation and noise removal. One of the key benefits of this algorithm is that by adjusting the parameter σ, different levels of features can be smoothed out effectively. An even more robust version, utilizing the same weighting terms as given above, is discussed in (Yangxing et al. 2019).

(f) Yadav et al. (2018a) offer an extension of (Yadav et al. 2018c) to point sets. The proposed iterative scheme consists of the following three stages: normal filtering, feature detection, and vertex update. The first two make use of a similar box filter as given in Eq. (22), here given as

$$g_\sigma(x) = \begin{cases} 1 & x \leq \sigma \\ 0 & \text{otrw.} \end{cases}$$

with input arguments

$$x = \arccos(n_i \cdot n_j) \qquad \text{and} \qquad \sigma \in \mathbb{R},$$

where n_i, n_j are unit-length normals and σ is an angle threshold for the neighbor selection (chosen by the user). The deviation from the weighting defined in (Yadav et al. 2018c) is because vertex normals are more sensitive to noise compared to face normals. Similar to the influence function and error norm derived in Eqs. (24) and (23), the anisotropic weights given above yield an influence function of

$$\psi_\sigma(x) = x g_\sigma(x) = \begin{cases} x & |x| < \sigma \\ 0 & \text{otrw.} \end{cases}$$

and an error norm of

$$\rho_\sigma(x) = \int_0^x x' g_\sigma(x') dx' = \begin{cases} x^2 & |x| < \sigma \\ 0 & \text{otrw.} \end{cases} .$$

The latter is a version of the truncated quadratic error norm, see the second row of Table 1. In contrast to (Yadav et al. 2018c), the influence function is both bounded and re-descending ($\psi \to 0$ when $x \to \infty$). The impact of outliers is therefore kept small as it scales down for larger arguments x and feature preservation is yielded. However, the performance of this algorithm is not optimal in the presence of moderate and high levels of noise due to the unavailability of a spatial filter $f_{\sigma_d}(d)$.

Discussion: Local versus Global Weighting Note that out of the methods for point set surface denoising presented here, only (Öztireli et al. 2009) utilizes a local vertex-based weight σ_d. In contrast, methods (Li 2009; Zheng et al. 2017; Digne and de Franchis 2017; Zheng et al. 2018) use global weighting terms σ_d. While localized terms can capture features on a finer level, they are harder to calibrate than global parameters. Furthermore, an implicit assumption of many algorithms is a noisy but uniformly dense sampling as input. Handling non-uniform densities requires additional work, see (Skrodzki et al. 2018). Finally, if the features of the input geometry are of comparable size, a global parameter is sufficient to capture them while still removing noise. Hence, most algorithms reduce to simple global parameters.

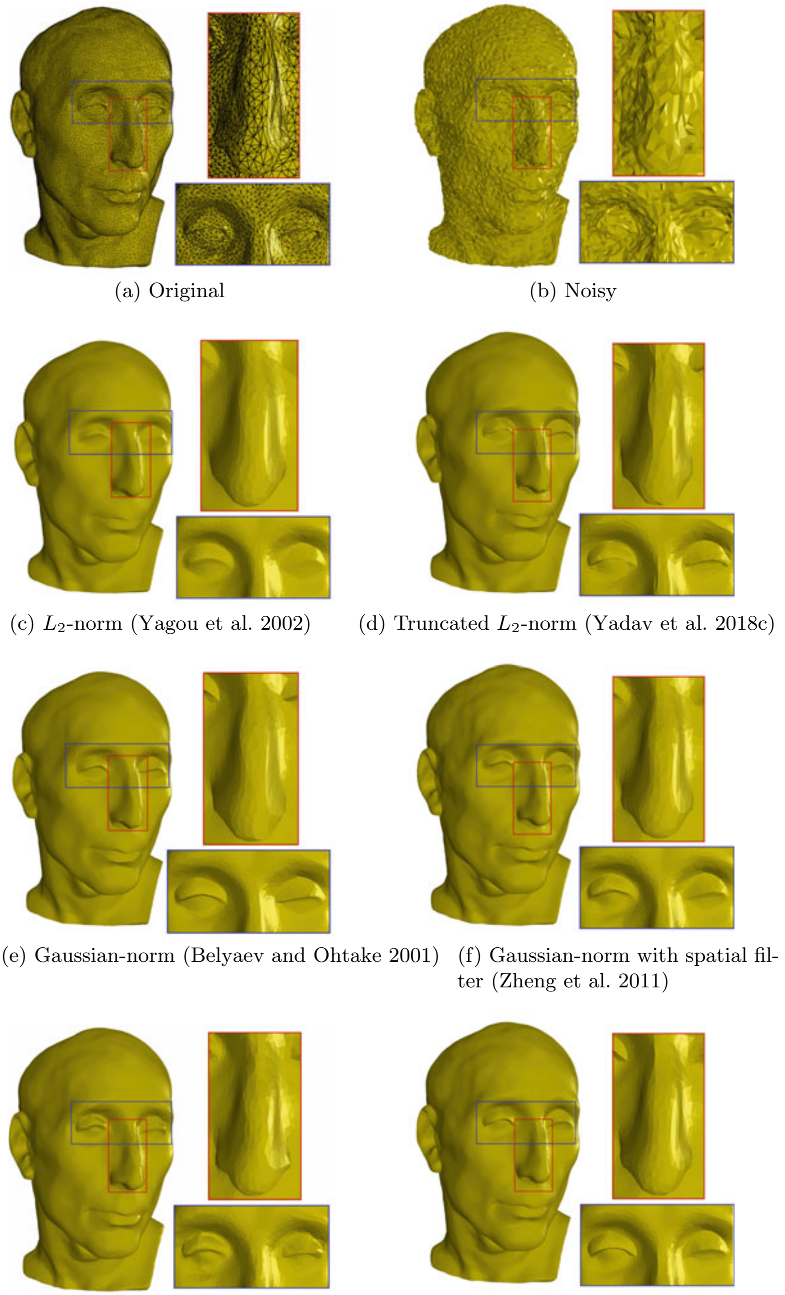
5 Experiments and Results

In this section, we present experimental results regarding the state-of-the-art methods as listed in the previous sections, which are using different robust error norms. We have chosen two different models (CAD and CAGD) with different levels of noise. Figure 2 shows the Nicola model corrupted with a moderate level of Gaussian noise (standard deviation $\sigma_n = 0.2\ell_e$, where ℓ_e is the average edge length). Using this model, we show the capability of feature preservation with the usage of different error norms. As shown in Fig. 2, the L_2-norm is not effective in terms of feature preservation (blurred eye region) because of the linear influence function and also as it is not bounded. The truncated L_2-norm preserves features in the eye region better compared to the L_2-norm as it has a truncated linear influence function. Figure 2e, f shows the outputs of using the Gaussian norm without and with spatial filter, respectively. The Gaussian error norm has a re-descending influence function, which makes the algorithm more effective compared to the L_2 related norms. The spatial filter is helping to remove noise effectively (eye and nose regions). Huber's minimax (Fig. 2e) and the Gaussian error norm (Fig. 2g) have quite similar outputs as they have re-descending influence functions and do not use spatial filters. Figure 2h shows the output of using Tukey's error norm, which has a sharper cut-off in the influence function compared to the Gaussian error norm. Therefore, feature preservation is better compared to other norms mentioned and the spatial filter is helping to remove noise components effectively.

Figure 3 shows the robustness of the mentioned norm against high level of noise. The Fandisk model is corrupted with a Gaussian noise ($\sigma_n = 0.3\ell_e$) in random direction. As it is shown, L_2 and Huber's minimax norms are able to remove the noise components effectively but feature preservation is not effective. In case of the Gaussian error norm, the spatial filter removes different components of noise including low-frequency ripples. However, the truncated L_2-norm is able to remove low-frequency components by introducing an additional processing step (binary optimization) in the pipeline. The algorithm (Yadav et al. 2019) uses Tukey's error norm, which helps to preserve features effectively and the spatial filter removes the noise components.

6 Conclusion

In this paper, we unified state-of-the-art methods for normal filtering in surface denoising using the robust statistics framework. We discussed different M-estimators, which are the main tools of robust statistics. These tools are defined by a robust error norm and a corresponding influence function, respectively. Based on the properties of the influence function (bounded and re-descending) and of the anisotropic weighting function, we discussed the robustness of state-of-the-art methods in terms of feature preservation and feature enhancement (see Table 2). Furthermore, we have shown that the introduction of spatial filters along with anisotropic filters will improve the robustness of the algorithm in terms of noise removal. The robust statistics framework not only provides a platform to bring new insight into the field of surface denoising and clarify the relation between different methods in the field. It can also be used for new methods to combine the advantages of the known filtering techniques. The application of robust statistics is not limited to surface denoising, and it can be used effectively in other areas of the field of geometry processing. Corresponding applications of this powerful tool are left as further research.



(a) Original

(b) Noisy

(c) L_2-norm (Yagou et al. 2002)

(d) Truncated L_2-norm (Yadav et al. 2018c)

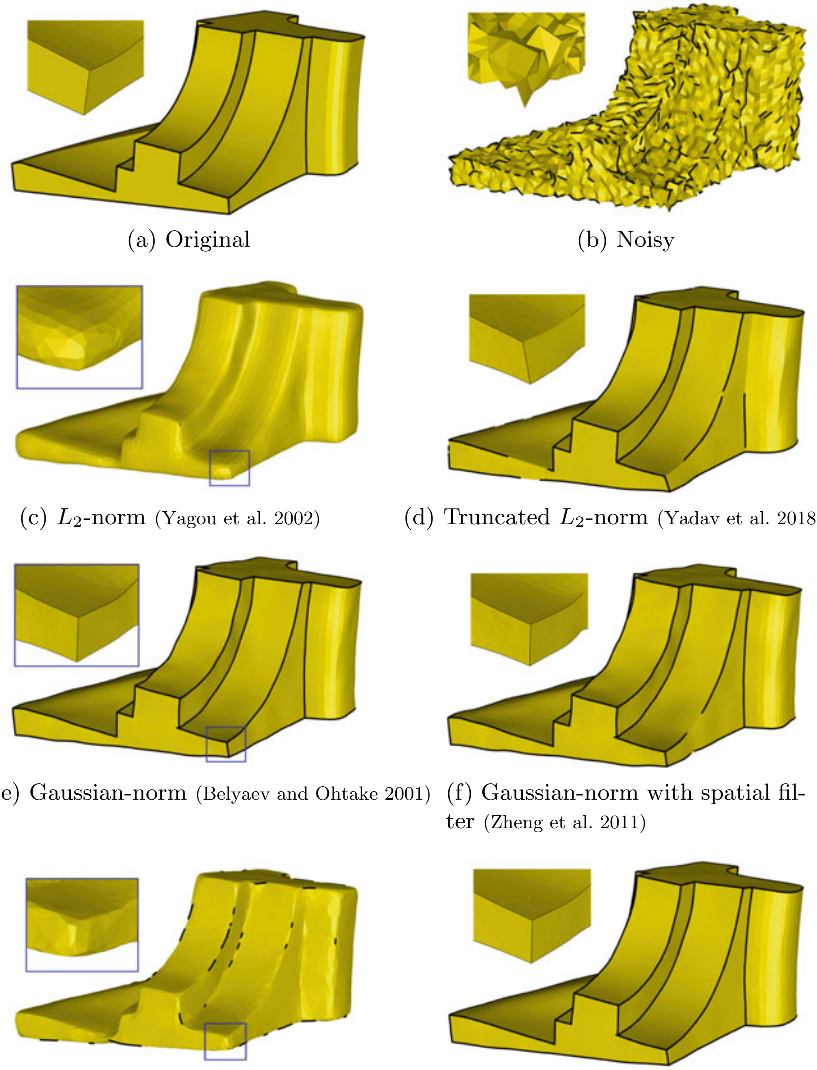
(e) Gaussian-norm (Belyaev and Ohtake 2001)

(f) Gaussian-norm with spatial filter (Zheng et al. 2011)

(g) Huber's minimax (Centin and Signoroni 2018)

(h) Tukey's-norm (Yadav et al. 2019)

Fig. 2 Nicola model corrupted with a Gaussian noise ($\sigma_n = 0.2l_e$) in random direction. Images **c**–*h* show the results produced by state-of-the-art methods, which are using different robust error norms (see Table 1)



(a) Original

(b) Noisy

(c) L_2-norm (Yagou et al. 2002)

(d) Truncated L_2-norm (Yadav et al. 2018c)

(e) Gaussian-norm (Belyaev and Ohtake 2001)

(f) Gaussian-norm with spatial filter (Zheng et al. 2011)

(g) Huber's minimax (Centin and Signoroni 2018)

(h) Tukey's-norm (Yadav et al. 2019)

Fig. 3 Fandisk model corrupted with a Gaussian noise ($\sigma_n = 0.3\ell_e$) in random direction. **c** –*h* show the results produced by state-of-the-art methods, which are using different robust error norms (see Table 1). The black curve highlights sharp edge information in the geometries and is detected using a dihedral angle threshold of $\theta = 70°$

Table 2 Overview on the discussed methods. For each method, we present the authors, year, citation, which input is processed (PSS or meshes), what error norm is used and whether a spatial weighting is applied. Furthermore, we collect the assessments from the above sections how the different methods perform in terms of feature preservation and noise removal

Method	Section	Input	Error norm	Spatial weights	Feature preservation	Noise removal
Belyaev and Ohtake (2001)	3.1 a	Mesh	Gaussian	No	Good	Ok
Yogou et al. (2002)	3.1 b	Mesh	L_1 and L_2	No	Ok	Ok
Yadav et al. (2018c)	3.1 d	Mesh	Truncated L_2	No	Good	Ok
Shen and Barner (2004)	3.1 e	Mesh	Gaussian	No	Good	Ok
Tasdizen et al. (2002)	3.1 f	Mesh	Gaussian	No	Good	Ok
Centin and Signoroni (2018)	3.1 g	Mesh	Huber's minimax*	No	Excellent	Ok
Zheng et al. (2011)	3.2 a	Mesh	Gaussian	Gaussian	Good	Good
Zhang et al. (2015)	3.2 b	Mesh	Gaussian	Gaussian	Good	Good
Yadav et al. (2019)	3.2 c	Mesh	Tukey's	Gaussian	Excellent	Good
Öztireli (2009)	4.1 a	PSS	Gaussian	Gaussian	Good	Good
Mattei and Castrodad (2016)	4.1 b	PSS	Gaussian	No	Good	Ok
Li et al. (2009)	4.2 a	PSS	Gaussian	Gaussian	Good	Good
Zheng et al. (2017)	4.2 b	PSS	Gaussian	Gaussian	Good	Good
Park et al. (2013)	4.2 c	PSS	Gaussian	No	Good	Ok
Digne and Franchis (2017)	4.2 d	PSS	Gaussian	Gaussian	Good	Good
Zheng et al. (2018)	4.2 e	PSS	Gaussian	Gaussian	Good	Good
Yadav et al. (2018a)	4.2 f	PSS	Truncated L_2	No	Good	Ok

$\star$ The error norm used in method (Centin and Signoroni 2018) is not equivalent to Huber's minimax. However, the utilized weighting term closely resembles the function $g_\sigma(x)$ of Huber's minimax, see Table 1 and the discussion in Sect. 3.1g

Acknowledgements This research was supported by the DFG Collaborative Research Center TRR 109, “Discretization in Geometry and Dynamics,” the Berlin Mathematical School, the Einstein Center for Mathematics Berlin, Nocturne GmbH, and the German National Academic Foundation. The authors would like to thank the anonymous reviewer for many helpful suggestions and comments on how to improve the paper.

References

Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT (2003) Computing and rendering point set surfaces. IEEE Trans Visual Comput Graph 9(1):3–15

Bajaj CL, Xu G (January, 2003) Anisotropic diffusion of surfaces and functions on surfaces. ACM Trans Graph 22(1):4–32, Association for Computing Machinery, New York, NY, USA. ISSN: 0730-0301. https://doi.org/10.1145/588272.588276

Barash D (2002) Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. IEEE Trans Pattern Anal Mach Intell 24(6):844–847

Beaton AE, Tukey JW (1974) The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. Technometrics 16(2):147–185

Belyaev AG, Ohtake Y (2001) Nonlinear diffusion of normals for crease enhancement. In: Vision geometry X, vol 4476, pp 42–48. International Society for Optics and Photonics

Black MJ, Rangarajan A (1996) On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. International Journal of Computer Vision 19(1):57–91

Black MJ, Sapiro G, Marimont DH, Heeger D (1998) Robust anisotropic diffusion. IEEE Trans Image Process 7(3):421–432

Botsch M, Kobbelt L, Pauly M, Alliez P, Lévy B (2010) Polygon mesh processing. Peters

Centin M, Signoroni A (2018) Mesh denoising with (geo)metric fidelity. IEEE Trans Visual Comput Graph 24(8):2380–2396

Chu CK, Glad IK, Godtliebsen F, Marron JS (1998) Edge-preserving smoothers for image processing. J Am Stat Assoc 93(442):526–541

Clarenz U, Diewald U, Rumpf M (2000) Anisotropic geometric diffusion in surface processing. In: Proceedings visualization 2000. VIS 2000 (Cat. No.00CH37145)

Desbrun M, Meyer M, Schröder P, Barr A (2001) Implicit fairing of irregular meshes using diffusion and curvature flow. SIGGRAPH

Digne J, de Franchis C (2017) The bilateral filter for point clouds. Image Process Line 7:278–287. https://doi.org/10.5201/ipol.2017.179

Durand F, Dorsey J (2002) Fast bilateral filtering for the display of high-dynamic-range images. ACM Trans Graph 21(3):257–266

Fleishman S, Drori I, Cohen-Or D (July, 2003) Bilateral mesh denoising. ACM Trans Graph 22(3):950–953. ISSN: 0730-0301. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/882262.882368

Hampel FR, Ronchetti EM, Rousseeuw PJ, Stahel WA (2005) Robust statistics: the approach based on influence functions. Wiley

Hildebrandt K, Polthier K (2004) Anisotropic filtering of non linear surface features. Comput Graph Forum. ISSN: 1467-8659. https://doi.org/10.1111/j.1467-8659.2004.00770.x

Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points. SIGGRAPH Comput Graph

Huber PJ (1981)Robust statistics. Wiley

Jones TR, Durand F, Desbrun M (2003) Non-iterative, feature-preserving mesh smoothing. ACM Trans Graph 22(3):943–949

Lange C, Polthier K (2005) Anisotropic smoothing of point sets. Comput Aided Geom Des 22(7):680–692

Li J (2009) Feature-preserving denoising of point-sampled surfaces. In: Proceedings of the 3rd WSEAS international conference on computer engineering and applications

Mattei E, Castrodad A (2016) Point cloud denoising via moving RPCA: Mrpca. *Computer Graphics Forum*, 2016

Mrázek, P, Weickert J, Bruhn A (2006) On robust estimation and smoothing with spatial and tonal kernels. Springer, Berlin, , pp 335–352

Ohtake Y, Belyaev AG, Seidel H-P (2002) Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normals. In: In vision modeling and visualization, Eurographics Association. https://www.semanticscholar.org/paper/Mesh-Smoothing-by-Adaptive-and-Anisotropic-Gaussian-Ohtake-Belyaev/19b431c843f4b37d2218e7efcd8f64b6ff589c1f

Öztireli X, Guennebaud G, Gross M (2009) Feature preserving point set surfaces based on non-linear kernel regression. Comput Graph Forum

Park MK, Lee SJ, Jang Y, Lee YY, Lee KH (2013) Feature-aware filtering for point-set surface denoising. Comput Graph

Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. IEEE Trans Pattern Anal Mach Intell 12(7):629–639. https://doi.org/10.1109/34.56205

Shen Y, Barner KE (2004) Fuzzy vector median-based surface smoothing. IEEE Trans Visual Comput Graph 10(3):252–265

Skrodzki M, Jansen J, Polthier K (2018) Directional density measure to intrinsically estimate and counteract non-uniformity in point clouds. Comput Aided Geom Des 64:73–89

Sun X, Rosin PL, Martin R, Langbein F (2007) Fast and effective feature-preserving mesh denoising. IEEE Trans Visual Comput Graph 13(5):925–938

Sun Y, Chen H, Qin J, Li H, Wei M, Zong H (2019) Reliable rolling-guided point normal filtering for surface texture removal. In: Computer graphics forum, vol 38 , issue no 7, pp 721–732. Wiley Online Library

Tasdizen T, Whitaker R, Burchard P, Osher S (2002) Geometric surface smoothing via anisotropic diffusion of normals. In: IEEE visualization, 2002. VIS 2002, pp 125–132

Taubin G (1999) A signal processing approach to fair surface design. Comput Graph (1999) (Proceedings of Siggraph '95)

Taubin G (2001a) Geometric signal processing on polygonal meshes. Eurographics State of the Art Reports

Taubin G (2001b) Linear anisotropic mesh filtering. In: IBM research report RC22213(W0110-051), IBM T.J. Watson Research

Tomasi C, R. Manduchi. Bilateral filtering for gray and color images. In: Iccv, vol 98, issue no 1

Trahanias PE, Venetsanopoulos AN (1993) Vector directional filters-a new class of multichannel image processing filters. IEEE Trans Image Process 2(4):528–534

Vallet B, Levy B (2008) Spectral geometry processing with manifold harmonics. Comput Graph Forum 27(2):251–260. https://doi.org/10.1111/j.1467-8659.2008.01122.x

Winkler G, Aurich V, Hahn K, Martin A, Rodenacker K (1998) Noise reduction in images: some recent edge-preserving methods. 138, sfb386. https://epub.ub.uni-muenchen.de/1527/. http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-1527-2

Yadav SK, Reitebuch U, Skrodzki M, Zimmermann E, Polthier K (2018a) Constraint-based point set denoising using normal voting tensor and restricted quadratic error metrics. Comput Graph 74:234–243. ISSN: 0097-8493. https://doi.org/10.1016/j.cag.2018.05.014. https://www.sciencedirect.com/science/article/pii/S0097849318300797

Yadav SK, Kadas EM, Motamedi S, Polthier K, Hausser F, Gawlik K, Paul F, Brandt A (2018b) Optic nerve head three-dimensional shape analysis. J Biomed Opt 23(10):1–13

Yadav SK, Reitebuch U, Polthier K (2018c) Mesh denoising based on normal voting tensor and binary optimization. IEEE Trans Visual Comput Graph 24(8):2366–2379

Yadav SK, Reitebuch U, Polthier K (2019) Robust and high fidelity mesh denoising. IEEE Trans Visua Comput Graph 25(6):2304–2310

Yagou H, Ohtake Y, Belyaev AG (2003) Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding. Proce Comput Grap Int 2003:28–33

Yagou H, Ohtake Y, Belyaev A (2002) Mesh smoothing via mean and median filtering applied to face normals. In: Proceedings of geometric modeling and processing. Theory and Applications. GMP 2002. pp 124–131

Zhang W, Deng B, Zhang J, Bouaziz S, Liu L (2015) Guided mesh normal filtering. Comput Graph Forum 34(7):23–34

Zheng Y, Fu H, Au OK, Tai C (2011) Bilateral normal filtering for mesh denoising. IEEE Trans Visual Comput Graph 17(10):1521–1530

Zheng Y, Li G, Wu S, Y. Liu, and Y. Gao. Guided point cloud denoising via sharp feature skeletons. Vis Comput (2017)

Zheng Y, Li G, Xu X, Wu S, Nie Y (2018) Rolling normal filtering for point clouds. Comput Aided Geom Des 62:16–28. ISSN (0167-8396). https://doi.org/10.1016/j.cagd.2018.03.004. https://www.sciencedirect.com/science/article/pii/S0167839618300189