

Article

A Learning-Free Noise-Adaptive Framework for Feature-Preserving Point Cloud Denoising

Artur Janowski ^{1,*} , Ahmet Emin Karkınlı ² , Mustafa Hüsrevoğlu ² , Talha Taşkanat ³ 
and Abdüsselam Kesikoğlu ⁴ 

¹ Department of Geodesy, Institute of Geodesy and Civil Engineering, Faculty of Geoengineering, University of Warmia and Mazury in Olsztyn, Oczapowskiego 1, 10-719 Olsztyn, Poland

² Department of Geomatics Engineering, Faculty of Engineering, Niğde Ömer Halisdemir University, Niğde 51240, Türkiye; akarkinli@ohu.edu.tr (A.E.K.); mhusrevoglu@ohu.edu.tr (M.H.)

³ Department of Geomatics Engineering, Faculty of Engineering, Erciyes University, Kayseri 38280, Türkiye; talhataskanat@erciyes.edu.tr

⁴ Tomarza MA Vocational School, Kayseri University, Kayseri 38280, Türkiye; kesikoglu@kayseri.edu.tr

* Correspondence: artur.janowski@uwm.edu.pl

Abstract

Point cloud denoising is a fundamental preprocessing task in 3D vision and geometry processing, where the main challenge is to suppress corruption while preserving sharp features, thin structures, and local surface fidelity. Classical geometric filters are computationally efficient and interpretable, but they commonly rely on fixed local supports or globally selected parameters, which limits their effectiveness under spatially heterogeneous corruption. More adaptive non-local and learning-based approaches can improve robustness, yet they often introduce higher computational complexity, stronger modeling assumptions, or substantial training-data dependency. In this work, we propose Noise-Adaptive Bilateral Normal Projection (NABNP), a learning-free point cloud denoising framework that introduces explicit patch-wise adaptation to local corruption conditions. NABNP estimates a robust dimensionless local noise level from point-to-plane residuals and uses this quantity to adapt the angular bandwidth of bilateral normal refinement, the balance between weighted local averaging and normal projection, and the magnitude of the positional update. This design enables conservative smoothing in locally reliable neighborhoods while applying stronger geometric correction in more severely corrupted regions. We evaluate NABNP on standard benchmark models under 13 stress scenarios covering additive Gaussian noise, outlier contamination, sparsity, and rotation perturbation, resulting in 650 trials per method. The experimental results show that NABNP provides strong aggregate behavior among the evaluated learning-free baselines, particularly under low-to-medium corruption, sparsity, and rotation perturbations, while its advantage becomes less pronounced under the most severe noise and outlier settings. The method remains training-free and interpretable, with a moderate computational cost associated with repeated neighborhood analysis and covariance-based local updates.

Keywords: point cloud denoising; feature-preserving filtering; bilateral normal refinement; normal projection; local noise estimation; learning-free geometry processing



Academic Editor: Thomas Lindner

Received: 29 May 2026

Revised: 22 June 2026

Accepted: 26 June 2026

Published: 2 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Point clouds have become a fundamental geometric representation in a broad range of 3D vision and geometry-processing applications, including surface reconstruction, reverse

engineering, robotic perception, autonomous navigation, and digital heritage preservation [1–3]. In practical acquisition pipelines, however, raw point clouds captured by depth sensors, structured-light devices, or LiDAR systems are inevitably affected by measurement noise, irregular sampling density, and outliers. Such degradation distorts the underlying surface structure and can substantially reduce the reliability of downstream tasks such as normal estimation, registration, reconstruction, and geometric analysis. Consequently, point cloud denoising remains a central and practically important preprocessing problem in modern 3D data processing pipelines.

The core challenge of point cloud denoising is to suppress corruption while preserving sharp features, thin structures, and local geometric fidelity. Naive smoothing tends to blur discontinuities, flatten corners, and induce the well-known shrinkage effect. To address this problem, classical geometry-driven approaches established strong foundations through bilateral filtering, point set surface modeling, moving least-squares fitting, projection-based reconstruction, differential-geometric regularization, and optimization-based local plane-fitting strategies [4–11]. These methods are attractive because they are mathematically interpretable, computationally efficient, and often effective under moderate corruption. Nevertheless, many of them rely on fixed local supports or globally selected parameters. When the corruption strength varies substantially across the point cloud, a single global setting typically forces an undesirable compromise between insufficient denoising in heavily corrupted regions and over-smoothing in relatively clean ones.

To improve robustness and feature preservation, subsequent research has explored more sophisticated filtering strategies. A first direction focuses on normal-guided and feature-aware processing, where the orientation field is refined before positional correction is applied [12–16]. A second direction seeks stronger structural modeling through non-local collaboration, graph-based regularization, and low-rank decomposition [17–22]. More recently, learning-based approaches have substantially advanced point cloud denoising by predicting filtered coordinates or displacement fields directly from data [2,23–28]. Although these methods have improved denoising quality under a variety of conditions, they also expose a persistent trade-off. Classical local filters often remain insufficiently adaptive to spatially heterogeneous corruption, whereas non-local and learning-based solutions generally introduce increased optimization complexity, stronger modeling assumptions, or dependence on large-scale annotated training data. Recent analyses of the field likewise indicate that robustness, interpretability, and local adaptivity remain difficult to achieve simultaneously within a single practical framework [3].

Motivated by this gap, we propose Noise-Adaptive Bilateral Normal Projection (NABNP), a learning-free point cloud denoising framework that preserves the transparency of classical geometric filtering while introducing explicit patch-wise adaptation to local corruption conditions. The central idea is to estimate, for each point, a dimensionless local noise level directly from robust point-to-plane residuals and then use this quantity to govern three tightly coupled components of the filtering process: the effective angular bandwidth of bilateral normal refinement, the balance between weighted local averaging and normal projection, and the allowable magnitude of the positional update. As a result, NABNP behaves conservatively in locally reliable neighborhoods to protect high-frequency geometric content, while becoming progressively more corrective in severely corrupted regions.

This work makes three main contributions. First, we introduce a robust patch-wise noise descriptor based on the median absolute deviation of normal-direction residuals, enabling local corruption strength to be estimated without assuming a fixed global noise scale. Second, we formulate a stable noise-adaptive update mechanism that blends weighted local averaging and normal projection under scale-aware clipping and damping, thereby improving flexibility without sacrificing geometric interpretability. Third, we provide

extensive experimental validation across multiple stress conditions, including additive Gaussian noise, outlier contamination, sparsity, and rotation perturbation, and analyze both parameter sensitivity and severe-corruption behavior to clarify the operating range of the method. The remainder of the paper is organized as follows. Section 2 reviews related studies, Section 3 presents the proposed methodology, Section 4 reports the experimental evaluation, Section 5 discusses the findings and limitations, and Section 6 concludes the paper.

2. Related Work

Early research on geometric point cloud denoising was largely built upon local smoothing and projection principles, where an underlying surface is first inferred from neighborhood geometry and the points are then regularized with respect to that estimate. The bilateral filtering paradigm introduced a feature-aware smoothing mechanism by combining spatial proximity with signal similarity, and subsequently inspired a broad family of geometry-processing methods aimed at suppressing noise while preserving structural discontinuities and fine-scale details [4,5]. In the context of point-based geometry, point set surface formulations, moving least-squares approximations, and algebraic point set surface models provided principled ways to represent local surfaces directly from unorganized points, and became highly influential in later denoising and reconstruction pipelines [6–8]. Projection-based formulations further strengthened this line by explicitly updating points toward estimated local surfaces rather than relying only on isotropic smoothing, as exemplified by the locally optimal projection (LOP) operator and its weighted extension WLOP [9,10]. Although these methods provide the basis for many geometric filtering pipelines, their reliance on fixed support sizes or globally selected parameters can make them less effective when noise intensity changes across the point cloud.

To alleviate the over-smoothing tendency of early projection and smoothing methods, subsequent studies increasingly focused on normal-guided filtering and explicit feature preservation. These methods typically improve the local orientation field before updating point positions, thereby attempting to suppress noise while retaining sharp edges and high-curvature structures. Representative examples include edge-aware point set resampling, guided denoising strategies based on sharp feature skeletons, rolling normal filtering, and iterative guidance normal filters [12–15]. Related developments also introduced more comprehensive geometric denoising frameworks that jointly address local smoothing, normal stability, and shrinkage control, as well as guided normal filtering formulations that further stabilize the estimated orientation field [16,29]. These approaches remain attractive because they are lightweight, interpretable, and free from training data. However, once their parameters are fixed, they often apply similar update behaviors across locally clean and severely corrupted regions, which may limit their adaptability under heterogeneous noise patterns.

Beyond purely local formulations, several studies have explored non-local redundancy, sparse regularization, graph-based modeling, and low-rank decomposition to improve denoising robustness. Non-local methods exploit patch similarity and repeated geometric structures beyond the immediate neighborhood, allowing the denoising process to benefit from collaborative information across distant regions of the point cloud [17,18]. More recent work has also investigated selective feature aggregation strategies to improve refinement stability under noisy observations [19]. In parallel, sparsity-driven optimization models introduced explicit regularization mechanisms, such as L_0 -based formulations, to preserve geometric sharpness while suppressing noise [30]. Graph-based approaches further formalized inter-point relationships through feature graphs and manifold-inspired Laplacian regularization, leading to denoising models that encode both local and semi-global structural dependencies [20,21]. Another related direction employed low-rank and RPCA-

inspired decompositions to separate structured geometric content from noise and sparse outliers [22]. While these methods often improve robustness under challenging corruption patterns, they generally introduce higher modeling complexity, heavier optimization procedures, or stronger structural assumptions than lightweight local geometric filters.

Learning-based denoising has also become an important direction in point cloud processing. PCPNet showed that local geometric quantities, such as normals and patch-level descriptors, can be estimated directly from raw point patches, providing a data-driven basis for local shape analysis under noisy sampling conditions [1]. PointCleanNet later extended this learning-based perspective to point cloud denoising by jointly addressing noise removal and outlier suppression in dense point sets [23]. Pointfilter adopted an encoder–decoder filtering strategy to regress refined point positions directly, reducing the need for hand-tuned filtering parameters while preserving geometric structures [2]. ScoreDenoise introduced a score-based formulation that estimates denoising gradients for point-set refinement [31]. Subsequent studies further explored recurrent, iterative, graph-based, and contrastive-learning formulations that combine point displacement prediction with normal estimation or feature refinement [24–28]. These recent studies are important reference points for the broader state of the art; however, they typically require trained models, training data, and implementation-specific inference protocols. Despite their advances, learning-based methods may be sensitive to domain shift when applied to unseen real-world scans and offer less direct geometric interpretability than classical model-based formulations.

Taken together, the reviewed literature points to a persistent trade-off in point cloud denoising. Classical geometric filters are computationally efficient and interpretable, but they often rely on fixed neighborhood scales or globally defined update parameters, which limits their ability to respond to spatially varying corruption. Non-local, graph-based, low-rank, and learning-based approaches can improve robustness under more challenging noise conditions, yet they usually introduce additional computational cost, stronger modeling assumptions, or dependence on training data. This trade-off motivates a learning-free denoising framework that retains the transparency of geometric filtering while adapting its behavior at the patch level according to local noise conditions. To this end, the proposed Noise-Adaptive Bilateral Normal Projection (NABNP) combines robust local noise estimation, bilateral normal refinement, and a noise-adaptive blending strategy between weighted local averaging and normal projection.

3. Proposed Methodology

Conventional point cloud denoising methods typically rely on a fixed update rule and a globally chosen scale parameter, which makes them sensitive to heterogeneous corruption patterns such as spatially varying noise, irregular sampling density, and the coexistence of smooth regions and sharp geometric structures. Although learning-based methods can alleviate some of these limitations by exploiting data-driven priors, they introduce substantial training cost, dataset dependency, and reduced geometric interpretability [2,6]. In this work, we instead seek a learning-free denoising framework that remains adaptive at the patch level while preserving a transparent mathematical formulation.

To this end, we propose Noise-Adaptive Bilateral Normal Projection (NABNP), an iterative filtering method that couples local surface estimation with robust patch-wise noise quantification. The central idea is to infer, for each point, a dimensionless local noise level directly from point-to-plane residuals, and then use this quantity to control the effective angular bandwidth of bilateral normal refinement, the blending ratio between weighted local averaging and normal projection, and the allowable update magnitude. Consequently, NABNP behaves conservatively in locally reliable neighborhoods while progressively strengthening geometric restoration in more severely corrupted regions.

3.1. Problem Formulation and Iterative Framework

The noisy input point cloud is defined in Equation (1):

$$\mathcal{P}^{(0)} = \{p_i^{(0)} \in \mathbb{R}^3\}_{i=1}^N, \quad (1)$$

where N denotes the number of points and $p_i^{(0)} \in \mathbb{R}^3$ is the initial position of the i -th point. The filtered output after T iterations is defined in Equation (2):

$$\mathcal{P}^{(T)} = \{p_i^{(T)} \in \mathbb{R}^3\}_{i=1}^N, \quad (2)$$

where T is the total number of filtering iterations.

At iteration t , the k -nearest-neighbor set of point $p_i^{(t)}$ is defined in Equation (3):

$$\mathcal{N}_i^{(t)} = \{p_j^{(t)} \mid p_j^{(t)} \text{ is among the } k \text{ nearest neighbors of } p_i^{(t)}\}, \quad (3)$$

where k is the neighborhood size. In the implementation, the k -NN search is accelerated using a kd-tree structure, which is standard in practical point cloud processing pipelines [32]. When the query includes the center point itself, the self-distance is excluded from the local-scale computation introduced later in Equation (4). For notational simplicity, the iteration superscript (t) is omitted in the remainder of this section whenever the meaning is clear.

At each iteration, NABNP first estimates a local spatial scale and an initial tangent structure from a weighted covariance analysis. It then computes a robust local noise descriptor from normal-direction residuals. This descriptor is used to adapt the bilateral normal refinement and the subsequent point update. The final displacement combines a weighted local mean shift with a scaled normal projection, and the resulting step is clipped and damped before being applied. Algorithm 1 summarizes the complete procedure.

Algorithm 1: Noise-Adaptive Bilateral Normal Projection (NABNP)

Input: Noisy point cloud $\mathcal{P}^{(0)} = \{p_i^{(0)}\}_{i=1}^N$, neighborhood size k , iteration count T , parameters $\lambda_s, \sigma_n, r_0, r_1, q, \gamma, \gamma_n, \tau_s, \beta$

Output: Filtered point cloud $\mathcal{P}^{(T)}$

```

1 for  $t \leftarrow 0$  to  $T - 1$  do
2   Compute the  $k$ -NN sets  $\mathcal{N}_i^{(t)}$  using Equation (3)
3   foreach point  $p_i^{(t)}$  do
4     Compute the local scale  $s_i$  and spatial bandwidth  $\sigma_{s,i}$  using Equations (4) and (5)
5     Compute the spatial weights  $w_{ij}^s$ , weighted centroid  $\mu_i$ , and covariance matrix  $C_i$  using
       Equations (6)–(8)
6     Extract the initial normal  $n_i$  as the eigenvector associated with the smallest eigenvalue of  $C_i$ 
7   foreach point  $p_i^{(t)}$  do
8     Compute the residuals  $r_{ij}$ , robust scale  $\hat{\sigma}_i$ , normalized ratio  $\rho_i$ , and bounded noise level  $nl_i$ 
       using Equations (9)–(12)
9     Compute the effective angular bandwidth  $\sigma_{n,i}^{\text{eff}}$  using Equation (13)
10  foreach point  $p_i^{(t)}$  do
11    Sign-align neighboring normals using Equation (14)
12    Compute the bilateral weights  $w_{ij}^b$  and refined normal  $\tilde{n}_i$  using Equations (15) and (16)
13    Compute the weighted local mean displacement  $\Delta_i^{\text{WLM}}$  using Equation (17)
14    Compute the normal-projection displacement  $\Delta_i^N$  using Equation (18)
15    Compute the effective projection gain  $\gamma_i$  and scaled projection  $\Delta_i^{\text{proj}}$  using Equations (19) and (20)
16    Compute the blending coefficient  $\alpha_i$  and raw displacement  $\Delta_i$  using Equations (21) and (22)
17    Compute the step bound  $m_i$  and clipped displacement  $\tilde{\Delta}_i$  using Equations (23) and (24)
18    Update the point position using Equation (25)

```

3.2. Local Scale and Initial Covariance Analysis

A reliable local geometric estimate should adapt to neighborhood scale rather than depend on a fixed global support radius. Therefore, for each point p_i , the local spatial scale is defined in Equation (4):

$$s_i = \text{median}_{p_j \in \mathcal{N}_i^*} \|p_j - p_i\|_2, \quad (4)$$

where $\mathcal{N}_i^*$ denotes the neighbor set with the self-point removed if present, and s_i represents the median Euclidean neighbor distance around p_i .

Using the local scale s_i , the spatial bandwidth is defined in Equation (5):

$$\sigma_{s,i} = \lambda_s s_i, \quad (5)$$

where $\lambda_s > 0$ is a global scaling factor controlling the spatial support width and $\sigma_{s,i}$ is the spatial bandwidth associated with point p_i .

Using the spatial bandwidth $\sigma_{s,i}$, the Gaussian spatial weights are defined in Equation (6):

$$w_{ij}^s = \exp\left(-\frac{\|p_j - p_i\|_2^2}{\sigma_{s,i}^2 + \varepsilon}\right), \quad p_j \in \mathcal{N}_i, \quad (6)$$

where w_{ij}^s is the spatial weight assigned to neighbor p_j relative to the center point p_i , and $\varepsilon > 0$ is a small numerical-stability constant.

Using the spatial weights w_{ij}^s , the weighted centroid is defined in Equation (7):

$$\mu_i = \frac{\sum_{p_j \in \mathcal{N}_i} w_{ij}^s p_j}{\sum_{p_j \in \mathcal{N}_i} w_{ij}^s + \varepsilon}, \quad (7)$$

where $\mu_i \in \mathbb{R}^3$ is the weighted local patch center around point p_i .

Using the same weights, the spatially weighted covariance matrix is defined in Equation (8):

$$C_i = \frac{\sum_{p_j \in \mathcal{N}_i} w_{ij}^s (p_j - \mu_i)(p_j - \mu_i)^\top}{\sum_{p_j \in \mathcal{N}_i} w_{ij}^s + \varepsilon}, \quad (8)$$

where $C_i \in \mathbb{R}^{3 \times 3}$ characterizes the second-order local geometric structure around p_i . The weighted centroid in Equation (7) defines the local patch center, while the covariance matrix in Equation (8) provides the initial tangent estimate. Let $n_i \in \mathbb{R}^3$ denote the eigenvector associated with the smallest eigenvalue of C_i . This PCA-based construction is standard for local normal estimation on unorganized point clouds [33]. This vector serves as the initial estimate of the local surface normal.

3.3. MAD-Based Local Noise-Level Estimation

The key adaptive component of NABNP is the estimation of local corruption strength directly from neighborhood geometry rather than from a user-specified global noise parameter. Using the initial tangent estimate (μ_i, n_i) , the absolute normal-direction residual of each neighboring point is defined in Equation (9):

$$r_{ij} = \left| (p_j - \mu_i)^\top n_i \right|, \quad p_j \in \mathcal{N}_i, \quad (9)$$

where r_{ij} measures the point-to-plane deviation of neighbor p_j from the tangent plane centered at μ_i with normal n_i . The residuals in Equation (9) quantify the dispersion of the local patch around the tangent structure inferred from the covariance analysis.

To obtain a robust local noise estimate, NABNP uses the median absolute deviation (MAD). The robust local scale estimate is defined in Equation (10):

$$\hat{\sigma}_i = \frac{\text{median}_{p_j \in \mathcal{N}_i}(r_{ij})}{0.67448975}, \quad (10)$$

where $\hat{\sigma}_i$ is the robust estimate of the normal-direction noise scale around point p_i , and the denominator 0.67448975 is the standard Gaussian consistency factor used for MAD-based scale estimation [34]. The estimate in Equation (10) is substantially less sensitive to heavy-tailed local deviations and isolated outliers than variance-based alternatives. In this sense, the local descriptor is related to the broader robust-estimation tradition, including bounded-influence and redescending estimators such as Huber, Hampel, Tukey biweight, Andrews sine, and Fair-type M-estimators [35–39]. NABNP does not solve an explicit M-estimation problem; instead, it uses a MAD-type scale estimate as a lightweight reliability signal that can be evaluated independently within each local patch.

Since $\hat{\sigma}_i$ in Equation (10) still depends on the size of the local patch, the normalized noise ratio is defined in Equation (11):

$$\rho_i = \frac{\hat{\sigma}_i}{s_i + \varepsilon}, \quad (11)$$

where ρ_i is a dimensionless local noise ratio.

Using the normalized ratio ρ_i , the bounded local noise-level coefficient is defined in Equation (12):

$$nl_i = \text{clip}\left(\frac{\rho_i - r_0}{r_1 - r_0 + \varepsilon}, 0, 1\right), \quad (12)$$

where r_0 and r_1 are the lower and upper transition thresholds, respectively, and $\text{clip}(\cdot, 0, 1)$ truncates its argument to the interval $[0, 1]$. Small values of nl_i in Equation (12) indicate locally reliable neighborhoods, whereas values close to one indicate stronger deviation from the tangent structure. This bounded, scale-normalized coefficient is the central control variable of NABNP. Since the residuals are measured relative to the initial covariance-based tangent estimate, nl_i should be interpreted as a local reliability indicator rather than an exact physical noise magnitude. When a neighborhood is severely contaminated, the initial tangent itself may become biased; in such cases, the clipped coefficient intentionally limits the update strength, but it cannot fully replace an explicit outlier-screening or robust model-fitting stage.

3.4. Noise-Adaptive Bilateral Normal Refinement

To obtain a more stable orientation field for projection, the initial normals are refined through bilateral aggregation. This refinement follows the general feature-aware bilateral filtering principle while adapting its angular support to local corruption conditions [4,5,14,15]. The effective angular bandwidth used in this refinement is defined in Equation (13):

$$\sigma_{n,i}^{\text{eff}} = \sigma_n(1 + 0.2nl_i), \quad (13)$$

where $\sigma_n > 0$ is the base angular bandwidth and $\sigma_{n,i}^{\text{eff}}$ is the noise-adaptive angular bandwidth at point p_i . Equation (13) enlarges the angular support in noisier neighborhoods and allows the normal estimation stage to gather more stable directional evidence when the local corruption level is high.

Because eigenvector signs are ambiguous, the sign-aligned neighboring normal is defined in Equation (14):

$$\tilde{n}_j^{(i)} = \text{sign}(n_j^\top n_i) n_j, \quad (14)$$

where n_j is the initial normal of neighbor p_j , $\bar{n}_j^{(i)}$ is its sign-aligned version with respect to n_i , and $\text{sign}(\cdot)$ denotes the sign function.

Using the spatial bandwidth $\sigma_{s,i}$ from Equation (5) and the effective angular bandwidth $\sigma_{n,i}^{\text{eff}}$ from Equation (13), the bilateral refinement weights are defined in Equation (15):

$$w_{ij}^b = \exp\left(-\frac{\|p_j - p_i\|_2^2}{\sigma_{s,i}^2 + \varepsilon} - \frac{\|\bar{n}_j^{(i)} - n_i\|_2^2}{(\sigma_{n,i}^{\text{eff}})^2 + \varepsilon}\right), \quad (15)$$

where w_{ij}^b is the bilateral normal weight assigned to neighbor p_j .

Using the bilateral weights w_{ij}^b , the refined normal is defined in Equation (16):

$$\tilde{n}_i = \frac{\sum_{p_j \in \mathcal{N}_i} w_{ij}^b \bar{n}_j^{(i)}}{\left\| \sum_{p_j \in \mathcal{N}_i} w_{ij}^b \bar{n}_j^{(i)} \right\|_2 + \varepsilon}, \quad (16)$$

where $\tilde{n}_i \in \mathbb{R}^3$ is the bilateral-refined unit normal at point p_i . Equation (16) suppresses unstable local normal fluctuations while remaining sensitive to genuine orientation changes near edges and thin structures.

3.5. Noise-Adaptive Blending and Stable Point Update

Once the local noise level and the refined normal have been estimated, NABNP computes two complementary displacements. The position update stage is related in spirit to projection-based point consolidation and feature-preserving geometric denoising methods, but differs by explicitly adapting the update magnitude to the estimated local corruption level [9,10,22,30].

The weighted local mean displacement is defined in Equation (17):

$$\Delta_i^{\text{WLM}} = \mu_i - p_i, \quad (17)$$

where $\Delta_i^{\text{WLM}} \in \mathbb{R}^3$ is the weighted local mean displacement and μ_i is the weighted centroid defined in Equation (7). The term in Equation (17) provides conservative positional regularization and should be distinguished from the arithmetic-mean LMF baseline used in the experiments.

The normal-projection displacement is defined in Equation (18):

$$\Delta_i^N = -\left((p_i - \mu_i)^\top \tilde{n}_i\right) \tilde{n}_i, \quad (18)$$

where $\Delta_i^N \in \mathbb{R}^3$ is the unscaled normal-projection displacement. Equation (18) directly reduces the normal-direction deviation of the point from the locally estimated tangent plane.

The effective projection gain is defined in Equation (19):

$$\gamma_i = \gamma(1 + \gamma_n n l_i), \quad (19)$$

where $\gamma > 0$ is the base projection scale, $\gamma_n \geq 0$ controls the noise-driven amplification, and γ_i is the effective projection scale at point p_i .

Using the effective gain γ_i , the scaled projection displacement is defined in Equation (20):

$$\Delta_i^{\text{proj}} = \gamma_i \Delta_i^N, \quad (20)$$

where $\Delta_i^{\text{proj}} \in \mathbb{R}^3$ is the scaled projection displacement.

The adaptive blending coefficient is defined in Equation (21):

$$\alpha_i = (nl_i)^q, \quad (21)$$

where $q > 0$ shapes the transition curve and $\alpha_i \in [0, 1]$ controls the relative contribution of the two update branches.

Using the coefficient α_i , the raw displacement is defined in Equation (22):

$$\Delta_i = (1 - \alpha_i)\Delta_i^{\text{WLM}} + \alpha_i\Delta_i^{\text{proj}}, \quad (22)$$

where $\Delta_i \in \mathbb{R}^3$ is the raw update before clipping. Equation (22) yields an interpretable continuum. For small nl_i , the displacement is dominated by weighted local averaging. As nl_i increases, the contribution of normal projection becomes progressively stronger.

To prevent unstable motion in highly corrupted regions, the maximum admissible step length is defined in Equation (23):

$$m_i = \tau_s(1 + 0.5nl_i)s_i, \quad (23)$$

where $\tau_s > 0$ is the step-cap ratio and m_i is the maximum admissible step length at point p_i .

Using the upper bound m_i , the clipped displacement is defined in Equation (24):

$$\tilde{\Delta}_i = \min\left(1, \frac{m_i}{\|\Delta_i\|_2 + \varepsilon}\right)\Delta_i, \quad (24)$$

where $\tilde{\Delta}_i \in \mathbb{R}^3$ is the clipped update. Equation (24) preserves the update direction while limiting its magnitude according to the local scale and corruption level.

Finally, the iterative point update is defined in Equation (25):

$$p_i^{(t+1)} = p_i^{(t)} + \beta\tilde{\Delta}_i, \quad (25)$$

where $\beta \in (0, 1]$ is a damping factor controlling the fraction of the clipped displacement applied at each iteration. Equation (25) is repeated for all points and all iterations, with the neighborhood graph and all local quantities recomputed after each iteration.

The contribution of NABNP lies in the coupling of robust patch-wise noise estimation, noise-adaptive bilateral normal refinement, and scale-aware blending between weighted local averaging and normal projection. In particular, the local descriptor nl_i defined in Equation (12) plays a central role because it governs the effective normal bandwidth in Equation (13), the blending behavior in Equation (21), and the step bound in Equation (23). This design enables a single learning-free formulation to adapt across different corruption regimes while maintaining a clear geometric interpretation.

For a fixed neighborhood size k and iteration count T , the dominant computational cost of NABNP arises from repeated k -NN queries and local weighted moment computations. The overall practical complexity is defined in Equation (26):

$$\mathcal{O}(T(N \log N + Nk)), \quad (26)$$

where the $N \log N$ term corresponds to neighborhood retrieval and the Nk term corresponds to patch-local computations. Equation (26) is consistent with the intended learning-free design of NABNP, since the method is entirely patch-local and requires no training stage.

4. Experimental Results

4.1. Experimental Setup and Evaluation Metrics

We evaluate the proposed Noise-Adaptive Bilateral Normal Projection (NABNP) on standard PCPNet benchmark models commonly used in point cloud filtering studies [1]: *Fandisk*, *Bunny*, *Dragon*, *Armadillo*, and *Boxunion*. These models include sharp edges, smooth regions, and fine-scale details, providing diverse stress cases for surface-preserving denoising.

The experimental protocol comprises 13 scenarios grouped into four stress categories. Additive Gaussian noise is applied with zero mean and standard deviation $\sigma \in \{0.005, 0.01, 0.015, 0.02, 0.025\}$ scaled by the bounding-box diagonal. For outlier robustness, random points are injected at ratios $\{1\%, 2\%, 5\%, 10\%\}$ of the original point count, uniformly sampled within a 1.2-times expanded bounding volume, and a mild Gaussian perturbation with $\sigma = 0.005$ is also added. To simulate sparsity, points are randomly decimated to keep $\{75\%, 50\%, 25\%\}$ of the data after adding mild Gaussian noise with $\sigma = 0.01$. For the rotation stress test, the input point cloud is randomly rotated by composing independent rotations about two axes, and a mild Gaussian perturbation with $\sigma = 0.01$ is applied.

Each model and scenario is evaluated over 10 independent random seeds, resulting in $5 \times 13 \times 10 = 650$ trials per method. We compare the proposed NABNP against representative non-learning geometric filtering baselines, including Local Mean Filter (LMF), Local Median Filter (LMedF), Normal Bilateral Filter (NBF), and Weighted Locally Optimal Projection (WLOP) [10]. WLOP is included as a representative projection-based baseline from the geometry-processing literature, while LMF, LMedF, and NBF provide simple and widely used local filtering references. The WLOP baseline uses fixed parameters $k = 35$, 2 iterations, $\mu = 0.35$, and $h_{\text{scale}} = 5.0$ across all experiments. These settings were selected from a small grid search over cached benchmark scenarios, summarized in Table 1, so that WLOP is represented by a competitive fixed configuration rather than by arbitrary default values. The final NABNP configuration used throughout the benchmark was determined via Optuna-based Bayesian hyperparameter optimization [40] and employs $k = 35$, $T = 2$, $\lambda_s = 1.0$, $\sigma_n = 0.5$, $r_0 = 0.03347589$, $r_1 = 0.16289218$, $q = 0.73559010$, $\gamma = 0.95$, $\gamma_n = 0.35$, $\tau_s = 1.0$, and $\beta = 0.9$. Within NABNP, the averaging branch corresponds to a spatially weighted local mean update, which is distinct from the standalone arithmetic-mean LMF baseline reported in the comparisons.

The main quantitative comparison is intentionally restricted to learning-free methods with fully reproducible inference rules and no training stage. Recent neural denoisers such as PointCleanNet, Pointfilter, ScoreDenoise, PCDNF, IterativePFN, and related architectures are discussed in Section 2, but they are not included in the 650-trial main benchmark because their performance depends on trained weights, training-domain assumptions, implementation-specific inference protocols, and dependency-constrained runtime environments. Consequently, the primary experimental claims in this study should be read as comparisons within the selected learning-free geometric baseline group rather than as a universal state-of-the-art ranking over all neural and non-neural denoising methods. To provide additional context, Section 4.3 reports a separate full-resolution reference check using runnable pretrained learning-based pipelines.

Quantitative evaluation is conducted using three standard geometric error measures computed via nearest-neighbor queries. MSE is the mean squared distance from the filtered set to the ground truth in the Pred $\rightarrow$ GT direction. Chamfer Distance (CD) is the sum of mean squared distances in both directions. HD95 is the symmetric 95th-percentile Hausdorff distance reported as $\max(P_{95}(\text{Pred} \rightarrow \text{GT}), P_{95}(\text{GT} \rightarrow \text{Pred}))$. We also report end-to-end runtime in seconds (Time_s) measured using wall-clock timing under a fixed

machine configuration. Table 2 summarizes the protocol. In all result tables, the symbol $\downarrow$ indicates that lower values are better.

Table 1. Preliminary WLOP grid-search summary.

Rank	k	Iter.	μ	h_{scale}	Mean CD $\downarrow$	Mean HD95 $\downarrow$
1	35	2	0.35	5.0	0.001654	0.0442
2	25	2	0.35	5.0	0.001661	0.0446
3	35	2	0.45	5.0	0.001677	0.0455
4	35	2	0.35	4.0	0.001679	0.0456
5	25	2	0.35	4.0	0.001680	0.0456

Table 2. Summary of the experimental protocol, stress scenarios, compared methods, and evaluation metrics.

Item	Values / Description
3D Models	Fandisk, Bunny, Dragon, Armadillo, Boxunion
Scenarios	13 scenarios in 4 categories (Noise, Outlier, Sparsity, Rotation)
Trials	10 random seeds per model–scenario pair ($5 \times 13 \times 10 = 650$ trials per method)
Compared Methods	Noisy, LMF, LMedF, NBF, WLOP, Ours (NABNP)
WLOP Settings	$k = 35$, iterations = 2, $\mu = 0.35$, $h_{\text{scale}} = 5.0$ (fixed across all experiments)
Additive Noise	$\sigma \in \{0.005, 0.01, 0.015, 0.02, 0.025\}$ scaled by bounding-box diagonal
Outliers	$\{1\%, 2\%, 5\%, 10\%\}$ random points in $1.2 \times$ expanded volume (+Gaussian noise, $\sigma = 0.005$)
Sparsity	Keep ratio $\{75\%, 50\%, 25\%\}$ after random decimation (+Gaussian noise, $\sigma = 0.01$)
Rotation	Random 3D rotation (two-axis composition) (+Gaussian noise, $\sigma = 0.01$)
Metrics	MSE (Pred $\rightarrow$ GT), CD (bi-directional), HD95 (symmetric 95th percentile), Time_s (seconds)

4.2. Quantitative Results

This section reports quantitative results for the proposed Noise-Adaptive Bilateral Normal Projection (NABNP) and the baselines under the predefined stress scenarios. Unless stated otherwise, reported values are averaged over all five models and ten random seeds. In the quantitative tables, values are reported as mean $\pm$ standard deviation and the best value for each scenario is highlighted in bold.

Under additive Gaussian noise, we evaluate five noise levels $\sigma \in \{0.005, 0.01, 0.015, 0.02, 0.025\}$ scaled by the bounding-box diagonal. Figure 1 reports the mean Chamfer Distance (CD) as a function of σ , where error bars denote one standard deviation across the aggregated trials. Table 3 provides the corresponding numerical results for CD and HD95 (mean $\pm$ standard deviation) for each method at each noise level.

Under outlier corruption, random points are injected at ratios 1%, 2%, 5%, and 10% of the original point count. Figure 2 reports HD95 as a function of the outlier ratio, with error bars indicating one standard deviation over all models and seeds. Table 4 provides the corresponding numerical results.

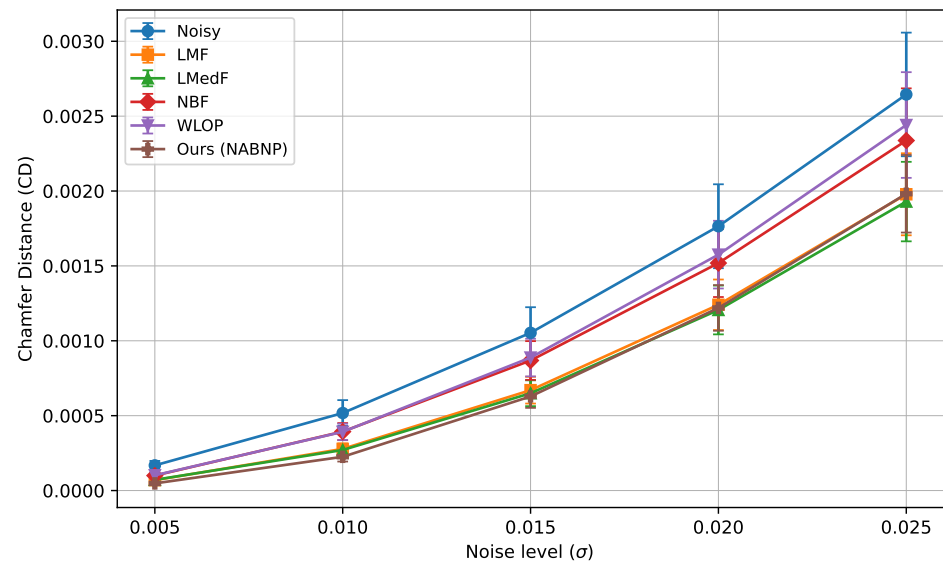


Figure 1. Chamfer Distance (CD) under additive Gaussian noise.

Table 3. Quantitative results under additive Gaussian noise.

σ	Method	CD ↓	HD95 ↓
0.005	Noisy	0.000168 ± 0.000030	0.021426 ± 0.001523
	LMF	0.000069 ± 0.000011	0.010717 ± 0.001010
	LMedF	0.000071 ± 0.000012	0.010555 ± 0.001030
	NBF	0.000100 ± 0.000014	0.013725 ± 0.001140
	WLOP	0.000102 ± 0.000015	0.014548 ± 0.001027
	Ours (NABNP)	0.000048 ± 0.000010	0.008539 ± 0.001336
0.010	Noisy	0.000518 ± 0.000086	0.041500 ± 0.003002
	LMF	0.000278 ± 0.000036	0.026944 ± 0.001657
	LMedF	0.000271 ± 0.000035	0.026005 ± 0.001619
	NBF	0.000394 ± 0.000057	0.033489 ± 0.002026
	WLOP	0.000392 ± 0.000054	0.034377 ± 0.002051
	Ours (NABNP)	0.000225 ± 0.000032	0.023697 ± 0.002222
0.015	Noisy	0.001053 ± 0.000171	0.061219 ± 0.004385
	LMF	0.000670 ± 0.000089	0.045111 ± 0.002675
	LMedF	0.000650 ± 0.000086	0.043760 ± 0.002585
	NBF	0.000868 ± 0.000129	0.053268 ± 0.003374
	WLOP	0.000888 ± 0.000126	0.054929 ± 0.003396
	Ours (NABNP)	0.000628 ± 0.000076	0.042541 ± 0.002861
0.020	Noisy	0.001765 ± 0.000280	0.080514 ± 0.005697
	LMF	0.001240 ± 0.000169	0.063760 ± 0.003801
	LMedF	0.001206 ± 0.000163	0.062150 ± 0.003702
	NBF	0.001519 ± 0.000227	0.072756 ± 0.004673
	WLOP	0.001575 ± 0.000226	0.075226 ± 0.004747
	Ours (NABNP)	0.001219 ± 0.000152	0.061708 ± 0.003812
0.025	Noisy	0.002645 ± 0.000413	0.099392 ± 0.007152
	LMF	0.001978 ± 0.000274	0.082314 ± 0.005018
	LMedF	0.001930 ± 0.000265	0.080522 ± 0.004877
	NBF	0.002337 ± 0.000348	0.091804 ± 0.006092
	WLOP	0.002441 ± 0.000353	0.094990 ± 0.006213
	Ours (NABNP)	0.001982 ± 0.000259	0.080775 ± 0.004804

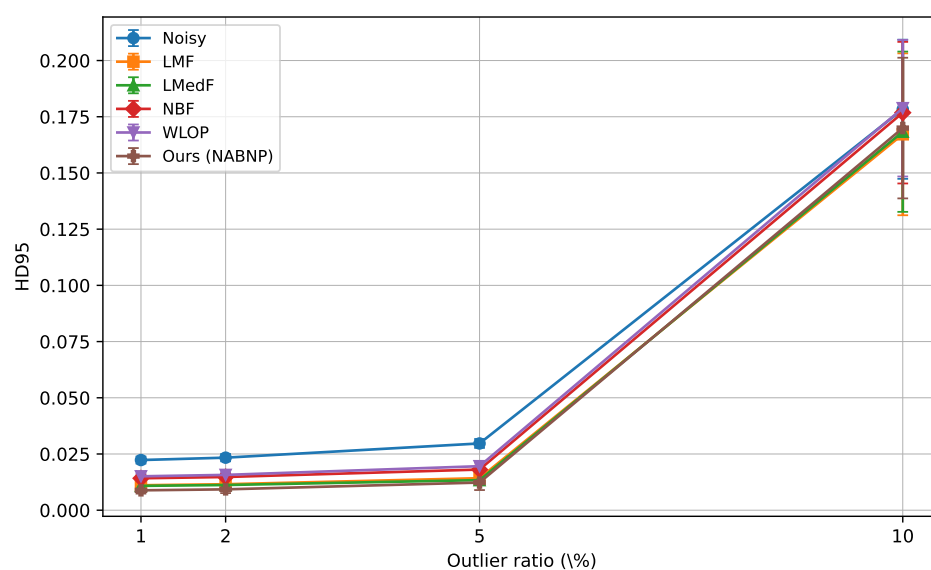


Figure 2. HD95 under increasing outlier contamination.

Table 4. Quantitative results under outlier contamination.

Outliers	Method	HD95 ↓
1%	Noisy	0.022307 ± 0.001575
	LMF	0.011093 ± 0.001066
	LMedF	0.010821 ± 0.001021
	NBF	0.014198 ± 0.001191
	WLOP	0.015072 ± 0.001073
	Ours (NABNP)	0.008840 ± 0.001462
2%	Noisy	0.023367 ± 0.001631
	LMF	0.011552 ± 0.001138
	LMedF	0.011199 ± 0.001067
	NBF	0.014766 ± 0.001255
	WLOP	0.015701 ± 0.001127
	Ours (NABNP)	0.009270 ± 0.001668
5%	Noisy	0.029687 ± 0.001981
	LMF	0.014239 ± 0.001602
	LMedF	0.013374 ± 0.001452
	NBF	0.018085 ± 0.001731
	WLOP	0.019575 ± 0.001578
	Ours (NABNP)	0.012298 ± 0.003340
10%	Noisy	0.178325 ± 0.030939
	LMF	0.167187 ± 0.036000
	LMedF	0.168340 ± 0.035644
	NBF	0.176811 ± 0.031490
	WLOP	0.178818 ± 0.030388
	Ours (NABNP)	0.169964 ± 0.031282

Under sparsity, the point clouds are randomly decimated to keep 75%, 50%, and 25% of points. Figure 3 reports Chamfer Distance (CD) across the keep ratios, with error bars indicating one standard deviation over all models and seeds. Table 5 provides the corresponding numerical results.

Under the rotation stress test, the input point clouds are randomly rotated and evaluated under the same evaluation pipeline. Table 6 reports the corresponding MSE, CD, and HD95 results (mean ± standard deviation) averaged over all models and seeds.

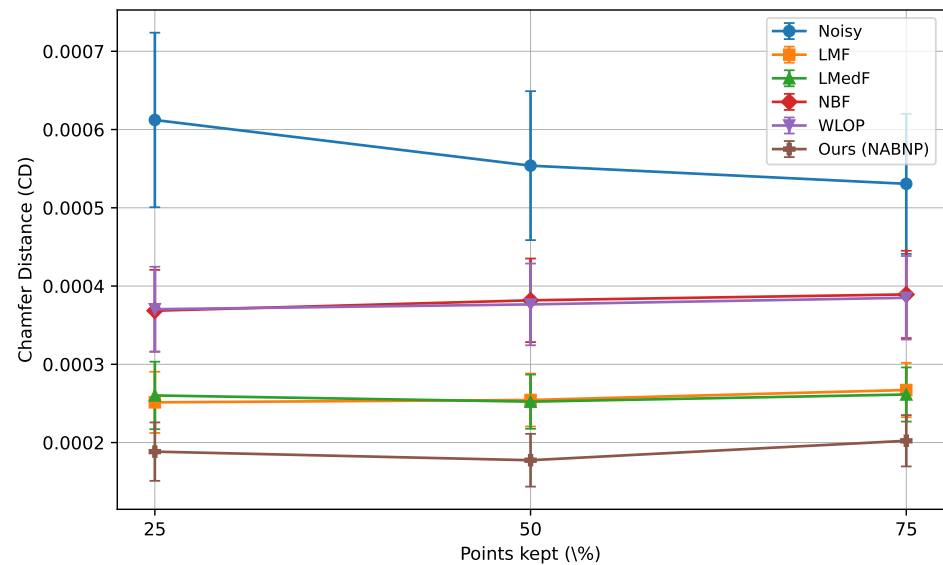


Figure 3. Chamfer Distance (CD) under point cloud sparsity.

Table 5. Quantitative results under point cloud sparsity.

Kept Points	Method	CD ↓
75%	Noisy	0.000531 ± 0.000090
	LMF	0.000267 ± 0.000035
	LMedF	0.000261 ± 0.000035
	NBF	0.000389 ± 0.000056
	WLOP	0.000385 ± 0.000053
	Ours (NABNP)	0.000202 ± 0.000033
50%	Noisy	0.000554 ± 0.000095
	LMF	0.000254 ± 0.000034
	LMedF	0.000252 ± 0.000035
	NBF	0.000382 ± 0.000053
	WLOP	0.000377 ± 0.000052
	Ours (NABNP)	0.000177 ± 0.000034
25%	Noisy	0.000612 ± 0.000112
	LMF	0.000251 ± 0.000039
	LMedF	0.000260 ± 0.000043
	NBF	0.000368 ± 0.000052
	WLOP	0.000370 ± 0.000054
	Ours (NABNP)	0.000188 ± 0.000037

Table 6. Rotation stress-test results.

Method	MSE ↓	CD ↓	HD95 ↓
Noisy	0.000457 ± 0.000071	0.000518 ± 0.000086	0.041500 ± 0.003002
LMF	0.000219 ± 0.000026	0.000278 ± 0.000036	0.026944 ± 0.001657
LMedF	0.000206 ± 0.000024	0.000271 ± 0.000036	0.026029 ± 0.001624
NBF	0.000323 ± 0.000041	0.000394 ± 0.000057	0.033489 ± 0.002026
WLOP	0.000330 ± 0.000041	0.000392 ± 0.000054	0.034377 ± 0.002051
Ours (NABNP)	0.000170 ± 0.000029	0.000225 ± 0.000032	0.023697 ± 0.002222

To provide an aggregate summary across the benchmark, we report win counts over all 65 scenario–model combinations. For each scenario and model, a win is assigned to the method achieving the lowest mean error, averaged over seeds, for the corresponding metric. Table 7 reports win counts for MSE, CD, and HD95.

Table 7. Win counts over the 65 scenario–model combinations.

Method	MSE Wins	CD Wins	HD95 Wins
LMF	11	12	2
LMedF	11	9	6
NBF	0	0	0
WLOP	0	0	0
Ours (NABNP)	43	44	57

4.3. External Learning-Based Reference Check

In addition to the main learning-free benchmark, we performed a scoped full-resolution reference check with two learning-based denoisers for which official pretrained inference pipelines could be executed reproducibly: Pointfilter [2] and ScoreDenoise [31]. This check was not designed as a complete neural-baseline ranking. Instead, it provides a protocol-matched indication of how the proposed learning-free method behaves relative to representative trained models on a small set of full-resolution benchmark cases. The evaluation used two models (*Fandisk* and *Dragon*), four representative scenarios (Noise_0.01, Noise_0.02, Outlier_5%, and Sparsity_50%), and one random seed. The neural methods were executed with CUDA using their pretrained inference pipelines; reported runtimes include preprocessing, patch construction or clustering, and model inference. Table 8 reports mean values over completed cases; ScoreDenoise completed seven of eight cases and failed on one Dragon Outlier_5% case.

Table 8. External learning-based reference check on eight full-resolution representative cases.

Method	Completed	CD ↓	HD95 ↓	CD Wins	HD95 Wins	Time (s) ↓
NABNP	8/8	0.001348	0.0289	2	2	4.64
Pointfilter	8/8	0.001205	0.0150	6	6	504.53
ScoreDenoise	7/8	0.004579	0.0981	0	0	193.51

Table 8 shows that Pointfilter achieves the lowest mean CD and HD95 in this limited reference check and is especially strong under the tested Gaussian-noise and sparsity cases. The win counts are descriptive and refer only to this scoped reference check. NABNP remains substantially faster and gives the best results in the two tested outlier cases, which is consistent with its design goal of providing a training-free and interpretable geometric filtering alternative. ScoreDenoise does not improve the aggregate behavior in this protocol and is unstable under outlier contamination. These results clarify that the proposed method should not be interpreted as universally outperforming trained neural denoisers; rather, it offers a different operating point characterized by reproducible learning-free inference, considerably lower runtime, and robust behavior in the tested outlier cases.

4.4. Diagnostic Sensitivity Analyses

To examine the stability of the fixed NABNP configuration, we performed a one-at-a-time parameter sensitivity check around the selected values. The analysis used deterministic 5000-point subsamples from two representative models, four stress scenarios, and two random seeds. The purpose of this diagnostic experiment is not to replace the full benchmark, but to quantify whether moderate parameter perturbations substantially change the error trends. Table 9 reports the observed ranges of CD and HD95 after normalizing each case by the default NABNP result. Most perturbations remain close to one, with the largest CD variation appearing for the spatial scale factor λ_s , which is expected because it directly controls the effective neighborhood support. The transition exponent q , angular bandwidth σ_n , and threshold pair (r_0, r_1) show comparatively narrow ranges in this diagnostic setting.

Table 9. One-at-a-time parameter sensitivity around the default NABNP configuration.

Parameter	Tested Values	CD Ratio Range	HD95 Ratio Range
β	0.8, 1.0	0.924–1.078	0.941–1.053
γ	0.85 $\times$, 1.15 $\times$	0.903–1.097	0.930–1.072
γ_n	0.5 $\times$, 1.5 $\times$	0.916–1.082	0.934–1.059
λ_s	0.8 $\times$, 1.2 $\times$	0.887–1.124	0.924–1.097
q	0.8 $\times$, 1.2 $\times$	0.979–1.024	0.996–1.005
r_0, r_1	0.85 $\times$, 1.15 $\times$	0.937–1.060	0.985–1.011
σ_n	0.8 $\times$, 1.2 $\times$	0.973–1.013	0.990–1.009
τ_s	0.75 $\times$, 1.25 $\times$	1.000–1.000	1.000–1.000

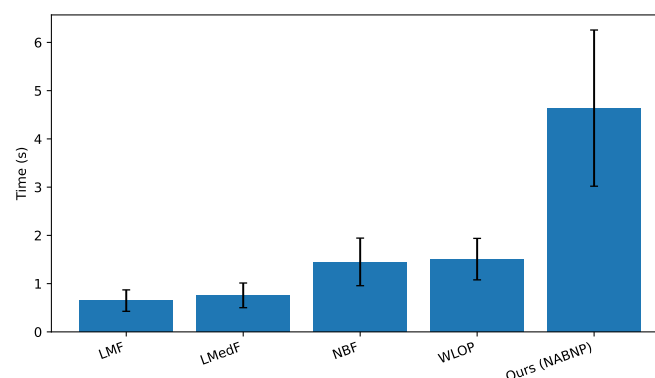
We also examined the behavior of the local noise descriptor under outlier contamination, because the descriptor is computed from the initial covariance-based tangent estimate. Table 10 summarizes the bounded noise-level coefficient nl_i on deterministic 10,000-point subsamples, together with the final NABNP HD95 values from the full benchmark. The descriptor is intentionally clipped to $[0, 1]$, and the high saturation percentages indicate that outlier-contaminated neighborhoods are frequently treated as unreliable. The marked increase in HD95 at 10% outliers, despite similar descriptor saturation, confirms the practical boundary of the local tangent-based mechanism: once many neighborhoods become geometrically ambiguous, the bounded descriptor cannot by itself distinguish moderate from severe contamination.

Table 10. Bounded local noise-level coefficient under outlier contamination.

Outliers	Mean nl_i	Mean P95 nl_i	Saturated $nl_i \geq 0.95$ (%)	HD95 $\downarrow$
1%	0.985	1.000	93.2	0.0088
2%	0.984	1.000	92.7	0.0093
5%	0.984	1.000	92.5	0.0123
10%	0.986	1.000	92.9	0.1700

4.5. Runtime and Statistical Significance

Figure 4 reports end-to-end runtime in seconds (Time_s) aggregated over all models and scenarios for the final method configurations, with error bars indicating one standard deviation. The corresponding numerical values are obtained from the same experimental runs used for accuracy evaluation. NABNP is slower than the simpler local filters because it recomputes neighborhoods, covariance statistics, noise descriptors, and bilateral normal weights at each iteration. Its runtime should therefore be interpreted as a moderate cost for adaptive learning-free filtering rather than as a speed advantage over LMF, LMedF, or NBF.

**Figure 4.** End-to-end runtime aggregated over all evaluated scenarios.

We further assess statistical significance using non-parametric tests. For each scenario, the Friedman test [41] is applied across methods using paired samples over model and

seed. Table 11 reports Friedman test statistics and p -values for CD and HD95. These tests establish whether method rankings differ within a scenario, but they are omnibus tests and should not be interpreted as pairwise proof that any single method dominates all alternatives.

Table 11. Friedman test results by scenario.

Scenario	N	χ^2_{CD}	p_{CD}	χ^2_{HD95}	p_{HD95}
Noise_0.005	50	189.04	8.52×10^{-40}	193.60	8.93×10^{-41}
Noise_0.01	50	191.02	3.19×10^{-40}	200.00	3.76×10^{-42}
Noise_0.015	50	180.80	5.02×10^{-38}	193.60	8.93×10^{-41}
Noise_0.02	50	173.31	2.03×10^{-36}	183.63	1.24×10^{-38}
Noise_0.025	50	190.40	4.35×10^{-40}	180.21	6.73×10^{-38}
Outlier_1%	50	184.30	8.87×10^{-39}	200.00	3.76×10^{-42}
Outlier_10%	50	197.06	1.61×10^{-41}	163.14	3.11×10^{-34}
Outlier_2%	50	185.74	4.35×10^{-39}	200.00	3.76×10^{-42}
Outlier_5%	50	178.45	1.61×10^{-37}	180.80	5.02×10^{-38}
Rotation_Inv	50	191.02	3.19×10^{-40}	200.00	3.76×10^{-42}
Sparsity_25%	50	190.40	4.35×10^{-40}	186.37	3.20×10^{-39}
Sparsity_50%	50	194.62	5.38×10^{-41}	200.00	3.76×10^{-42}
Sparsity_75%	50	193.60	8.93×10^{-41}	200.00	3.76×10^{-42}

Because the relative ranking among competing baselines varies across scenarios and metrics, we report the omnibus Friedman test as an overall distributional check and use the scenario-wise tables and win counts as descriptive summaries rather than as pairwise dominance claims.

4.6. Qualitative Comparison

Figure 5 presents qualitative comparisons on four representative stress cases using a common reference-surface visualization. For each row, a reference mesh is reconstructed once from the ground-truth (GT) point cloud. The GT column is shown as a neutrally shaded reference surface, whereas the remaining columns are evaluated by coloring the vertices of this shared reference mesh according to their nearest-neighbor distance to the corresponding method output. To improve visual readability, distances are clipped by a fixed fraction of the GT bounding-box diagonal and mapped to a normalized color scale, where cooler colors indicate lower deviation and warmer colors indicate larger discrepancy. The color scale is normalized within each row to support method comparison for the same model–scenario pair; therefore, color intensities should not be compared directly across different rows.

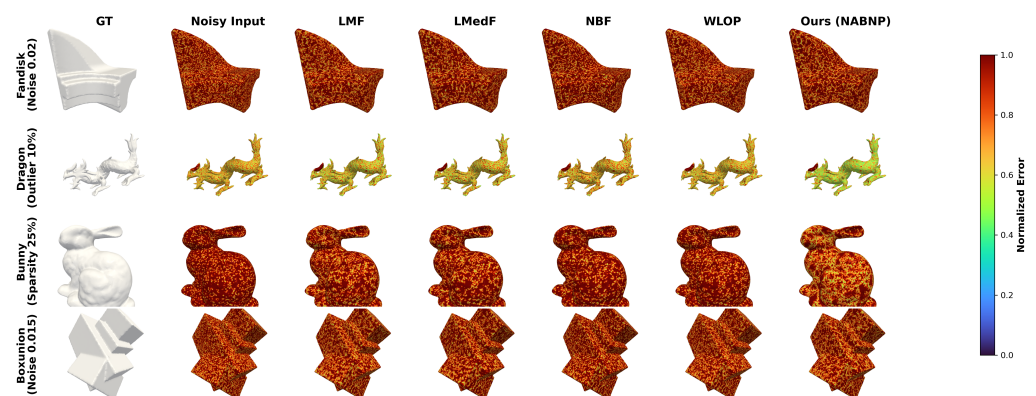


Figure 5. Qualitative comparison using a shared GT-reference visualization for four representative stress cases.

The comparison includes the GT, the noisy input, and all evaluated methods (LMF, LMedF, NBF, WLOP, and the proposed NABNP) for the following model–scenario pairs: *Fandisk* (Noise $\sigma = 0.02$), *Dragon* (Outlier 10%), *Bunny* (Sparsity 25% kept), and *Boxunion* (Noise $\sigma = 0.015$). The same viewpoint is used across all methods within each row to ensure direct visual comparison of geometric deviations.

5. Discussion

The results presented in Section 4 show that NABNP provides strong aggregate denoising behavior across the evaluated learning-free baselines while remaining geometrically interpretable and free from training data. In aggregate terms, the proposed method achieves the highest number of wins across the evaluated scenario–model combinations, particularly for HD95, which suggests that the framework is effective not only in reducing average geometric error but also in limiting larger local deviations. This behavior is important because point cloud denoising is not solely a smoothing problem; it also requires preserving the structural fidelity of the underlying surface. These win counts are descriptive summaries of the selected benchmark rather than a claim of universal superiority over all existing learning-based and non-learning methods. In this respect, the observed performance of NABNP is consistent with the long-standing objective of geometry-aware filtering methods, which seek to balance noise suppression and feature preservation without introducing excessive shrinkage or structural blurring [5,9,10,16].

The experimental trends also help explain why the proposed formulation works well in practice. Unlike classical local filters that rely on fixed support parameters or uniform update rules, NABNP adapts its behavior according to the locally estimated corruption level. This design is especially relevant in point clouds where the severity of degradation is spatially heterogeneous. The residual-based local noise descriptor controls the effective angular bandwidth used during bilateral normal refinement and also governs the balance between weighted local averaging and normal projection, together with the allowable step size. As a result, the method behaves conservatively in locally reliable regions while becoming progressively more corrective in unstable neighborhoods. This interpretation agrees with the broader literature on normal-guided and feature-preserving filtering, where the quality of the normal field is often the key determinant of the subsequent positional update [12,14,15,42].

A closer inspection across corruption types reveals a meaningful pattern. Under additive Gaussian noise, NABNP performs particularly well in the low-to-moderate regime, where local tangent estimation remains sufficiently stable and the adaptive blending mechanism can regularize points without sacrificing fine-scale geometry. Under sparsity, the method also remains highly competitive, which indicates that its local-scale normalization and weighted averaging branch respond well to irregular sampling density. The rotation experiment further confirms that the formulation is effectively rotation-invariant, as expected from its reliance on Euclidean distances, covariance-based local structure, and normal-direction residuals rather than on any preferred global orientation. These observations suggest that the main strength of NABNP lies in scenarios where local geometry is still recoverable, even if it is moderately corrupted. In such settings, the method benefits from remaining simpler and more interpretable than non-local or learning-based alternatives while still retaining substantial adaptive capacity [2,17,20,21,23].

The external learning-based reference check further contextualizes this interpretation. Pointfilter achieved lower errors than NABNP in several representative full-resolution Gaussian-noise and sparsity cases, confirming that trained neural denoisers can provide very strong accuracy when the test protocol is compatible with their learned priors. At the same time, the reference check also showed that this gain came with substantially

higher end-to-end runtime and did not extend to the tested outlier-contamination cases, where NABNP obtained the best CD and HD95 values. ScoreDenoise was less stable in the same reference protocol, particularly under outlier contamination. These observations support a more precise conclusion: NABNP is not intended to replace all trained denoisers in maximum-accuracy settings; rather, it provides a training-free and geometrically transparent alternative with favorable runtime and robust behavior under the evaluated learning-free benchmark and selected outlier reference cases.

At the same time, the results identify a clear boundary condition that should be stated explicitly. Under very strong degradation, particularly at the highest outlier ratio and at the most severe tested Gaussian noise level, the advantage of NABNP becomes smaller and may be matched or slightly exceeded by more aggressive robust statistical baselines in specific metrics. This behavior is consistent with the structure of the pipeline itself. Since the method ultimately depends on the quality of local covariance analysis and the initial tangent estimate, severe contamination of the neighborhood can degrade the reliability of both the residual statistics and the projection direction. The diagnostic results in Table 10 support this interpretation: Once the bounded local descriptor becomes saturated over most sampled neighborhoods, it acts primarily as an instability flag rather than as a finely graded measure of contamination strength. In other words, NABNP improves local robustness, but it is not an explicit outlier-removal framework and does not impose a global sparse corruption model of the kind considered in RPCA- or sparsity-driven formulations [22,30]. This is an important practical limitation, and acknowledging it strengthens rather than weakens the interpretation of the results.

From a broader perspective, NABNP occupies a useful middle ground in the current point cloud denoising landscape. Relative to lightweight classical filters, it offers substantially better adaptation to heterogeneous corruption, although this comes with higher runtime than very simple local averaging or median filters. Relative to non-local optimization methods and neural denoisers, it avoids heavy modeling overhead, training-data dependency, and reduced interpretability [18,24–28]. This trade-off is practically meaningful in settings where annotated data are unavailable, where filtering behavior must remain explainable, or where scans exhibit varying noise characteristics that are difficult to capture with a single trained model. Nevertheless, the present study is still limited to benchmark models under controlled synthetic perturbations and to a learning-free comparison group. Future work should therefore examine the behavior of the proposed patch-wise adaptation mechanism under real sensor artifacts, anisotropic acquisition noise, structured missing regions, and more severe neighborhood contamination, and should also include direct protocol-matched comparisons with recent neural denoisers where trained models and evaluation settings can be made fully reproducible. Another promising direction is to couple the current adaptive projection framework with a more explicit neighborhood reliability or outlier-screening mechanism so that the method can retain its interpretability while improving robustness in the extreme cases where purely local geometric inference becomes least reliable.

6. Conclusions

In this work, we presented Noise-Adaptive Bilateral Normal Projection (NABNP), a learning-free point cloud denoising framework designed to handle spatially heterogeneous corruption while preserving geometric interpretability. The proposed method addresses the limitations of fixed-rule geometric filters by combining robust patch-wise noise estimation, noise-adaptive bilateral normal refinement, and a scale-aware blending strategy between weighted local averaging and normal projection. Through this design, NABNP adapts its

behavior to the local reliability of the neighborhood rather than applying a single uniform update rule over the entire point set.

The proposed formulation was evaluated on standard benchmark models under four stress categories, including additive Gaussian noise, outlier contamination, sparsity, and rotation. The experimental results showed that NABNP provides strong aggregate performance within the evaluated learning-free baseline group. In particular, the method demonstrated clear advantages in preserving geometric fidelity under low-to-moderate noise levels, irregular sampling densities, and rotated inputs, while also yielding especially strong aggregate behavior in HD95. A separate full-resolution reference check with pretrained learning-based denoisers further indicated that trained models can achieve lower errors in some Gaussian-noise and sparsity cases, whereas NABNP offers much lower end-to-end runtime and strong behavior in the tested outlier cases without requiring training data. These findings support the main premise of this study, namely that explicit patch-wise adaptation can substantially improve the flexibility and feature-preserving capability of classical geometric filtering without requiring offline training or shape-specific parameter retuning.

At the same time, the results also revealed an important boundary condition. Under extremely heavy outlier contamination and the most severe tested noise level, the relative advantage of NABNP became less pronounced. This behavior reflects a practical limitation of purely local geometric inference, since the quality of the subsequent refinement and projection stages ultimately depends on the reliability of the initial local tangent estimate. Future work will therefore focus on strengthening robustness in highly contaminated neighborhoods, integrating explicit neighborhood reliability or outlier-screening mechanisms, and extending the current framework to real-world scans characterized by anisotropic sensor noise, structured missing regions, and more complex acquisition artifacts.

The performance of NABNP demonstrates that a mathematically transparent and learning-free denoising framework can effectively handle diverse geometric degradations when the local tangent structure remains sufficiently recoverable. By combining adaptability, interpretability, and practical robustness, the proposed method provides a useful solution for point cloud processing pipelines that require reproducible geometric filtering without training data.

Author Contributions: Conceptualization, A.J., A.E.K., M.H., T.T. and A.K.; methodology, A.J., A.E.K., M.H., T.T. and A.K.; software, A.J., A.E.K., M.H., T.T. and A.K.; validation, A.J., A.E.K., M.H., T.T. and A.K.; formal analysis, A.J., A.E.K., M.H., T.T. and A.K.; investigation, A.J., A.E.K., M.H., T.T. and A.K.; resources, A.J., A.E.K., M.H., T.T. and A.K.; data curation, A.J., A.E.K., M.H., T.T. and A.K.; writing—original draft preparation, A.J., A.E.K., M.H., T.T. and A.K.; writing—review and editing, A.J., A.E.K., M.H., T.T. and A.K.; visualization, A.J., A.E.K., M.H., T.T. and A.K.; supervision, A.J., A.E.K., M.H., T.T. and A.K.; project administration, A.J., A.E.K., M.H., T.T. and A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The benchmark data analyzed in this study are publicly available from the PCPNet project page at <https://geometry.cs.ucl.ac.uk/projects/2018/pcpnet/> (accessed on 25 June 2026) and the associated repository at <https://github.com/paulguerrero/pcpnet> (accessed on 25 June 2026). The synthetic perturbation scenarios were generated as described in the manuscript. The source code and reproducibility scripts are available at <https://github.com/Aekarkinli/Noise-Adaptive-Bilateral-Normal-Projection-NABNP-Point-Cloud-Denoising> (accessed on 25 June 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Guerrero, P.; Kleiman, Y.; Ovsjanikov, M.; Mitra, N.J. PCPNET: Learning Local Shape Properties from Raw Point Clouds. *Comput. Graph. Forum* **2018**, *37*, 75–85. <https://doi.org/10.1111/cgf.13343>.
- Zhang, D.; Lu, X.; Qin, H.; He, Y. Pointfilter: Point Cloud Filtering via Encoder-Decoder Modeling. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 2015–2027. <https://doi.org/10.1109/TVCG.2020.3027069>.
- Beşdok, E.; Civicioğlu, P. Analysing detail preserving capabilities of bilateral, laplacian and taubin mesh filtering methods. *Mersin Photogramm. J.* **2023**, *5*, 67–74. <https://doi.org/10.53093/mephoj.1349021>.
- Tomasi, C.; Manduchi, R. Bilateral Filtering for Gray and Color Images. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), Bombay, India, 4–7 January 1998; pp. 839–846. <https://doi.org/10.1109/ICCV.1998.710815>.
- Fleishman, S.; Drori, I.; Cohen-Or, D. Bilateral Mesh Denoising. *ACM Trans. Graph.* **2003**, *22*, 950–953. <https://doi.org/10.1145/882262.882368>.
- Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; Silva, C.T. Computing and Rendering Point Set Surfaces. *IEEE Trans. Vis. Comput. Graph.* **2003**, *9*, 3–15. <https://doi.org/10.1109/TVCG.2003.1175093>.
- Fleishman, S.; Cohen-Or, D.; Silva, C.T. Robust Moving Least-Squares Fitting with Sharp Features. *ACM Trans. Graph.* **2005**, *24*, 544–552. <https://doi.org/10.1145/1073204.1073227>.
- Guennebaud, G.; Gross, M. Algebraic Point Set Surfaces. *ACM Trans. Graph.* **2007**, *26*, 23:1–23:11. <https://doi.org/10.1145/1239451.1239474>.
- Lipman, Y.; Cohen-Or, D.; Levin, D.; Tal-Ezer, H. Parameterization-Free Projection for Geometry Reconstruction. *ACM Trans. Graph.* **2007**, *26*, 22:1–22:5. <https://doi.org/10.1145/1276377.1276405>.
- Huang, H.; Li, D.; Zhang, H.; Ascher, U.; Cohen-Or, D. Consolidation of Unorganized Point Clouds for Surface Reconstruction. *ACM Trans. Graph.* **2009**, *28*, 176:1–176:7. <https://doi.org/10.1145/1618452.1618522>.
- Kurban, T.; Beşdok, E. 3D Dimensional Point Cloud Filtering Using Differential Evolution Algorithm. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 653–657.
- Huang, H.; Wu, S.; Gong, M.; Cohen-Or, D.; Ascher, U.; Zhang, H.R. Edge-Aware Point Set Resampling. *ACM Trans. Graph.* **2013**, *32*, 9:1–9:12. <https://doi.org/10.1145/2421636.2421645>.
- Zheng, Y.; Li, G.; Wu, S.; Liu, Y.; Gao, Y. Guided Point Cloud Denoising via Sharp Feature Skeletons. *Vis. Comput.* **2017**, *33*, 857–867. <https://doi.org/10.1007/s00371-017-1391-8>.
- Zheng, Y.; Li, G.; Xu, X.; Wu, S.; Nie, Y. Rolling Normal Filtering for Point Clouds. *Comput. Aided Geom. Des.* **2018**, *62*, 16–28. <https://doi.org/10.1016/j.cagd.2018.03.004>.
- Han, X.F.; Jin, J.S.; Wang, M.J.; Jiang, W. Iterative Guidance Normal Filter for Point Cloud. *Multimed. Tools Appl.* **2018**, *77*, 16887–16902. <https://doi.org/10.1007/s11042-017-5258-9>.
- Liu, Z.; Xiao, X.; Zhong, S.; Wang, W.; Li, Y.; Zhang, L.; Xie, Z. A feature-preserving framework for point cloud denoising. *Comput.-Aided Des.* **2020**, *127*, 102857. <https://doi.org/10.1016/j.cad.2020.102857>.
- Rosman, G.; Dubrovina, A.; Kimmel, R. Patch-Collaborative Spectral Point-Cloud Denoising. *Comput. Graph. Forum* **2013**, *32*, 1–12. <https://doi.org/10.1111/cgf.12139>.
- Wang, J.; Jiang, J.; Lu, X.; Wang, M. Rethinking Point Cloud Filtering: A Non-Local Position Based Approach. *Comput.-Aided Des.* **2022**, *144*, 103162. <https://doi.org/10.1016/j.cad.2021.103162>.
- Wang, W.; Pan, W.; Liu, X.; Su, K.; Rolfe, B.; Lu, X. Random Screening-Based Feature Aggregation for Point Cloud Denoising. *Comput. Graph.* **2023**, *116*, 64–72. <https://doi.org/10.1016/j.cag.2023.08.013>.
- Hu, W.; Gao, X.; Cheung, G.; Guo, Z. Feature Graph Learning for 3D Point Cloud Denoising. *IEEE Trans. Signal Process.* **2020**, *68*, 2841–2856. <https://doi.org/10.1109/TSP.2020.2978617>.
- Zeng, J.; Cheung, G.; Ng, M.; Pang, J.; Yang, C. 3D Point Cloud Denoising Using Graph Laplacian Regularization of a Low Dimensional Manifold Model. *Trans. Img. Proc.* **2020**, *29*, 3474–3489. <https://doi.org/10.1109/TIP.2019.2961429>.
- Mattei, E.; Castrodad, A. Point Cloud Denoising via Moving RPCA. *Comput. Graph. Forum* **2017**, *36*, 123–137. <https://doi.org/10.1111/cgf.13068>.
- Rakotosaona, M.J.; La Barbera, V.; Guerrero, P.; Mitra, N.J.; Ovsjanikov, M. PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds. *Comput. Graph. Forum* **2020**, *39*, 185–203. <https://doi.org/10.1111/cgf.13753>.
- Chen, H.; Wei, Z.; Li, X.; Xu, Y.; Wei, M.; Wang, J. RePCD-Net: Feature-Aware Recurrent Point Cloud Denoising Network. *Int. J. Comput. Vis.* **2022**, *130*, 615–629. <https://doi.org/10.1007/s11263-021-01564-7>.
- Liu, Z.; Zhao, Y.; Zhan, S.; Liu, Y.; Chen, R.; He, Y. PCDNF: Revisiting Learning-Based Point Cloud Denoising via Joint Normal Filtering. *arXiv* **2022**, arXiv:2209.00798. [2209.00798].

26. Edirimuni, D.d.S.; Lu, X.; Shao, Z.; Li, G.; Robles-Kelly, A.; He, Y. IterativePFN: True Iterative Point Cloud Filtering. *arXiv* **2023**, arXiv:2304.01529. [[2304.01529](https://arxiv.org/abs/2304.01529)].
27. Edirimuni, D.d.S.; Lu, X.; Li, G.; Wei, L.; Robles-Kelly, A.; Li, H. StraightPCF: Straight Point Cloud Filtering. *arXiv* **2024**, arXiv:2405.08322. [[2405.08322](https://arxiv.org/abs/2405.08322)].
28. de Silva Edirimuni, D.; Lu, X.; Li, G.; Robles-Kelly, A. Contrastive Learning for Joint Normal Estimation and Point Cloud Filtering. *IEEE Trans. Vis. Comput. Graph.* **2024**, *30*, 4527–4541. <https://doi.org/10.1109/TVCG.2023.3263866>.
29. Feng, Z.A.; Han, X.F. Guided Normal Filter for 3D Point Clouds. *Multimed. Tools Appl.* **2023**, *82*, 13797–13810. <https://doi.org/10.1007/s11042-022-13751-w>.
30. Sun, Y.; Schaefer, S.; Wang, W. Denoising point sets via L0 minimization. *Comput. Aided Geom. Des.* **2015**, *35–36*, 2–15. <https://doi.org/10.1016/j.cagd.2015.03.011>.
31. Luo, S.; Hu, W. Score-Based Point Cloud Denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 4583–4592.
32. Rusu, R.B.; Cousins, S. 3D Is Here: Point Cloud Library (PCL). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4. <https://doi.org/10.1109/ICRA.2011.5980567>.
33. Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W. Surface Reconstruction from Unorganized Points. In Proceedings of the SIGGRAPH '92: 19th Annual Conference on Computer Graphics and Interactive Techniques, Chicago, IL, USA, 26–31 July 1992; pp. 71–78. <https://doi.org/10.1145/133994.134011>.
34. Rousseeuw, P.J.; Croux, C. Alternatives to the Median Absolute Deviation. *J. Am. Stat. Assoc.* **1993**, *88*, 1273–1283. <https://doi.org/10.1080/01621459.1993.10476408>.
35. Huber, P.J. Robust Estimation of a Location Parameter. *Ann. Math. Stat.* **1964**, *35*, 73–101. <https://doi.org/10.1214/aoms/1177703732>.
36. Hampel, F.R. The Influence Curve and Its Role in Robust Estimation. *J. Am. Stat. Assoc.* **1974**, *69*, 383–393. <https://doi.org/10.1080/01621459.1974.10482962>.
37. Beaton, A.E.; Tukey, J.W. The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. *Technometrics* **1974**, *16*, 147–185. <https://doi.org/10.1080/00401706.1974.10489171>.
38. Andrews, D.F. A Robust Method for Multiple Linear Regression. *Technometrics* **1974**, *16*, 523–531. <https://doi.org/10.1080/00401706.1974.10489233>.
39. Fair, R.C. On the Robust Estimation of Econometric Models. *Ann. Econ. Soc. Meas.* **1974**, *3*, 667–677.
40. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631. <https://doi.org/10.1145/3292500.3330701>.
41. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. <https://doi.org/10.1080/01621459.1937.10503522>.
42. Lu, X.; Wu, S.; Chen, H.; Yeung, S.K.; Chen, W.; Zwicker, M. GPF: GMM-Inspired Feature-Preserving Point Set Filtering. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 2315–2326. <https://doi.org/10.1109/TVCG.2017.2725948>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.