Research Article

Structural Feature-Preserving Point Cloud Denoising Method for Aero-Engine Profile

Jieqiong Yan,1 Laishui Zhou,1 Jun Wang,1 Xiaoping Wang,1 and Xia Liu2

1College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
2School of Mechanical Engineering, Nanjing Vocational University of Industry Technology, Nanjing 210023, China

Correspondence should be addressed to Jieqiong Yan; yanjieqiong@nuaa.edu.cn and Laishui Zhou; zlsme@nuaa.edu.cn

Received 11 October 2021; Revised 1 January 2022; Accepted 28 January 2022; Published 16 March 2022

Academic Editor: Teng Wu

Copyright © 2022 Jieqiong Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The ex-service and old type aero-engines are valuable for education. In many cases, these aero-engines only have physical objects, but lack geometric models. This brings difficulties to talent cultivation. Therefore, the education department needs to reconstruct geometric models of above aero-engines. The laser scanning devices provide raw data of aero-engine profile, but noise directly affects reconstruction accuracy. In order to ensure that noise is removed without blurring or distorting structural features, a structural feature-preserving point cloud denoising method is proposed. The noisy point cloud is divided into casing feature data, pipeline feature data and complex shape feature data. According to shape characteristics of each feature data, three denoising networks are designed to estimate position correction vectors of noisy points and project them back onto underlying surfaces. Qualitative and quantitative experiments show that our method significantly outperforms state-of-the-art methods, both in terms of preservation and restoration of structural features.

1. Introduction

The ex-service and old type aero-engines integrate the latest scientific and technological achievements in aerodynamics, engineering thermophysics, automatic control and other related fields at that time. These aero-engines are of great value in education and scientific research. At present, digital technology is increasingly being introduced into teaching process. However, due to the lack of reasonable data management and storage methods, these aero-engines only have physical objects without three-dimensional(3D) digital models. These models have wide application prospects in the teaching of aeronautics and astronautics engineering courses. Therefore, the education department put forward a demand for reverse engineering researchers to reconstruct 3D digital models of above aero-engines.

Most commonly, fully developed laser scanners produce raw point cloud representing aero-engine profile. The sequence of geometry processing methods is used to reconstruct geometry model of aero-engine profile. However, even with high-fidelity scanners, the acquired raw data contains inevitable noise from various sources. Thus, the first stage of geometry processing workflow typically involves a feature-preserving denoising procedure to preserve and even recover various features of measured objects.

The point cloud of aero-engine profile is comprised of various structural features with complex shapes, such as casing shells, reinforcing ribs, bolted flanges, pipelines, bosses, bolts, accessories, clamps, pipe joints, etc (Figures 1(d) and 1(e)). The above structural features play the roles of bearing, containing, connecting, sealing and fixing. Since the ultimate goal is to reconstruct the geometric model of aero-engine profile as accurately as possible, the main challenge is to eliminate noise while maximally preserving detailed shapes of different structural features. For example, the thin-walled structure of bolted flanges and the sharp features between reinforcing ribs and casing shells need to be preserved. The shape, size and the position relative to casing need to be preserved for each boss.

The quality of reconstructed geometry models strongly depends on the denoised point clouds. The problems of
over-smoothing and unnatural artifacts (e.g., feature blurring, shape distortion, and vertex drifts) caused by upstream denoising method will affect reconstruction accuracy. Moreover, most of existing denoising methods require fine tuning of various parameters in order to produce satisfactory results, which is difficult for non-professional users. Inspired by recent successes of deep learning for object classification [1–3], part segmentation [1–3], semantic labeling [1–3] and denoising [4], we present a structural feature-preserving point cloud denoising method named as SFP-PDNet. Specifically, the core of SFP-PDNet is a structural feature-preserving idea of denoising after segmentation. This approach differs from previous methods, which remove noise and preserve features simultaneously.

The point cloud of aero-engine profile is composed of multistage assembly of aero-engine casings and aero-engine accessory system (Figure 1). The detections of various structural features from the point cloud of aero-engine profile are formulated as segmentation problems. Identifying complex shape features from multistage assembly of aero-engine casings and aero-engine accessory system, structural feature segmentation network (SFSN) of SFP-PDNet divides the noisy point cloud of aero-engine profile into casing feature data, pipeline feature data and complex shape feature data.

SFP-PDNet includes three different denoising networks: casing feature denoising network (CFDN), pipeline feature denoising network (PFDN) and complex shape feature denoising network (CSFDN). The denoising task of each feature data is considered to be a local problem. Given a noisy point cloud $X' = [x'_1, x'_2, \ldots, x'_N]^T \in \mathbb{R}^{N \times 3}$, each kind of denoising network is applied to local patch $X'_i \in X'$, centered at point $x'_i$ with a fixed radius $r$ proportional to the point cloud’s bounding box diagonal. The patch-focused learning method of SFP-PDNet exploits all shape information collected from noisy patch $X'_i$ to estimate the position correction vector of its center $x'_i$. Along these correction vectors, noisy points are projected onto original surfaces.

A series of experiments are performed to demonstrate the robustness and effectiveness of SFP-PDNet. The main contributions of this paper are summarized as follows:

(1) SFSN produces semantically-meaningful segmentation result for the noisy point cloud of aero-engine profile. Various structural features of aero-engine profile are divided into three sets of feature data. The characteristic properties of points in the same local neighborhood of each feature data are consistent. It is remarkably accurate to estimate the correction vector of patch center from patch without interference of different feature data.

(2) The core difference of three feature denoising networks is the local neighborhood encoder. These encoders are specifically tailored according to shape characteristics of each kind of feature data. These encoders produce patch feature vectors containing shape information to subsequent regressors. The regressor of each denoising network estimates the position correction vector of each noisy point based on above shape information.
2. Related Work

We briefly review research efforts that are most relevant to our work. The previous feature-preserving denoising methods can be roughly classified into six categories as follows:

2.1. Bilateral Filtering-Based Methods. There are over-smoothed effects existing in the results of [5–9]. The reason is that the weights of smoothing filter factor and feature retention factor for different regions cannot be adjusted adaptively. Motivated by guided image filtering algorithm [10], the positions of noisy points are adjusted to match the filtered normals [11]. However, it takes a relatively long time to deal with point cloud containing large amount of points [12].

2.2. MLS-Based Methods. Moving least squares (MLS) [13] and its extensions (Robust MLS (RMLS) [14], Implicit MLS (IMLS) [15]) remove noise by iteratively projecting points to local polynomial fitting surfaces. These methods are robust to small-scale noise, but cannot handle outliers well. Projection errors occur in areas with high curvature or low sampling rate. Furthermore, convex features of input shape might not be preserved. To address the instability of projection operator, Robust Implicit (RIMLS) [16] performs bilateral filtering on the normal field before projection, which preserves sharp features to some extent. However, it is unlikely to perform well when the input shape is corrupted with high level of noise. Besides, APSS [17] suggests fitting the local neighborhoods into algebraic spheres for projection, but it may produce unnatural artifacts for detail features.

2.3. Particle-Based Methods. Locally optimal projection (LOP) [18] and its modifications (Weighted LOP (WLOP) [19], Anisotropic WLOP (AWLOP) [20], Continuous LOP (CLOP) [21]) aim at first sampling some points from input shape to become particles, then these particles are enforced to update positions and gradually approach underlying surfaces. AWLOP achieves sharp feature-preservation by anisotropically projecting points to local $L_1$-medians according to the orientations of points. But when the noise level is relatively high, the result may be over-sharpened. Besides, when input point clouds are relatively sparse, it is difficult to produce satisfactory results.

2.4. Feature-Based Methods. It should be emphasized that the aforementioned methods require users to carefully tune parameters case by case. Furthermore, the parameter settings of these methods are valid for entire point cloud, and it is difficult to adjust parameters adaptively for different areas. Thus, some researchers proposed feature-based methods [7, 22–25] to recognize and classify feature areas and non-feature areas. On this basis, different filtering methods are used to remove noise for different areas. The main disadvantage is that time cost is relatively high.

2.5. Graph-Based Methods. This class of methods represent point clouds on graphs, and perform denoising via graph filters. Osher et al. [26] proposed a low-dimensional manifold model (LDMM). Inspired by the work in [26], Zeng et al. [27] proposed a patch-based Graph Laplacian Regularizer algorithm (GLR). This algorithm exploits the inherent self-similarities between surface patches in input point cloud to preserve sharp features. Hu et al. [28] employ feature graph learning (FGL) to 3D point cloud denoising. These methods blur geometric features in varying degrees.

2.6. Learning-Based Methods. When dealing with point clouds containing complex shape features, it is difficult to determine proper input parameters of above approaches, which can eliminate noise while preserving features. In order to solve the problem of excessive smoothing and avoid repeated adjustment of parameters, researchers turned to learning-based method. Inspired by the successes of deep learning for image denoising [29–31] and mesh denoising [32, 33], and especially seminal works designed for deep learning directly on point cloud [1], Rakotosaona et al. [4] proposed PointCleanNet to remove outliers and reduce noise. PointCleanNet directly consumes raw point clouds of various shapes and noise models without parameter adjustment. Compared with above methods, the denoising effect has been greatly improved.

3. SFP-PDNet Model

The multistage assembly of aero-engine casings is connected with aero-engine accessory system through bosses, islands, accessories and clamps. The above features are identified as complex shape features by SFSN. Complex shape features are composed of several different levels of detail features. The shape of each detail feature is complex, and the position relationship between multiple detail features is not easy to determine. It is difficult to accurately describe the position relationship of each detail feature relative to the whole complex shape feature (Figure 2(a)). The casing shells, bolted flanges, rear/front mounting edges and reinforcing ribs are identified as casing features (Figure 2(b)). The pipelines are identified as pipeline features (Figure 2(c)).
We assume the following point cloud formation model:

\[
X' = C' \cup P' \cup S' = \left\{ c'_i \right\}_{i=1}^{N_C'} \cup \left\{ s'_i \right\}_{i=1}^{N_S'} \cup \left\{ p'_i \right\}_{i=1}^{N_P'}
\]

\[
= \left\{ c_i + n_i \right\}_{i \in \hat{C}} \cup \left\{ s_i + n_i \right\}_{i \in \hat{S}} \cup \left\{ p_i + n_i \right\}_{i \in \hat{P}}
\]

where \(X' = \{x'_i\}_{i=1}^{N} \in \mathbb{R}^3\) is noisy point cloud, \(C', P'\) and \(S'\) are noisy point clouds of casing, pipeline and complex shape features, respectively. \(n_i\) is additive noise caused by various reasons. The clean point cloud \(X = \{x_i\}_{i=1}^{N} \in \mathbb{R}^3\) is perfect surface samples (i.e., \(c_i, s_i, p_i \in \mathbb{R}^3\) lying on the underlying surfaces).

Due to the different shape characteristics of casing and pipeline features, SFSN defines two non-linear functions \(g\) and \(h\) to identify complex shape features from multistage assembly of aero-engine casings and aero-engine accessory system, respectively

\[
\hat{c}_i = g(X'_i)
\]

(2)

where \(c^o_i\) is the estimated probability that point \(x'_i\) is recognized as a noisy point of complex shape feature from multistage assembly of aero-engine casings. We add \(x'_i\) to \(S'\) if \(c^o_i > 0.5\). We add \(x'_i\) to \(C'\) if \(c^o_i < 0.5\)

\[
\hat{p}_i = h(X'_i)
\]

(3)

where \(p^o_i\) is the estimated probability that point \(x'_i\) is recognized as a noisy point of complex shape feature from aero-engine accessory system. We add \(x'_i\) to \(P'\) if \(p^o_i > 0.5\). We add \(x'_i\) to \(P'\) if \(p^o_i < 0.5\).

A number of local surface patches containing various structural features are constructed from the denoising dataset of SFP-PDNet by patch-focused learning method (Section 5.1). Three functions \(a, b\) and \(f\) are defined to estimate the position correction vectors of each type of feature noisy point. In the training process, the denoising network learns how to move noisy points to get as close to underlying surfaces as possible. These functions \(a, b\) and \(f\) correspond to the denoising networks of each feature data

\[
d_i^c = a(C^o_i)
\]

(4)

\[
d_i^p = b(S^o_i, S^r_i, S^l_i)
\]

(5)

\[
d_i^p = f(P^o_i)
\]

(6)

where noisy patch \(C^o_i\) is the set of \(k_c\) nearest neighbors of \(c'_i\) in \(C'\) within the radius \(r_c\). In the same way, the small patch \(S^o_i\), medium patch \(S^r_i\) and large patch \(S^l_i\) are constructed from \(S'\). The noisy patch \(P^o_i\) is constructed from \(P'\). Finally, the denoised feature points are combined into a denoised point cloud \(X^o\):

\[
\hat{X} = \hat{C} \cup \hat{S} \cup \hat{P} = \left\{ \hat{c}_i \right\}_{i=1}^{N_C'} \cup \left\{ \hat{s}_i \right\}_{i=1}^{N_S'} \cup \left\{ \hat{p}_i \right\}_{i=1}^{N_P'}
\]

\[
= \left\{ c'_i + d'_c \right\}_{c_i \in \hat{C}} \cup \left\{ s'_i + d'_s \right\}_{s_i \in \hat{S}} \cup \left\{ p'_i + d'_p \right\}_{p_i \in \hat{P}}
\]

(7)

4. SFP-PDNet Architecture

Given a noisy point cloud \(X' = [x'_1, x'_2, \ldots, x'_N] \in \mathbb{R}^{N \times 3}\), the ultimate goal is to produce a denoised point cloud \(X^o = [x^o_1, x^o_2, \ldots, x^o_N] \in \mathbb{R}^{N \times 3}\) that is closer to the clean point cloud \(X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{N \times 3}\). An overview of SFP-PDNet architecture is shown in Figure 3. Identifying complex shape features from multistage assembly of aero-engine casings and aero-engine accessory system, SFSN divides the noisy point cloud of aero-engine profile into
the noisy point clouds of casing features, pipeline features and complex shape features.

Analyzing and summarizing the shape characteristics of each type of feature data, three denoising networks estimate the position correction vectors of noisy points. Projecting noisy points back onto the underlying surfaces, the denoised point clouds are obtained with reliable structural feature preservation and recovery.

4.1. Structural Feature Segmentation Network. The point cloud of aero-engine profile contains a number of structural features. The distribution of these structural features is irregular and greatly different in volume from each other. In addition to casing features and pipeline features, complex shape features possess rich detail features (Figure 4). Thus, there are over-segmentation and under-segmentation problems in the results of existing segmentation methods for the point cloud of aero-engine profile. In contrast, SFSN neglects the shape changes inside feature data and clusters them into one segment. At the same time, SFSN focuses on the shape differences between different types of feature data and divides them into different segments (Figure 5). The results of structural feature segmentation are shown in Figure 6. As we can see, the segmentation boundaries of complex shape features are clean and clear. The accuracy of SFSN is 93.42% for 0.5% noisy point cloud, which achieves high-precision semantic segmentation. The $b$ is the original shape bounding box diagonal.

4.2. Denoising Networks. As shown in Figure 3, the denoising network of each type of feature has its corresponding neighborhood encoder, which is used to collect shape information from patches to generate patch feature vectors. According to the shape characteristics of each type of feature, three different neighborhood encoders are designed, including casing feature neighborhood encoder (CFNE), pipeline feature neighborhood encoder (PFNE) and complex shape feature neighborhood encoder (CSFNE). The multilayer perceptions added with skip connections are used as the basic feature extractor named as BasicBlock (Figure 7).

(i) The CSFNE takes a mini-batch of complex shape feature as input and outputs a patch feature vector $B \times 4224$ of complex shape feature
Figure 4: Continued.
The CFNE takes a mini-batch of casing feature as input and outputs a patch feature vector $B \times 1408$ of casing feature.

The PFNE takes a mini-batch of pipeline feature as input and outputs a patch feature vector $B \times 1024$ of pipeline feature.

Figure 4: The shape characteristics of aero-engine profile point cloud (a) complex and irregular shape (b) rich detail features (c) irregular distribution.

Figure 5: The shape changes and shape differences of aero-engine casings.
The patch centers correction vectors $d_i^c$, $d_i^p$, $d_i^s$ of each type of feature are estimated by three different regressors based on above feature vectors. Given the noisy patches $P_i^p$, $C_i^c$, $S_i^m$, $S_i^l$, the PFNE (Figure 8), CFNE (Figure 9) and CSFNE (Figure 10) first predict an affine transformation matrix by a quaternion spatial transformer network (QSTN in Figure 11). The transformation is directly applied to the coordinates of input patch points. This rotates input patches to the canonical orientation (note that the final estimated correction vectors are rotated back from canonical orientation to world space). After QSTN, three encoders use BasicBlock [3 → 64] and BasicBlock[64 → 64] to encode mini-batch into 64-dim feature space. Three intermediate feature vectors are generated as follows: $B \times 64 \times (k_i + k_m + k_l)$ (complex shape feature), $B \times 64 \times k_c$ (casing feature) and $B \times 64 \times k_p$ (pipeline feature).

4.2.1. Pipeline Feature Neighborhood Encoder. In general, after the removal of accessories, pipe joints and clamps from aero-engine accessory system, the shapes of remaining pipelines are relatively simple and do not possess detail features.
Most of pipelines are depicted as curved cylinders, which belong to smooth surfaces. Some irregular pipelines are generated based on sketch curves by stretching and sweeping. In the aero-engine accessory system, there are few free-form pipelines. Therefore, the network structure of PFNE does not need to be complicated.

The second spatial transformer operates on the intermediate feature vector in the neighborhood encoder of PointCleanNet (red box in Figure 12). Different from PointCleanNet, PFNE directly maps $B \times 64 \times k_p$ into 1024-dim feature space. The patch feature vector $B \times 1024$ is formed by extracting the maximum value from final dimension ($k_p$). The high-level feature learned from single-scale neighborhood $P_{r}$ is delivered to noisy patch regressor of pipeline feature. It expresses the overall shape change trend of current noisy patch (Figure 13). It is sufficient to predict the displacement of noisy patch center $p_r$. After testing, PFNE removal of the second spatial transformer does not significantly affect the denoising effect compared to PointCleanNet. On the contrary, it reduces the complexity of network and improves the denoising efficiency of noisy point cloud of pipeline features.

4.2.2. Casing Feature Neighborhood Encoder. The two casings are connected by several node-to-node bolted joints along the longitudinal mounting edges (Figure 14). The aero-engine casing is a kind of rotary shell part. Its overall shape is annular. A casing without complex shape features can be abstracted as a combination of multiple cylinders and cones along their axes. The casing shells, reinforcing ribs, longitudinal mounting edges and front/rear mounting edges together constitute casing body. The casing body is the basis for determining the positions of remaining structural features such as bosses, islands, grooves and bolts. The casing shells are generally cylindrical or conical surfaces. The shape transitions between casing shells are reinforcing ribs, which enhance the rigidity of blade installation positions. The mounting edges are bolt-flange structures, which are used for circumferential (longitudinal mounting edges) and axial connections (front/rear mounting edges) between
**Figure 10: CSFNE.**

**Figure 11: QSTN.**

**Figure 12: The neighborhood encoder of PointCleanNet.**
multiple casings. The sharp features between reinforcing ribs and casing shells need to be preserved. The thin-walled structure of mounting edges shall be retained during the denoising process.

In CFNE, the intermediate feature vector $B \times 64 \times k_c$ is mapped into different dimensions by BasicBlock $[64 \rightarrow 128]$, BasicBlock $[128 \rightarrow 256]$ and BasicBlock $[256 \rightarrow 1024]$. We max pool outputs of the last three layers of BasicBlock to obtain multiple dimensional features: high-level feature ($f_h=1024$), mid-level feature ($f_m=256$), low-level feature ($f_l=128$). The high-level feature expresses the overall shape change trend of sharp feature. The mid-level feature abstracts the transitional relationship between shape mutation and surrounding shape. The low-level feature shows shape mutation itself (Figure 15). In order to make shape mutation and overall shape change trend of sharp feature consistent, the assistance of transitional relationship is necessary. These $f_h$, $f_m$ and $f_l$ are concatenated to form patch feature vector $B \times 1408$. The feature vector learned from the single-scale neighborhood $C^{r}_{c}$ is delivered to noisy patch regressor of casing feature. After testing, light noise remains at sharp features between reinforcing ribs and casing shells. This can ensure that sharp edges are not over-smoothed to the utmost extent. CFNE also prevents over-sharpened artifacts at the thin-walled structure of mounting edges.

4.2.3. Complex Shape Feature Neighborhood Encoder. The complex multi-feature structure of casing is determined by the position and function of casing in aero-engine. Various aero-engine accessory systems need to be connected outside of casings, including oil system, cooling system, control system, pipelines and pumps. The islands in Figure 14 are used to connect the gliding system for attitude adjustment of aero-engine turbine blades. The bosses in Figure 14 are used to connect aero-engine pipeline system for conveying various specified fluids. Therefore, along the circumference of casing, a number of islands and bosses are distributed at some certain angles (Figure 14). The pipeline system is usually located outside of casings with brackets and clamps for vibration prevention. The accessories are involved in aero-engine start control, fuel/lubricating oil supply, working state control and other functions (Figure 16).

The shape characteristic of complex shape feature is rich detail features. Constructing multi-scale neighborhoods as the input of CSFNE has three purposes. The first is to gather as much information as possible about the shapes of detail features themselves. Second, it is to collect the position relationship between different detail features. Third, it is to collect the relationship between multiple detail features and the whole complex shape features.

Combining with Figures 10 and 17, we explain why multi-scale neighborhood is needed to recover accurate complex shape features from noisy data. The detail features of a casing boss in Figure 17 include four bolts and a hemispherical feature. The small patches describe the shape of a
The medium patches express the position relationship between a bolt and the hemispherical feature. The large patches reveal the relationship between four bolts, a hemispherical feature and the whole boss.

The high-level, mid-level and low-level features of each patch correspond to detail shapes with different degrees. The low-level feature is closest to underlying surfaces and expresses real detail shapes. The high-level feature abstracts out overall shape change trend within current local neighborhood. The mid-level feature is essential transitional information. Only by paying sufficient attention to the detail shapes of these detail features during the denoising process, the complex shape features recovered by CSFDN can be more reliable.

Consequently, as shown in Figure 10, three patches are stacked into a group, and B groups are selected as the input of CSFNE. The intermediate feature vector $B \times 64 \times (k_s + k_m + k_l)$ is mapped into different dimensions by BasicBlock [64→128], BasicBlock [128→256] and BasicBlock [256→1024]. When max pooling outputs of the last three layers of BasicBlock, we use three different max pooling layers to extract maximum value from the final dimension $(k_s + k_m + k_l)$. The low-level feature, mid-level feature and high-level feature of different scale neighborhoods are obtained. The feature vectors of each scale neighborhood are obtained by summarizing these features from feature spaces of different dimensions. These feature vectors from different patches are concatenated to form a patch feature...
vector $B \times 4224$. The feature vector learned from multi-scale neighborhoods $S^r_s, S^m_i, S^l_i$ is delivered to noisy patch regressor of complex shape feature.

5. Implementation Details

5.1. Patch-Focused Learning Method. When the neural network collects shape information directly from entire point cloud, it is highly likely to ignore the shapes of complex shape features, because there is a large volume difference between complex shape features and casing features. When complex shape features are composed of multiple detail features, the shapes of detail features are ignored more seriously. Therefore, patch-focused learning method is proposed for the training of denoising networks.

We introduce the process of constructing local surface patches by using the pipeline feature as an example. Firstly, a subset of noisy points is selected as patch centers, i.e., \( \{ p'_i \}_{i=1}^{N_p} \subset P' \) from noisy point cloud of pipeline feature $P' = \{ p'_i \}_{i=1}^{N_p}, p'_i \in \mathbb{R}^3$, and then noise patch $P'_i$ is the set of $k_p$ nearest neighbors of $p'_i$ in $P'$ with the radius $r_p$. The union of noisy patches should cover the whole pipeline feature, i.e., $\bigcup_{i=1}^{N_p} P'_i = P'$. After partitioning the whole pipeline feature into overlapping local regions, PFNE is used to abstract the patch feature vector of pipeline feature. Finally, all information collected from the patch is used to estimate displacement $d'_i$ of its center $p'_i$. The patches of each type of feature data are shown in Figure 18.

Patch-focused learning method is inspired by PointNet+ [2]. The difference is that the embedded feature of patch center is not propagated to other points in the patch. The internal shape change of patch may not conform to distance-based interpolation rule. Therefore, the interpolated features of other points in the patch obtained by point feature propagation cannot express the shapes of these points. According to wrong information, the displacement of noisy points predicted by denoising networks may yield wrong results. Patch-focused learning method does contribute to avoid this kind of error.

Another advantage of training denoising network on patches instead of entire point cloud is that we are able to sample a large number of patches from each shape. Since the point cloud of aero-engine profile possesses various

---

**Figure 17**: Illustration of complex shape feature denoising.
structural features at different positions, two patches with near-by centers may still be significantly different from each other. Thus, we utilize a relatively small set of shapes to train these denoising networks effectively.

To evenly cover entire point cloud, patch centers are selected among input shape by the iterative farthest point sampling (IFPS) [2] algorithm. In Section 6, we determine the appropriate radius of local neighborhood after taking into account the volume of each type of feature. To remove unnecessary degree of freedom from input space, we translate patch to the origin and normalize its radius by multiplying with \(1/r\). \(r\) denotes the radius of local neighborhood. Our denoising networks take a fixed number of points as input. Patches that have too few points are padded with zeros (patch center), and we pick \(k\) points closest to the patch center for patches with too many points [34].

5.2. Loss Function.

Choosing a good loss function is critical, as this choice has direct impact on the properties of the denoised point cloud. The point clouds denoised by SFP-PDNet are expected to be as close as possible to underlying surfaces. Additionally, the denoised point clouds are supposed to be evenly distributed over underlying surfaces. The initial noisy point \(x'_i\) (before denoising) performs a ball-query in clean point cloud \(X\). The local neighborhood \(X_i^r \in X\) is generated centered at point \(x'_i\) with a radius \(r\).

\[
\begin{align*}
    x_{\min} &= \arg \min_{x_j \in X_i^r} \|x'_i - x_j\| \\
    x_{\max} &= \arg \max_{x_j \in X_i^r} \|x'_i - x_j\|
\end{align*}
\]

where \(x_{\min}\) is the closest point to \(x'_i\) in \(X_i^r\), \(x_{\max}\) is the farthest point to \(x'_i\) in \(X_i^r\).

The noisy point \(x'_i\) moves along the displacement predicted by denoising network to \(x'^{\Lambda}\). We strive to relocate \(x'^{\Lambda}\) toward \(x_{\min}\) and simultaneously prevent noisy points from clustering into filament structure along the tangent direction.
Therefore, the full loss function is a weighted combination of two loss terms:

$$\text{loss} = \beta \left\| \hat{x} - x_{\min} \right\| + (1 - \beta) \left\| \hat{x} - x_{\max} \right\|$$

we set $\beta$ to 0.99 in our experiments [4].

5.3. Iterative Denoising. The prediction results of denoising network strongly depend on the contents of local neighborhoods. Two adjacent noisy patches have similar contents, and similar predictions. Therefore, residual noise still exists in the point cloud after applying displacements computed from a single iteration of SFP-PDNet. During the test process, we observe that residual noise has similar noise model as original noise, but with smaller magnitude. This means we can iterate our network on residual noise to keep improving our estimates (i.e., feeding the output of the network as the input again).

SFP-PDNet is applied iteratively to the point cloud of aero-engine profile with different noise scales (Figure 19). In order to quantitatively evaluate residual noise of denoised point clouds, we assume that clean point cloud $X$ is the ideal point cloud, and evaluate the magnitude of residual noise by calculating the chamfer distance (CD) between denoised point cloud $\hat{X}$ and clean point cloud $X$. The CD between $X$ and $\hat{X}$ is defined as follows:

$$p_{\text{cor \ ce}} = \frac{1}{|X^\wedge|} \sum_{\hat{x}_i \in \hat{X}} \min_{x \in X} \left\| \hat{x}_i - x \right\|_2$$

$$s_{\text{cor \ ce}} = \frac{1}{|X|} \sum_{x_i \in X} \min_{\hat{x}_j \in \hat{X}} \left\| x_i - \hat{x}_j \right\|_2$$

$$\text{CD} = \frac{1}{2} (p_{\text{cor \ ce}} + s_{\text{cor \ ce}})$$

where $|X^\wedge|$ and $|X|$ denote the cardinality of $X^\wedge$ and $X$, respectively. The result is shown in Figure 20, as the number of iterations increases, the magnitude of residual noise gradually decreases. After the third iteration, the noise scale of casing tends to stabilize.

In practice, we observed shrinking of the point cloud after several iterations. To counteract this shrinking, the displacement of each noisy patch center is corrected after each iterative denoising.

$$d_i^{\text{corr}} = d_i - \frac{1}{k} \sum_{x_j \in N^\wedge(\hat{x}_i)} \hat{d}_j$$

where $d_i^{\text{corr}}$ is the corrected displacement and $N(\hat{x}_i)$ are the $k$ nearest neighbors of $\hat{x}_i$. In denoised point cloud $X^\wedge$, we set $k=100$.

6. Dataset

The dataset of denoising phase of SFP-PDNet is divided into six groups (Figure 21). Three different denoising networks

![Figure 20: The relationship between residual noise and number of iterations. Three denoising networks are trained only on the basic training set (Section 6).](image-url)
Figure 21: The datasets of denoising phase for SFP-PDNet. (a) basic denoising training set. The denoising training sets with different structural features (b) complex shape feature (c) pipeline feature (d) casing feature (e) validation set (f) test set.
Table 1: The number of patches contained in each dataset.

| Dataset                              | Number of patches |
|--------------------------------------|-------------------|
| Basic denoising training set         | 56000             |
| Complex shape feature denoising      | 74000             |
| training set                         |                   |
| Casing feature denoising training    | 14000             |
| set                                   |                   |
| Pipeline feature denoising training  | 60000             |
| set                                   |                   |
| Validation set for CSFDN             | 31000             |
| Validation set for CFDN              | 6000              |
| Validation set for PFDN              | 25000             |
| Test set for CSFDN                   | 25000             |
| Test set for CFDN                    | 5000              |
| Test set for PFDN                    | 20000             |

complete the first stage of training in basic training set. They perform feature shape learning in the corresponding training sets, respectively. Three denoising networks have corresponding validation set and test set. The former is used to observe the learning process of network and to determine when to terminate training. The latter is used to evaluate the denoising capability of trained model.

We obtained the clean point clouds of 28 different shapes from PointCleanNet’s dataset (Figure 21(a)). Each point in clean point cloud is set a random movement along the normal direction to generate corresponding noisy point cloud. We set four upper limits for the magnitude of movements: 0.5%, 1.0%, 1.5% and 2.0% of original shape’s bounding box diagonal.

On the basis of measured data and photos of aero-engines (Figures 1, 2 and 4), The UG NX12.0 is used to model 25(train: validation: test = 14:6:5) pairs of casing feature and complex shape feature and 105(train: validation: test = 60:25:20) pairs of pipeline feature and complex shape feature original triangle meshes. The volume difference of different structural features is relatively large. Therefore, the clean point clouds are generated by sampling evenly from the meshes: 5k-10k points (complex shape feature), 10k-100k points (casing feature) and 3k-5k points (pipeline feature).

After adding noise with four scales to clean point clouds, the remaining datasets are constructed. An example is shown in Figure 19. We select 400 points from each shape in the basic denoising training set and 200 points from each shape in three types of feature denoising training sets to be patch centers. By observing the volume of each type of feature data, we set \( r_z = 0.03b, r_p = 0.05b, r_s = 0.01b, r_m = 0.03b, r_l = 0.07b \). The number of patches contained in each dataset is summarized in Table 1.

7. Results

We conduct the following verification experiments:

(i) We compare our method quantitatively and qualitatively to state-of-the-art deep-learning based method as well as non-deep-learning based methods, including PointCleanNet [4], AWLOP [5], GRL [27], FGL [28](Section 7.1, 7.2 and 7.3).

(ii) The comparison between different neighborhood encoder includes three parts: quantitative comparison, training time comparison and test time comparison (Section 7.4).

Quantitative evaluation metric CD distance between denoised point cloud and clean point cloud in Section 5.3 is able to measure proximity to the surface and regular distribution on the surface.

Qualitative comparison metric The visual results of above approaches are colored based on errors compared to mesh surfaces. Green denotes low error. Red denotes high error.

7.1. Comparisons of Different Methods with Four Noise Scales. Figure 22 and Table 2 are qualitative and quantitative comparison of different methods applied to four noise scales, respectively. We carefully tune the parameters of AWLOP [5], GRL [27], FGL [28] to yield satisfactory results, while PointCleanNet [4] and SFP-PDNet work across all noise scales with same trained model. These point clouds are denoised by SFP-PDNet and PointCleanNet through three iterations. In order to better observe the detail features of various structural features, the larger versions of Figure 22 are given in Figure 23.

When dealing with small-scale noise point cloud, there is residual noise (23c) in the sharp features between casing shells and reinforcing ribs (22b red circle). The order of residual amplitude from small to large is PointCleanNet, AWLOP, FGL, SFP-PDNet and GLR. The above sharp features have been largely over-smoothed in the results of AWLOP. For structural feature-preserving, our method and FGL method are superior to other three methods, among which the results of our method are better in preserving the sharp edges of island 3. The thin-walled structure of front mounting edge of the casing is the difficulty of denoising. Our method achieves a relatively good balance between noise removal and thin-walled structure retention. Front mounting edge of the casing shrinks to a plane by AWLOP, GLR and FGL.

We notice that the superior of our method is more significant when dealing with large-scale noise (22d, 22e, 22f and Table 2). The trapezoidal cross section of boss 1 (22b) obtained by our method is more faithful to the real shape. Boss 2 includes detailed features of a cylinder and a cone (22b). Our method guarantees the structural completeness of boss 2 to the utmost extent. As the noise level increases, AWLOP, GRL, FGL are unable to effectively remove noise (22e, 22f). PointCleanNet removes noise but boss 1 boss 2 and island 3 are blurred. In general, SFP-PDNet preserves sharp features between casing shells and reinforcing ribs and thin-walled structures of mounting edges and simultaneously faithfully recovers multiple bosses and islands from noisy data.

7.2. Comparisons of Different Methods with Four Aero-Engine Casings. Figure 24 shows the denoising results on
four aero-engine casings corrupted by relatively high level of noise (2.0%b). In order to better observe and compare the recovery capabilities of different methods, the larger versions of structural features (red box in Figure 24) are given in Figure 25. As we can see, all of the tested methods are able to remove noise. The following content mainly discusses the preservation of structural features.

The detail feature of boss1 is cylinder. SFP-PDNet restores its overall shape, while the cylinder is distorted into a tower feature by PointCleanNet. GLR and FGL produce unnatural artifacts in denoised results of boss1. The cylindrical casing shell where the boss2 is located has a step with other casing shell, forming a convex shoulder structure. SFP-PDNet not only preserves the sharp edges of convex shoulder, but also retains the thin-walled structure of front/rear mounting edges. On the contrary, PointCleanNet does not extract the overall shape of boss2 from noise data. The right part of boss2 is distorted to collapse, resulting in the loss of bolts. GLR and FGL blur the hemispherical detail feature of boss2 inevitably and over-smooth the sharp edges of convex shoulder.

The bosses of casing3 are located in sunken casing shell and are close to each other. In the result of our method, the denoised point cloud of each boss is independent. The overall shape of denoised boss3 is correct. On the contrary, the other three methods do not show competitive results, leaving irregular protruding point clouds. In addition, SFP-PDNet also preserves the sharp edges of casing shells on

![Figure 22: Qualitative comparison of different methods.](image)

**Table 2: Quantitative comparison of different methods (CD between denoised point clouds and clean point clouds).**

| Noise scale | 0.5%b | 1.0%b | 1.5%b | 2.0%b |
|-------------|-------|-------|-------|-------|
| Noisy point cloud | 93.5292 | 250.9772 | 397.3565 | 552.9951 |
| AWLOP | 213.0293 | 229.2286 | 257.7298 | 336.1399 |
| GLR | 85.8618 | 158.6838 | 202.2552 | 237.5155 |
| FGL | 77.3829 | 154.9883 | 202.1608 | 237.3584 |
| PointCleanNet | 87.9174 | 161.4370 | 199.0544 | 225.4781 |
| SFP-PDNet | 79.4725 | 156.3495 | 196.4974 | 221.6739 |
both sides of these bosses. The other three methods failed to recover the position relationship of these bosses in the sunken casing shell.

The casing contains both casing features (casing shells, mounting edges, reinforcing ribs) and complex shape features (bosses, islands, stiffeners). As we can see, all of the tested methods are capable of removing noise except FGL. SFP-PDNet recovers complex shape features (boss, stiffeners, island, island) with accurate shape, size and position from noise data. In contrast, the other three methods not only blur above complex shape features in varying degrees but also over-smooth sharp edges evidently (reinforcing ribs). These methods cause the shape distortion of the thin-walled structure of front/rear mounting edges.

Figure 23: The larger versions of qualitative comparison. (a) original triangle mesh (b) the larger version of features (c)0.5%b (d)1.0%b (e)1.5%b (f)2.0%b.

There are two reasons for the compelling structural feature-preserving effect of SFP-PDNet: on the one hand, SFSN obtain reasonable structural segmentation results even in face of large-scale noise. After the noisy point cloud of aero-engine profile is segmented, there is no shape interference between different feature data. This prevents detail features of complex shape features from being ignored because of the large volume difference with casing features. It lays the critical foundation for the follow-up denoising procedure. On the other hand, the denoising networks with different types of feature data have learned the shape characteristics of corresponding structural features from corresponding denoising training set. Therefore, these structural features recovered by our method are more reasonable and more reliable.
7.3. Comparisons of Different Methods with Aero-Engine Profile. SFP-PDNet and PointCleanNet are applied to the noisy point clouds of aero-engine accessory system (Figure 26) and multistage assembly of aero-engine casings (Figures 27, 28). From Figure 26, we clearly observe that SFP-PDNet generates substantially better results than PointCleanNet in terms of structural feature preservation and denoising. The boundaries of accessories with irregular shapes (area 1, 2, 5) denoised by SFP-PDNet are clearer than PointCleanNet. Comparing to PointCleanNet, the hexagonal shape of bolts from these accessories (area 3, 4, 5) and the rectangular shape of clamps from area 6, 7 are not blurred after SFP-PDNet denoising. Our approach also produces good quality results for pipelines (Area 1, 2, 3, 6, 7, 8, 9). It is easier to analyze accurate pipelines radiuses from our denoising results, which can be used to reconstruct the geometric models of pipelines.

The aero-engine in Figure 27(a) is modeled by connecting 10 casings axially. It refers to the size of an ex-service aero-engine. The noise scale of Figure 27(b) is \(0.5\%b=20.6\) mm. It means that each point in the point cloud has a position offset of about 20 mm. The selection of noise scale can demonstrate the effectiveness of SFP-PDNet to some extent. In order to better observe the bolt-flange connection structure of casing and casing assembly as well as various irregular-shaped bosses, we select the point clouds of corresponding areas for enlargement (Figure 28).

In the denoising results by PointCleanNet (Figure 28(b)), vertex drifts occur in the bolt-flange connection structure of casing and casing assembly. The sharp edges of reinforcing ribs are excessively smoothed. In contrast, SFP-PDNet maintains a high level of structural feature-preservation capability (Figure 28(a)). Furthermore, PointCleanNet blurs four irregular-shaped bosses in casing 8 (Figure 28(d)) and seven cylindrical bosses in casing 10 (Figure 28(f)), and conversely the size, position and shape of these bosses recovered by SFP-PDNet are clear and accurate (Figure 28(c), 28(e)).

7.4. Quantitative Comparisons of Different Neighborhood Encoders. In this paper, three different denoising networks are proposed. Including PointCleanNet, the difference between above four networks is the architecture of neighborhood encoder. The neighborhood encoder of PointCleanNet is shown in Figure 12. We highlight the best denoising results of four kinds of networks in Table 3. Additionally, we record comparisons of training time and test time between different networks (Table 4 and Table 5).

As shown in Table 3, our PFNE performs slightly worse than PointCleanNet. However, we can see that PFNE decreases the time cost during network training (Table 4) and reduces the complexity of network model (Figure 8 and Table 5). We can see that CFDN produces better denoising results than PointCleanNet. The reason is that patch
feature vectors yielded by CFDN contain low-level, mid-level, and high-level features, simultaneously. On the basis of CFNE, CSFNE collects shape information from multiscale neighborhoods. When dealing with small-scale noise (0.5% and 1.0%), the denoising effect is better than CFDN.

As we can see from Table 3, the denoising effect of CSFDN is inferior to CFDN when applied to large-scale noise (1.5% and 2.0%). The reason is that, in order to make a fair comparison, these four networks only train on the basic training set and act directly on entire noisy point cloud. As the input of CSFDN, the points of other structural features are coupled with the noisy points of current structural feature in the large patches. The shape information obtained from large patches may mislead the predictions of displacements to some extent. In order to preserve the shapes of other structural features, CSFDN may retain some noise of current structural feature to some extent.
This also explains the key reason why the denoising results of SFP-PDNet are obviously better than PointClean-Net. SFP-PDNet chooses to divide noisy point cloud of aero-engine profile into different types of feature data by SFSN, and then perform denoising. The advantage is that the local neighborhood of current structural feature does not contain noisy points of other structural features. This strongly ensures that points in same local neighborhood have same characteristic properties. This prevents the feature shape learning of current structural feature from being interfered by other structural features.

In terms of training time, as shown in Table 4, CFDN increases the computation cost because it needs to extract feature from different depths of network. CSFDN needs to integrate shape information from multi-scale neighborhoods, which brings impact on the training efficiency. The hardware conditions are as follows: GPU: Tesla V100-PCIE-32GB, GPU number: 2, memory: 32GB, CPU thread: 20.

From Table 5, we observe that the model complexity of CSFDN is the highest, followed by CFDN, and PFDN model is the simplest. The hardware conditions are as follows: GPU: NVIDIA GeForce GTX 1050 TI, GPU number: 1, memory: 4096 M, CPU thread: 12.

8. Conclusion

In order to remove noise without blurring or distorting structural features of aero-engine profile, a structural feature-preserving point cloud denoising method named as SFP-PDNet is proposed. Specifically, the core of SFP-PDNet is a structural feature-preserving idea of denoising after segmentation. SFP-PDNet yields reasonable structural segmentation results even in face of large-scale noise. It prevents various structural features from over-segmentation.

Table 3: Quantitative comparison of different networks (CD between denoised point clouds and clean point clouds).

| Noise scale | 0.5%b | 1.0%b | 1.5%b | 2.0%b |
|-------------|-------|-------|-------|-------|
| Noisy point cloud | 25.9509 | 61.8829 | 101.3253 | 147.6456 |
| PointCleanNet | 21.2030 | 31.7430 | 35.6857 | 39.3215 |
| PFDN | 21.5467 | 31.9884 | 35.8912 | 39.7581 |
| CFDN | 18.9026 | 30.0490 | 33.8574 | 36.9377 |
| CSFDN | 18.4810 | 29.8555 | 34.0884 | 37.7451 |
| SFP-PDNet | 16.0326 | 27.5073 | 32.4082 | 35.2707 |

Table 4: Quantitative comparison of training time, the batchsize for training is 64.

| Network | Running time of training an epoch |
|---------|----------------------------------|
| PointCleanNet | 93.60201 s |
| PFDN | 75.64919 s |
| CFDN | 108.1791 s |
| CSFDN | 122.74275 s |

Table 5: Quantitative comparison of test time, the noisy point cloud of casing contains 25,952 points.

| Network | Running time of testing a model |
|---------|--------------------------------|
| PointCleanNet | 136.96406 s |
| PFDN | 119.01124 s |
| CFDN | 144.65983 s |
| CSFDN | 330.47302 s |
and under-segmentation and lays the critical foundation for follow-up denoising procedure. The quantitative comparison shows that SFP-PDNet generates substantially better results than state-of-the-art denoising methods. The visual results show that SFP-PDNet effectively removes noise while preserving the sharp features between casing shells and reinforcing ribs and the sharp edges of stiffeners. It also retains the thin-walled structures of mounting edges. In addition, SFP-PDNet recovers the bosses, islands, clamps and accessories with accurate shape, size and position from noise data. The point cloud denoised by SFP-Net is conducive to reconstruct the geometry model of aero-engine profile.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Funding

The authors gratefully acknowledge the support of the National Key Research and Development Program of China (Grant No. 2020YFB2010702).

References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3D classification and segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 652–660, Honolulu, HI, USA, 2017.

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: deep hierarchical feature learning on point sets in a metric space,” Advances in Neural Information Processing Systems, vol. 30, 2017 http://arxiv.org/abs/1706.02413.

[3] J. H. Wang and D. L. James, “Dynamic graph CNN for learning on point clouds,” ACM Transactions on Graphics, vol. 38, no. 5, pp. 1–12, 2019.

[4] M. J. Rakotosona, V. La Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, “PointCleanNet: learning to Denoise and remove outliers from dense point clouds,” Computer Graphics Forum, vol. 39, no. 1, pp. 185–203, 2020.

[5] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang, “Edge-aware point set resampling,” ACM Transactions on Graphics, vol. 32, no. 1, pp. 1–12, 2013.

[6] X. Du, X. Jiang, C. Hao, and Y. Wang, “Bilateral filtering denoising algorithm for point cloud model,” Computer applications and software, vol. 27, no. 7, pp. 245–246, 2010.

[7] C. Shuang, Y. Jianping, and M. Wen, “Bilateral filtering denoise algorithm for point cloud based on feature selection,” Journal of Southeast University (Natural Science Edition), vol. 43, pp. 351–354, 2013.

[8] Y. Hua, P. Jiankeng, and M. Jianwen, “Denoising algorithm for bilateral filtered point cloud based on noise classification,” Journal of Computer Applications, vol. 35, no. 8, pp. 2305–2310, 2015.

[9] L. I. Pengfei, W. U. Haie, and J. I. N. G. Junfeng, “Noise classification denoising algorithm for point cloud model,” Computer Engineering and Applications, vol. 52, no. 20, pp. 188–192, 2016.

[10] K. He, J. Sun, and X. Tang, “Guided image filtering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 6, pp. 1397–1409, 2013.

[11] X. F. Han, J. S. Jin, M. J. Wang, and W. Jiang, “Iterative guidance normal filter for point cloud,” Multimedia Tools and Applications, vol. 77, no. 13, pp. 16887–16902, 2018.

[12] X.-F. Han, J. S. Jin, M. J. Wang, and W. Jiang, “Guided 3D point cloud filtering,” Multimedia Tools and Applications, vol. 77, no. 13, pp. 17397–17411, 2018.

[13] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, “Computing and rendering point set surfaces,” IEEE Transactions on Visualization and Computer Graphics, vol. 9, no. 1, pp. 3–15, 2003.

[14] S. Fleishman, D. Cohen-Or, and C. T. Silva, “Robust moving least-squares fitting with sharp features,” ACM Transactions on Graphics, vol. 24, no. 3, pp. 544–552, 2005.

[15] R. Kolliari, “Provably good moving least squares,” ACM Trans. Algorithms, vol. 4, no. 2, pp. 1–25, 2008.

[16] A. C. Öztireli, G. Guennebaud, and M. Gross, “Feature preserving point set surfaces based on non-linear kernel regression,” Computer Graphics Forum, vol. 28, no. 2, pp. 493–501, 2009.

[17] G. Guennebaud and M. Gross, “Algebraic point set surfaces,” ACM Transactions on Graphics, vol. 26, no. 3, p. 23, 2007.

[18] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, “Parameterization-free projection for geometry reconstruction,” ACM Transactions on Graphics, vol. 26, no. 3, p. 22, 2007.

[19] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, “Consolidation of unorganized point clouds for surface reconstruction,” ACM Transactions on Graphics, vol. 28, no. 5, pp. 1–7, 2009.

[20] F. Shachar, D. Iddo, and C. O. Daniel, “Bilateral mesh Denoising,” ACM Transactions on Graphics, vol. 22, no. 3, pp. 950–953, 2003.

[21] R. Preiner, O. Mattausch, M. Arikan, R. Pajarola, and M. Wimmer, “Continuous projection for fastL1reconstruction,” ACM Transactions on Graphics, vol. 33, no. 4, pp. 1–13, 2014.

[22] X. Gu, Y. Liu, and Q. Wu, “A filtering algorithm for scattered point cloud based on curvature features classification,” Journal of Information and Computational Science, vol. 12, no. 2, pp. 525–532, 2015.

[23] W. U. Lushen, H. H. Haoiang, and C. H. E. N. Huawei, “Denoising of three-dimensional point data based on classification of feature information,” Optics and Precision Engineering, vol. 24, no. 6, pp. 1465–1473, 2016.

[24] C. Liu, D. Yuan, and H. Zhao, “3D point cloud denoising and normal estimation for 3D surface reconstruction,” in 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 820–825, Zhuhai, China, 2015.

[25] Z. Liu, X. Xiao, S. Zhong et al., “A feature-preserving framework for point cloud denoising,” Computer-Aided Design, vol. 127, article 102857, 2020.

[26] S. Osher, Z. Shi, and W. Zhu, “Low dimensional manifold model for image processing,” SIAM Journal on Imaging Sciences, vol. 10, no. 4, pp. 1669–1690, 2017.

[27] J. Zeng, G. Cheung, M. Ng, J. Pang, and C. Yang, “3D point cloud Denoising using graph Laplacian regularization of a
low dimensional manifold model,” *IEEE Transactions on Image Processing*, vol. 29, pp. 3474–3489, 2019.

[28] W. Hu, X. Gao, G. Cheung, and Z. Guo, “Feature graph learning for 3D point cloud Denoising,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2841–2856, 2020.

[29] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian Denoiser: residual learning of deep CNN for image Denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[30] K. Zhang, W. Zuo, S. Gu, and L. Zhang, “Learning Deep CNN Denoiser Prior for Image Restoration,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3929–3938, Honolulu, HI, USA, 2017.

[31] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, “Deep convolutional neural network for inverse problems in imaging,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[32] P.-S. Wang, Y. Liu, and X. Tong, “Mesh denoising via cascaded normal regression,” *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 1–12, 2016.

[33] R. Benouini, I. Batioua, K. Zenkouar, S. Najah, and H. Qjidaa, “Efficient 3D object classification by using direct Krawtchouk moment invariants,” *Multimedia Tools and Applications*, vol. 77, no. 20, pp. 27517–27542, 2018.

[34] P. Guerrero, Y. Kleiman, M. Ovsjanikov, and N. J. Mitra, “PCPNet: Learning local shape properties from raw point clouds,” *Computer Graphics Forum*, vol. 37, no. 2, pp. 75–85, 2018.