3D POINT CLOUD DENOISING VIA DEEP NEURAL NETWORK BASED LOCAL SURFACE ESTIMATION

Chaojing Duan [1] *Siheng Chen* [2] *Jelena Kovačević* [3]

[1] Department of Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh, USA
[2] Mitsubishi Electric Research Laboratories, Boston, USA
[3] Tandon School of Engineering, New York University, Brooklyn, USA

ABSTRACT

We present a neural-network-based architecture for 3D point cloud denoising called neural projection denoising (NPD). In our previous work, we proposed a two-stage denoising algorithm, which first estimates reference planes and follows by projecting noisy points to estimated reference planes. Since the estimated reference planes are inevitably noisy, multi-projection is applied to stabilize the denoising performance. NPD algorithm uses a neural network to estimate reference planes for points in noisy point clouds. With more accurate estimations of reference planes, we are able to achieve better denoising performances with only one-time projection. To the best of our knowledge, NPD is the first work to denoise 3D point clouds with deep learning techniques. To conduct the experiments, we sample 40000 point clouds from the 3D data in ShapeNet to train a network and sample 350 point clouds from the 3D data in ModelNet10 to test. Experimental results show that our algorithm can estimate normal vectors of points in noisy point clouds. Comparing to five competitive methods, the proposed algorithm achieves better denoising performance and produces much smaller variances. Our code is available at https://github.com/chaojingduan/Neural-Projection.

***Index Terms*—** point cloud, denoising, deep learning, normal vector, reference plane

1. INTRODUCTION

The rapid development of 3D sensing techniques and the emerging field of 2D image-based 3D reconstruction make it possible to sample or generate millions of 3D points from surfaces of objects [1–3]. 3D point clouds are discrete representations of continuous surfaces and are widely used in robotics, virtual reality, and computer-aided shape design. 3D point clouds sampled by 3D scanners are generally noisy due to measurement noise, especially around edges and corners [4]. 3D point clouds reconstructed from multi-view images contain noise since the reconstructed algorithms fail to manage matching ambiguities [5, 6]. The inevitable noise in 3D point clouds undermines the performance of surface reconstruction algorithms and impairs further geometry processing tasks since the fine details are lost and the underlying manifold structures are prone to be deformed [7].

However, 3D point clouds denoising or processing is challenging because 3D point clouds are permutation invariant and the neighboring points representing the local topology interact without any explicit connecting information. To denoise 3D point clouds, we aim to estimate the continuous surface localized around each 3D point and remove noise by projecting noisy points to the corresponding local surfaces. The intuition is that noiseless points are sampled from surfaces. To estimate local surfaces, we parameterize them by 2D reference planes. By projecting noisy points to the estimated reference planes, we ensure that all the denoised points come from the underlying surfaces.

Deep neural networks have shown ground-breaking performances in various domains, such as speech processing and image processing [8]. Recently, several deep-learning architectures have been proposed to deal with 3D point clouds in tasks such as classification, segmentation, and upsampling [9–11]. In this work, we learn reference planes from noisy point clouds and further reduce noise with deep learning; we name the proposed algorithm as neural projection denoising (NPD). Estimated reference planes are represented by normal vectors and interceptions. The reason for using deep learning is that previous algorithms are not robust enough to noise intensity, sampling density, and curvature variations. These algorithms need to define neighboring points to capture local structures. However, it is difficult to choose the neighboring points adaptively according to the sampling density or curvature variation.

In our experiments, the point clouds used for training are sampled from the 3D dataset ShapeNet and the point clouds used for testing are sampled from the 3D dataset ModelNet10 [12, 13]. The experimental results show that NPD outperforms four of five other denoising algorithms in all seven categories and achieves the lowest variance in both evaluation metrics.

Contributions. 1) To the best of our knowledge, NPD is the first to directly deal with 3D point clouds for denois-

ing tasks with deep learning techniques; 2) NPD can estimate normal vector for each point with both local and global information and is less affected by noise intensity and curvature variation; 3) NPD can denoise noisy point clouds without defining neighboring points for noisy point clouds or calculating the eigendecomposition to estimate local geometries; 4) NPD provides the possibility of 3D point cloud parameterization with the combination of local and global information.

2. RELATED WORK

Point cloud denoising. 3D point cloud denoising has been tackled by various approaches: mesh-based denoising, graph-based denoising, and projection-based denoising. Bilateral filter (BF) and partial differential equations (PDE) are widely used in mesh and point cloud denoising [14, 15], but these mesh-based algorithms cause shrinkage and deformation [16]. Graph-based denoising (GBD) algorithms have received increasing attention because the Laplace-Beltrami operator of manifolds can be approximated by the graph Laplacian [17, 18]; however, a graph constructed from a noisy point cloud is also noisy and cannot reflect the true manifold, causing deformation issues [19]. Projection-based denoising algorithm named weighted multi-projection (WMP) in [20] estimates reference planes for 3D points and project noisy 3D points to the planes multiple times for denoising purpose. NPD estimates the reference planes with deep learning techniques and project noisy point clouds only once.

Deep learning on 3D data. Researchers first attempted to handle 3D data with deep learning techniques by voxelizing 3D shapes and applying 3D convolutional neural networks [13]. Voxelization as a 3D data pre-processing procedure is computationally intensive and leads in quantization artifacts [9]. The authors in [9] proposed an architecture that directly consumed 3D point clouds without voxelizing or converting for classification and segmentation tasks. In the paper, we adopt the idea and redesign the framework to estimate normal vectors of points in a noisy point cloud and directly denoise 3D point cloud via deep learning techniques.

3. 3D POINT CLOUD DENOISING ALGORITHM

Problem formulation. Let $\mathcal{S} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, ..., N\}$ be a 3D noiseless point cloud, where N is the total number of points in $\mathcal{S}$, and each point $\mathbf{p}_i = [x_i, y_i, z_i]^T$ is a coordinate vector. Let $\widetilde{\mathcal{S}} = \{\widetilde{\mathbf{p}}_i \in \mathbb{R}^3 \mid i = 1, ..., N\}$ represent a noisy 3D point cloud, and each point $\widetilde{\mathbf{p}}_i = [\widetilde{x_i}, \widetilde{y_i}, \widetilde{z_i}]^T$ is a coordinate vector. Note that $\widetilde{\mathbf{p}}_i = \mathbf{p}_i + \mathbf{n}_i$, where $\mathbf{n}_i$ is a 3D noise vector attached to the point $\mathbf{p}_i$ and $\mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$.

Since 3D point clouds are sampled from surfaces, they are essentially 2D manifolds embedded in 3D space. These surfaces can be locally approximated by 2D reference planes. One of our goals is learning reference planes for points with the existence of noise, but the accurate reference planes for the ground-truth surface are unknown. Unlike the work in [20] which calculate the reference planes by constructing weighted covariance matrices from noisy point clouds, NPD estimates the reference planes by learning the global and local geometries via deep learning techniques.

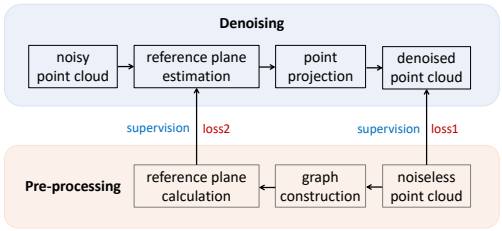

Fig. 1. **System pipeline:** During pre-processing, we calculate reference planes for points in noiseless point clouds as in [20]. During denoising, we process noisy point clouds to estimate reference planes for points. We then project noisy points to their corresponding estimated reference planes to denoise the point clouds. The noiseless point clouds and the calculated reference planes are used to supervise the denoised point clouds and the estimated reference planes, respectively.

Algorithm. The system pipeline is shown in Fig. 1. During pre-processing, we use graph-based techniques to calculate reference planes for points as in [20]. Let T_i be the reference plane with normal vector $\mathbf{a}_i$ and interception c_i calculated from noiseless point cloud for point $\mathbf{p_i}$. Let $\widehat{T}_i$ be the reference plane with normal vector $\widehat{\mathbf{a}}_i$ and interception $\widehat{c}_i$ estimated from our neural network.

During denoising procedure, we project noisy point $\widetilde{\mathbf{p}}_i$ to the estimated plane $\widehat{T}_i$ to obtain the denoised point $\widehat{\mathbf{p}}_i$ as

$$\widehat{\mathbf{p}}_i = \widetilde{\mathbf{p}}_i - \widehat{\mathbf{a}}_i^T \widetilde{\mathbf{p}}_i \widehat{\mathbf{a}}_i + \widehat{c}_i \widehat{\mathbf{a}}_i.$$

The reference plane and the noiseless point cloud are used to constraint or supervise the reference plane estimation and the denoised point cloud, respectively.

NPD architecture is shown in Fig. 2. The architecture directly takes noisy point clouds as inputs and passes through shared multi-layer perceptron (MLP), which means each perceptron is applied to the feature vectors of each point independently and identically as in [9]. The max-pooling is selecting the max value of each column of the matrix; the global information is contained in a 1D vector for each point cloud and represents a whole point cloud [9].

The local feature vector contains local information for each point. The extracted global information is concatenated to make our architecture less sensitive to sampling densities and able to learn the number of neighboring points adaptively. 3D point cloud denoising requires local information of each point since reference planes are different for different points due to their local geometries. For example, a point on a flat surface and a point on a sharp edge contain different local information and they should be treated differently. 3D point cloud denoising also requires global information for each point in a point cloud. For example, the edge distribution and

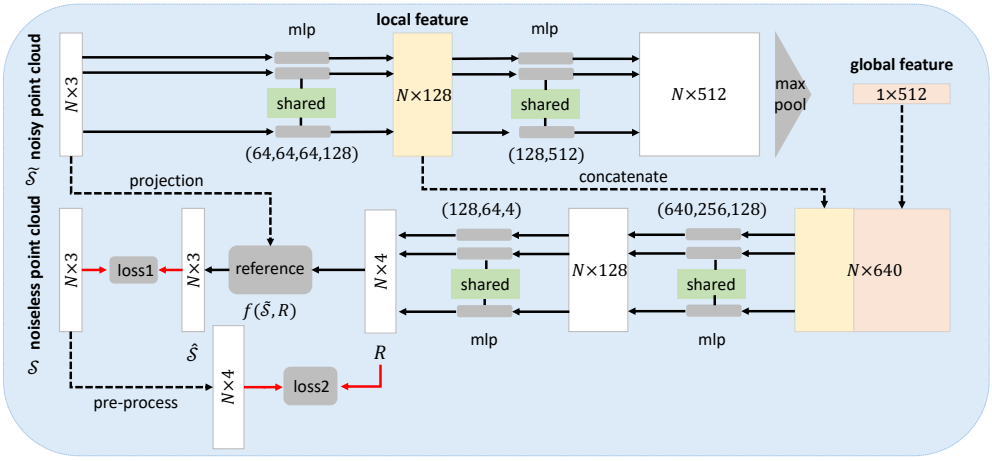

Fig. 2. **NPD architecture**: The proposed network takes noisy point clouds $\widetilde{\mathcal{S}}$ with N points as input, then passes the inputs through the shared multi-layer perceptron (MLP) to obtain local features. We mark MLP as shared because each point goes through the same MLP and the feature vector is obtained independently and identically as in [9]. The global features are captured by max-pooling, which selects the max value of each of 512 columns to produce a 1×512 global feature vector. We concatenate global and local features, then pass them through MLP to obtain reference planes represented as an $N \times 4$ matrix. A function $f(\widetilde{\mathcal{S}}, R)$ is applied to project the noisy points to the estimated reference planes to denoise point clouds. The denoised point cloud $\widehat{\mathcal{S}}$ and estimated reference planes $\widehat{T}_i$ are supervised by the noiseless point cloud $\mathcal{S}$ and the calculated reference plane T_i, respectively. We use MSE loss to calculate *loss1* (Eq. 1), which constraints the denoised point clouds to stay close to the noiseless point clouds. We use the cosine similarity loss to calculate *loss2* (Eq. 2), which constrains the estimated reference planes $\widehat{T}_i$ to be similar to the calculated reference planes T_i [20]. The number of MLP layers are shown in brackets.

the curvature variation of an airplane and a table are distinct. It is essential that each point is aware of the skeleton and the global view of the surface from which it is sampled.

We pass the combined information through shared MLPs to learn reference planes. We then project the points in noisy point clouds to their corresponding reference planes. The projected points are the denoised points sampled from the reference planes. *loss1* is treated as a point-cloud loss and we calculate mean-squared-error (MSE) between the noiseless point cloud $\mathcal{S}$ and the denoised point cloud $\widehat{\mathcal{S}}$,

$$\text{loss1}(\widehat{\mathbf{p}}_\text{i}) = \text{MSE}(\widehat{\mathcal{S}}, \mathcal{S}) = \frac{1}{\text{N}} \sum_\text{i} \|\widehat{\mathbf{p}}_\text{i} - \mathbf{p}_\text{i}\|_2^2. \quad (1)$$

loss2 is treated as a reference-plane loss and we calculate the absolute value of the cosine similarity.

$$\text{loss2}(\widehat{\mathbf{a}}_\text{i}, \widehat{\text{c}}_\text{i}) = \frac{1}{\text{N}} \sum_\text{i} |1 - \frac{[\mathbf{a}_\text{i}^\text{T}, \text{c}_\text{i}]^\text{T} [\widehat{\mathbf{a}}_\text{i}^\text{T}, \widehat{\text{c}}_\text{i}]}{\|[\mathbf{a}_\text{i}^\text{T}, \text{c}_\text{i}]\|_2 \|[\widehat{\mathbf{a}}_\text{i}^\text{T}, \widehat{\text{c}}_\text{i}]\|_2}|. \quad (2)$$

The training loss is calculated as

$$\text{loss} = (\alpha * \text{loss1} + (1 - \alpha) * \text{loss2}),$$

where $\alpha \in [0, 1]$ is a parameter that we can tune during the training for the best performance. We also show the effect of α with three small datasets in Fig. 3.

4. EXPERIMENTAL RESULTS

Dataset. We train our network with ShapeNet data, which contains 55 object categories with more than 50,000 3D mesh models [12]. We test our net with the dataset in ModelNet10, which contains 10 object categories with more than 3000 3D mesh models [13]. We select 40000 3D meshes in ShapeNet and sampled point clouds for the training procedure and each point cloud contains 2048 points; we randomly select 50 3D meshes in seven categories in ModelNet10 and sample point clouds for the testing procedure. The point clouds are rescaled into a unit cube and centered at the origin. We pre-process the training data to obtain reference planes of points and add i.i.d. Gaussian noise to construct noisy point clouds. We compare the denoising results from NPD with the state-of-the-art denoising algorithms, including BF algorithm [14], GDB algorithm [21], PDE algorithm [15], Non-local Denoising (NLD) algorithm [16] and MWP [20]. For all the algorithms, we tune parameters to produce their best performances.

Results and analysis. To quantify the performances of different algorithms, we use the metrics MSE (Eq.(1)) and Chamfer distance (CD) defined as,

$$\text{CD}(\widehat{\mathcal{S}}, \mathcal{S}) = \frac{1}{N} \Big(\sum_{\widehat{\mathbf{p}}_i \in \widehat{\mathcal{S}}} \min_{\mathbf{p}_j \in \mathcal{S}} ||\mathbf{p}_i - \widehat{\mathbf{p}}_j||_2^2 + \sum_{\mathbf{p}_i \in \mathcal{S}} \min_{\widehat{\mathbf{p}}_j \in \widehat{\mathcal{S}}} ||\widehat{\mathbf{p}}_j - \mathbf{p}_i||_2^2 \Big),$$

where $\widehat{\mathcal{S}}$ and $\mathcal{S}$ denote denoised point clouds and noiseless point clouds, respectively. We have $\widehat{\mathbf{p}}_i \in \widehat{\mathcal{S}}$, $\mathbf{p}_i \in \mathcal{S}$ and $N = |\widetilde{\mathcal{S}}| = |\mathcal{S}|$ is the total number of points.

To show the effects of α in our loss function, we vary $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$ to train the neural network with smaller datasets. MSE and CD errors are shown in Fig. 3.

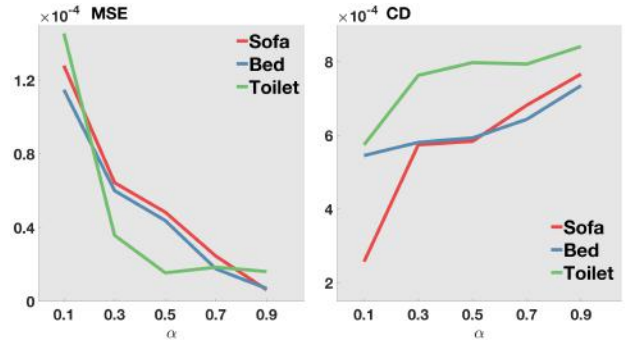

Fig. 3. For all three categories (sofa, bed, and toilet), MSE (left) decreases and CD (right) increases as α increases.

The tendencies of MSE error and CD error are different as the value of α increases. CD promotes the plane-to-plane matching; MSE promotes the point-to-point matching. 1) When $\alpha \to 0$, we restrict the predicted reference planes stay closer to the calculated reference planes. It makes the manifolds that the denoised point clouds are lying on close to the manifolds that the noiseless point clouds are lying on, which produces the smaller CD values; 2) When $\alpha \to 1$, we restrict the point clouds $\widehat{\mathcal{S}}$ calculated from the predicted reference planes stay close to the noiseless point clouds $\mathcal{S}$. It makes the denoised point clouds stay close to the noiseless point clouds, which produces the smaller MSE values.

We also train a network without providing the global feature in Fig. 2 and the errors are shown in Table 1.

Loss (10^{-4})	Local+global features		local feature only	
Category	MSE	CD	MSE	CD
Sofa	**0.627**	**0.7670**	10.845	10.4758
Bed	**0.700**	**0.7356**	13.5513	9.9097
Toilet	**1.617**	**0.8418**	11.307	9.8542

Table 1. MSE error and CD error for three categories (sofa, bed, and toilet) with and without global features. It shows that the global features significantly improve the denoising performance by providing global contextual information, which is rarely considered in most previous works on denoising.

In our experiments, we start the training with small α and increase the value of α gradually. Figs. 4 and Fig. 5 show the denoising performances in seven categories evaluated by MSE and CD (mean and standard deviation), respectively. We see that NPD algorithm outperforms four of its competitors in all seven categories by the defined metrics. NPD algorithm also produces the smallest variance for all the categories and both evaluation metrics. We analyze the reason to be that the number of point clouds and the number of object categories in the training dataset are large enough and the trained network is powerful enough to denoise all the point clouds in our testing dataset.

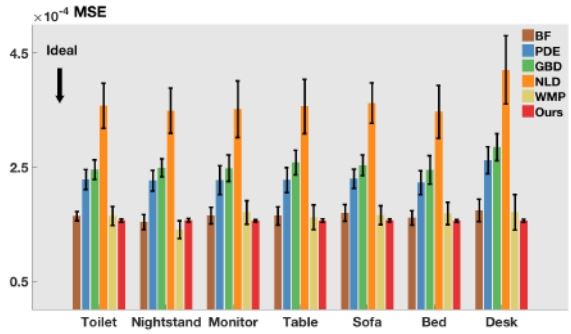

Fig. 4. The proposed algorithm (in red) outperforms all of its competitors in terms of MSE. The denoised point clouds produced by our architecture stay point-wise closer to the noiseless point clouds compared to the denoised point clouds produced by other algorithms.

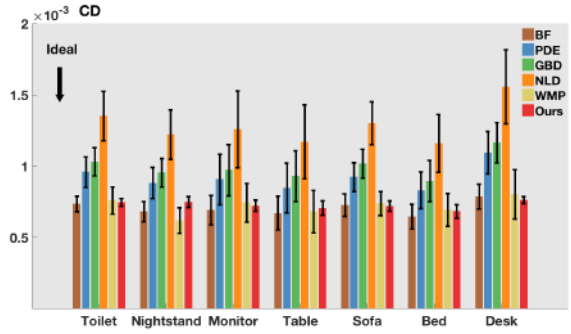

Fig. 5. The proposed algorithm (in red) outperforms its competitors in terms of CD. The denoised point clouds produced by our architecture are closer to the original point clouds or the manifolds point clouds are lying on compared to the denoised point clouds produced by other algorithms.

5. CONCLUSIONS

We propose a novel algorithm for 3D point cloud denoising by estimating reference planes for points with deep learning techniques called neural projection algorithm (NPD). Without searching for neighboring points for each point in noisy point clouds as previous algorithms, we directly estimate reference planes and project noisy points to their corresponding reference planes. We validate the proposed architecture on real datasets and the proposed architecture beats BF, PDE, GBD, and NLD algorithms. The proposed algorithm produces a much smaller variance in all seven categories for both two evaluation metrics.

6. REFERENCES

[1] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4, 2011.

[2] A. Aldoma, Z. C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 80–91, Sept 2012.

[3] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3d-tof cameras," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 1623–1630.

[4] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, Oct 2013.

[5] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, Aug 2010.

[6] H. Chen and J. Shen, "Denoising of point cloud data for computer-aided design, engineering, and manufacturing," *Engineering with Computers*, vol. 34, no. 3, pp. 523–541, 2018.

[7] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević, "Fast resampling of three-dimensional point clouds via graphs," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 666–681, Feb 2018.

[8] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 160–167.

[9] 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*, 2017, pp. 652–660.

[10] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Interpretable unsupervised learning on 3d point clouds," *arXiv preprint arXiv:1712.07262*, 2018.

[11] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-net: Point cloud upsampling network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2790–2799.

[12] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1912–1920.

[14] J. Digne and C. de Franchis, "The Bilateral Filter for Point Clouds," *Image Processing On Line*, vol. 7, pp. 278–287, 2017.

[15] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," *Trans. Img. Proc.*, vol. 17, no. 7, pp. 1047–1060, Jul. 2008.

[16] J.-E. Deschaud and F. Goulette, "Point cloud non local denoising using local surface descriptor similarity," *IAPRS*, vol. 38, no. 3A, pp. 109–114, 2010.

[17] M. Belkin and P. Niyogi, "Towards a theoretical foundation for laplacian-based manifold methods," in *Learning Theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 486–500.

[18] 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," *arXiv preprint arXiv:1803.07252*, 2018.

[19] C. Dinesh, G. Cheung, I. V. Bajic, and C. Yang, "Fast 3d point cloud denoising via bipartite graph approximation & total variation." arXiv:1804.10831, 2018.

[20] C. Duan, S. Chen, and J. Kovačević, "Weighted multi-projection: 3d point cloud denoising with tangent planes," *IEEE Global Conference on Signal and Informationn Processing*, 2018.

[21] Y. Schoenenberger, J. Paratte, and P. Vandergheynst, "Graph-based denoising for time-varying point clouds," in *2015 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. IEEE, 2015, pp. 1–4.