POINT CLOUD DENOISING USING NORMAL VECTOR-BASED GRAPH WAVELET SHRINKAGE

Ryosuke Watanabe †, *Keisuke Nonaka* †, *Haruhisa Kato* †, *Eduardo Pavez*‡, *Tatsuya Kobayashi*†, *Antonio Ortega* ‡

†KDDI Research, Inc. ‡ University of Southern California

ABSTRACT

Many applications that use point clouds, such as 3D immersive tele-presence, suffer from geometric quality degradation. This noise may be caused by measurement errors of the capturing device or by the point cloud estimation method. In this paper, we propose a novel graph-based point cloud denoising approach using the spectral graph wavelet transform (SGWT) and graph wavelet shrinkage. Unlike conventional SGWT-based denoising methods, the proposed wavelet shrinkage thresholds are determined based on the normal vector at each point and are thus based on the local geometric structure of the point cloud. This approach avoids excessive wavelet shrinkage, which can lead to the loss of complex geometric structure. Experimental results show that the proposed method achieves the best accuracy as compared with recent deep-learning-based and graph-based state-of-the-art denoising methods.

Index Terms— point cloud denoising, wavelet shrinkage, spectral graph wavelets, graph signal processing

1. INTRODUCTION

Point clouds are important in a variety of applications such as 3D immersive tele-presence [1], obstacle detection in automatic driving [2], and holographic 3D television [3]. Acquired point clouds can be constructed from depth sensors or multiple RGB cameras and are often contaminated with noise caused by the sensor's measurement error. Thus, point cloud denoising methods are needed to improve point cloud quality.

To tackle the denoising problem, methods based on graph signal processing (GSP) [4] have gained a lot of attention [5, 6, 7, 8] because they can achieve high denoising accuracy and, unlike deep-learning based methods [9, 10], do not require training data. All GSP-based denoising techniques start by selecting a graph signal transformation having properties that can facilitate separating signal and noise components. For example, the graph Fourier transform (GFT) [4] has been proposed for denoising in [5], where it was shown that it can reconstruct smooth point clouds by suppressing the high-frequency components. However, a GFT-based method cannot preserve well the local geometric structure of a point cloud because the GFT is in general not localized in the vertex domain. Block-based representations have been proposed as alternatives [11]. While the block structure ensures localization in the vertex domain, it also leads to blocking artifacts (e.g., discontinuities), which may appear at the boundary between blocks because the transformations are applied for each block independently.

Spectral graph wavelet transforms (SGWTs) [12] have become popular alternatives, as they can provide a trade-off between spectral and vertex domain localization. For this reason, denoising methods using SGWTs [6, 7, 8] can take advantage of global smoothness while preventing over-smoothing of discontinuities. Existing SGWT-based denoising methods, such as manifold spectral graph wavelet (MSGW) [6], robust piece-wise smooth manifolds (RPSM) [7], and geometry-and-color graph (GAC) [8] have achieved promising results, but also have some limitations. In MSGW, all wavelet coefficients in high frequency bands above a certain scale are discarded so that the shape of the point cloud after denoising can become inaccurate when the original noise-free signal had energy in the high frequency bands. As for RPSM, it may not be suitable for large point clouds because it requires a significant amount of memory [8]. In GAC, BayesShrink [13], a popular wavelet shrinkage technique for image denoising, is applied for soft-thresholding. Soft-thresholding has the advantage of allowing large coefficients in high frequency bands to be preserved, and the experiments in [8] reveal that memory requirements of GAC are below those of RPSM [7]. However, while GAC addresses some of the problems in MSGW and in RPSM, it makes use of color information to construct a graph for the point cloud. Our work is motivated by the fact that in some cases using color information may not be desirable. For example, the color information may be noisy or may exhibit high frequency variation even though the geometry is smooth. Moreover, depending on the sensor and the point cloud reconstruction method (e.g., LiDAR sensor, image silhouette-based 3D reconstruction [14]), there are some cases where color information cannot be obtained.

In this paper, we propose a novel denoising method, which is called the normal vector-based graph wavelet shrinkage (NVGWS). To prevent the loss of original signal components by soft-thresholding, we propose to select wavelet shrinkage thresholds that are determined based on an estimate of the normal vector to the point cloud at each point. As shown in Fig. 1, when the noiseless point cloud corresponds to a smooth surface, geometric errors (shifts in point positions) following the direction of the surface normal are likely to be highly visible, while errors that keep the points within the surface can be tolerated. Thus, we can select a smaller shrinkage threshold (stronger denoising) for coordinate directions that are aligned with the surface normal, and larger thresholds for coordinates along the surface tangent plane. Our goal is to reduce the loss of complex geometric structure due to excessive wavelet shrinkage, by taking into account the local structure of the point cloud.

2. PRELIMINARIES

2.1. Graph Construction and Frequency

We start by introducing mathematical notations and definitions of graph frequency [4]. An undirected and weighted graph $G =$

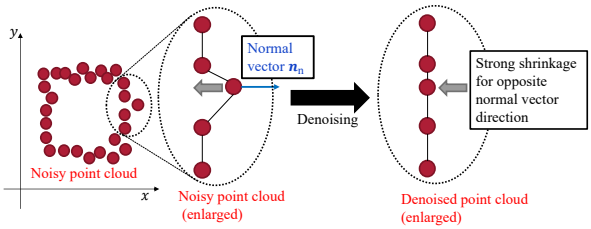


Fig. 1. Denoising based on normal vector direction.

(V, E) is constructed by a geometric point cloud $P = \{p_n\}, n = 1, ..., N, p_n \in \mathbb{R}^3$, where V and E indicate the set of nodes and edges on the graph, respectively. The adjacency matrix $\mathbf{W} = \{w_{ij}\}$, where the $ij-th$ entry, w_{ij}, is the weight of the edge between nodes i and j, is calculated by using the Gaussian kernel function as:

$$w_{ij} = \begin{cases} \exp(\frac{-||\mathbf{p_i}-\mathbf{p_j}||_2^2}{\theta^2}) & (\mathbf{p_j} \in \mathcal{N}_k(\mathbf{p_i}) \text{ or } \mathbf{p_i} \in \mathcal{N}_k(\mathbf{p_j})) \\ 0 & (\text{otherwise}), \end{cases} \tag{1}$$

where $\mathbf{p_i} \in \mathcal{N}_k(\mathbf{p_j})$ if $\mathbf{p_i}$ is one of the k nearest neighbors (kNN) of $\mathbf{p_j}$ and where θ represents the average of all pair-wise distances [1]. On the constructed graph, the combinatorial graph Laplacian $\mathbf{L}$ is calculated by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ represents the diagonal degree matrix. The combinatorial graph Laplacian characterizes the global smoothness of a graph signal $f \in \mathbb{R}^N$ because the graph Laplacian quadratic form is derived from $|\nabla \mathbf{f}|^2 = \sum_{i\sim j} w_{ij}(f(i) - f(j))^2 = \mathbf{f}^\top \mathbf{L}\mathbf{f}$, where the sum is over all pairs of nodes i and j that are connected, denoted by $i\sim j$. For this reason, the GFT $\hat{f}$ is defined as $\hat{f}(\lambda_l) = \sum_n f(n)\phi_l(n)$, where λ_l and $\phi_l(n)$ denote the l-th eigenvalue and eigenvector of $\mathbf{L}$, respectively. As in [6], we consider three graph signals f_x, f_y, f_z corresponding to each of the 3-dimensional (3D) coordinates of the points.

2.2. Spectral Graph Wavelet Transform (SGWT)

The SGWT [12] is constructed using a kernel operator $T_g = g(L)$, which acts on a graph signal f by modulating each graph Fourier mode: $T_g\hat{f}(l) = g(\lambda_l)\hat{f}(l)$. A scaled operator $T_g^t = g(t\mathbf{L})$ indicates the scaling in the spectral domain at scale t. At this time, the wavelets are calculated by applying T_g^t to a single vertex operator: $\psi_s^n = T_g^t\delta_n$ where δ_n is the impulse on a single vertex n. Spectral graph wavelet coefficients are calculated by

$$\Psi_f(t, n) = (T_g^t f)(n) = \sum_{l=1}^{N} g(t\lambda_l)\hat{f}(\lambda_l)\phi_l(n). \tag{2}$$

3. PROPOSED METHOD

3.1. Overall calculation flow

As shown in Fig. 2, the overall calculation flow of NVGWS includes the following 6 steps.

1. The combinatorial graph Laplacian $\mathbf{L}$ is calculated by the graph constructed by Eq. (1).

[1]The general principles in our approach would still be applicable even if a different graph construction method was chosen (e.g., different choice of θ).

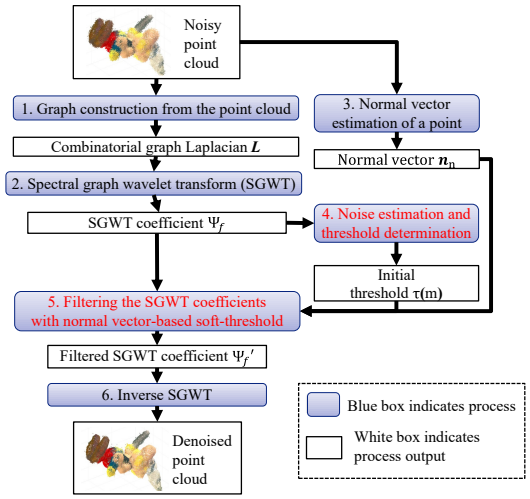


Fig. 2. Overall calculation flow of the proposed NVGWS.

2. The wavelet coefficient $\Psi_f(m, n)$ is calculated from graph Laplacian $\mathbf{L}$, where $m(m = 0, ..., M-1)$ shows the index of the wavelet sub-band.
3. The normal vector $\mathbf{n}_n \in \mathbb{R}^3$ at point p_n is calculated from the oriented tangent plane at that point, estimated from its k nearest neighbors $\mathcal{N}_k(\mathbf{p}_n)$ following the approach in [15].
4. The threshold for wavelet shrinkage is determined by BayesShrink [13] and the estimated normal vector (see Section 3.2).
5. The filtered wavelet coefficient $\Psi'_f(m, n)$ is obtained from $\Psi_f(m, n)$ with the normal vector-based soft-threshold (see Section 3.2).
6. The filtered coefficient $\Psi'_f(m, n)$ is transformed to a denoised positional signal by inverse-SGWT.

The main novelty in our work is in the SGWT coefficient filtering, steps 4 and 5 above (highlighted in red in Fig. 2) and described next.

3.2. Normal vector-based graph wavelet shrinkage

For accurately denoising a point cloud, a method for obtaining the desirable filtered coefficient $\Psi'_f(m, n)$ is essential. In the conventional soft-thresholding approach [8], excessive wavelet shrinkage leads to the loss of complex geometric structure. To address this problem, we propose a new threshold determination method for a point cloud. Figure 1 explains our denoising algorithm. As shown in Fig. 1, the position correction for the normal vector direction is more important than that for other directions to obtain a smooth point cloud and preserve the local structure. Based on this assumption, the soft-threshold for wavelet shrinkage is weighted by the magnitude of the normal vector at each point. Our threshold determination consists of 3 steps: 1) geometric noise level estimation, 2) initial threshold determination inspired by conventional image processing [13, 16], and 3) final threshold determination by using the normal vector at each point.

First, the geometric noise level σ is estimated by the median estimator [16] as:

$$\sigma = \frac{\text{Median}(|\Psi_f(m, n)|)}{0.6745}, \quad \text{for} \quad 5 \le m < M. \tag{3}$$

As with the conventional graph-based method [8], several high frequency wavelet sub-bands are utilized to estimate the noise level σ.

However, unlike in the conventional image processing setting where intensities on the regular pixel grid are independent of the pixel values, here frequency is dependent on the graph structure, because the graph signals (the coordinate signals f_x, f_y, f_z) are also used to construct the graph. As a consequence, since signal and graph are related, it is unlikely that even a noisy graph signal will have a significant amount of high frequency information. In our experiments we have observed only small amounts of energy in the high frequency bands, even under high noise conditions. In particular, when we have observed the energy under noise-free condition, almost all the energy had been concentrated on the low frequency sub-bands. Therefore, the noise level can be estimated by several high frequency sub-bands because any energy of noise-free (original) coordinate signals hardly appears in the high frequency sub-bands.

Second, the initial threshold $\tau(m)$ is calculated based on BayesShrink [13]. When the standard deviation of SGWT coefficients in each sub-band is given as $\sigma_p(m)$, the threshold $\tau(m)$ is computed as

$$\tau(m) = \min\left(\frac{\sigma^2}{\sigma'_p}, \sigma\sqrt{2\log N}\right), \tag{4}$$

where

$$\sigma'_p = \sqrt{(\max(\sigma_p^2 - \sigma^2, 0))}. \tag{5}$$

In NVGWS, the initial threshold $\tau(m)$ is selected from σ_n^2/σ'_p for BayesShrink [13], or universal threshold $\sigma\sqrt{2\log N}$ [16], as shown in Eq. (4). As described above, since the energy at the high frequency sub-band is too small, σ_n^2/σ'_p gets closer to $+\infty$ in high frequency sub-bands. As a result, all the high frequency components become 0 by using BayesShrink only. To preserve the geometric structure of an original point cloud represented by the coefficients in high frequency, the universal threshold is selected when the σ_n^2/σ'_p becomes too large.

Finally, the filtered coefficient $\Psi'_f(m,n)$ is calculated by the normal vector-based soft-threshold. In the SGWTs, the coefficient $\Psi_f(m,n)$ associates with the point n as indicated in Eq. (2). Thus, the thresholds were weighted by the magnitudes for each of the 3D coordinates $\bar{\mathbf{n}}_n = \{\bar{n}_f(n)\} \in \mathbb{R}^3$ corresponding to the normal vector $\mathbf{n}_n$ as follows:

$$\Psi'_f = \begin{cases} \Psi_f - \tau'_f & (\tau'_f < \Psi_f) \\ 0 & (-\tau'_f \leq \Psi_f \leq \tau'_f) \\ \Psi_f + \tau'_f & (\Psi_f < -\tau'_f), \end{cases} \tag{6}$$

where,

$$\tau'_f(m,n) = \tau(m)\bar{n}_f(n). \tag{7}$$

For example, when the graph signal f_x is considered, $\bar{n}_f(n)$ indicates the magnitude of the x component of $\mathbf{n}_n$.

4. EXPERIMENTAL RESULTS

4.1. Experimental conditions

We now evaluate the proposed NVGWS.

Dataset: We use the Greyc dataset [17], the Microsoft voxelized upper bodies (MVUB) dataset [18], and the ModelNet-40 [19]. To prepare the noisy point clouds, their geometric signal values f_x, f_y, f_z were corrupted with additive white Gaussian noise $\sigma = 0.5$ for the Greyc and $\sigma = 1.0$ for the MVUB. When the same σ was utilized for the Greyc and MVUB datasets, the noise level of the MVUB looked apparently small because the scales of coordinate values of two datasets were different. For this reason, the larger σ was adopted for the MVUB to equalize the noise level when we observed the point clouds. Besides, the point clouds of the ModelNet-40 were perturbed by Gaussian noise with standard deviation of 2.5% of the bounding box diagonal.

Table 1. Memory usage comparison between the propose method and RPSM [7] using the MVUB dataset [18]

Data name	The number of points	Memory usage [GByte]	
		Proposed	RPSM [7]
Andrew	279,664	0.50	582.72
David	330,797	0.59	815.29
Phil	370,798	0.67	1024.39
Ricardo	214,656	0.39	343.30
Sarah	302,437	0.54	681.49

Comparison methods: We compared NVGWS with several conventional denoising methods: IBR [5], MSGW [6], GAC [8], GPDNet [9], and DMRDenoise [10]. For the learning-based methods [9, 10], the pre-trained models provided by the authors were utilized for evaluation. The pre-trained models were trained with datasets [19, 20] which include person, furniture, and vehicle point clouds. Note that RPSM [7] was excluded from the comparison experiments. In our preliminary experiment, we investigated the memory usage of the proposed method and RPSM [7] with 5 types of point clouds in the MVUB dataset as shown in Table 1. As the memory usage of the RPSM becomes terabyte-order for a large point cloud, it is not feasible to use in a general-purpose computer or an electronic device such as a smartphone. Thus, we excluded the RPSM from the comparison experiment. Besides, since the ModelNet-40 dataset [19] does not have color attribute, the evaluation of GAC was excluded for the dataset.

Evaluation metrics: The C2C (cloud-to-cloud) and the C2P (cloud-to-plane) errors [21] were measured between the ground-truth point cloud and the denoised point cloud.

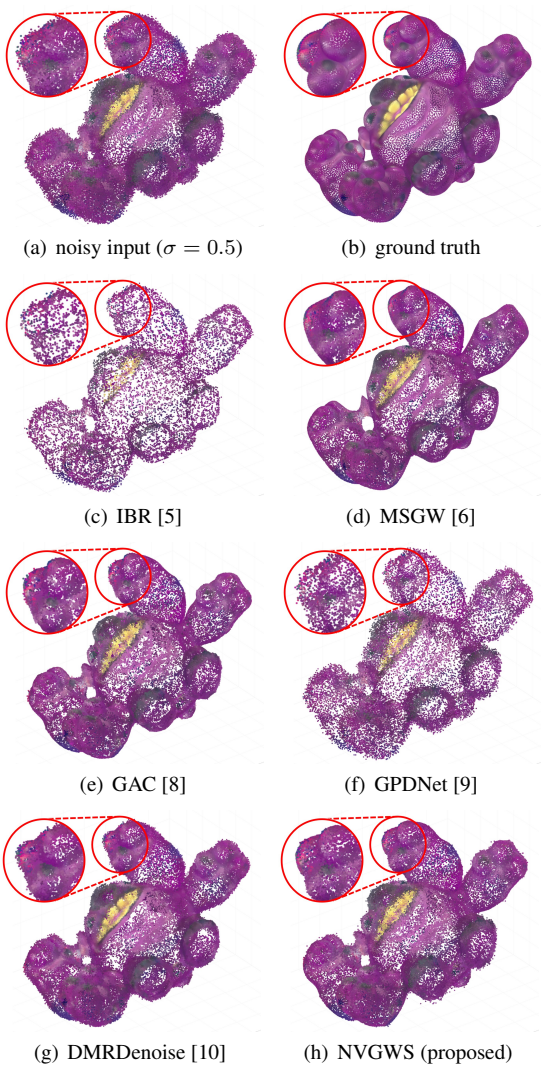
Implementation details: In NVGWS, the number of SGWT sub-bands M and k was set to 6 and 60, respectively. To ensure a fair comparison, the parameters k and M were selected to be the same as those used in the previous graph wavelet-based denoising studies [6, 8, 12]. As a band-pass filter T_g^t for SGWT, the filter having the tight frame feature [22] was adopted to preserve the energy of signals which represent the 3D position at each point through the transformation unlike the previous SGWT-based denoising methods [6, 8] which used non-tight frame. To accelerate the computation of SGWT, Chebyshev polynomial approximation whose order was $k/2$ was utilized as in [12].

4.2. Results and discussion

Quantitative evaluation: Table 2 shows the average quantitative results calculated with all point clouds in the Greyc, MVUB, and ModelNet-40 datasets. For both C2C and C2P errors, NVGWS achieved the best accuracy in the Greyc and MUVB datasets compared with other denoising methods. Moreover, while both NVGWS and GAC [8] obtained high quality denoised point clouds, our NVGWS is more broadly applicable, since GAC [8] requires color information and NVGWS does not use it. In contrast, DMRDenoise [10] outperformed the proposed method in the ModelNet-40 dataset because the pre-trained model is trained with this dataset. While the learning-based method obtained the best accuracy when a suitable training dataset is available, such datasets are not always available in practice. Thus, the proposed method is effective when it is difficult

Table 2. Quantitative evaluation results using C2C error and C2P error ($\times 10^{-3}$). Bold characters indicate the least error among all the methods.

	Greyc [17]		MVUB [18]		ModelNet-40 [19]	
	C2C	C2P	C2C	C2P	C2C	C2P
IBR [5]	0.521	165.635	0.783	24.965	0.077	4.083
MSGW [6]	0.464	291.675	0.734	43.273	0.034	1.389
GAC [8]	0.347	72.916	0.615	18.734	- (NO COLOR)	- (NO COLOR)
GPDNet [9]	0.503	156.718	1.240	104.267	0.028	1.015
DMRDenoise [10]	0.375	114.657	0.610	19.494	**0.016**	**0.245**
NVGWS (Proposed)	**0.330**	**69.036**	**0.592**	**17.983**	0.020	0.397



(a) noisy input ($\sigma = 0.5$) (b) ground truth

(c) IBR [5] (d) MSGW [6]

(e) GAC [8] (f) GPDNet [9]

(g) DMRDenoise [10] (h) NVGWS (proposed)

Fig. 3. Visualization of the denoised point clouds (σ = 0.5, 4arms_monstor model).

to prepare similar training data.

Subjective observation: Figure 3 shows the denoised point clouds of the "4arms_monstor model" in the Greyc dataset, which are rendered from a certain viewpoint. The point clouds of IBR [5], GPDNet [9], and DMRDenoise [10] were still noisy (Fig. 3(c), Fig. 3(f), and Fig. 3(g)). Also, in Fig. 3(d), the denoised point cloud obtained from MSGW [6] became too smooth and lost the detailed structures because all the high frequency components were discarded by MSGW scheme. When we compared NVGWS in Fig. 3(h) with GAC [8] in Fig. 3(e), NVGWS could reconstruct complicated structures such as the tip of a finger of the monster as shown in the red circles in Fig. 3. Moreover, while some methods such as IBR caused the holes on the point cloud surface, NVGWS is unlikely to cause many holes. This is because the distribution of points can be maintained along a surface in NVGWS because points are not shrunk in the direction along the surface. However, we confirmed that some isolated fine noise remained on the surface of point clouds denoised by NVGWS. This is because the denoising using soft-thresholding with the normal-vector did not work as well due to errors in the estimation of the normal vector at the boundary between faces whose angles change significantly.

5. CONCLUSION

This paper proposed an accurate point cloud denoising method using normal vector-based graph wavelet shrinkage (NVGWS). By focusing on the shrinkage for normal vector direction, accurate denoising was accomplished compared to conventional denoising methods. In the future, the effect of NVGWS will be confirmed by using the practical point cloud obtained with a real depth sensor. Furthermore, we will explore a more accurate noise estimation method for graph coordinate signals to enhance the denoising accuracy.

6. REFERENCES

[1] Gerd Bruder, Frank Steinicke, and Andreas Nüchter, "Immersive point cloud virtual environments," in *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, 2014, pp. 161–162.

[2] Duarte Fernandes, João Monteiro, António Silva, Rafael Névoa, Cláudia Simões, Dibet Gonzalez, Miguel Guevara, Paulo Novais, and Pedro Melo-Pinto, "Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy," *Information Fusion*, 11 2020.

[3] Mostafa Agour and Thomas Kreis, "Experimental investigation of holographic 3D-TV approach," in *2009 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, 2009, pp. 1–4.

[4] David I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[5] Yann Schoenenberger, Johan Paratte, and Pierre 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)*, 2015, pp. 1–4.

[6] Shay Deutsch, Antonio Ortega, and Gérard Medioni, "Manifold denoising based on spectral graph wavelets," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4673–4677.

[7] Shay Deutsch, Antonio Ortega, and Gérard Medioni, "Robust denoising of piece-wise smooth manifolds," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2786–2790.

[8] Muhammad Abeer Irfan and Enrico Magli, "Joint geometry and color point cloud denoising based on graph wavelets," *IEEE Access*, vol. 9, pp. 21149–21166, 2021.

[9] Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli, "Learning graph-convolutional representationsfor point cloud denoising," in *The European Conference on Computer Vision (ECCV)*, 2020, pp. 103–118.

[10] Shitong Luo and Wei Hu, "Differentiable manifold reconstruction for point cloud denoising," in *The 28th ACM International Conference on Multimedia*, 2020, pp. 1330–1338.

[11] Cha Zhang, Dinei Florêncio, and Charles Loop, "Point cloud attribute compression with graph transform," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 2066–2070.

[12] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[13] S.G. Chang, Bin Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, 2000.

[14] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 150–162, 1994.

[15] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, 1992.

[16] David L Donoho and Iain M Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[17] Anass Nouri, Christophe Charrier, and Olivier Lezoray, "Greyc 3D colored mesh database," 2017.

[18] C. Loop, Q. Cai, S. O. Escolano, and P. A. Chou, "Microsoft voxelized upper bodies - a voxelized point cloud dataset," *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012*, 2016.

[19] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3d shapenets: A deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1912–1920.

[20] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu, "ShapeNet: An Information-Rich 3D Model Repository," Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University, Princeton University, Toyota Technological Institute at Chicago, 2015.

[21] Dong Tian, Hideaki Ochimizu, Chen Feng, Robert Cohen, and Anthony Vetro, "Geometric distortion metrics for point cloud compression," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3460–3464.

[22] Nora Leonardi and Dimitri Van De Ville, "Tight wavelet frames on multislice graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3357–3367, 2013.