Point Cloud Denoising using Joint Geometry/Color Graph Wavelets

Point cloud denoising using joint geometry/color graph wavelets

Muhammad Abeer Irfan and Enrico Magli
Dept. of Electronics and Telecommunications – Politecnico di Torino, Italy

***Abstract*—A point cloud is a 3D geometric signal representation associated with other attributes such as color, normal, transparency. Point clouds usually suffer from noise due to imperfect acquisition systems. Based on the notion that geometry and color are correlated, we present a novel non-iterative framework for point cloud denoising using Spectral Graph Wavelet transform (SGW) that takes advantage of this correlation and performs denoising in the graph frequency domain. The proposed approach is based on the design of a joint geometry and color graph that compacts the energy of smooth graph signals in low-frequency bands. We then apply soft-thresholding to remove the noise from the spectral graph wavelet coefficients. Experimental results show that the proposed technique significantly outperforms state-of-the-art methods.**

***Index Terms*—Graph signal processing, point cloud denoising, Spectral Graph Wavelets**

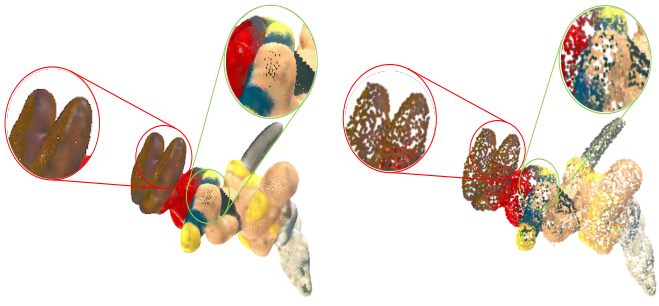
I. Introduction

A point cloud is an effective way to represent volumetric data in 3D space [1], [2], [3]. Point clouds are unordered collections of points in 3D space, where each point is comprised of 3D coordinates plus attributes such as color, normals, transparency. Point clouds have now been extensively used in many fields such as culture and heritage reconstruction, 3D broadcasting, navigation of unmanned vehicles [4]. The acquired point clouds from the active sensors such as Microsoft Kinect© or generated by stereo matching algorithms [5] often suffer from geometry noise and therefore point cloud denoising should be performed to improve their quality.

Significant amount of work has been done for point cloud denoising [6], [7], [8], [9]. The local tangent space based graph is considered for robust denoising of piece-wise smooth manifolds (RPSM) [10]. Data-driven approaches [5], [11] have shown good results for point cloud denoising, but may not be applicable where no dataset of noise-free point clouds is available. In recent literature, an iterative-based regularization technique (IBR) constructs geometry-only based graph and uses it for point cloud denoising using convex optimization [6]. The same graph has also been exploited for manifold denoising based on Spectral Graph Wavelet (MSGW) [12]. The drawback of these techniques is that the estimation of the correct position of the points is based on noisy geometry; this may lead to erroneous estimation of the local surface, and consequently to generate artefacts in the resulting denoised point cloud (see Fig. 1).

In this paper, we address the point cloud geometry denoising problem by introducing a novel framework which jointly uses geometry and color attribute of points. The rationale is that on a smooth surface the color is also typically smooth; this notion has been recognized in various point cloud applications [13], [14], [15]. The proposed algorithm uses Spectral Graph Wavelet (SGW) [16], providing a trade-off between vertex and spectral-domain localization. This allows us to take advantage of global smoothness i.e., lower frequency bands contain most of the energy, while avoiding to over-smooth discontinuities which corresponds to large magnitude coefficients in the high frequency bands.

In particular, the proposed method constructs a graph based on *both* geometry and color attribute of each point, and then applies the SGW to the corresponding graph signal. SGW based technique is effective for point cloud denoising, since in the graph signal defined on constructed graph, most of the energy is concentrated in lower frequencies. Furthermore, the joint geometry/color graph provides improved spectral separation between signal and noise, which make the noise removal easier. Most recent point cloud denoising algorithms are iterative and may also be sensitive to the parameters selection [17], [18], [6], [7]. On the other hand, our proposed technique is non-iterative and has low computational cost, which scales linearly with the data size; it is also robust to parameter selection. Experimental results show that our algorithm effectively exploits the correlation between geometry and color to relocate each point towards its original position, avoiding the artefacts generated by other techniques. We show that the proposed algorithm outperforms state-of-the-art methods using both qualitative and quantitative evaluation metrics.



(a) (b)

Fig. 1: Asterix model: (a) Ground-truth, (b) denoised with IBR [6]. The noisy points are moved towards their nearest neighbors rather than their original positions, generating holes in the denoised point cloud.

II. Proposed approach

A. Graph nomenclature

An undirected weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consists of a finite set of m vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$ defined as $(i, j, w_{i,j})$, where $i, j \in \mathcal{V}$ and each edge has a non-negative weight $w_{i,j} \in \mathbb{R}^+$ that reflects the affinity between node i and j. The corresponding *adjacency matrix* $\mathbf{W}(i, j) = w_{i,j}$ is a real symmetric $m \times m$ matrix. Let $\mathbf{D}$ be the diagonal degree matrix with entries $\mathbf{D}(i, i) = \sum_j w_{i,j}$. Given $\mathbf{W}$ and $\mathbf{D}$, the combinatorial graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The graph signal $g(\mathcal{G})$ for a given graph $\mathcal{G}$ is defined on the vertices of a graph, as $g : \mathcal{V} \rightarrow \mathbb{R}^D$ for some dimension D.

B. Spectral graph wavelet preliminaries

In this section, we establish a mathematical notation and definition of SGW [16]. For a function g defined on $\mathcal{G}$, its Graph Fourier Transform (GFT) $\hat{g}$ is defined as $\hat{g}(l) = \langle \chi_l, g \rangle = \sum_{i=1}^{m-1} g(i)\chi_l(i)$, where χ_l represents the eigenvectors of the graph Laplacian $\mathbf{L}$ and eigenvalues are denoted as λ_l for $l = 0, \ldots, m-1$. SGW [16] defines a scaling operator in the Graph Fourier domain based on λ_l. In particular, SGW are constructed using a kernel f, the wavelet operator $T_f = f(\mathbf{L})$ acts on a given function g by modulating each Fourier mode as:

$$\widehat{T_f\, g}(l) = f(\lambda_l)\hat{g}(l) \tag{1}$$

The inverse transform is defined as:

$$(T_f\ g)(n) = \sum_{l=0}^{m-1} f(\lambda_l)\hat{g}(l)\chi_l(n) \tag{2}$$

The wavelet operator at scale s is defined as $T_f^s = f(s\mathbf{L})$, the SGW is then computed by localizing the operators T_f^s by applying them to the impulse δ on a single vertex: $\psi_{s,n} = T_f^s \delta_n$. The wavelet coefficients of a given graph signal g are computed by taking the inner product with these wavelets as $\Psi_g(s, n) = \langle \psi_{s,n}, g \rangle$. The $\Psi_g(s, n)$ can be achieved directly from the wavelet operators T_f^s using the orthonormality of χ as:

$$\Psi_g(s, n) = (T_f^s g)(n) = \sum_{l=0}^{m-1} f(s\lambda_l)\hat{g}(l)\chi_l(n) \tag{3}$$

The naive approach for computing SGW from Eq. 3 requires explicit calculation of the entire χ, which scales poorly for large graphs. Another approach is to diagonalize $\mathbf{L}$ for direct computation of SGW transform, which is feasible only for graphs with less than a thousand vertices [16]. Dealing with real-world and synthetic point clouds with large number of points, the SGW can be calculated with a fast algorithm using low order polynomials, which approximate the scaled generating kernels. At each scale, the wavelet coefficients can be computed by applying a polynomial of $\mathbf{L}$ to the input data. This results in lower computational cost when the graph is sparse. The scaling and wavelet coefficients of an input $[m \times D]$ graph signal are efficiently computed using Chebyshev polynomial approximation [19], [16] and then mapped to $[m * (I - 1) \times D]$, where I represents the number of wavelet decomposition levels.

C. Joint geometry and color k-NN graph

A point cloud is represented as $\mathcal{P} = \{p_1, p_2, p_3, ..., p_N\}$ with $p_i \in \mathbb{R}^6$ containing 3D geometry and RGB color information for point p_i. The six-dimensional feature of each point is $p_i = [X_i, C_i]$, where $X_i = [x_i\ y_i\ z_i] \in \mathbb{R}^3$ is the geometric coordinate vector and $C_i = [R_i\ G_i\ B_i] \in \mathbb{R}^3$ are the color attributes. A common approach is to construct a k nearest neighbor (k-NN) graph based on Euclidean distance to makes geometric structure explicit [20]. Our denoising algorithm generates k-NN graph based on both coordinates proximity and similarity in color of points p_i.

Construction of k-NN color-only graph is not a good choice as geometrically distant points may have similar color but different semantic content, which may lead to construction of a wrong graph, while geometry-only k-NN graph generates artefacts as anticipated in Sec. I. The joint geometry and color graph is helpful as it exploits more information about the point cloud. In order to construct such graph for a given point cloud $\mathcal{P}$, the weight of the edge between node i and j has to be computed. A common option is the threshold Gaussian kernel [21]:

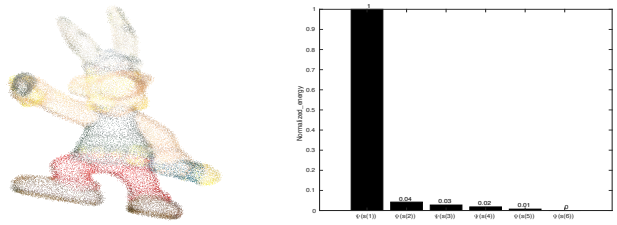
$$w_{i,j} = \begin{cases} \exp\left(-\frac{\|X_i - X_j\|^2}{2\theta_X^2} - \frac{\|C_i - C_j\|^2}{2\theta_C^2}\right) & \text{if } p_j \in \phi_k(i) \\ & \text{or } p_i \in \phi_k(j) \\ 0 \qquad \text{otherwise} \end{cases} \tag{4}$$

Here, θ_X and θ_C determine the relative contribution of geometry and color in the construction of joint geometry and color graph. $\phi_k(i)$ is the set of k nearest neighbors to point p_i, and $\phi_k(j)$ is the set of k nearest neighbors to point p_j. The resulting k-NN graph is denoted as $\mathcal{G}$.

D. Geometry denoising

The objective of the proposed algorithm is to denoise the geometry by exploiting the constructed joint geometry/color k-NN graph $\mathcal{G}$. The given noisy geometry attribute of each point p_i can be expressed as $X_i = x_i + n$, x_i being the unknown true position of a point p_i and n is the geometry noise, with $X_i,\ x_i,\ n \in \mathbb{R}^3$. The objective is to estimate x_i for each point in a point cloud. This can be performed using the proposed denoising technique based on SGW. After the construction of an undirected graph $\mathcal{G}$ using Eq. 4 and defining the graph Laplacian $\mathbf{L}$ from $\mathbf{W}$, we take the SGW transform using low order polynomials to establish tight vertex localization of SGW coefficients. For each SGW bands i.e., $\Psi_{X_i}(s(i))$ for $1 \leq i \leq I$, SGW coefficients are computed for the corresponding noisy signal X_i and preserve all the scaling and wavelet coefficients that corresponds to a low frequency wavelet band s. Denoising of $\Psi_{X_i}(s(i), n)$ is then performed by soft-thresholding [22]

based on the property that energy of the signal is concentrated in the low frequency spectral wavelet bands as anticipated in Sec. I and shown in Fig. 2. In the last, an inverse spectral wavelet transform of the denoised SGW coefficients $\Psi^*_{X_i}(s(i))$ is taken to obtain the denoised point cloud $\mathcal{Q}$.



(a) (b)

Fig. 2: (a) Noisy Asterix model with $\mu = 0, \sigma = 0.2$ (b) Normalized energy of $\Psi_{X_i}(s(i), n)$, where $1 \leq i \leq I$ and $n = 1 \ldots . N$.

III. Experimental Results

A. Evaluation metrics

The metrics used for the performance evaluation of the proposed algorithm are the same as in [23]. Assume that $\mathcal{P}$ and $\mathcal{Q}$ represent the geometry of the original and denoised point cloud respectively, where $\mathcal{P} = \{p_i\}_{i=1}^{N_1}$ $\mathcal{Q}$=$\{q_i\}_{i=1}^{N_2}$, such that p_i, $q_i \in \mathbb{R}^3$.

1) Mean-square-error (MSE): It is computed as an average of the squared Euclidean distance between each point in $\mathcal{P}$ and its corresponding nearest point in $\mathcal{Q}$, and also between each point in $\mathcal{Q}$ and its corresponding nearest point in $\mathcal{P}$:

$$\text{MSE} = \frac{1}{2N_1} \sum_{p_i \in \mathcal{P}} \min_{q_i \in \mathcal{Q}} \|p_i - q_i\|_2^2 + \frac{1}{2N_2} \sum_{q_i \in \mathcal{Q}} \min_{p_i \in \mathcal{P}} \|q_i - p_i\|_2^2 \tag{5}$$

2) Mean city-block distance (MCD): MCD uses l_1 norm instead of l_2 norm.

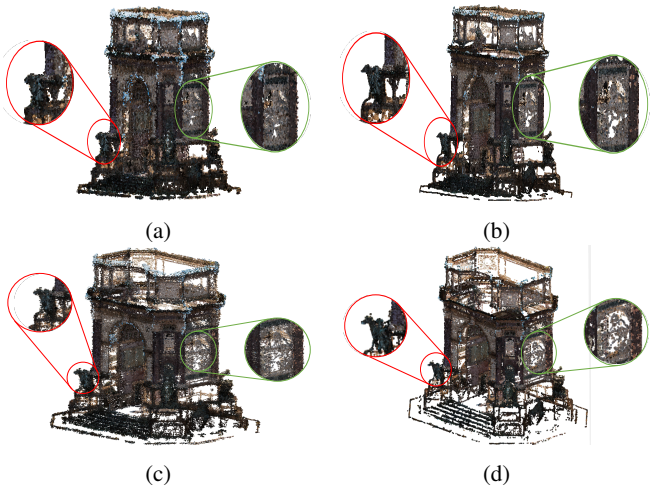
$$\text{MCD} = \frac{1}{2N_1} \sum_{p_i \in \mathcal{P}} \min_{q_i \in \mathcal{Q}} \|p_i - q_i\| + \frac{1}{2N_2} \sum_{q_i \in \mathcal{Q}} \min_{p_i \in \mathcal{P}} \|q_i - p_i\| \tag{6}$$

B. Experimental setup

We set $I = 6$ wavelet decomposition levels, and preserve the wavelet coefficients of $s(1)$ scale and then add with the wavelet coefficients which are above the threshold τ for the corresponding noise level in the $s(i)$ for $2 \leq i \leq I$. In order to denoise the point cloud locally by approximating the spectral wavelet coefficients, the Chebyshev polynomial approximation order is set to $k/2$ for a k-NN graph. All the parameters used in this paper are given in Table I.

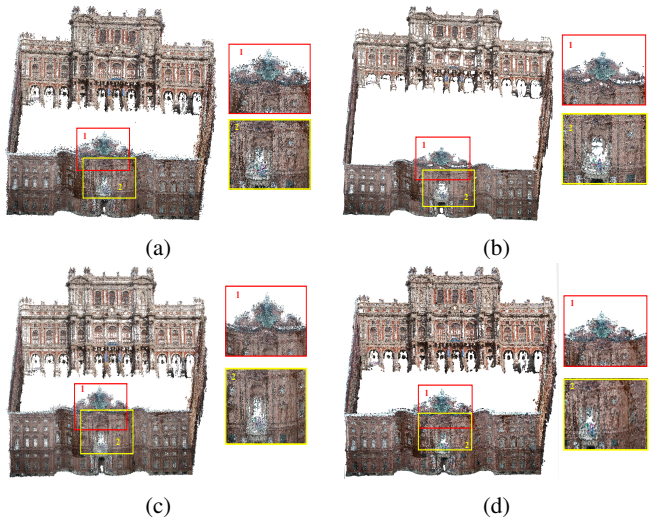
TABLE I: Parameter values used for the proposed, MSGW [12], IBR [6] and RPSM [10].

Parameter	Proposed	MSGW [12]	IBR [6]	RPSM [10]
k	30	60	20	10
θ_X	0.8	–	–	–
θ_C	0.2	–	–	–
γ	–	–	0.50 ($\sigma = 0.2$, 0.3) 0.70 ($\sigma = 0.4$)	–
τ	0.2 ($\sigma = 0.2$) 0.3 ($\sigma = 0.3$) 0.4 ($\sigma = 0.4$)	0.2 ($\sigma = 0.2$) 0.3 ($\sigma = 0.3$) 0.4 ($\sigma = 0.4$)	–	–



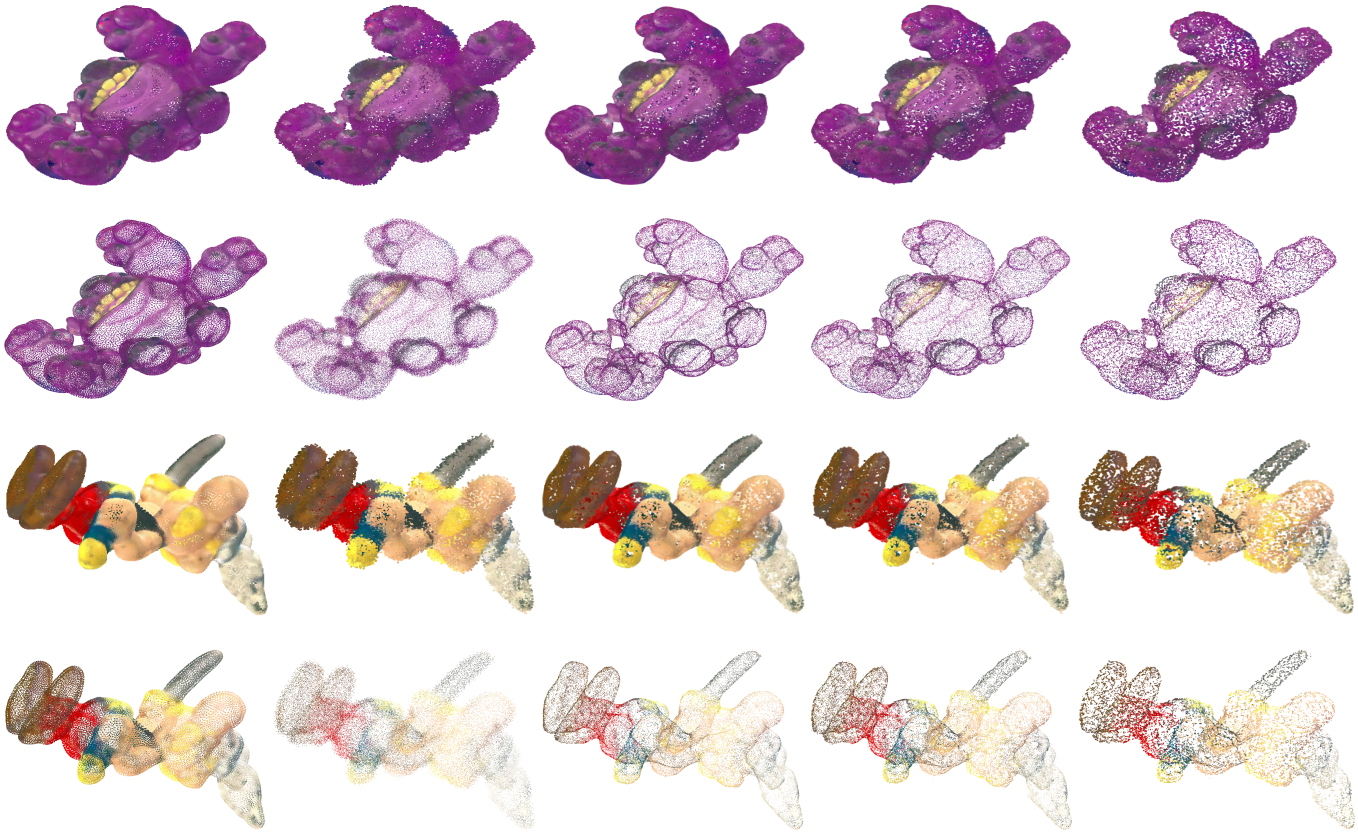
(a) (b) (c) (d)

Fig. 3: Arco_Valentino model. (a) Noisy input, denoised results by (b) Proposed algorithm, (c) MSGW [12] applied to outlier-free input, and (d) IBR performed after the outlier removal step [6].



(a) (b) (c) (d)

Fig. 4: Palazzo_Carignano model. (a) Noisy input, denoised results by (b) Proposed algorithm, (c) MSGW [12] applied to outlier-free input, and (d) IBR performed after the outlier removal step [6].



(a) (b) (c) (d) (e)

Fig. 5: Point cloud models with color: (a) ground-truth, (b) noisy input ($\mu = 0$ and σ = 0.3), denoised results by (c) proposed algorithm, (d) MSGW [12], and (e) IBR [6].

C. Denoising of real-world point clouds

We show a visual comparison for the point cloud denoised by the proposed algorithm with a graph constructed from only geometry and regularization as in [6] , as well as the algorithm described in [12]. The experiment is performed on real-world natural point clouds, for which we do not have a noiseless reference, hence the results are only qualitative. Fig. 3-a and Fig. 4-a show the point clouds with real noise; it can be seen that the points with same color are typically spread in a small neighborhood during the acquisition process, which may blur the details. Fig. 3-b and Fig. 4-b report the resulting denoised point cloud using proposed algorithm. Here the noisy points are moved close to their original position by exploiting the correlation of their geometry and color, hence preserving the details which were hidden in the noisy point cloud. Fig. 3-c and Fig. 4-c show the denoised point cloud using [12]; it can be seen that the algorithm is less effective at denoising the geometry even after applying the outlier removal step prior to perform denoising. Fig. 3-d and Fig. 4-d show the denoised output using IBR [6] applied to the outlier-free input; it can be seen in same region that, using no color information, the noisy points are not moved to their correct location, enlarging gaps and generally providing a noisier result near object boundaries. This algorithm is iterative and for real-world point cloud it is very computationally expensive; moreover the outlier removal step is required for both MSGW and IBR which makes these algorithm more computational complex.

D. Denoising of synthetic point clouds

The proposed denoising approach has also been applied to noise-free point clouds from the *Greyc* dataset [24], corrupted with zero-mean Gaussian synthetic geometry noise applied to the points with $\sigma = 0.2, 0.3$ and 0.4. In Fig. 5, the results are shown in two rows for *4arms_monstre* and *Asterix* models respectively. The first row of figures for each point cloud in Fig. 5 is the natural representation of the point cloud, where the artifacts (i.e. creation of holes) in the denoised point cloud using MSGW and IBR with respect to proposed algorithm can be clearly seen. Moreover, the second and fourth rows provide an alternative view; they are the zoomed-in versions of the same point clouds displayed with the same size as in the first and third row. This type of visualization is more sparse and allows to better discriminate the noise removal and fine details at the boundaries. Fig. 5-a shows the noise-free point cloud models. Fig. 5-b shows the noisy point clouds with σ = 0.3. The denoised point clouds obtained by the proposed algorithm are shown in Fig. 5-c; it can be seen that the geometry noise

TABLE II: MSE and MCD comparison of various algorithms for *Greyc* dataset.

Gaussian Noise	Methods	4arms monstre	Asterix	Cable car	Dragon	Duck	Green Dinasour	Green monster	Horse	Jaguar	Long Diansour	Mario	Mario car	Pokeman ball	Rabbit	Red horse	Statue	**Average**
MSE																		
σ = 0.2	Proposed	**0.35**	**0.27**	**0.20**	**0.20**	0.73	**0.36**	**0.17**	**0.41**	**0.13**	**0.16**	**0.13**	**0.13**	0.38	**0.37**	**0.21**	**0.36**	**0.29**
	MSGW[12]	0.36	**0.27**	0.23	0.23	0.85	0.37	0.18	0.43	0.14	0.17	0.14	0.14	**0.36**	0.37	0.23	0.37	0.30
	IBR [6]	0.38	0.30	0.29	0.31	**0.63**	0.41	0.23	0.45	0.20	0.24	0.19	0.19	0.41	0.40	0.30	0.37	0.33
σ = 0.3	Proposed	**0.36**	**0.27**	**0.22**	**0.21**	0.76	**0.37**	**0.18**	**0.42**	**0.14**	**0.17**	**0.14**	**0.15**	**0.40**	**0.38**	**0.22**	**0.36**	**0.30**
	MSGW[12]	0.38	0.28	0.24	0.24	0.92	0.39	0.18	0.44	0.15	0.18	0.15	0.15	**0.40**	0.40	0.25	0.38	0.32
	IBR [6]	0.39	0.30	0.30	0.32	**0.71**	0.42	0.24	0.46	0.20	0.25	0.20	0.19	0.44	0.42	0.32	0.38	0.35
σ = 0.4	Proposed	**0.37**	**0.27**	**0.26**	**0.28**	**1.15**	**0.40**	**0.19**	**0.44**	**0.15**	**0.19**	**0.16**	**0.16**	**0.41**	**0.41**	**0.28**	**0.38**	**0.34**
	MSGW[12]	0.38	0.28	0.30	0.32	1.35	0.43	0.21	0.45	0.16	0.21	0.17	0.17	0.41	0.41	0.32	0.39	0.37
	IBR [6]	0.39	0.31	0.31	0.33	1.16	0.43	0.24	0.45	0.21	0.26	0.21	0.20	0.46	0.42	0.33	0.38	0.38
MCD																		
σ = 0.2	Proposed	**0.51**	**0.39**	**0.30**	**0.30**	1.07	**0.52**	**0.25**	**0.60**	**0.19**	**0.24**	**0.19**	**0.19**	0.56	**0.53**	**0.31**	**0.52**	**0.42**
	MSGW[12]	0.53	0.39	0.33	0.33	1.24	0.53	0.26	0.61	0.20	0.25	0.21	0.20	**0.52**	0.54	0.34	0.53	0.44
	IBR [6]	0.55	0.43	0.42	0.44	**0.94**	0.59	0.33	0.65	0.29	0.35	0.28	0.28	0.61	0.59	0.44	0.54	0.48
σ = 0.3	Proposed	**0.52**	**0.39**	**0.32**	**0.31**	1.13	**0.54**	**0.27**	**0.61**	**0.21**	**0.25**	**0.21**	**0.21**	0.59	**0.55**	**0.32**	**0.53**	**0.44**
	MSGW[12]	0.54	0.40	0.35	0.35	1.36	0.56	0.27	0.63	0.21	0.26	0.22	0.22	**0.58**	0.58	0.36	0.55	0.47
	IBR [6]	0.56	0.44	0.43	0.46	**1.06**	0.61	0.34	0.66	0.30	0.36	0.29	0.28	0.65	0.61	0.46	0.55	0.50
σ = 0.4	Proposed	**0.54**	**0.40**	**0.38**	**0.40**	**1.68**	**0.58**	**0.28**	**0.63**	**0.23**	**0.28**	**0.23**	**0.23**	**0.60**	**0.59**	**0.40**	**0.53**	**0.50**
	MSGW[12]	0.54	0.41	0.43	0.45	1.96	0.61	0.31	0.64	0.24	0.30	0.24	0.25	0.61	0.59	0.47	0.56	0.54
	IBR [6]	0.57	0.45	0.44	0.47	1.71	0.62	0.35	0.68	0.31	0.37	0.30	0.29	0.67	0.61	0.48	0.55	0.55

TABLE III: MSE and MCD comparison between proposed algorithm and RPSM [10] on sub-sampled *Greyc* dataset for different noise levels.

Gaussian Noise	Methods	4arms monstre	Asterix	Cable car	Dragon	Duck	Green Dinasour	Green monster	Horse	Jaguar	Long Diansour	Mario	Mario car	Pokeman ball	Rabbit	Red horse	Statue	**Average**
MSE																		
σ = 0.2	Proposed	**0.51**	**0.34**	**0.73**	**0.52**	**0.80**	**0.58**	**0.49**	**0.65**	**0.38**	**0.47**	**0.45**	**0.44**	**0.41**	**0.65**	**0.63**	**0.52**	**0.54**
	RPSM [10]	0.90	0.74	1.25	1.02	1.33	0.99	0.87	1.19	0.78	0.74	1.02	0.86	0.73	0.94	1.18	1.03	0.97
σ = 0.3	Proposed	**0.54**	**0.36**	**0.77**	**0.53**	**0.84**	**0.61**	**0.51**	**0.68**	**0.39**	**0.49**	**0.46**	**0.47**	**0.44**	**0.68**	**0.66**	**0.54**	**0.56**
	RPSM [10]	0.95	0.76	1.34	1.07	1.35	1.04	0.93	1.24	0.83	0.80	1.07	0.92	0.79	1.00	1.26	1.06	1.03
σ = 0.4	Proposed	**0.56**	**0.37**	**0.80**	**0.56**	**0.87**	**0.64**	**0.52**	**0.71**	**0.41**	**0.50**	**0.48**	**0.48**	**0.46**	**0.69**	**0.68**	**0.56**	**0.58**
	RPSM [10]	0.98	0.79	1.41	1.13	1.39	1.12	0.98	1.28	0.89	0.86	1.13	0.98	0.90	1.05	1.31	1.11	1.08
MCD																		
σ = 0.2	Proposed	**0.75**	**0.50**	**1.08**	**0.76**	**1.19**	**0.85**	**0.72**	**0.96**	**0.56**	**0.70**	**0.66**	**0.65**	**0.60**	**0.95**	**0.93**	**0.76**	**0.79**
	RPSM [10]	1.44	1.04	1.93	1.57	1.91	1.60	1.32	1.79	1.18	1.24	1.49	1.25	1.52	1.63	1.79	1.56	1.52
σ = 0.3	Proposed	**0.79**	**0.52**	**1.14**	**0.79**	**1.25**	**0.90**	**0.75**	**1.00**	**0.58**	**0.72**	**0.68**	**0.69**	**0.65**	**0.99**	**0.98**	**0.80**	**0.83**
	RPSM [10]	1.51	1.08	2.05	1.63	1.94	1.67	1.41	1.87	1.24	1.32	1.56	1.34	1.60	1.72	1.90	1.61	1.59
σ = 0.4	Proposed	**0.82**	**0.53**	**1.18**	**0.82**	**1.30**	**0.95**	**0.77**	**1.05**	**0.60**	**0.74**	**0.70**	**0.70**	**0.67**	**1.02**	**1.00**	**0.82**	**0.85**
	RPSM [10]	1.55	1.12	2.15	1.71	2.00	1.77	1.47	1.92	1.32	1.40	1.63	1.41	1.74	1.78	1.97	1.66	1.66

has been regularized and the noisy points are moved close to their original positions with less adverse effect of creating holes. Fig. 5-d shows the denoised output of MSGW [12]; it can be seen that the geometry is not denoised properly and also causes the opening of holes in the resulting output. The resulting denoised point clouds using the IBR algorithm [6] are shown in 5-e. It can be seen that the geometry is not quite as much regularized with respect to the proposed algorithm, while still better than MSGW [12]; moreover, the resulting point cloud of the IBR algorithm [6] has big holes; the reason is the value of γ parameter, higher γ is required for denoising well, but it causes severe artefacts. Overall, it can be seen from the qualitative results of both real-world and synthetic point clouds that the point clouds denoised by the proposed algorithm have better quality and fewer artefacts.

The comparative analysis has also been performed with respect to RPSM [10] and the results are shown in Fig. 6. In this particular case, we performed the experiment on sub-sampled point clouds from the same dataset due the large memory requirement of RPSM. The sub-sampling performed here is on spatial basis, where the average minimum distance between the two points in each point cloud is set to 0.80. The number of points in each point cloud model of Greyc dataset [24] is around 20,000 on average. The proposed algorithm and RPSM [10] are applied to the sub-sampled noisy inputs shown in Fig. 5-b; the ground-truth point clouds are shown in Fig. 5-a. The denoised outputs of the proposed algorithm are shown in Fig. 6-a and Fig. 6-c, and the resulting denoised outputs of RPSM [10] are shown in Fig. 6-b and Fig. 6-d; it can be seen that RPSM [10] over-regularized the sub-sampled point clouds which tends to generate the holes in the resulting denoised point cloud.

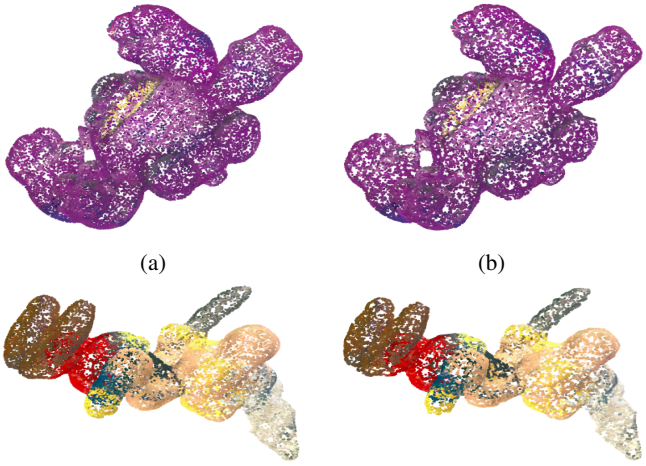
E. Objective evaluation on Greyc color mesh database

The proposed method has also been verified via quantitative evaluation on the complete *Greyc* dataset. Each point cloud has been corrupted with zero-mean Gaussian synthetic geometry noise, applied to each point with σ = 0.2, 0.3 and 0.4. The MSE and MCD comparisons between the proposed algorithm and the denoising approaches used in IBR [6] using total variation regularization and MSGW in [12] is shown in Table II. The results show that the proposed denoising technique performed better than MSGW [12] and IBR [6] for all the models for noise level $\sigma = 0.2,\ 0.3$ and 0.4 except *duck* (where

IBR performed better) and *Pokeman_ball* (where MSGW performed better).

The objective evaluation has also been performed for the comparison between the proposed algorithm and RPSM [10] on the sub-sampled point cloud models of *Greyc* dataset [24]. Table III shows the MSE and MCD comparison; it can be seen that the proposed algorithm outperformed the RPSM [10] in terms of both metrics for $\sigma = 0.2$, 0.3 and 0.4.

The average MSE and MCD (last column in Table II and Table III) shows that the gain is larger as the noise level increases, showing that the proposed denoising methods is indeed better at removing geometry noise.



(a) (b) (c) (d)

Fig. 6: 4arms_monstre model: (a) Denoised results by proposed algorithm, (b) RPSM [10]. Asterix model: (c) denoised results by proposed algorithm, and (d) RSPM [10].

IV. Conclusions

We proposed a novel and efficient point cloud denoising technique in the frequency domain using spectral graph wavelet transform. Our approach takes advantage of correlation between the color attribute and geometry of points and use them jointly for denoising the geometry of a point cloud.

For subjective evaluation we showed that the proposed algorithm performs well on real-world point clouds, performing geometry denoising and avoiding artifacts typical of other techniques. We also performed an extensive quantitative analysis on each models in *Greyc* datasets, evaluating the performance of the proposed algorithm using MSE and MCD metrics. Both subjective and objective results show that the proposed technique performs very well for point cloud denoising, outperforming state-of-the-art techniques.

References

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

[2] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1765–1778, 2016.

[3] 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, 2017.

[4] C. Tulvan, R. Mekuria, Z. Li, and S. Laserre, "Use cases for point cloud compression. ISO," IEC JTC1/SC29/WG11 MPEG (2016), Tech. Rep., 2016.

[5] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "Surfacenet: An end-to-end 3D neural network for multiview stereopsis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2307–2315.

[6] 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.

[7] E. Mattei and A. Castrodad, "Point cloud denoising via moving RPCA," in *Computer Graphics Forum*, vol. 36, no. 8. Wiley Online Library, 2017, pp. 123–137.

[8] C. Dinesh, G. Cheung, and I. V. Bajic, "3D point cloud denoising via bipartite graph approximation and reweighted graph laplacian," *arXiv preprint arXiv:1812.07711*, 2018.

[9] M.-J. Rakotosaona, V. La Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, "Pointcleannet: Learning to denoise and remove outliers from dense point clouds," in *Computer Graphics Forum*. Wiley Online Library, 2019.

[10] S. Deutsch, A. Ortega, and G. Medioni, "Robust denoising of piecewise smooth manifolds," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2786–2790.

[11] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Ec-net: an edge-aware point set consolidation network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 386–402.

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

[13] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, "Fusion of geometry and color information for scene segmentation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 5, pp. 505–521, 2012.

[14] C. Dinesh, G. Cheung, and I. V. Bajić, "3D point cloud color denoising using convex graph-signal smoothness priors," in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2019, pp. 1–6.

[15] E. K. Matti and S. Nebiker, "Geometry and colour based classification of urban point cloud scenes using a supervised self-organizing map," *Photogrammetrie-Fernerkundung-Geoinformation*, vol. 2014, pp. 161–173, 2014.

[16] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[17] G. Guennebaud and M. Gross, "Algebraic point set surfaces," in *ACM SIGGRAPH 2007 papers*, 2007, pp. 23–es.

[18] A. C. Öztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 493–501.

[19] G. M. Phillips, *Interpolation and approximation by polynomials*. Springer Science & Business Media, 2003, vol. 14.

[20] J. Wang, *Geometric structure of high-dimensional data and dimensionality reduction*. Springer, 2012.

[21] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. 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.

[22] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, "Wavelet toolbox," *The MathWorks Inc., Natick, MA*, vol. 15, p. 21, 1996.

[23] 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.

[24] A. Nouri, C. Charrier, and O. Lézoray, "Technical report: Greyc 3D colored database," Ph.D. dissertation, Normandie Université, Unicaen, EnsiCaen, CNRS, GREYC UMR 6072, 2017.