3D Point Cloud Denoising Using a Joint Geometry and Color k-NN Graph

Muhammad Abeer Irfan, Enrico Magli
Dept. of Electronics and Telecommunications-Politecnico di Torino, Italy
{abeer.irfan,enrico.magli}@polito.it

***Abstract*—Many point cloud acquisition methods, e.g. multi-viewpoint image stereo matching and acquisition of depth data from active light sensors, suffer from significant geometry noise in the data. In the existing literature, denoising of this geometry noise has been performed using only geometry information. In this paper, based on the notion that color attributes are correlated with the geometry, we propose a novel geometry denoising technique that takes advantage of this correlation via a graph-based optimization process. In particular, we construct a graph based on both color and geometry information, and use it for graph-based Tikhonov regularization. Results on synthetic and real-world point clouds show that the proposed denoising method significantly outperforms existing geometry-only techniques.**

***Index Terms*—convex optimization, graph signal processing, point cloud denoising**

I. Introduction

Point cloud is an important representation of volumetric objects in three-dimensional (3D) space, which allows visualization from any viewpoint [1]–[3]. A point cloud consists of a set of points, representing the geometry and some attributes like color, normal, transparency and size, related to an object or scene. Due to the efficient representation of 3D objects, point clouds have been now extensively adopted in various fields, such as 3D broadcasting, culture and heritage reconstruction, navigation of unmanned vehicles, and 3D immersive tele-presence [4]. The acquisition of a point cloud can be performed using active sensors, or computed indirectly from multi-viewpoint images [5]. In both cases, the obtained point cloud suffers from noise, and hence denoising should be performed in order to improve its quality.

Many techniques have therefore been proposed for geometry denoising of a point cloud. Normally, denoising is performed by applying statistical methods [6], or by evaluating surface normals around a small neighborhood and taking average along their normal direction [7], [8]. The denoising process can be categorized into outlier removal and noise removal; this latter often applies surface smoothing moving points towards their correct positions based on a smoothness prior. The local tangent space based graph is used for robust denoising of piece-wise smooth manifolds (RPSM) [9]. Recent data-driven approaches [10], [11] have shown interesting results, but these methods may not be applicable when a dataset of noise-free point clouds is not available.

In recent literature, the geometry of a point cloud has been expressed as a graph, which is used for denoising by convex optimization [12]. Manifold denoising based on Spectral Graph Wavelet (MSGW) [13] also used geometry-only graph for denoising. The disadvantage of these approaches is that the correct position of a point is estimated based on the *noisy* geometry; this can lead to errors in estimating the local surface, and as a result holes are typically formed in the denoised point cloud (see Fig. 1).

In this paper, we introduce a denoising method which jointly uses the geometry and the color attribute of points to remove geometry noise. The rationale is that on a smooth surface the color is also typically smooth; this is well-known and it has been used in point cloud segmentation [14]–[16]; moreover, in this paper we are only interested in geometry noise, which is the prevalent type of noise in point clouds. Therefore, in the proposed method we carefully exploit both the geometry and color to relocate each point to its correct position, avoiding the artifacts generated by denoising techniques employing only geometry information. In particular, our proposed technique constructs a graph based on both the geometry and the color attribute of each point, and subsequently applies convex optimization to perform geometry denoising. We show experimentally that the joint use of geometry and color outperforms existing methods using both subjective and objective quality metrics.

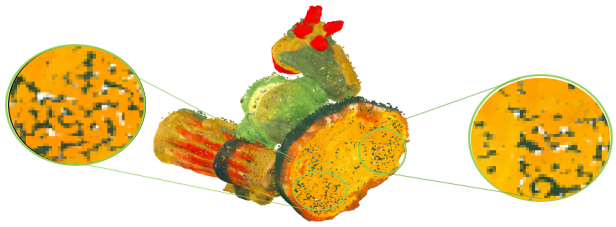

Fig. 1: Green_monster model: Geometry denoised from geometry-only graph. The noisy points are moved towards their nearest neighbors rather than their correct positions, opening holes in the surface [12].

II. Proposed method

A. Outlier removal

The initial step is to eliminate outliers from the entire point cloud. Outliers have distinct characteristics in that each outlier has a sparse neighborhood; therefore, the detection of outliers is density based as in [12], [17].

B. Graph taxonomy

An undirected weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ is defined for a finite set of vertices $\mathcal{V}$ of cardinality $|\mathcal{V}| = N$, $\mathcal{E}$ representing a set of edges connecting vertices of the form $(v_i, v_j) \in \mathcal{V}$, each edge having a non-negative weight $w_{i,j}$. The corresponding adjacency matrix $\mathbf{W}$ is a real symmetric *m* x *m* matrix. A graph signal $g(\mathcal{G})$ for a given graph $\mathcal{G}$ is defined on the vertices of a graph, as $g : \mathcal{V} \rightarrow \mathbb{R}^n$ for some dimension n.

C. Graph construction from combined geometry and color

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. A common approach is to construct a k nearest-neighbor (k-NN) graph based on Euclidean distance as it makes geometric structure explicit [18]. Our approach generates k-NN graph based on *both* color similarity and geometric proximity of points p_i. This is done by defining six-dimensional features for each point as $f_{1i} = [\omega_1 X_i \; \omega_2 C_i]$, with $X_i = [x_{1i} \; x_{2i} \; x_{3i}]$ and $C_i = [c_{1i} \; c_{2i} \; c_{3i}]$, where c_{1i}, c_{2i} and c_{3i} are the color attributes and x_{1i}, x_{2i} and x_{3i} are the geometric coordinates of point p_i. In this work we consider Euclidean coordinates and the RGB colorspace; ω_1 and ω_2 are weights that determine how much geometry and color contribute to the generation of the k-NN graph.

A color-only graph is not a good choice because geometrically distant points may have same color and different contents, which may leads to the construction of a wrong graph. The color combined with the geometry is helpful because it provides more information about the manifold with respect to using only the geometry.

Each edge connects a vertex to its k nearest neighbors with an assigned weight which is computed with some metric. A common option is the threshold Gaussian kernel [19]:

$$w_{i,j} = \begin{cases} \exp\left(-\frac{\|f_{1i} - f_{1j}\|^2}{2\theta^2}\right) & \text{if } f_{1j} \in \phi_k(i) \text{ or } f_{1i} \in \phi_k(j) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Here, θ represents variance and $\phi_k(i)$ 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. In calculating $||f_{1i} - f_{1j}||$, the weights ω_1 and ω_2 determine the relative contribution of geometry and color in the construction of the resulting k-NN graph, which is denoted as $\mathcal{G}$.

D. Geometry denoising

The algorithm presented in this paper has the objective to perform geometry denoising exploiting the graph $\mathcal{G}$ constructed from both geometry and color information of the noisy point cloud. To this end, we define a graph signal $g(\mathcal{G})$, where each vertex of $\mathcal{G}$ is associated to the geometry information X_i of the corresponding point p_i of point cloud $\mathcal{P}$. Each value can be expressed as $X_i = x_i + w_i$, x_i being the unknown exact geometry of point p_i and w_i the geometry noise, with $X_i, x_i, w_i \in \mathbb{R}^3$. The objective is to estimate x_i for each point of the point cloud. This can be done by means of the proposed denoising algorithm described in Sec. II-E, which moves points closer to their true location based on a smoothness assumption applied to the joint geometry and color information embedded in graph $\mathcal{G}$.

In particular, graph $\mathcal{G}$ can be considered a noisy approximation of a 3D manifold. The regularity of the combined geometry and color is therefore associated with the proximity of the points to the manifold. Graph gradient can be used to measure the degree of smoothness of a graph signal [20]. We propose to employ convex optimization to enforce smoothness of the geometry graph signal defined above on $\mathcal{G}$, exploiting the combined geometry and color k-NN graph for improved quality of the resulting denoised point cloud.

E. Geometry denoising from joint geometry and color k-NN graph

We considered convex optimization for denoising the graph signal with the constraint that the signal must be smooth on a graph. Hence, the denoising problem can be written as follows:

$$\widehat{x} = \arg\min_x \|x - g\|_2^2 + \gamma \|\nabla_{\mathcal{G}} x\|_2^2 \tag{2}$$

Here, the estimated denoised geometry is referred to as $\widehat{x}$, the observed noisy signal is represented by the graph signal g defined above, γ is a parameter for regularization and $\nabla_{\mathcal{G}} x$ represents the gradient of the signal x on the graph $\mathcal{G}$. Eq. 2 has two terms; the first is a fidelity term that enforces the denoised point to be not too far from its observed position, while the second term promotes smoothness of the denoised point cloud on $\mathcal{G}$ via Tikhonov regularization. The same method can also employ Total Variation (TV) regularization with the constraint that the underlying manifold of a point cloud be piecewise smooth. This leads to the following convex optimization problem:

$$\widehat{x} = \arg\min_x \|x - g\|_2^2 + \gamma \|\nabla_{\mathcal{G}} x\|_1 \tag{3}$$

The problems in Eq. 2 and 3 can be solved efficiently by alternating direction method of multipliers [21].

III. Experimental Results

A. Evaluation Metrics

The metrics used for the performance evaluation of the proposed algorithm are the same as in [22]. Assume that $\mathcal{R}$ and $\mathcal{Q}$ represent the geometry of the original and denoised point cloud respectively, where $\mathcal{R} = \{r_i\}_{i=1}^{N_1}$ $\mathcal{Q}$=$\{q_i\}_{i=1}^{N_2}$, such that r_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 $\mathcal{R}$ and its corresponding nearest point in $\mathcal{Q}$, and also between each point in $\mathcal{Q}$ and its corresponding nearest point in $\mathcal{R}$:

$$\text{MSE} = \frac{1}{2N_1} \sum_{r_i \in \mathcal{R}} \min_{q_i \in \mathcal{Q}} \|r_i - q_i\|_2^2 + \frac{1}{2N_2} \sum_{q_i \in \mathcal{Q}} \min_{r_i \in \mathcal{R}} \|q_i - r_i\|_2^2 \tag{4}$$

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

$$\text{MCD} = \frac{1}{2N_1} \sum_{r_i \in \mathcal{R}} \min_{q_i \in \mathcal{Q}} \|r_i - q_i\| + \frac{1}{2N_2} \sum_{q_i \in \mathcal{Q}} \min_{r_i \in \mathcal{R}} \|q_i - r_i\| \tag{5}$$

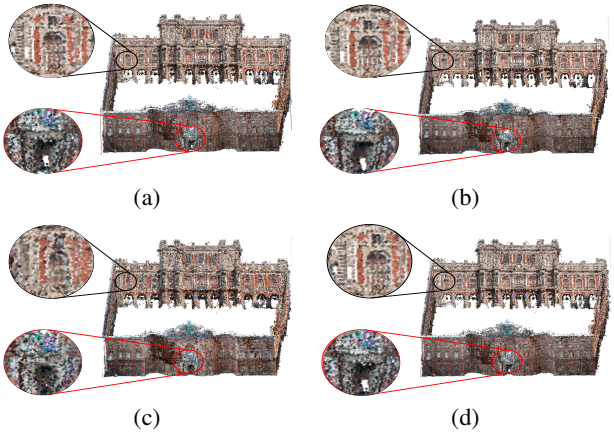

Fig. 2: Palazzo_Carignano_Dense model. (a) Noisy input, (b) outlier-free input, denoised results by (c) proposed algorithm, and (d) geometry-only graph [12].

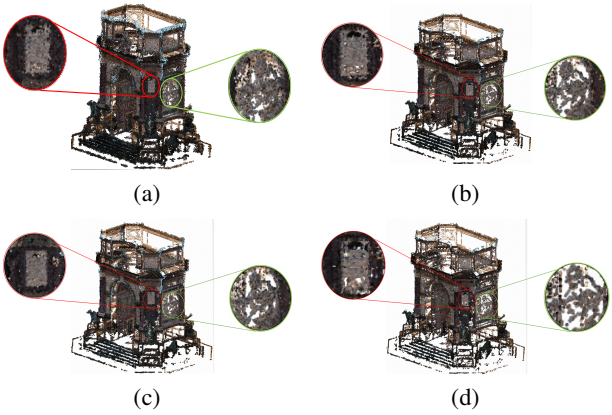

Fig. 3: Arco_Valentino model. (a) Noisy input, (b) outlier-free input, denoised results by (c) proposed algorithm, and (d) geometry-only graph [12].

The graph signal processing in our denoising algorithm has been implemented using GSPBOX [23] and for the convex optimization we have used UNLocBox [24].

B. Experimental setup

For the outlier removal we have set $\epsilon = 0.01$, $k = 5$ and $\tau = 1$ as in [12]. For the noise removal we have found that setting $k = 15$ and $\gamma = 0.075$ for uniform distributed noise with μ = 0 and σ = 0.3 and 0.4 consistently provides very good results. However, $k = 15$ and $\gamma = 0.1$ gives good results for the noise with μ = 0 and σ = 0.5. After extensive experimentation we have found that the weights $\omega_1 = 0.70$ and $\omega_2 = 0.30$ provide best results for geometry denoising.

C. Natural point clouds with real noise

We show a visual comparison between the point cloud denoised by the proposed algorithm, and a graph constructed from only geometry as 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. 2-a and Fig. 3-a show the point clouds with real noise; it can be seen that the points with same color are typically in a small neighborhood. Fig. 2-b and Fig. 3-b are the resulting outputs after outlier removal. Fig. 2-c and Fig. 3-c depict the denoised point cloud using the proposed algorithm. Here the noisy points are moved close to their original position by exploiting the correlation of their color, and hence they fill the gaps that were present in the noisy point cloud and are due to geometry noise. Fig. 2-d and Fig. 3-d show the denoised point cloud using the geometry-only graph approach in [12]; 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.

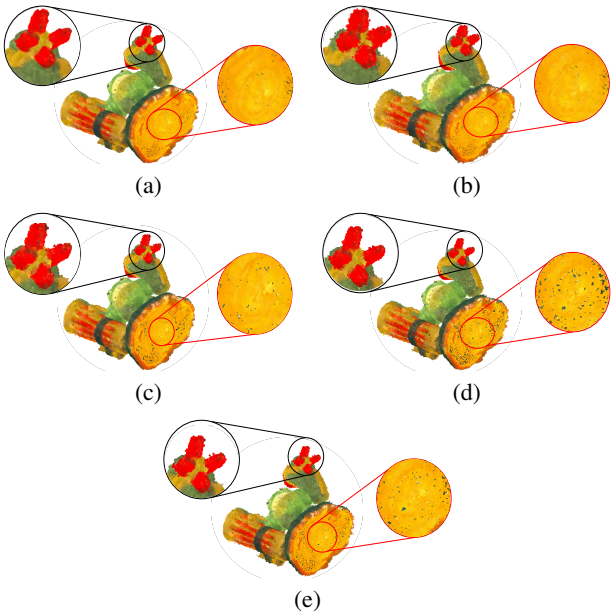

Fig. 4: Green_monstre model: (a) ground-truth (b) noisy input (μ = 0 and σ = 0.4), denoised results by (c) proposed algorithm (d) geometry-only graph [12], and (e) MSGW [13].

D. Point clouds with synthetic noise

The proposed denoising approach has also been applied to noise-free point clouds from the *Greyc* dataset [25], corrupted with zero-mean uniform synthetic geometry noise applied to 50% of the points with σ = 0.3, 0.4 and 0.5. Fig. 4-a and Fig. 5-a show the noise-free point clouds. Fig. 4-b and Fig. 5-b show the noisy point clouds. The denoised point clouds obtained by our proposed algorithm are shown in Fig. 4-c and 5-c; it can be seen that the geometry noise has been regularized and the noisy points are moved close to their original positions. The resulting denoised point clouds using the geometry-only algorithm [12] are shown in Fig. 4-d and 5-d. It can be seen

TABLE I: MSE comparison for *Greyc* dataset.

Uniform Distribution	Methods	4arms monstre	Asterix	Cable car	Dragon	Duck	Green dinosaur	Green monstre	Horse	Jaguar	Long dinosaur	Mario	Mario car	Pokemon ball	Rabbit	Red horse	Statue
σ=0.3	Proposed algorithm	**0.2760**	**0.2340**	**0.2179**	**0.2172**	**0.7138**	**0.2932**	**0.1730**	**0.3138**	0.1550	**0.1815**	**0.1443**	**0.1384**	**0.3184**	**0.2784**	**0.2361**	**0.2755**
	Geometry-only graph [12]	0.3160	0.2448	0.2378	0.2476	0.8812	0.3393	0.2116	0.3761	**0.1535**	0.1856	0.1534	0.1447	0.3634	0.3399	0.2506	0.3116
	MSGW [13]	0.4161	0.325	0.3835	0.4564	0.9533	0.5969	0.4958	0.5036	0.2629	0.2468	0.2420	0.2804	0.523	0.6108	0.3579	0.4781
σ=0.4	Proposed algorithm	**0.2830**	**0.2398**	**0.2273**	**0.2330**	**0.7330**	**0.3038**	**0.1873**	**0.3228**	**0.1713**	**0.1979**	**0.1600**	**0.1592**	**0.3349**	**0.2892**	**0.2504**	**0.2814**
	Geometry-only graph [12]	0.3190	0.2485	0.2453	0.2580	0.9072	0.3478	0.1828	0.3769	0.1753	0.2034	0.1699	0.1746	0.3735	0.3417	0.2599	0.3114
	MSGW [13]	0.4107	0.4433	0.3678	0.4074	0.8956	0.4790	0.3303	0.4574	0.3276	0.3494	0.3206	0.3832	0.4955	0.4129	0.3793	0.3857
σ=0.5	Proposed algorithm	**0.2867**	**0.2477**	**0.2435**	**0.2510**	**0.7541**	**0.3116**	**0.2003**	**0.3323**	**0.1893**	**0.2156**	**0.1767**	**0.1765**	**0.3336**	**0.2969**	**0.2655**	**0.2887**
	Geometry-only graph [12]	0.3231	0.2568	0.2668	0.2575	0.9027	0.9027	0.3470	0.2956	0.3807	0.2027	0.2253	0.1981	0.3635	0.3445	0.2766	0.3229
	MSGW [13]	0.4095	0.4369	0.3708	0.4095	0.8990	0.4773	0.3340	0.4549	0.3327	0.3554	0.3278	0.3839	0.4951	0.4112	0.3830	0.3832

TABLE II: MCD comparison for *Greyc* dataset.

Uniform Distribution	Methods	4arms monstre	Asterix	Cable car	Dragon	Duck	Green dinosaur	Green monstre	Horse	Jaguar	Long dinosaur	Mario	Mario car	Pokemon ball	Rabbit	Red horse	Statue
σ=0.3	Proposed algorithm	**0.4075**	**0.3443**	**0.3189**	**0.3196**	**1.058**	**0.4342**	**0.2440**	**0.4622**	0.2263	**0.2668**	**0.2108**	**0.2008**	**0.4722**	**0.4112**	**0.3468**	**0.4066**
	Geometry-only graph [12]	0.4620	0.3590	0.3457	0.3617	1.2974	0.4989	0.2518	0.5490	**0.2247**	0.2728	0.2241	0.2107	0.5329	0.4959	0.3668	0.4553
	MSGW [13]	0.4925	0.5191	0.4268	0.6388	1.8430	0.8086	0.4582	0.6665	0.5820	0.5936	0.4910	0.5604	0.8084	0.5812	0.6812	0.6643
σ=0.4	Proposed algorithm	**0.4176**	**0.3526**	**0.3306**	**0.3415**	**1.0862**	**0.4489**	**0.2746**	**0.4751**	**0.2482**	**0.2894**	**0.2321**	**0.2292**	**0.4960**	**0.4266**	**0.3666**	**0.4147**
	Geometry-only graph [12]	0.4669	0.3650	0.3577	0.3767	1.3340	0.5107	0.2681	0.5496	0.2547	0.2984	0.2468	0.2511	0.5480	0.4984	0.3802	0.4548
	MSGW [13]	0.5895	0.6250	0.5198	0.5820	1.3218	0.6881	0.4676	0.6484	0.4577	0.4971	0.4506	0.5229	0.7144	0.5941	0.5443	0.5541
σ=0.5	Proposed algorithm	**0.4220**	**0.3638**	**0.3527**	**0.3667**	**1.1191**	**0.4598**	**0.2928**	**0.4889**	**0.2721**	**0.3132**	**0.2546**	**0.2519**	**0.4928**	**0.4374**	**0.3872**	**0.4250**
	Geometry-only graph [12]	0.4724	0.3766	0.3877	0.4019	1.3299	0.5095	0.3058	0.5551	0.2915	0.3286	0.2763	0.2821	0.5350	0.5025	0.4036	0.4704
	MSGW [13]	0.5891	0.6185	0.5249	0.5852	1.3232	0.6865	0.4745	0.6512	0.4651	0.5056	0.4611	0.5245	0.7153	0.5931	0.5492	0.5512

that the geometry is not quite as much regularized; moreover, as anticipated in Sec. I, these approaches have an adverse effect on overall geometry, as it tends to open holes in the output denoised point cloud. Overall, it can be seen from the qualitative results of both the real-world and synthetic point clouds that the point clouds denoised by the proposed algorithm have better quality and fewer artifacts.

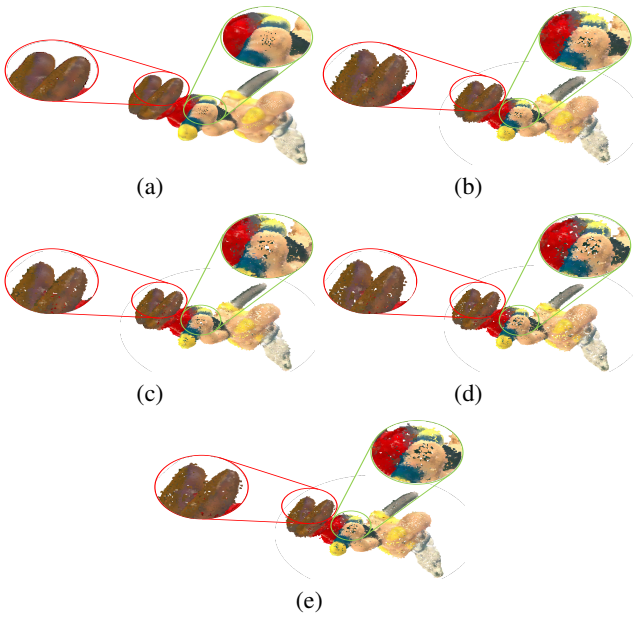

Fig. 5: Asterix model: (a) ground-truth (b) noisy input ($\mu = 0$ and σ = 0.4), denoised results by (c) proposed algorithm (d) geometry-only graph [12], and (e) MSGW [13].

E. Evaluation on Greyc color mesh database

The qualitative results have also been verified via quantitative evaluation on the complete *Greyc* dataset. Each point cloud has been corrupted with zero-mean uniform synthetic geometry noise, applied to 50% of the points with σ = 0.3, 0.4 and 0.5. The MSE and MCD comparisons between the proposed algorithm and the denoising approaches used in [12] using Tikhonov regularization and MSGW in [13] are shown in Table. I and Table. II respectively. The results show that the proposed denoising technique performed better than [12] in terms of both metrics for noise level $\sigma = 0.3$, $\sigma = 0.4$ and $\sigma = 0.5$. The gain is larger as the noise level increases, showing that the proposed denoising methods is indeed better at removing geometry noise.

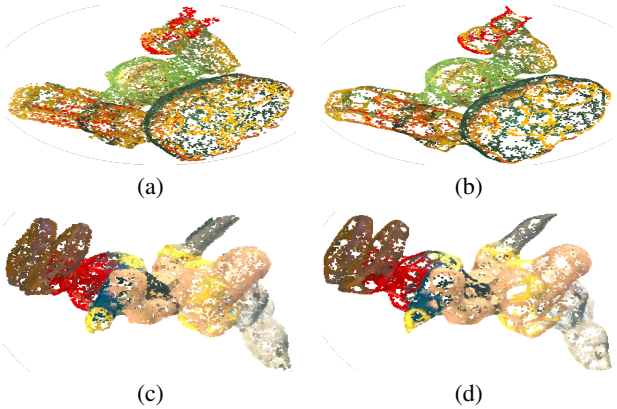

Fig. 6: Green_monster model: (a) Denoised results by proposed algorithm, (b) RPSM [9] – Asterix model: (c). denoised results by proposed algorithm, and (d) RSPM [9]

TABLE III: MSE comparison on sub-sampled *Greyc* dataset.

Model	$\sigma = 0.3$		$\sigma = 0.4$		$\sigma = 0.5$	
	Proposed	RPSM [9]	Proposed	RPSM [9]	Proposed	RPSM [9]
4arms_Monstre	**0.2916**	0.4811	**0.3389**	0.5047	**0.4206**	0.5232
Asterix	**0.2605**	0.3084	**0.2751**	0.3412	**0.2996**	0.3725
Cable_car	**0.3558**	0.7440	**0.3811**	0.7990	**0.4019**	0.8177
Dragon	**0.2496**	0.4432	**0.2622**	0.4769	**0.2832**	0.4967
Green_monster	**0.3318**	0.6179	**0.3589**	0.6478	**0.3734**	0.6739
Rabbit	**0.3096**	0.5771	**0.3197**	0.6004	**0.3480**	0.6339
Red_horse	**0.3385**	0.5138	**0.3641**	0.5487	**0.3870**	0.5773

We have also compared the proposed algorithm with RPSM [9]. Due to the large memory requirement of RPSM, we performed experiments on sub-sampled point clouds. The sub-sampling performed here is on spatial basis, setting a minimal distance between the two points equal to 0.80. The number of points in the sub-sampled clouds are around 20000 on average. Table III and Table IV show the MSE and MCD comparisons between the proposed algorithm and RPSM [9] on sub-sampled point clouds.

The proposed algorithm and RPSM [9] are applied to the noisy inputs shown in Fig. 4-a and Fig. 5-a; the ground-truth point clouds are shown in Fig. 4-b and Fig. 5-b. The denoised outputs of the proposed algorithm are depicted in Fig. 6-a and Fig. 6-c, and the resulting point clouds of RPSM [9] are shown in Fig. 6-b and Fig. 6-d. It can be seen that the RPSM [9] over-regularized the sub-sampled point clouds which tends to generate artifacts.

TABLE IV: MCD comparison on sub-sampled *Greyc* dataset.

Model	$\sigma = 0.3$		$\sigma = 0.4$		$\sigma = 0.5$	
	Proposed	RPSM [9]	Proposed	RPSM [9]	Proposed	RPSM [9]
4arms_Monstre	**0.4315**	0.7007	**0.5677**	0.8993	**0.6187**	1.0080
Asterix	**0.3835**	0.4499	**0.4106**	0.4776	**0.4399**	0.5299
Cable_car	**0.5064**	0.9154	**0.5397**	0.9802	**0.5718**	1.0549
Dragon	**0.3688**	0.6463	**0.3869**	0.6811	**0.4176**	0.7036
Green_monster	**0.4878**	0.7138	**0.5193**	0.7674	**0.5480**	0.8025
Rabbit	**0.4587**	0.7975	**0.4887**	0.8244	**0.5139**	0.8790
Red_horse	**0.4963**	0.7452	**0.5371**	0.7790	**0.5656**	0.8101

IV. Conclusions

We proposed a novel and efficient point cloud denoising technique based on graph signal processing, where the color attribute and geometry of points are used 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, avoiding artifacts typical of other techniques. We also performed an extensive quantitative analysis using multiple datasets, evaluating the performance of the proposed algorithm using MSE and MCD metrics. Both the subjective and objective results show that the proposed technique performs very well for point cloud denoising, outperforming state-of-the-art techniques.

References

[1] G. Lafruit, S. Quackenbush, S. Foessel, and A. Hinds, "Technical report of the joint ad hoc group for digital representations of light/sound fields for immersive media applications," 2016.

[2] C. Zhang, Q. Cai, P. A. Chou, Z. Zhang, and R. Martin-Brualla, "Viewport: A distributed, immersive teleconferencing system with infrared dot pattern," *IEEE MultiMedia*, vol. 20, no. 1, pp. 17–27, 2013.

[3] C. Loop, C. Zhang, and Z. Zhang, "Real-time high-resolution sparse voxelization with application to image-based modeling," in *Proceedings of the 5th High-Performance Graphics Conference*. ACM, 2013, pp. 73–79.

[4] C. Tulvan, R. N. Mekuria, Z. Li, and S. Laserre, "Use cases for point cloud compression (PCC)," 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] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[7] N. J. Mitra and A. Nguyen, "Estimating surface normals in noisy point cloud data," in *Proceedings of the nineteenth annual symposium on Computational geometry*. ACM, 2003, pp. 322–328.

[8] H. Yagou, Y. Ohtake, and A. Belyaev, "Mesh smoothing via mean and median filtering applied to face normals," in *Geometric Modeling and Processing. Theory and Applications. GMP 2002. Proceedings*. IEEE, 2002, pp. 124–131.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] F. Verdoja, D. Thomas, and A. Sugimoto, "Fast 3D point cloud segmentation using supervoxels with geometry and color for 3D scene understanding," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 1285–1290.

[15] Q. Zhan, Y. Liang, and Y. Xiao, "Color-based segmentation of point clouds," *Laser scanning*, vol. 38, no. 3, pp. 155–161, 2009.

[16] M. C. Dal, 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.

[17] X. Gao, W. Hu, and Z. Guo, "Graph-based point cloud denoising," in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. IEEE, 2018, pp. 1–6.

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

[19] D. I. Shumaemph n, 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.

[20] 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," *arXiv preprint arXiv:1211.0053*, 2012.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[22] 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.

[23] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "Gspbox: A toolbox for signal processing on graphs," *arXiv preprint arXiv:1408.5781*, 2014.

[24] N. Perraudin, V. Kalofolias, D. Shuman, and P. Vandergheynst, "Unlocbox: A matlab convex optimization toolbox for proximal-splitting methods," *arXiv preprint arXiv:1402.0779*, 2014.

[25] A. Nouri, C. Charrier, and O. Lézoray, "Technical report: Greyc 3D colored mesh database," 2017.