Exploiting color for graph-based 3D point cloud denoising

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

journal homepage: www.elsevier.com/locate/jvci

Exploiting color for graph-based 3D point cloud denoising

Muhammad Abeer Irfan[a,*], Enrico Magli[a,1]

[a]Dept. of Electronics and Telecommunications – Politecnico di Torino, 10129, Italy

ARTICLE INFO

Article history:

Keywords: Point cloud denoising, Color denoising, Convex optimization, Tikhonov regularization, Total variation, Graph signal processing

ABSTRACT

A point cloud is a representation of a 3D scene as a discrete collection of geometry plus other attributes such as color, normal, transparency associated with each point. The traditional acquisition process of a 3D point cloud, e.g. using depth information acquired directly by active sensors or indirectly from multi-viewpoint images, suffers from a significant amount of noise. Hence, the problem of point cloud denoising has recently received a lot of attention. However, most existing techniques attempt to denoise only the geometry of each point, based on the geometry information of the neighboring points; there are very few works at all considering the problem of denoising the color attributes of a point cloud. In this paper, we move beyond the state of the art and we propose a novel technique employing graph-based optimization, taking advantage of the *correlation* between geometry and color, and using it as a powerful tool for several different tasks, i.e. color denoising, geometry denoising, and combined geometry and color denoising. The proposed method is based on the notion that the correct location of a point also depends on the color attribute and not only the geometry of the neighboring points, and the correct color also depends on the geometry of the neighbors. The proposed method constructs a suitable k-NN graph from geometry and color and applies graph-based convex optimization to obtain the denoised point cloud. Extensive simulation results on both real-world and synthetic point clouds show that the proposed denoising technique outperforms state-of-the-art methods using both subjective and objective quality metrics.

1. Introduction

In the new era of graphics technology, 3D point clouds have gained increasing attention as a signal representation of volumetric objects, real scenes, or 3D objects [1, 2, 3]. A point cloud comprises a set of points with 3D geometric information along with attribute data, i.e., color. Point clouds have been widely applied in many fields such as 3D broadcasting, 3D immersive telepresence, culture and heritage reconstruction, and navigation of unmanned vehicles [4]. A point cloud can be acquired directly using low-cost depth sensors like Microsoft Kinect or high-resolution 3D scanner like LiDAR. Moreover, in the past few years, multi-view stereo-matching techniques have been extensively studied to recover a 3D model from videos and images, where the output is typically

[*]Corresponding author: Tel.: +39-348-250-5489;
e-mail: `abeer.irfan@polito.it` (Muhammad Abeer Irfan)

a point cloud [5]. However, in each case, the generated point cloud is intrinsically noisy, which has called for new methods for point cloud denoising [6, 7, 8, 9]. Moving least squares (MLS)-based methods [10, 11], and locally optimal projection (LOP)-based approaches [12, 13] are considered as two main classes of point cloud denoising techniques. Still, these are usually prone to over-smoothing [8, 9] due to the utilization of local operators. Sparse-based methods [8, 14] assume sparse representation of the point normals and achieve good performance [15]. However, at the higher noise levels, the inaccurate estimation of normal or the local plane can lead to over-smoothing or over-sharpening [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 smoothing prior.

With the improvement of computer graphics, computer vision technology, and optical components, in addition to laser scanning sensors, economical low-cost RGB-D cameras have been developed such as the Astra, Astra S, Astra Pro, Intel RealSense [16, 17, 18], and Microsoft Kinect [19, 20]. With the advent of RGB-D cameras, it is quite easy to generate the point cloud of an object. However, the point cloud obtained with these cameras will have significant noise in geometry and color, exhibiting artifacts because of various view angles, reflective material or characteristics of the surfaces of the objects, light intensities, as well as the limitations of sensors [21]. Alongside the geometry of objects and scenes, which is important for many applications, surface colors and details play an essential role in several virtual and augmented reality applications [22]. The color is an essential attribute of a point cloud, and it has been considered a key feature for point cloud segmentation [23, 24, 25] and retrieval of 3D models [26, 27]. Noise in the colors of a point cloud may lead e.g. to wrong segmentation. In recent works, geometry denoising has received a lot of attention, whereas there is just one work for denoising the color of a point cloud using Graph Laplacian regularizer (GLR) coupled with alternating direction method of multipliers [28]; yet, there are numerous applications employing both the geometry and color attribute of a point cloud.

In current literature, the geometry of a point cloud is expressed as a graph, which is used for denoising by convex optimization [29]. In this paper, we propose the joint use of the geometry and the color attribute of the points in a point cloud to remove noise. We note that the color attribute is a powerful and very informative feature that is indeed correlated with the geometry, as also observed in [25, 28, 30]. Knowledge of the color can be exploited to improve the denoising process for geometry noise, and indeed geometry and color can be jointly employed to remove geometry *and* color noise. Based on this notion, the main contributions of our work are as follows:

1. Construction of a k-Nearest Neighbor (k-NN) graph based on geometry and color and use of convex optimization for denoising the color of a point cloud.
2. Construction of a joint geometry and color k-NN graph and denoising of the geometry of a point cloud.
3. Construction of a joint geometry and color k-NN graph and denoising of both color and geometry of a point cloud.

We note that this paper is the first to construct joint geometry-color graphs, as well as the first to address the problem of joint geometry-color denoising. Extensive simulation results on synthetic and real point clouds show that our proposed algorithm for color, geometry, and joint geometry-color denoising outperforms state-of-the-art techniques using both subjective and objective quality metrics.

The rest of the paper is organized as follows. In Sec. 2, an overview of related works is presented. Sec. 3 reviews the basics of graph signals and describes the construction of various graphs. Sec. 4 explains the proposed algorithms, which incorporate color-only, geometry-only, and joint geometry and color denoising. Performance evaluation metrics are discussed in Sec. 5. The subjective and objective experimental results are presented in Sec. 6. Finally, conclusions are drawn in Sec. 7.

2. Related Work

In the literature, point cloud denoising techniques can be classified into two categories: outlier removal and noise removal, i.e., surface smoothing techniques. Outlier removal techniques can be classified into two main approaches: statistical and model-based.

Statistical approaches: The primary purpose of the statistical methods is to remove the outliers based on the distribution of each point with respect to its neighbors or the number of neighbors to each point. The statistical outlier removal approach proposed in [31] computes the mean distance of each point from all its neighbors. The mean and standard deviation identify likely intervals of the global distances, and all the points whose mean distance is outside the defined range are counted as noise and eliminated from the point cloud. An alternate extensively used approach called radius outlier removal (ROR) is based on the number of neighbors; here, the number of neighbors of each point is computed in a defined radius and the points whose number of neighbors are less than a threshold are considered outliers and are eliminated from the point cloud [1].

Model-based approaches: These approaches employ the notion that noisy points are generally distant from the surface of the object. Due to the unknown underlying surface, the general idea is to approximate the surface with some model, e.g., sphere, plane, square, and so forth, and then compute the distance of each point to the surface of the model. The points having significant distance are considered as noise and removed [32]. A progressive plane algorithm is described in [33], whereby using the average normal and 3D coordinates of a given point set, a plane is estimated. A least-square plane fitting technique is used for computing the distances between each point to the plane and construct a progressive plane; a hybrid algorithm is introduced in [34] based on [33]. The issue with these methods is that in complex geometry, details can be missed because it is hard to estimate complex regions with simple models.

Surface smoothing techniques mainly contain moving least squares (MLS), locally optimal projection (LOP)-based methods, sparsity-based methods, and graph based-methods.

MLS-based methods: MLS-based methods typically use an estimated smooth surface from the given input to fit the point cloud, and then the points are projected onto the fitted surface. The MLS projection operator in [35] is used by [36] to compute the optimal MLS surface of the point cloud; this is considered a reference surface, and then the points are moved around the surface towards it.

Spherical fitting denoising based on MLS algebraic point set surfaces is proposed in [10], along with its variant [37]. This method overcomes unstable reconstruction in case of high curvature and enhances stability at a low sampling rate in comparison to the MLS-based approach. Several extensions of MLS, such as robust MLS [38] and robust implicit MLS [11], have also been proposed. These MLS-based methods can provide a smooth surface from significantly noisy input but are usually prone to over-smoothing, and are very sensitive to outliers [8, 9].

LOP-based methods: Unlike MLS-based methods, these methods do not measure specific surface parameters; LOP [12] enforces a uniform distribution over the given input point cloud and provides a set of points that represent the underlying surface. Its variant weighted LOP (WLOP) [39] provides a more uniformly distributed output by modifying the repulse term according to the local density. Moreover, anisotropic WLOP [13] remodels WLOP by using an anisotropic weighting function for better preservation of sharp features. Due to the use of local operators, LOP-based methods are also affected by over-smoothing [8, 9].

Sparsity-based methods: Sparsity-based methods seek a sparse representation of some features of geometry based on local planarity assumption. These methods solve a global minimization problem in l_1 [14] or l_0 [8] norm for the sparse reconstruction of surface normals and minimize a cost function to update the point positions with surface normals. Moving robust principal components analysis [7] is an efficient method for solving a minimization problem to update the position of each point. Weighted

l_1 minimization helps preserving sharp features. These approaches provide state-of-the-art performance [15]; however, the normal approximation can be affected by high-level noise, which leads to over-sharpening and over-smoothing [8].

Graph-based methods: Recently, graph-based techniques have been used for denoising point clouds. The conventional approach is to construct a k-NN graph as it makes geometric structure explicit [40]. The points in a given point cloud are considered as nodes, and each node is connected through edges to its k nearest neighbors with weights that reflect inter-node similarities based on geometric information [29, 41], and employ convex optimization to enforce smoothness on the geometry-only graph signal [29]. Manifold denoising based on spectral graph wavelet (MSGW) [42] also used a geometry-only graph for denoising. The local tangent space-based graph is used for robust denoising of piece-wise smooth manifolds (RPSM) [43]. The disadvantage of these approaches is that holes are typically formed in the denoised output (see Fig. 1) as the correct position of a point is estimated based on the *noisy* geometry. This can lead to errors in estimating the local surface as the information of the manifold is based only on geometry. In this paper, we employ the same convex optimization as in [29], but we carefully exploit both geometry and color to relocate each point to its correct position, avoiding the artifacts generated by denoising techniques employing only geometry information.

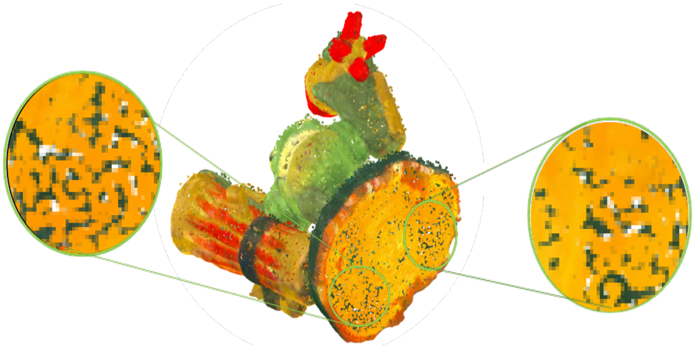


Fig. 1: Green_monster model: Geometry denoised from geometry-only graph. The noisy points are moved towards their nearest neighbors rather than their correct positions, opening holes in the surface [29].

2.1. Graph taxonomy

We provide a short review of basic concepts in graph signal processing [44, 45, 29, 41] that will be utilized in our denoising approach. 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 N x N matrix. A graph signal $g(\mathcal{G})$ for a given graph $\mathcal{G}$ is defined on the vertices of $\mathcal{G}$ as $g : \mathcal{V} \rightarrow \mathbb{R}^n$ for some dimension n. Fig. 2 shows an illustration of a graph for a point cloud.

3. Proposed method - Graph construction

3.1. Joint geometry/color graph construction from noise-free geometry and noisy color

For the color denoising of a point cloud, the graph is constructed from the geometry and color attribute of the points. In this case, we are given an input point cloud having geometry and color information; this is denoted as $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, ..., \boldsymbol{p}_N\}$ with $\boldsymbol{p}_i = [\boldsymbol{X}_i \ \boldsymbol{C}_i] \in \mathbb{R}^6$ containing 3D Euclidean geometry coordinates $\boldsymbol{X}_i \in \mathbb{R}^3$ and RGB color information $\boldsymbol{C}_i \in \mathbb{R}^3$ for point $\boldsymbol{p}_i$ and N is the total number of points. In the case of joint geometry/color graph construction, we employ a k-NN graph to express an explicit geometric structure [40]. Every vertex is connected through an edge to its k nearest neighbors with an associated weight, which is

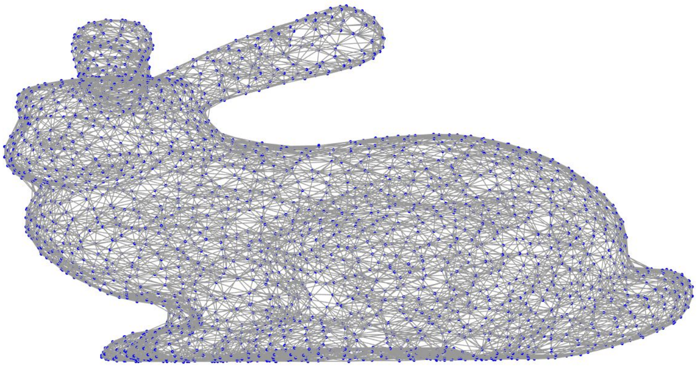


Fig. 2: Bunny model: An illustration of graph construction from geometry of a 3D point cloud.

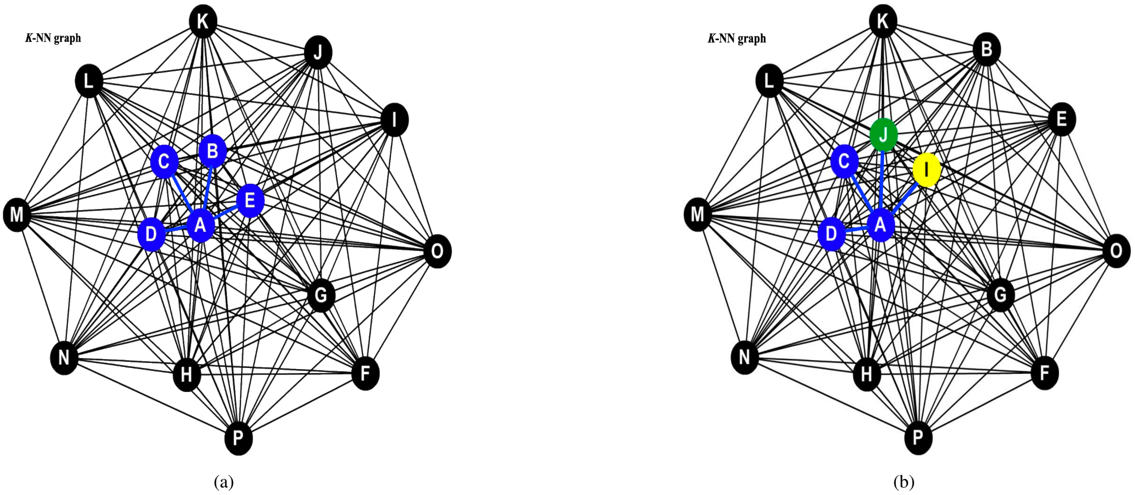


Fig. 3: (a) Illustration of a joint geometry and color k-NN graph for $k = 4$; node A is connected to the nodes that are closer to it by geometric and color distance, blue colored nodes are analogous to same color within proximity. (b) Illustration of a geometry only k-NN graph for $k = 4$; node A is connected to nodes that are within its proximity regardless of color of each connected node.

computed using some metric. In this context, we choose to use the Euclidean distance. A very standard choice for the weighting function is to use the thresholded Gaussian kernel [46]:

$$w_{i,j} = \begin{cases} \exp\left(-\frac{\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2}{2\theta_X^2} - \frac{\|\boldsymbol{C}_i - \boldsymbol{C}_j\|^2}{2\theta_C^2}\right) & \text{if } \boldsymbol{p}_j \in \phi_k(i) \\ & \text{or } \boldsymbol{p}_i \in \phi_k(j) \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{1}$$

Parameters θ_X and θ_C determine the relative contribution of geometry and color in the construction of the resulting k-NN graph, $\phi_k(i)$ is the set of k nearest neighbors to point $\boldsymbol{p}_i$, and $\phi_k(j)$ is the set of k nearest neighbors to point $\boldsymbol{p}_j$. We stress that in this setting, a color-only graph is not a good choice because geometrically distant points may have the same color and different contents, which may lead to the construction of a wrong graph. Fig. 3 illustrates how color can positively affect the graph construction by generating a set of neighbors that is more semantically meaningful than a geometry-only graph. The resulting k-NN graph is denoted as $\mathcal{G}_1$.

3.2. A joint geometry/color graph construction from noisy geometry and noise-free color

In the setting considered here, the given point cloud $\mathcal{P}$ contains noisy geometry and noise-free color information for point $\boldsymbol{p}_i$. The proposed approach described in Eq. 1 generates a k-NN graph based on color similarity and geometry proximity in a 3D plane, with a suitable choice of θ_X and θ_C. The resulting k-NN graph is denoted as $\mathcal{G}_2$.

3.3. A joint geometry/color graph construction from noisy geometry and noisy color

A joint k-NN graph constructed from the noisy geometry and color is used for the denoising of both the geometry and color of a point cloud. The given noisy point cloud $\mathcal{P}$ consists of noisy geometry and noisy color information for point $\boldsymbol{p}_i$, with a potentially different choice of θ_X and θ_C. The resulting k-NN graph is denoted as $\mathcal{G}_3$.

4. Proposed algorithm - Optimization

Having introduced the mechanism of graph construction for the different denoising scenarios we are interested in, we now present the proposed approach for denoising the color-only, geometry-only, and combined geometry and color noise of a point cloud.

The very first step is to remove the outliers from the entire point cloud. The outliers have distinct characteristics in that each outlier has a sparse neighborhood; therefore, the detection and removal of outliers are density-based [29, 32]. In the following, we follow the ROR approach, in which a sphere with a predefined radius r is formed, having each point $\boldsymbol{p}_i$ as its center. u_i denotes the number of points contained in each sphere. We then calculate $\bar{u} = \frac{\sum_{i=1}^{n} u_i}{N}$, where N is the total number of points. A point $\boldsymbol{p}_j$ is considered as an outlier if $u_j < \bar{u}$.

4.1. Color denoising

The algorithm presented in the following has the objective to perform color denoising by exploiting the graph $\mathcal{G}_1$ from geometric and color information of the noisy point cloud. We define a graph signal $g(\mathcal{G}_1)$, where each vertex of $\mathcal{G}_1$ is associated with the geometry $\boldsymbol{X}_i$ and color $\boldsymbol{C}_i$ of the corresponding point $\boldsymbol{p}_i$ of a point cloud $\mathcal{P}$. The input noisy point cloud consists of both geometry and color. Each point can be expressed as $\boldsymbol{p}_i$= $[\boldsymbol{X}_i, \boldsymbol{C}_i + \boldsymbol{W}_i]$, $\boldsymbol{X}_i$ being the geometry, $\boldsymbol{C}_i$ being the unknown true color, and $\boldsymbol{W}_i$ the color noise. The objective is to estimate $\boldsymbol{C}_i$ for each point of the cloud.

For denoising, we exploit the regularity of the color and its correlation with the proximity of the points. Graph gradient can be used to measure the degree of smoothness of a graph signal [46]. In the following, we propose a convex optimization technique that enforces the regularity of the denoised color attributes on $\mathcal{G}_1$. In particular, the denoising problem can be written as follows:

$$\widehat{\boldsymbol{C}} = \arg\min_{C} ||C - g||_2^2 + \gamma||\nabla_{\mathcal{G}_1} C||_2^2 \tag{2}$$

Here, $C = C(\mathcal{G}_1)$ is a graph signal on $\mathcal{G}_1$ containing an RGB color attribute on each node; the estimated denoised color for the whole point cloud is referred to as $\widehat{\boldsymbol{C}}$, the observed noisy signal is represented by the graph signal g defined above, γ is a parameter for regularization and $\nabla_{\mathcal{G}_1} C$ represents the gradient of the graph signal C on the graph $\mathcal{G}_1$ (see [46]). Eq. 2 has two terms; the first is a fidelity term that enforces the denoised point to be not too far from its observed color, while the second term promotes smoothness of the denoised point cloud on $\mathcal{G}_1$ via Tikhonov regularization. The same method can also employ Total Variation (TV) regularization with the constraint that the underlying manifold of a point cloud is piece-wise smooth. This leads to the following convex optimization problem:

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

The problems in Eq. 2 and 3 can be solved by alternating direction method of multipliers (ADMM) [47]. This yields the denoised point cloud $\widehat{\boldsymbol{p}}_i = [\boldsymbol{X}_i\ \widehat{\boldsymbol{C}}_i]$, where $\widehat{\boldsymbol{C}}_i$ is the denoised color attribute for point $\boldsymbol{p}_i$.

4.2. Geometry denoising

For geometry denoising, we exploit the graph $\mathcal{G}_2$ constructed from both geometry and color information of the noisy point cloud defined in Sec. 3.2. We define a graph signal $g(\mathcal{G}_2)$, where each vertex of $\mathcal{G}_2$ is associated with the geometry information $\boldsymbol{X}_i$ and color information $\boldsymbol{C}_i$ of the point cloud. The point $\boldsymbol{p}_i$ can be expressed as $\boldsymbol{p}_i = [\boldsymbol{X}_i + \boldsymbol{Z}_i, \boldsymbol{C}_i]$, $\boldsymbol{X}_i$ being the unknown true geometry of $\boldsymbol{p}_i$, $\boldsymbol{Z}_i$ the geometry noise, and $\boldsymbol{C}_i$ the color attribute, with $\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{C}_i \in \mathbb{R}^3$. The objective is to estimate $\boldsymbol{X}_i$ for each point. This can be done using the following denoising algorithm described in Eq. 4, which moves the points closer to their exact location based on a smoothness assumption applied to the joint geometry and color information embedded in graph $\mathcal{G}_2$. Graph $\mathcal{G}_2$ can be considered an 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.

$$\widehat{\boldsymbol{X}} = \arg\min_X ||X - g||_2^2 + \gamma ||\nabla_{\mathcal{G}_2} X||_2^2 \tag{4}$$

Here, $X = X(\mathcal{G}_2)$ is a graph signal on $\mathcal{G}_2$ containing geometry values on each node; the estimated denoised geometry for the whole point cloud is referred to as $\widehat{\boldsymbol{X}}$. The convex optimization problem in Eq. 4 promotes the smoothness of the graph signal X defined on $\mathcal{G}_2$ utilizing the combined geometry and color k-NN graph. The fidelity term in Eq. 4 enforces the denoised points to move to their observed position. The smoothness of the graph can be measured by graph gradient $\nabla_{\mathcal{G}_2} X$ of the signal X. This can also be done by Total variation using Eq. 5. This yields the denoised point cloud $\widehat{\boldsymbol{p}}_i = [\widehat{\boldsymbol{X}}_i\ \boldsymbol{C}_i]$, where $\widehat{\boldsymbol{X}}_i$ is the denoised geometry for point $\boldsymbol{p}_i$.

$$\widehat{\boldsymbol{X}} = \arg\min_X ||X - g||_2^2 + \gamma ||\nabla_{\mathcal{G}_2} X||_1 \tag{5}$$

4.3. Combined geometry and color denoising

Up to this point, we have taken advantage of the correlation between geometry and color to remove one type of noise, either on the geometry or on the color. In the following, we consider the case of simultaneous denoising of geometry and color noise employing the constructed k-NN graph $\mathcal{G}_3$ described in Sec. 3.3. The point $\boldsymbol{p}_i$ can be expressed as $\boldsymbol{p}_i = [\boldsymbol{X}_i + \boldsymbol{Z}_i, \boldsymbol{C}_i + \boldsymbol{W}_i]$, $\boldsymbol{X}_i$ being the unknown true geometry and $\boldsymbol{C}_i$ the noisy RGB color of $\boldsymbol{p}_i$, $\boldsymbol{Z}_i$ the geometry noise, and $\boldsymbol{W}_i$ is the color noise, with $\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{C}_i, \boldsymbol{W}_i \in \mathbb{R}^3$. The objective is to estimate $\boldsymbol{X}_i$ and $\boldsymbol{C}_i$ for each point. In the proposed algorithm, we use a weighting procedure for each component in the feature vector representing a point in the cloud to generate a weighted input signal $\tilde{\boldsymbol{X}}_i = [\Omega_1 \boldsymbol{X}_i, \Omega_2 \boldsymbol{C}_i]$ where, Ω_1 and $\Omega_2 \in \mathbb{R}$. $\tilde{\boldsymbol{X}}_i$ is then denoised using Eq. 6, which moves the points closer to their actual position and their true color based on the smoothness assumption in graph $\mathcal{G}_3$. The weights Ω_1 and Ω_2 enforce the fidelity term in Eq. 6 that the geometry and color of the denoised points do not move too far from their observed position and their actual color.

$$\widehat{\tilde{\boldsymbol{X}}} = \arg\min_{\tilde{X}} ||\tilde{X} - g||_2^2 + \gamma ||\nabla_{\mathcal{G}_3} \tilde{X}||_2^2 \tag{6}$$

Here, $\tilde{X} = \tilde{X}(\mathcal{G}_3)$ is a graph signal on $\mathcal{G}_3$ containing a geometry value and color value on each node; $\widehat{\tilde{\boldsymbol{X}}}$ represents the estimated geometry and color denoised values. The convex optimization problem in Eq. 6 promotes the smoothness of the graph signal $\tilde{X}$ defined on $\mathcal{G}_3$ utilizing the combined geometry and color k-NN graph. The smoothness of the graph can be measured by graph gradient $\nabla_{\mathcal{G}_3} \tilde{X}$ of the signal $\tilde{X}$. This yields the denoised point cloud $\widehat{\boldsymbol{p}}_i = [\widehat{\boldsymbol{X}}_i\ \widehat{\boldsymbol{C}}_i]$, with $\widehat{\boldsymbol{X}}_i$ being obtained from the first three components of $\widehat{\tilde{\boldsymbol{X}}}$ divided by Ω_1, and $\widehat{\boldsymbol{C}}_i$ being obtained from the last three components of $\widehat{\tilde{\boldsymbol{X}}}$ divided by Ω_2.

Table 1: Parameter setting of the proposed denoising techniques for both the synthetic and natural point cloud models with different noise levels.

Techniques	Noise level in color distribution	Noise added in color attribute	Noise added in geometry attribute	Noise level in geometry distribution	k (Synthetic point clouds)	γ (Synthetic point clouds)	k (Natural point clouds)	γ (Natural point clouds)	θ_X	θ_C	Ω_1	Ω_2
Color denoising	$\sigma = 10$	100%	0%	—	8	0.05	25	1.5	0.88	3.5	—	—
	$\sigma = 15$				8	0.08						
	$\sigma = 20$				8	0.13						
	$\sigma = 25$				8	0.16						
	$\sigma = 30$				8	0.20						
	$\sigma = 40$				8	0.28						
Geometry denoising	—	0%	50%	$\sigma = 0.3$	15	0.075	10	0.075	1.0	2.4	—	—
				$\sigma = 0.4$	15	0.075						
				$\sigma = 0.5$	15	0.1						
Combined geometry & color denoising	$\sigma = 20$	100%	50%	$\sigma = 0.3$	100	0.025	70	0.075	0.7	14	0.20	0.80
	$\sigma = 30$			$\sigma = 0.4$	100	0.025						
	$\sigma = 40$			$\sigma = 0.5$	100	0.04						

5. Evaluation Metrics

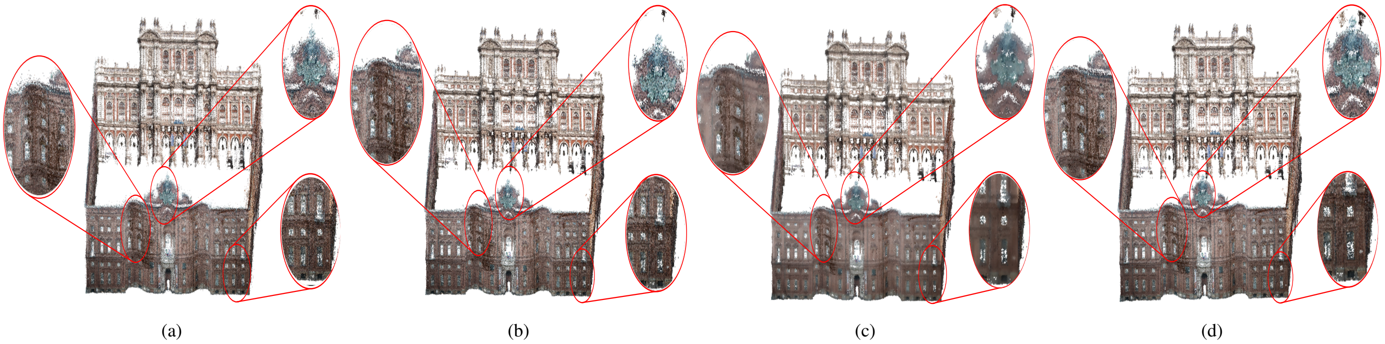
5.1. Evaluation Metrics for color denoising

The metrics used for the objective evaluation of the proposed color denoising algorithm are mean-squared-error (MSE) and peak signal-to-noise ratio (PSNR):

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{C}_i - \widehat{\boldsymbol{C}}_i\|^2 \tag{7}$$

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right) \tag{8}$$

where $\boldsymbol{C}_i$ and $\widehat{\boldsymbol{C}}_i$ represent the color attribute of the points in ground-truth and denoised point cloud, respectively, N is the number of points in a point cloud $\mathcal{P}$.


(a)

(b)

(c)

(d)

Fig. 4: Palazzo_Carignano_Dense model illustration. (a) noisy input, (b) outlier-free input, color denoised results by (c) proposed algorithm using Tikhonov regularization, and (d) using TV.

5.2. Evaluation Metrics for geometry denoising

For image denoising, quality metrics are based on a one-to-one correspondence between the samples of ground-truth and denoised data. However, in the case of point clouds, such constraint would be practically too restrictive. The Hausdorff distance overcomes the notion of a vertex to vertex distance [48, 49, 50, 51]. A triangular mesh $\mathcal{M}$ consists of a set $\boldsymbol{p}_i$ of points and a set $\mathcal{T}$ of triangles defining how the vertices from $\boldsymbol{p}_i$ are associated together, denoted by $\mathcal{M} = (\boldsymbol{p}_i, \mathcal{T})$. We consider a mesh corresponding to $\boldsymbol{p}_i$ as $\mathcal{M} = (\boldsymbol{p}_i, \mathcal{T})$ and a denoised point cloud $\widehat{\boldsymbol{p}}_i$. We are interested in measuring the distances between sets of points in the two point clouds.

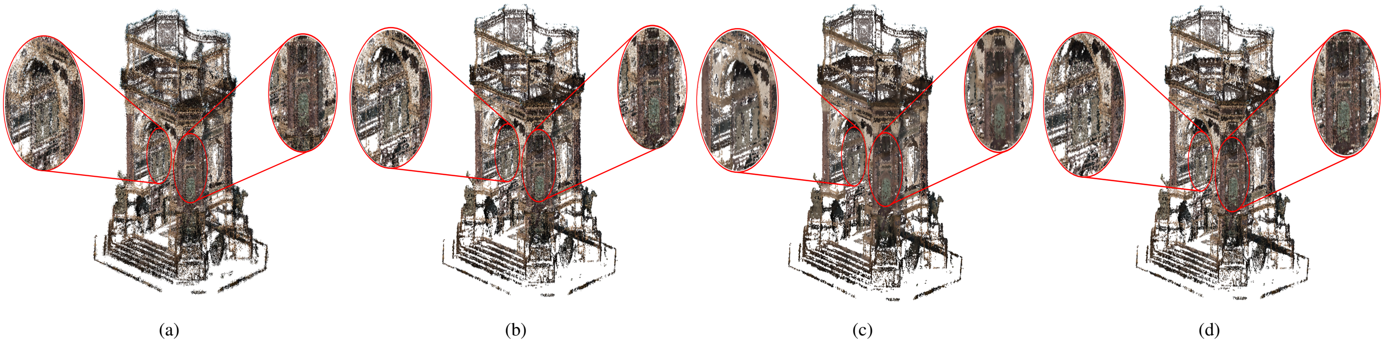


Fig. 5: Arco_Valentino model illustration. (a) noisy input, (b) outlier-free input, color denoised results by (c) proposed algorithm using Tikhonov regularization, and (d) using TV.

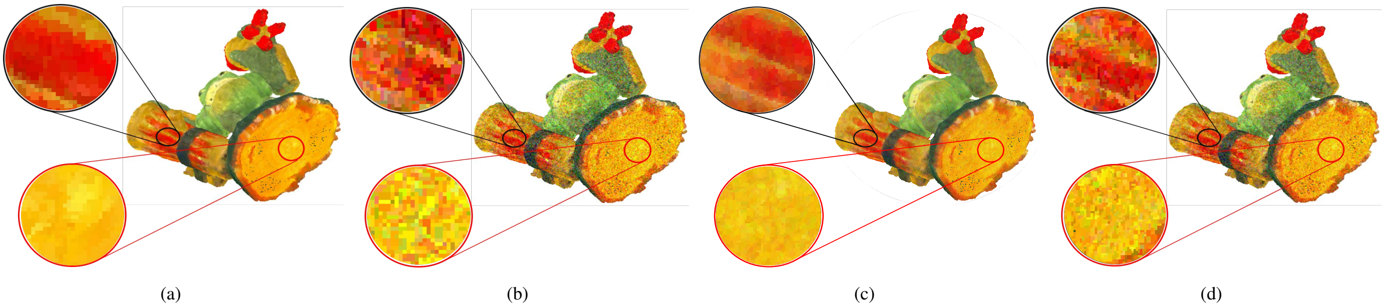


Fig. 6: Green_monster model: (a) ground-truth (b) noisy point cloud with noise level of $\mu = 0$ and $\sigma = 30$, color denoised results by (c) proposed algorithm using Tikhonov, and (d) using TV.

5.2.1. *Hausdorff distance*

The distance $d(\widehat{\boldsymbol{X}}_i, \mathcal{M})$ between a given point $\widehat{\boldsymbol{X}}_i \in \widehat{\boldsymbol{p}}_i$ and any point $\boldsymbol{X}_i \in \mathcal{M}$ is defined as:

$$d(\widehat{\boldsymbol{X}}_i, \mathcal{M}) = \min_{\boldsymbol{X}_i \in \mathcal{M}} \|\boldsymbol{X}_i - \widehat{\boldsymbol{X}}_i\|_2 \tag{9}$$

The Hausdorff distance between $\widehat{\boldsymbol{X}}$ and $\mathcal{M}$ is denoted as $d_H(\widehat{\boldsymbol{X}},\ \mathcal{M})$ and is given by:

$$d_H(\widehat{\boldsymbol{X}}, \mathcal{M}) = \max_{\widehat{\boldsymbol{X}}_i \in \widehat{\boldsymbol{p}}_i} d(\widehat{\boldsymbol{X}}_i, \mathcal{M}) \tag{10}$$

The $d_H(\widehat{\boldsymbol{X}}, \mathcal{M})$ and $d_H(\mathcal{M}, \widehat{\boldsymbol{X}})$ are referred to as forward and backward distance, respectively. These distances are not symmetrical, i.e., $d_H(\widehat{\boldsymbol{X}}, \mathcal{M}) \neq d_H(\mathcal{M}, \widehat{\boldsymbol{X}})$. The symmetrical Hausdorff distance $d_S(\widehat{\boldsymbol{X}}, \mathcal{M})$ can be computed as:

$$d_S(\widehat{\boldsymbol{X}}, \mathcal{M}) = \max\left(d_H(\widehat{\boldsymbol{X}}, \mathcal{M}), d_H(\mathcal{M}, \widehat{\boldsymbol{X}})\right) \tag{11}$$

The distance between any point $\boldsymbol{X}_i$ belonging to $\mathcal{M}$ and $\widehat{\boldsymbol{X}}$ can be computed analytically, as it can be reduced to the minimum of the distances between $\boldsymbol{X}_i$ and all the triangles $T \in \mathcal{T}$. If the orthogonal projection $\widehat{\boldsymbol{X}}_i$ of $\boldsymbol{X}_i$ on the plane of T is inside the triangle, the point-to-triangle distance is nothing but a point-to-plane distance. When the projection lies outside T, the point-to-triangle distance is the distance between $\boldsymbol{X}_i$ and the closest point $\widehat{\boldsymbol{X}}_i'$ of T, which lies necessarily on one of the sides of T [48, 49, 50].

The point-to-mesh distance in Eq. 9 can also be used to calculate the mean distance d_m between $\widehat{\boldsymbol{X}}$ and $\mathcal{M}$:

$$d_m(\widehat{\boldsymbol{X}}, \mathcal{M}) = \frac{1}{N} \sum_{\widehat{\boldsymbol{X}}_i \in \widehat{\boldsymbol{X}}} d(\widehat{\boldsymbol{X}}_i, \mathcal{M}) \tag{12}$$

$$\zeta = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - d_m)^2} \tag{13}$$

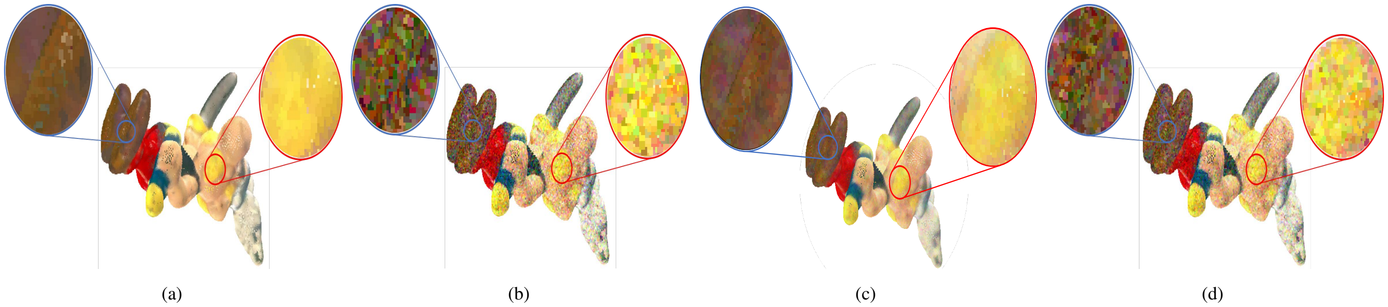


Fig. 7: Asterix model: (a) ground-truth (b) noisy point cloud with noise level of $\mu = 0$ and $\sigma = 30$, color denoised results by (c) proposed algorithm using Tikhonov, and (d) using TV.

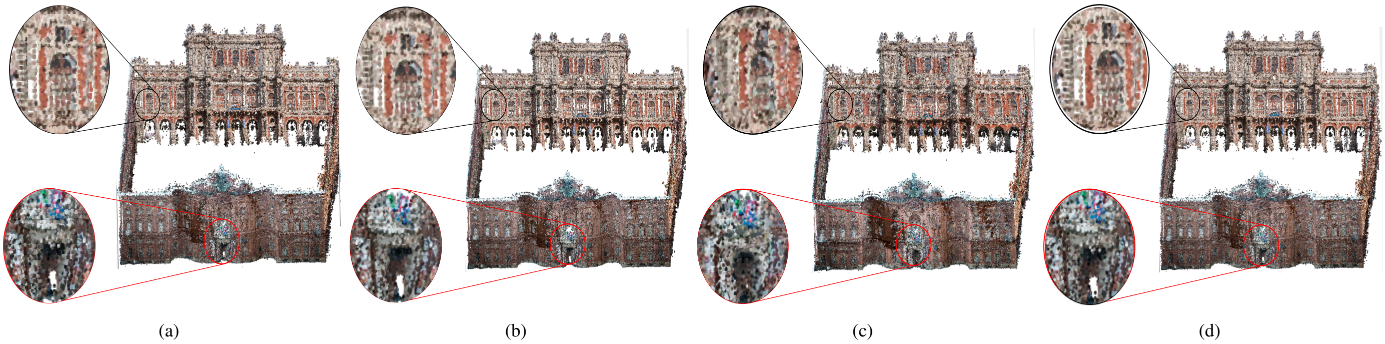


Fig. 8: Palazzo_Carignano_Dense model illustration. (a) noisy input, (b) outlier-free input, geometry denoised results by (c) proposed algorithm, and (d) geometry-only graph [29].

where $d_i = \|\boldsymbol{p}_i - \widehat{\boldsymbol{p}}_i\|^2$; ζ represents the standard deviation of the distance between the point $\widehat{\boldsymbol{p}}_i$ and the corresponding point $\boldsymbol{p}_i$. We computed the Hausdorff distance using the cloud-to-mesh (C2M) metric in CloudCompare [52]. The ground-truth 3D models act as the reference meshes to their respective denoised point clouds. The outputs of the C2M metric are d_H, d_m.

For performance comparison between the proposed algorithm and RPSM [43], we need a ground-truth mesh for each point cloud of the Greyc dataset [53]. Due to the large memory requirement of RPSM, we performed experiments on sub-sampled point clouds, for which the sub-sampled ground-truth mesh is not available. Hence, we employ on these sub-sampled point clouds the same metrics as in [54]. Assume Q and Q' represent the geometry of the noise-free and denoised point cloud respectively, where $Q = \{\boldsymbol{q}_i\}_{i=1}^{N_1}$ Q'=$\{\boldsymbol{q}'_i\}_{i=1}^{N_2}$, such that $\boldsymbol{q}_i$, $\boldsymbol{q}'_i \in \mathbb{R}^3$. We define distance metrics as follows.

Mean-square-error (MSE): It is computed as an average of the squared Euclidean distance between each point in Q and its corresponding nearest point in Q', and also between each point in Q' and its corresponding nearest point in Q:

$$\text{MSE} = \frac{1}{2N_1} \sum_{\boldsymbol{q}_i \in Q} \min_{\boldsymbol{q}'_i \in Q'} \|\boldsymbol{q}_i - \boldsymbol{q}'_i\|_2^2 + \frac{1}{2N_2} \sum_{\boldsymbol{q}'_i \in Q'} \min_{\boldsymbol{q}_i \in Q} \|\boldsymbol{q}'_i - \boldsymbol{q}_i\|_2^2 \tag{14}$$

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

$$\text{MCD} = \frac{1}{2N_1} \sum_{\boldsymbol{q}_i \in Q} \min_{\boldsymbol{q}'_i \in Q'} \|\boldsymbol{q}_i - \boldsymbol{q}'_i\| + \frac{1}{2N_2} \sum_{\boldsymbol{q}'_i \in Q'} \min_{\boldsymbol{q}_i \in Q} \|\boldsymbol{q}'_i - \boldsymbol{q}_i\| \tag{15}$$

5.3. *Experimental setup*

For our experiments, we assume that the noise follows a uniform and Gaussian distribution for geometry and color of a point cloud, respectively. This is a common assumption, see e.g. [55, 28]. The graph signal processing in our denoising algorithm has been implemented using GSPBOX [56] , and for the convex optimization we have used UNLocBox [57].

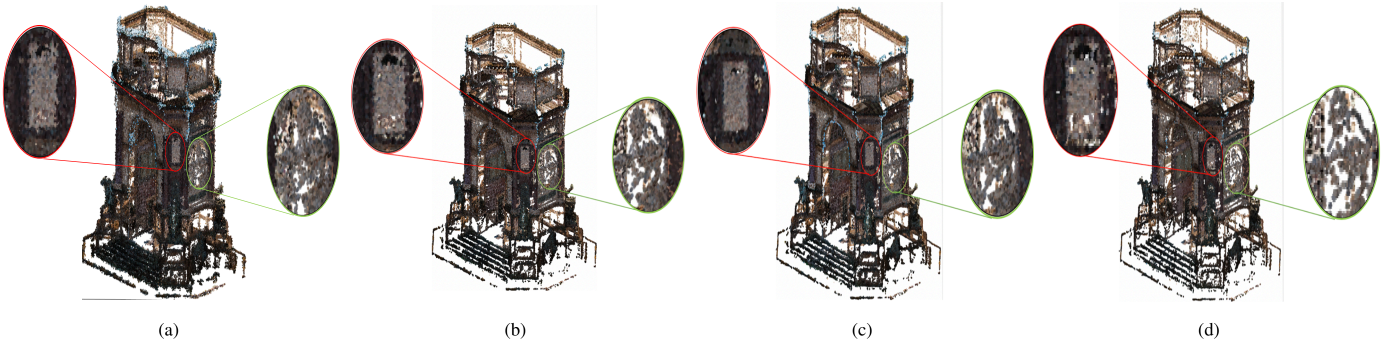


Fig. 9: Arco_Valentino model illustration. (a) noisy input, (b) outlier-free input, geometry denoised results by (c) proposed algorithm, and (d) geometry-only graph [29].

For the outlier removal we have set $\epsilon = 0.01$, $k = 5$ and $\tau = 1$ as in [29]. The parameters setting that provides the best results for the proposed algorithms for both the synthetic and real-world point clouds is shown in Tab. 1.

6. Experimental Results

This section focuses on the analysis of the experimental results, both subjective and objective on static real-world point cloud dataset available in the JPEG PLENO (GTI-UTM) database [58] and Greyc dataset of synthetic point clouds [53]. Each point cloud in real-world and synthetic dataset contains geometry and color attributes.

6.1. Subjective assessment of color denoising algorithm

The subjective analysis of color denoising algorithms has been performed for both synthetic and natural point clouds.

6.1.1. Color denoising of natural point clouds

The visual results of the proposed color denoising algorithm on natural point clouds are presented here. To the best of our knowledge, there is not much literature available for color denoising of the point cloud as anticipated in Sec. 1. Here, the qualitative results show the comparison between the proposed algorithm using Tikhonov and TV regularization. Fig. 4a and Fig. 5a show the real noisy point clouds; it can be seen that the details are not very clear due to a large amount of color noise. In some areas, two distinct regions are overlapped with each other due to the color noise.

Fig. 4b and Fig. 5b show the resulting point clouds after outlier removal. For the color denoising, the outliers must be removed before applying the proposed algorithm as they may lead to construct a wrong graph, which in turn affects the color denoising process. Fig. 4c and Fig. 5c depict the point cloud denoised using the proposed algorithm. Here, the colors are much smoother and natural by exploiting the relation of the color of the points within proximity. Due to the noise in the point cloud, details are missing, and one can not see the contours in the real point cloud. The denoised point clouds look sharper in comparison to the input noisy point clouds. The color denoising procedure helps to preserve object boundaries. Fig. 4d and Fig. 5d show the denoised point cloud using TV; it can be seen that the color is still noisy, and there is lack of details in the output point clouds. TV is not very effective at enforcing color smoothness in comparison to the proposed algorithm using Tikhonov regularization.

6.1.2. Color denoising of synthetic point clouds

The proposed algorithm for color denoising has been applied to noise-free point clouds affected by synthetic color noise; Gaussian distribution is used to add noise to the color attribute of every point in a reference point cloud while keeping the geometry noise-free. Fig. 6a and Fig. 7a present the ground-truth point cloud having noise-free geometry and color. Fig. 6b and Fig. 7b show

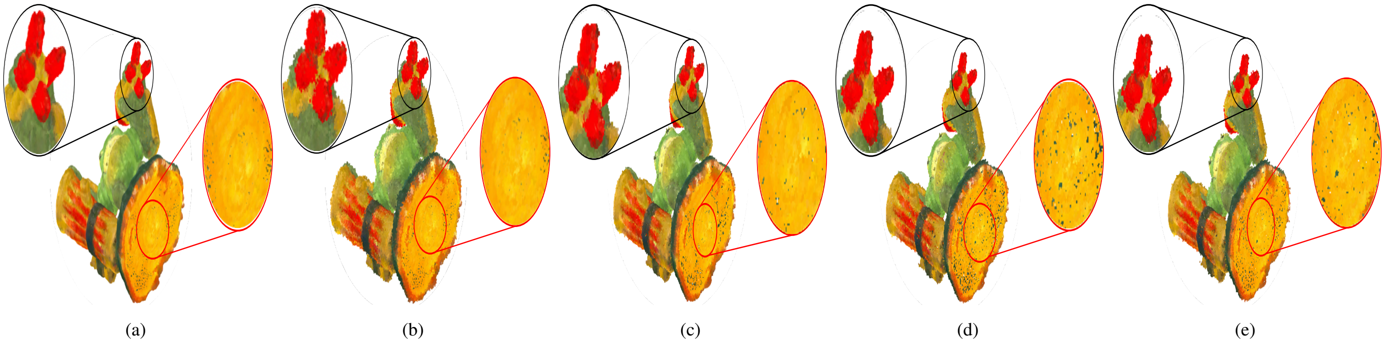


Fig. 10: Green_monster model: (a) ground-truth (b) noisy input, geometry denoised results by (c) proposed algorithm using Tikhonov regularization (d) geometry-only graph [29], and (e) MSGW [42].

the point cloud affected by Gaussian noise with $\mu = 0$ and $\sigma = 30$; adding noise to the color affects the details and causes blurring of boundaries. Fig. 6c and 7c depict the denoised output of the proposed algorithm using Tikhonov regularization. The color of the output point cloud is denoised by exploiting the correlation of color within the proximity, and the points in the k-neighborhood have a high probability of having a similar color as the surface has smooth color. Fig. 6d and Fig. 7d illustrate the denoised output using TV. The output point clouds are still noisy, and the details are not preserved. The TV technique is the least effective in terms of color denoising.

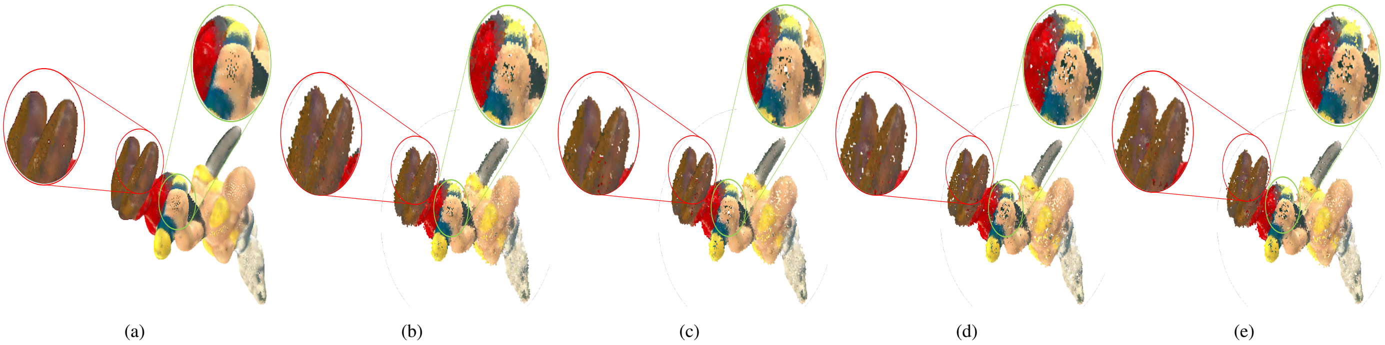


Fig. 11: Asterix model: (a) ground-truth (b) noisy input, geometry denoised results by (c) proposed algorithm using Tikhonov regularization (d) geometry-only graph [29], and (e) MSGW [42].

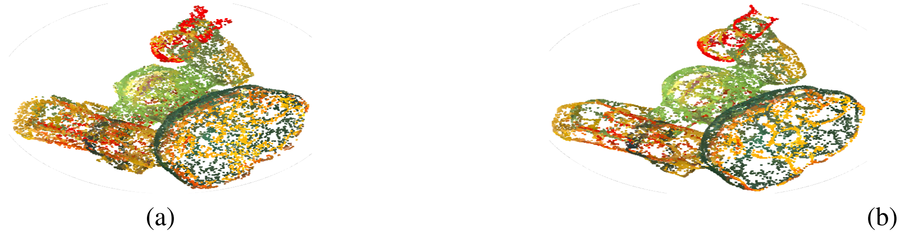


Fig. 12: Green_monster model: (a) denoised results by proposed algorithm, and (b) RPSM [43].

6.2. *Subjective assessment of geometry denoising algorithm*

The visual results of the geometry denoising algorithm applied to *Greyc* datasets and natural point clouds are discussed here.

6.2.1. *Geometry denoising of natural point clouds*

We show a visual comparison between the point cloud denoised by the proposed algorithm and denoised using a graph constructed from only geometry as in [29]. The experiment is performed on real-world natural point clouds, for which we do not have

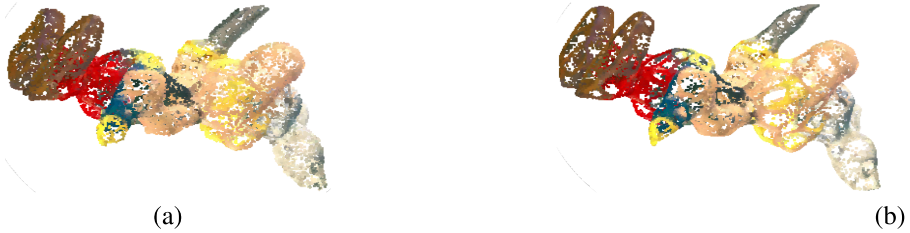


Fig. 13: Asterix model: (a) denoised results by proposed algorithm, and (b) RPSM [43].

a noiseless reference; hence, the results are only qualitative. Fig. 8a and Fig. 9a show the point clouds with real noise; it can be seen that the points with the same color are typically in a small neighborhood. Fig. 8b and Fig. 9b are the resulting output after outlier removal. Fig. 8c and Fig. 9c depict the denoised point cloud using the proposed algorithm. Here the noisy points are moved close to their original position by exploiting the correlation between the geometric coordinates and the color attribute. Fig. 8d and Fig. 9d show the denoised point cloud using the geometry-only graph approach in [29]; it can be seen in the same region that, using no color information, the noisy points are not moved to their correct location, leaving gaps as anticipated in Fig. 1, and generally providing a noisier result near object boundaries. As can be seen, the proposed method does not generally create gaps and provides a visually more natural result.

6.2.2. *Geometry denoising of synthetic point clouds*

The proposed denoising approach has also been applied to noise-free point clouds from the *Greyc* dataset [53], corrupted with uniform zero-mean synthetic geometry noise applied to 50% of the points with $\sigma = 0.3, 0.4$, and 0.5. Fig. 10a and Fig. 11a show the noise-free point clouds. Fig. 10b and Fig. 11b show the noisy point clouds with $\sigma = 0.4$. The denoised point clouds obtained by our proposed algorithm are shown in Fig. 10c and 11c; 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 [29] are shown in Fig. 10d and 11d. The output point clouds of MSGW [42] are shown in Fig. 10e and Fig. 11e. It can be seen that the geometry is not quite as much regularized; moreover, as anticipated in Sec. 1, these approaches have an adverse effect on overall geometry, as they tend to open holes in the denoised point clouds. 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.

We have also compared the proposed algorithm with RPSM [43]. Due to the limitations of RPSM as anticipated in Sec. 5.2, we performed experiments on sub-sampled point clouds. The sub-sampling performed here is on a spatial basis, setting a minimum distance between the two points equal to 0.80. The larger the distance, the fewer points will be retained in the sub-sampled output. The number of points in the sub-sampled clouds is around 20000 on average. The subjective results are shown in Fig. 12 and Fig. 13. The proposed algorithm and RPSM [43] are applied to the noisy inputs shown in Fig. 10b and Fig. 11b; the reference noise-free point clouds are shown in Fig. 10a and Fig. 11a. It can be seen that, while RPSM yields rather natural denoised color, it also tends to over-regularize the point cloud and thereby opens large holes.

6.3. *Subjective assessment of combined geometry and color denoising algorithm*

Here, the proposed algorithm has been evaluated subjectively for both real-world and synthetic point clouds.

6.3.1. *Denoising of natural point clouds*

The proposed denoising algorithm is applied to the natural point cloud with real noise in geometry and color. The visual results of the comparison between the point cloud denoising by the proposed technique using Tikhonov and TV are described here. Fig.

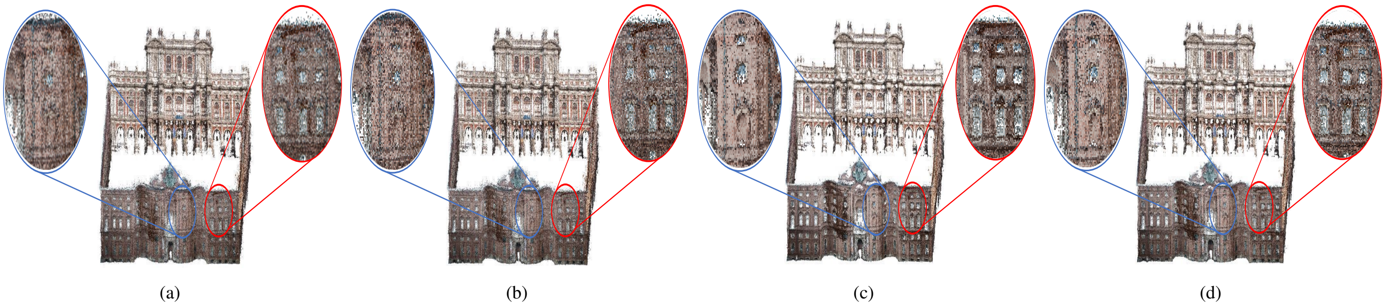


Fig. 14: Palazzo_Carignano_Dense model illustration. (a) noisy input, (b) outlier-free input, combined geometry and color denoised results by (c) proposed algorithm using Tikhonov regularizarion, and (d) using TV.

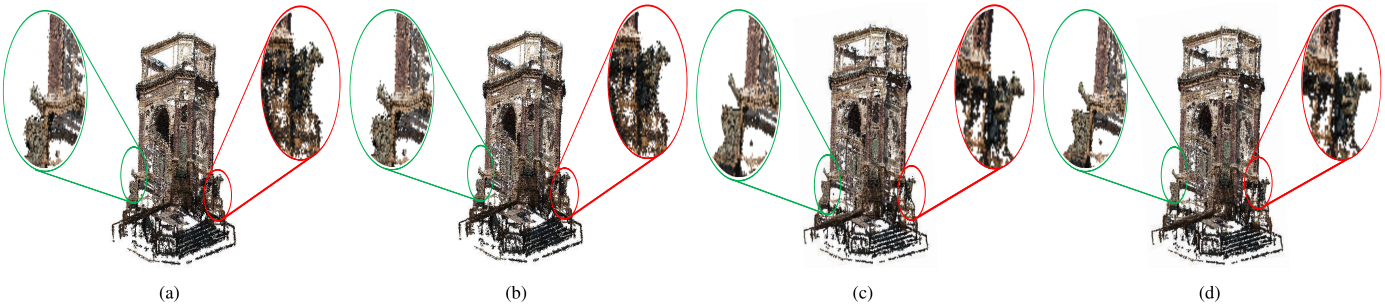


Fig. 15: Arco_Valentino model illustration. (a) noisy input, (b) outlier-free input, combined geometry and color denoised results by (c) proposed algorithm using Tikhonov regularizarion, and (d) using TV.

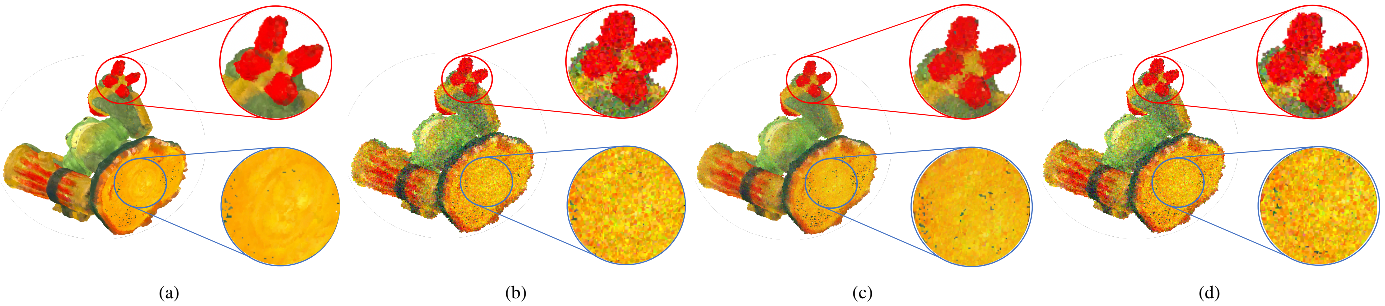


Fig. 16: Green_monster model: Highlighting only the color denoising effect in combined geometry and color denoising algorithm: (a) ground-truth (b) noisy point cloud with $\sigma = 30$ in color attribute, color denoised results in combined geometry and color by (c) proposed algorithm using Tikhonov, and (d) using TV.

14a and Fig. 15a show a real-world point cloud containing noise both in geometry and color. Due to the noise, the structural information in the real point clouds is lost, and fine details are not visible. Fig. 14b and Fig. 15b are the outcome after applying the outlier removal algorithm, which eliminates the noisiest points, thereby helping in the construction of a useful k-NN graph. Fig. 14c and Fig. 15c depict the denoised point clouds obtained by the proposed algorithm using Tikhonov regularization. The geometry noise is removed by moving the noisy points closer to their original position, and also the color becomes smoother by exploiting the correlation of geometry and color of points in a point cloud. Structural information is preserved in the resulting point cloud, and one can see more details, which were partly hidden in the noisy point cloud. Fig. 14d and Fig. 15d show the denoised point cloud by using TV. The TV also performs geometry denoising but has a very small denoising effect on the color noise; it can be seen that the details are not very well-defined in the output point clouds because the colors are not smooth enough.

6.3.2. Geometry and color denoising of synthetic point clouds

Here we present the subjective results of the proposed combined geometry and color denoising approach applied to noise-free point clouds affected by synthetic geometry and color noise. The noise in the geometry is added to 50% of the points using a

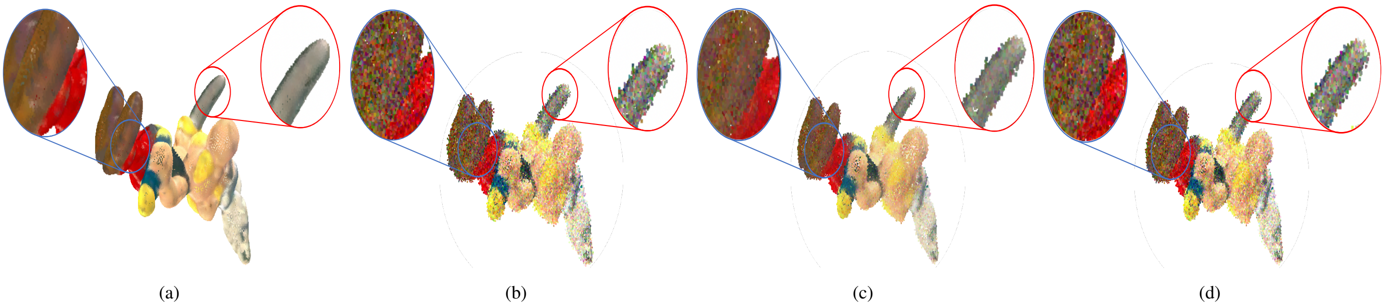


Fig. 17: Asterix model: Highlighting only the color denoising effect in combined geometry and color denoising algorithm: (a) ground-truth (b) noisy point cloud with σ = 30 in color attribute, color denoised results in combined geometry and color by (c) proposed algorithm using Tikhonov, and (d) using TV.

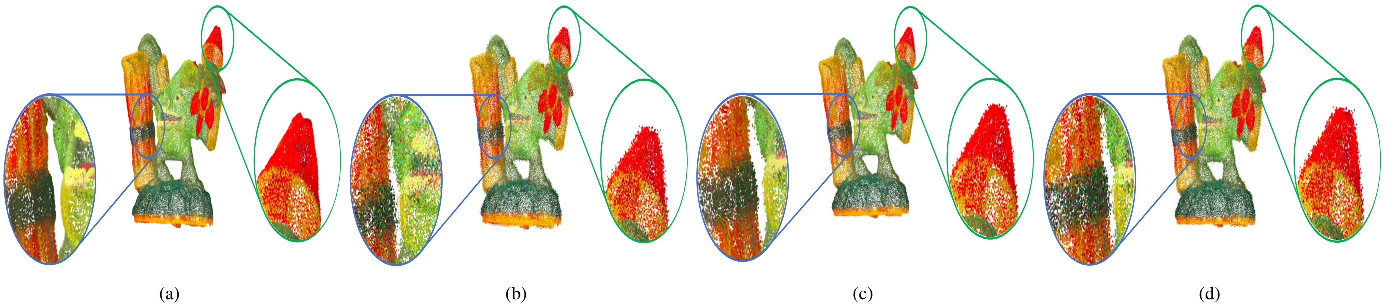


Fig. 18: Green_monster model: Same output point cloud as Fig. 16, highlighting geometry denoised effect in combined geometry and color denoising technique, (a) ground-truth (b) noisy point cloud with noise level of $\mu = 0, \sigma = 0.4$ in geometry attribute, geometry denoised results in combined geometry and color by (c) proposed algorithm, and (d) using TV.

zero-mean uniform noise with $\mu = 0$ and $\sigma = 0.3$, 0.4, and 0.5; the color of each point is corrupted by zero-mean Gaussian noise with $\sigma = 20$, 30, and 40. For clarity, the subjective results of combined geometry and color denoising are shown in separate figures focusing on the color and geometry individually. The effect of color denoising in the combined geometry and color denoising algorithm is shown in Fig. 16 and Fig. 17. The noise-free point cloud is shown in Fig. 16a and Fig. 17a. Fig. 16b and Fig. 17b depict the noisy point clouds highlighting only the color noise added in the noise-free synthetic point cloud. The denoised point clouds obtained by our proposed algorithm using Tikhonov regularization are shown in Fig. 16c and 17c, highlighting the color denoising effect in combined geometry and color denoising. It can be seen that the denoised output is cleaned from the color noise, and the colors in the output point cloud are closer to the exact color of the ground-truth input point cloud. Fig. 16d and Fig. 17d are the resulting denoised point clouds of the proposed algorithm using TV. The TV has a small effect at removing the noise from the color.

The effect of the proposed algorithm for geometry denoising can be seen in Fig. 18 and Fig. 19. The noise-free input point clouds are shown in Fig. 18a and Fig. 19a. The noisy point clouds highlighting only the geometry noise can be seen in Fig. 18b and Fig. 19b. The outcome of the proposed denoising algorithm is depicted in Fig. 18c and Fig. 19c, focusing only on the denoising effect on geometry. It can be seen that the geometry noise has been regularized, and the noisy points are moved close to their original positions; moreover, as anticipated in Sec. 1, our approach avoids generating artifacts by leaving fewer holes in the output denoised point cloud. Fig. 18d and Fig. 19d are the denoised output of TV regularization. The TV also helps in regularizing the noisy points. Overall we can perceive from the qualitative results of the synthetic point clouds that the point clouds denoised by the proposed algorithm using Tikhonov have better quality both in color and geometry denoising and exhibit fewer artifacts.

6.4. Objective evaluation on Greyc color mesh dataset

The quantitative evaluation has also been performed on the *Greyc* noise-free synthetic point clouds dataset [53].

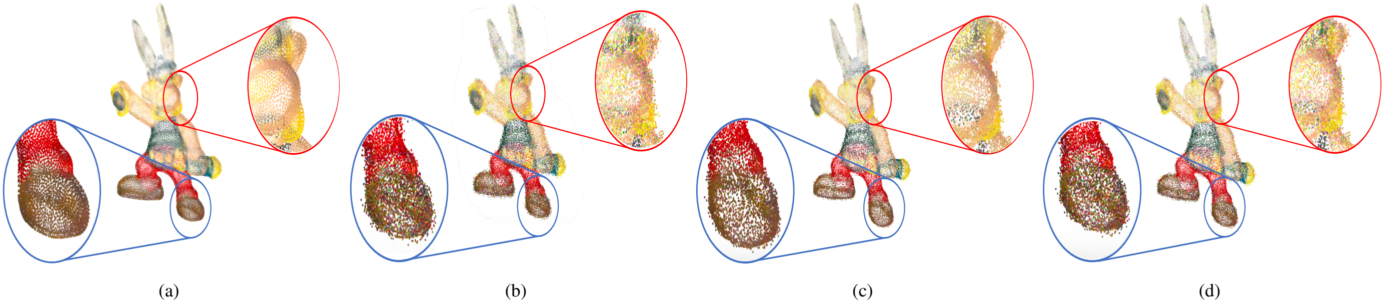


Fig. 19: Asterix model: Same output point cloud as Fig. 17, highlighting geometry denoised effect in combined geometry and color denoising technique, (a) ground-truth (b) noisy point cloud with noise level of $\mu = 0, \sigma = 0.4$ in geometry attribute, geometry denoised results in combined geometry and color by (c) proposed algorithm, and (d) using TV.

Table 2: MSE comparison of color denoising algorithm 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
$\sigma = 20$	Noisy	398.25	361.28	373.52	393.64	339.57	397.24	365.38	383.29	368.56	387.52	321.09	375.24	309.14	335.40	377.39	398.13
	Proposed	**77.75**	**103.56**	**145.23**	**82.80**	**130.65**	**63.04**	**91.64**	**150.11**	**85.83**	**77.90**	**95.91**	**89.56**	**149.53**	**93.75**	**82.52**	**84.77**
	TV	303.05	279.02	289.48	298.13	286.05	301.50	279.04	299.68	279.21	293.61	242.14	283.15	241.15	254.99	293.32	302.99
$\sigma = 30$	Noisy	869.27	781.84	816.58	867.14	722.06	875.25	796.19	850.03	797.42	850.88	692.54	810.94	645.11	713.61	811.92	882.35
	Proposed	**109.74**	**148.53**	**214.89**	**120.77**	**189.47**	**89.12**	**138.24**	**217.08**	**130.09**	**112.13**	**157.44**	**136.40**	**250.35**	**148.58**	**133.63**	**123.06**
	TV	590.97	540.25	564.67	589.46	548.87	594.55	542.77	596.83	539.70	575.89	467.70	544.00	449.27	485.12	561.85	602.88
$\sigma = 40$	Noisy	1506.10	1329.00	1384.60	1462.90	1241.90	1469.80	1347.80	1462.10	1355.50	1462.60	1171.90	1372.90	1092.10	1217.80	1376.20	1519.90
	Proposed	**137.31**	**210.90**	**281.89**	**149.67**	**262.51**	**133.94**	**184.53**	**279.98**	**175.90**	**144.71**	**228.37**	**185.25**	**385.53**	**213.93**	**242.06**	**154.26**
	TV	1119.50	995.38	1033.80	1081.70	979.85	1088.40	1000.40	1107.10	1000.90	1083.80	862.00	1005.10	816.02	903.14	1026.80	1131.90

6.4.1. *Color denoising*

The color attribute of each point cloud is corrupted with Gaussian noise applied to each point in a point cloud with $\sigma = 20, 30$, and 40. The MSE and PSNR comparisons between the proposed color denoising algorithm using Tikhonov and TV regularization are shown in Tab. 2 and Tab. 3. The results of both metrics show that the proposed technique via Tikhonov regularization performed far better than the TV regularization. The average gain in MSE and PSNR is shown in Fig. 20 for three different noise levels. It can be seen that with the increase in the intensity of noise level, the proposed color denoising method using Tikhonov performs much better than the TV regularization.

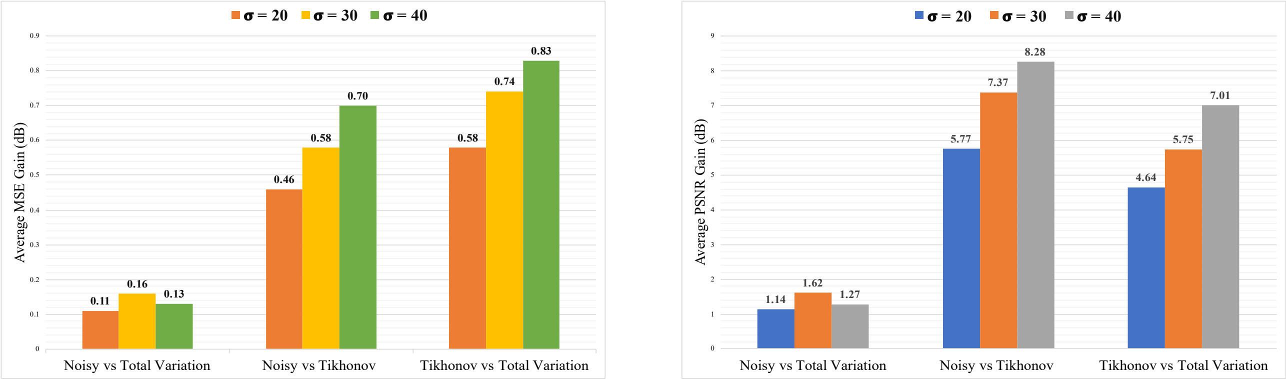
Gaussian noise with zero-mean and $\sigma = 10, 15, 20$, and 25 is added to the color attribute of the noise-free point cloud models of *Greyc* dataset for the comparison of the proposed color denoising algorithm with GLR [28, 60] and GTV [59]. Quantitative results in terms of PSNR are shown in Tab. 4, where GTV and GLR show the highest average PSNR value for noise level $\sigma = 10$ and $\sigma = 15$, respectively. With the increase in the noise level, the proposed algorithm performs better than GLR and GTV, with an

Table 3: PSNR comparison of color denoising algorithm 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
$\sigma = 20$	Noisy	22.13	22.55	22.41	22.18	22.82	22.14	22.50	22.30	22.47	22.25	23.07	22.39	23.23	22.88	22.36	22.13
	Proposed	**29.22**	**27.98**	**26.53**	**28.95**	**26.97**	**30.13**	**28.50**	**26.38**	**28.80**	**29.19**	**28.33**	**28.61**	**26.43**	**28.41**	**28.96**	**28.84**
	TV	23.32	23.67	23.52	23.39	23.57	23.34	23.67	23.36	23.67	23.45	24.29	23.61	24.31	24.07	23.46	23.32
$\sigma = 30$	Noisy	18.74	19.20	19.01	18.75	19.55	18.71	19.12	18.84	19.11	18.83	19.73	19.04	20.03	19.60	19.04	18.67
	Proposed	**27.72**	**26.41**	**24.81**	**27.31**	**25.35**	**28.63**	**26.72**	**24.76**	**26.98**	**27.63**	**26.15**	**26.78**	**24.14**	**26.33**	**26.87**	**27.23**
	TV	20.42	20.81	20.61	20.43	20.74	20.39	20.79	20.37	20.81	20.53	21.43	20.78	21.60	21.27	20.64	20.33
$\sigma= 40$	Noisy	16.35	16.90	16.72	16.48	17.19	16.46	16.84	16.48	16.81	16.48	17.44	16.75	17.75	17.28	16.74	16.31
	Proposed	**26.36**	**24.90**	**23.63**	**26.38**	**23.95**	**26.86**	**25.47**	**23.66**	**25.68**	**26.53**	**24.54**	**25.45**	**22.27**	**24.83**	**24.29**	**26.25**
	TV	17.64	18.15	17.97	17.79	18.22	17.76	18.13	17.69	18.13	17.78	18.78	18.11	19.01	18.57	18.01	17.59

Table 4: Color denoising comparison for Gaussian noise $\sigma = 10, 15, 20,$ and 25 with GLR-based [28] and GTV-based [59] in terms of PSNR and AET (s)

Model	$\sigma = 10$				$\sigma = 15$				$\sigma = 20$				$\sigma = 25$				AET (s)		
	Noise	Proposed	GLR	GTV	Noise	Proposed	GLR	GTV	Noise	Proposed	GLR	GTV	Noise	Proposed	GLR	GTV	Proposed	GLR	GTV
Asterix	28.38	31.32	**32.03**	31.61	24.94	29.21	**29.66**	29.53	22.53	27.98	**28.10**	27.56	20.68	26.69	26.12	**27.05**	**2.55**	5.20	211.00
Duck	28.53	**30.77**	30.40	30.60	25.20	**28.63**	28.42	28.14	22.71	**26.97**	26.89	26.35	20.90	**25.81**	25.68	25.04	**0.792**	1.70	58.00
Green_Dinosaur	28.14	33.17	33.28	**33.36**	24.60	31.28	**31.64**	31.31	22.13	30.13	30.30	**30.34**	20.23	29.23	28.52	**29.62**	**3.99**	11.20	389.00
Red_Horse	28.29	**32.43**	32.15	32.20	24.74	**30.17**	29.72	30.01	22.35	**28.96**	27.85	28.57	20.47	**27.38**	25.66	27.33	**8.13**	19.50	851.00
Average	28.30	31.92	31.87	**31.94**	24.85	29.82	**29.86**	29.75	22.44	**28.51**	28.29	28.20	20.61	**27.27**	26.495	27.26	**3.86**	9.40	377.25



(a) (b)

Fig. 20: (a): Average gain in MSE (dB) for the color denoising algorithm (b): Average gain in PSNR (dB) for the color denoising algorithm.

average PSNR increased by 0.25dB and 0.09dB, respectively. Besides PSNR, the average execution time (AET) per noise level for the proposed, GLR and GTV algorithms is also shown in Tab. 4. The proposed method is faster because it denoises the color using smaller γ, which in turn performs regularization in a smaller number of iterations. AET has been measured in MATLAB 2017b on a 3.5Ghz MacBook Pro with Intel Core i7 processor and 16GB memory.

6.4.2. Geometry denoising

Each point cloud has been corrupted with uniform synthetic geometry noise, applied to 50% of the points with $\sigma = 0.3, 0.4,$ and 0.5. The comparison between the proposed algorithm and the denoising approach used in [29] and MSGW [42] is shown in Tab. 7. The results show that the proposed denoising technique performs better than [29] and [42] in terms of all the metrics in the cloud to mesh distance at all noise levels. For $\sigma = 0.3$, the *Green monster, Mario, Pokeman ball, and Statue* show good results for the geometry-only algorithm [29]. Still, with the increase in noise intensity, the proposed algorithm performs better for all the models in the *Greyc* mesh dataset. The gain is more significant as the noise level increases, showing that the proposed denoising method is indeed better at removing geometry noise.

To check the robustness of the proposed algorithm to other noise distribution, Gaussian noise is added to each point cloud of the *Greyc* dataset with $\sigma = 0.2, 0.3,$ and 0.4. The comparative results are shown in Tab. 8, proving that the proposed algorithm yields good results with respect to the denoising techniques described in [29] and [42] at all noise levels for C2M distance. At noise level $\sigma = 0.2$, the geometry-only [29] algorithm performs better for *Asterix*, *Pokeman ball*, and *Rabbit* models; however, only the *duck* shows good results for MSGW [42] for all the noise intensities. The proposed denoising algorithm performs better for all the models except *Duck* as the noise level increases.

Tab. 5 and Tab. 6 show the MSE and MCD comparisons between the proposed algorithm and RPSM [43] on sub-sampled point clouds. The results clearly indicate that the proposed algorithm performs better than RPSM [43].

Table 5: MSE comparison on sub-sampled *Greyc* dataset.

Model	$\sigma = 0.3$		$\sigma = 0.4$		$\sigma = 0.5$	
	Proposed	RPSM [43]	Proposed	RPSM [43]	Proposed	RPSM [43]
4arms Monstre	**0.29**	0.48	**0.34**	0.51	**0.42**	0.52
Asterix	**0.26**	0.31	**0.28**	0.34	**0.30**	0.37
Cable car	**0.36**	0.74	**0.38**	0.80	**0.40**	0.82
Dragon	**0.25**	0.44	**0.26**	0.48	**0.28**	0.50
Green monster	**0.33**	0.62	**0.36**	0.65	**0.37**	0.67
Rabbit	**0.31**	0.58	**0.32**	0.60	**0.35**	0.63
Red horse	**0.34**	0.51	**0.36**	0.55	**0.39**	0.58

Table 6: MCD comparison on sub-sampled *Greyc* dataset.

Model	$\sigma = 0.3$		$\sigma = 0.4$		$\sigma = 0.5$	
	Proposed	RPSM [43]	Proposed	RPSM [43]	Proposed	RPSM [43]
4arms Monstre	**0.43**	0.70	**0.57**	0.90	**0.62**	1.00
Asterix	**0.38**	0.45	**0.41**	0.48	**0.44**	0.53
Cable car	**0.51**	0.92	**0.54**	0.98	**0.57**	1.05
Dragon	**0.37**	0.65	**0.37**	0.68	**0.42**	0.70
Green monster	**0.49**	0.71	**0.52**	0.77	**0.55**	0.80
Rabbit	**0.46**	0.80	**0.49**	0.82	**0.51**	0.88
Red horse	**0.50**	0.75	**0.54**	0.78	**0.57**	0.81

6.4.3. Combined geometry and color denoising

The geometry and color of each reference point cloud have been altered with uniform synthetic geometry noise and Gaussian color noise, respectively. The geometry of 50% of the points in each point cloud is affected with noise level $\sigma = 0.3, 0.4$, and 0.5, however, the color of each point in a point cloud is affected with $\sigma = 20, 30$, and 40. The C2M metric with respect to the corresponding reference point cloud has been computed for the denoised point cloud by the proposed algorithm using Tikhonov regularization and using TV. The comparative quantitative results of geometry denoising in a combined geometry and color denoising algorithm is shown in Tab. 9. For a better understanding of the results, we mapped the C2M distance on the reference point cloud. The distances are represented by the color scale; blue, green, yellow, and red display the range of distances from minimum to maximum. Fig. 21 shows the results of C2M distance for the *Asterix_model*. The performance evaluation of the proposed algorithm for color denoising has been done by computing the MSE and PSNR. The results of MSE and PSNR comparison with different noise levels are shown in Tab. 10 and Tab. 11, respectively. Fig. 22 shows the average gain in MSE and PSNR, which further verifies the better performance of the proposed algorithm. The results show that the proposed technique using Tikhonov regularization performs significantly better than the TV for noise levels $\sigma = 0.3, 0.4$, and 0.5 except for the *Duck*. The results also indicate that the proposed algorithm is robust to the increase in noise intensity.

Table 7: C2M metric comparison of the proposed geometry denoising algorithm with the geometry-only graph approach [29] and MSGW [42] for uniform noise.

Noise level in geometry	Methods	Parameters	4arms monstre	Asterix	Cable car	Dragon	Duck	Green Dinasour	Green monster	Horse	Jaguar	Long Diansour	Mario	Mario car	Pokeman ball	Rabbit	Red horse	Statue
$\sigma = 0.3$	Proposed	$\boldsymbol{d_H}$	**0.96**	0.97	**1.17**	**1.29**	**1.56**	**1.50**	1.18	**0.94**	**1.17**	1.45	1.18	**0.98**	1.01	**1.32**	**1.27**	0.90
		$\boldsymbol{d_m}$	**< 0.01**	**0.01**	**0.03**	**0.01**	**0.02**	**≈ 0.00**	0.02	**< 0.01**	**0.03**	**0.01**	0.04	**0.03**	0.03	**< 0.01**	**< 0.01**	0.01
		ζ	**0.07**	**0.07**	**0.16**	**0.11**	**0.18**	**0.04**	0.10	**0.06**	**0.14**	**0.13**	0.15	**0.14**	0.11	**0.06**	**0.10**	0.09
	Geometry-only [29]	$\boldsymbol{d_H}$	0.96	**0.75**	1.17	1.29	1.56	1.50	**0.75**	0.94	1.17	**1.02**	**1.02**	**0.98**	**0.64**	1.33	1.28	**0.90**
		$\boldsymbol{d_m}$	0.01	0.02	0.04	0.02	0.02	< 0.01	**0.02**	< 0.01	0.03	0.02	**0.04**	**0.03**	**0.03**	< 0.01	0.02	**0.01**
		ζ	0.08	0.08	0.18	0.14	0.19	0.08	**0.10**	0.08	0.15	0.15	**0.15**	**0.14**	**0.11**	0.08	0.14	**0.08**
	MSGW [42]	$\boldsymbol{d_H}$	1.10	0.98	1.31	1.55	1.97	1.86	1.25	1.31	1.29	1.20	1.11	1.34	0.99	1.59	1.65	1.50
		$\boldsymbol{d_m}$	0.02	0.10	0.09	0.08	0.11	0.01	0.08	0.01	0.07	0.05	0.07	0.18	0.07	0.01	0.04	0.01
		ζ	0.07	0.13	0.27	0.22	0.35	0.13	0.17	0.10	0.20	0.21	0.19	0.26	0.15	0.10	0.18	0.10
$\sigma = 0.4$	Proposed	$\boldsymbol{d_H}$	**0.96**	**0.97**	**1.17**	**1.29**	**1.56**	**1.50**	**1.18**	**0.94**	**1.17**	**1.45**	**1.18**	**0.98**	**1.01**	**1.32**	**1.27**	**0.90**
		$\boldsymbol{d_m}$	**0.01**	**0.02**	**0.05**	**0.03**	**0.03**	**< 0.01**	**0.04**	**0.01**	**0.05**	**0.03**	**0.07**	**0.05**	**0.03**	**0.01**	**0.02**	**0.02**
		ζ	**0.09**	**0.11**	**0.21**	**0.17**	**0.20**	**0.09**	**0.15**	**0.10**	**0.20**	**0.19**	**0.20**	**0.19**	**0.13**	**0.11**	**0.17**	**0.12**
	Geometry Only [29]	$\boldsymbol{d_H}$	1.18	0.97	1.17	1.29	1.57	2.12	**1.18**	1.32	1.18	1.45	1.32	1.20	1.01	1.32	1.80	1.27
		$\boldsymbol{d_m}$	0.02	0.03	0.06	0.04	0.03	0.01	**0.04**	0.01	0.06	0.04	0.08	0.06	0.04	0.02	0.03	0.02
		ζ	0.14	0.11	0.22	0.19	0.21	0.13	**0.15**	0.11	0.21	0.20	0.21	0.20	0.14	0.12	0.20	0.12
	MSGW [42]	$\boldsymbol{d_H}$	1.26	1.25	1.48	1.82	2.13	2.12	1.91	1.87	1.56	1.76	1.68785	1.86	1.35	1.86	1.87	1.79
		$\boldsymbol{d_m}$	0.04	0.18	0.14	0.13	0.18	0.02	0.13	0.01	0.14	0.10	0.15	0.27	0.09	0.02	0.06	0.02
		ζ	0.17	0.23	0.31	0.32	0.50	0.19	0.23	0.11	0.31	0.30	0.29	0.32	0.19	0.15	0.28	0.13
$\sigma = 0.5$	Proposed	$\boldsymbol{d_H}$	**1.37**	**1.07**	**1.43**	**1.30**	**1.59**	**1.51**	**1.19**	**1.33**	**1.18**	**1.46**	**1.32**	**1.55**	**1.11**	**1.32**	**1.80**	**1.28**
		$\boldsymbol{d_m}$	**0.02**	**0.04**	**0.09**	**0.06**	**0.03**	**0.01**	**0.07**	**0.02**	**0.08**	**0.05**	**0.11**	**0.08**	**0.05**	**0.02**	**0.05**	**0.03**
		ζ	**0.14**	**0.14**	**0.26**	**0.23**	**0.21**	**0.14**	**0.18**	**0.15**	**0.25**	**0.22**	**0.23**	**0.23**	**0.15**	**0.16**	**0.24**	**0.16**
	Geometry Only [29]	$\boldsymbol{d_H}$	1.53	1.31	1.44	1.32	1.63	2.13	1.34	1.35	1.44	1.47	1.45	1.56	1.12	1.33	1.89	1.28
		$\boldsymbol{d_m}$	0.03	0.05	0.12	0.07	0.05	0.02	0.08	0.03	0.09	0.07	0.12	0.10	0.06	0.03	0.07	0.04
		ζ	0.14	0.15	0.26	0.24	0.27	0.16	0.20	0.19	0.26	0.25	0.25	0.25	0.16	0.16	0.245	0.17
	MSGW [42]	$\boldsymbol{d_H}$	2.05	1.77	1.75	1.98	2.45	2.50	1.99	1.89	1.86	1.77	1.96	1.87	1.63	2.09	1.90	1.48
		$\boldsymbol{d_m}$	0.04	0.18	0.14	0.14	0.18	0.03	0.15	0.04	0.16	0.12	0.17	0.30	0.09	0.03	0.10	0.03
		ζ	0.18	0.24	0.32	0.33	0.51	0.20	0.26	0.24	0.34	0.51	0.32	0.34	0.19	0.17	0.29	0.20

7. Conclusion

In this paper, we proposed a novel and efficient framework for point cloud denoising based on graph signal processing. Unlike any other previous method, the proposed approach takes advantage of the correlation between the geometry and color attribute of a point cloud. Such correlation is encoded in a k-NN graph, which can be used to denoise only the color, only the geometry, or both of them jointly, by simply adapting the parameters to each denoising scenario. Denoising is performed as a convex optimization problem on a suitably defined graph signal.

We have provided a large set of experimental results using real as well as synthetic point clouds, which support the conclusion that the joint use of color and geometry is beneficial in all three denoising scenarios, providing denoised point clouds having higher subjective and objective quality and being devoid of artifacts.

For the subjective evaluation, we showed results on both real-world and synthetic point clouds for color denoising, geometry denoising, and combined geometry and color denoising procedure. The proposed color denoising algorithm has performed significantly well in both cases. The qualitative results for geometry denoising by the proposed algorithm are also very good, avoiding the artifacts typically caused by other techniques. The last set of results is associated with the combined geometry and color denoising; the proposed algorithm performs very well using Tikhonov regularization. The TV performs similarly well in denoising the geometry in the real-world point cloud, but not quite as well on color denoising. The overall results clearly show that Tikhonov regularization outperforms TV.

We also performed an extensive quantitative analysis using multiple datasets, evaluating the performance of the proposed algorithm for color denoising of point clouds using MSE and PSNR metrics. For geometry denoising, we computed the C2M metric between the reference and denoised point cloud. Both the subjective and objective results show that the proposed techniques

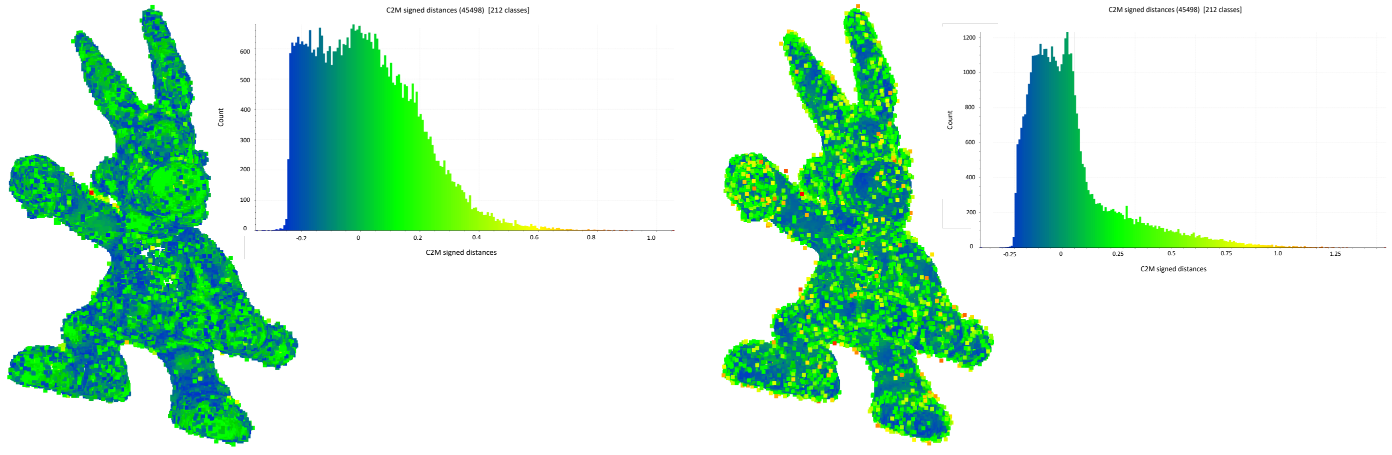
Table 8: C2M metric comparison of the proposed geometry denoising algorithm with the geometry-only graph approach [29] and MSGW [42] for Gaussian noise.

Noise level in geometry	Methods	Parameters	4arms monstre	Asterix	Cable car	Dragon	Duck	Green Dinasour	Green monster	Horse	Jaguar	Long Diansour	Mario	Mario car	Pokeman ball	Rabbit	Red horse	Statue
$\sigma = 0.2$	Proposed	d_H	**0.68**	0.61	**1.16**	**0.91**	1.56	**1.49**	**1.06**	**0.94**	**1.17**	**1.02**	**0.98**	**1.18**	0.64	0.93	**1.26**	**0.90**
		d_m	< **0.01**	0.01	**0.02**	**0.01**	0.02	< **0.01**	**0.01**	< **0.01**	**0.01**	**0.01**	**0.02**	**0.02**	0.03	< 0.01	< **0.01**	< **0.01**
		ζ	**0.04**	0.06	**0.12**	**0.08**	0.19	**0.02**	**0.07**	**0.03**	**0.11**	**0.09**	**0.10**	**0.12**	0.11	0.05	**0.07**	**0.05**
	Geometry Only [29]	d_H	0.78	**0.61**	1.27	0.98	1.56	1.71	1.17	0.94	1.38	1.23	0.98	1.07	**0.45**	**0.93**	1.47	1.09
		d_m	0.01	**0.01**	0.04	0.02	0.02	< 0.01	0.01	< 0.01	0.02	0.01	0.02	0.03	**0.03**	< **0.01**	0.01	< 0.01
		ζ	0.05	**0.05**	0.16	0.12	0.19	0.05	0.09	0.53	0.12	0.13	0.12	0.14	**0.11**	**0.04**	0.12	0.07
	MSGW [42]	d_H	1.37	0.97	1.18	0.92	**1.56**	1.51	1.07	0.95	1.41	1.02	0.99	1.18	0.91	0.94	1.81	1.28
		d_m	0.02	0.02	0.03	0.02	**0.02**	< 0.01	0.02	0.01	0.03	0.02	0.03	0.05	0.04	0.02	0.01	0.02
		ζ	0.11	0.09	0.17	0.12	**0.16**	0.07	0.11	0.11	0.15	0.12	0.15	0.17	0.12	0.12	0.13	0.12
$\sigma = 0.3$	Proposed	d_H	**0.96**	**1.05**	**1.17**	**1.28**	2.21	**1.50**	**1.18**	**1.30**	**1.17**	**1.03**	**1.38**	**1.19**	0.90	**1.32**	**1.27**	**0.90**
		d_m	**0.01**	**0.02**	**0.04**	**0.03**	0.05	< **0.01**	< **0.01**	< **0.01**	**0.03**	**0.02**	**0.04**	**0.06**	0.04	**0.01**	**0.02**	**0.01**
		ζ	**0.08**	**0.09**	**0.19**	**0.15**	0.27	**0.08**	**0.12**	**0.07**	**0.17**	**0.16**	**0.17**	**0.18**	0.13	**0.08**	**0.14**	**0.08**
	Geometry Only [29]	d_H	0.96	1.06	1.66	1.29	1.57	1.96	1.60	1.32	1.66	1.65	1.58	1.33	**0.87**	1.32	1.68	1.28
		d_m	0.01	0.04	0.07	0.04	0.05	0.01	0.04	0.01	0.05	0.03	0.06	0.09	**0.03**	0.01	0.02	0.01
		ζ	0.10	0.10	0.21	0.17	0.24	0.11	0.13	0.08	0.19	0.16	0.18	0.20	**0.12**	0.09	0.17	0.10
	MSGW [42]	d_H	1.53	1.31	1.67	1.83	**1.56**	2.13	1.20	1.88	1.86	1.50	1.56	1.34	1.30	1.88	1.98	1.82
		d_m	0.03	0.03	0.05	0.035	**0.03**	0.01	0.05	0.02	0.06	0.03	0.07	0.10	0.05	0.03	0.02	0.03
		ζ	0.14	0.12	0.23	0.16	**0.23**	0.09	0.16	0.14	0.22	0.18	0.22	0.23	0.14	0.16	0.16	0.15
$\sigma = 0.4$	Proposed	d_H	**1.13**	**1.48**	**1.34**	**1.53**	2.41	**1.83**	**1.37**	**1.58**	**1.29**	**1.19**	**1.61**	**1.33**	**0.98**	**1.46**	**1.35**	**1.02**
		d_m	**0.02**	**0.05**	**0.07**	**0.05**	0.07	**0.01**	**0.05**	**0.02**	**0.05**	**0.03**	**0.06**	**0.07**	**0.05**	**0.01**	**0.03**	**0.01**
		ζ	**0.12**	**0.15**	**0.22**	**0.19**	0.38	**0.11**	**0.14**	**0.13**	**0.23**	**0.18**	**0.21**	**0.20**	**0.17**	**0.11**	**0.16**	**0.13**
	Geometry Only [29]	d_H	1.24	1.55	1.89	1.56	2.04	1.94	1.75	1.59	1.92	1.84	1.79	1.50	0.99	1.48	1.73	1.39
		d_m	0.03	0.06	0.10	0.06	0.08	0.02	0.07	0.03	0.08	0.04	0.08	0.11	0.06	0.02	0.04	0.03
		ζ	0.15	0.18	0.26	0.23	0.31	0.15	0.16	0.15	0.27	0.19	0.29	0.24	0.17	0.13	0.22	0.18
	MSGW [42]	d_H	1.66	1.77	1.94	2.01	**1.97**	2.39	1.46	1.84	2.17	1.76	1.83	1.51	1.47	2.07	2.03	1.97
		d_m	0.05	0.06	0.09	0.06	**0.05**	0.01	0.08	0.05	0.09	0.05	0.11	0.13	0.07	0.06	0.04	0.04
		ζ	0.19	0.21	0.27	0.21	**0.29**	0.13	0.19	0.20	0.30	0.21	0.32	0.28	0.19	0.22	0.19	0.26

Table 9: C2M metric comparison between the proposed combined geometry and color denoising algorithm using Tikhonov regularization and TV.

Noise level in geometry	Methods	Parameters	4arms monstre	Asterix	Cable car	Dragon	Duck	Green Dinasour	Green monster	Horse	Jaguar	Long Diansour	Mario	Mario car	Pokeman ball	Rabbit	Red horse	Statue
$\sigma = 0.3$	Proposed algorithm using Tikhonov regularization	d_H	**0.68**	**0.61**	1.17	0.91	2.71	**1.50**	**0.52**	**0.94**	**0.82**	**1.02**	**0.82**	**0.69**	**0.45**	**0.93**	**1.27**	**0.89**
		d_m	< **0.01**	< **0.01**	0.05	0.03	0.37	≈ **0.00**	< **0.01**	≈ **0.00**	**0.01**	**0.01**	**0.02**	**0.01**	**0.03**	≈ **0.00**	**0.01**	< **0.01**
		ζ	**0.04**	**0.06**	0.19	0.15	0.68	**0.03**	**0.07**	**0.02**	**0.10**	**0.10**	**0.11**	**0.10**	**0.11**	**0.03**	**0.11**	**0.04**
	TV regularization	d_H	0.96	0.61	**0.82**	**0.91**	**1.56**	1.50	0.75	0.94	1.17	1.02	0.83	0.97	0.64	0.93	1.27	0.90
		d_m	0.01	0.01	**0.04**	**0.02**	**0.07**	< 0.01	0.02	< 0.01	0.03	0.02	0.05	0.03	0.03	< 0.01	0.01	0.01
		ζ	0.08	0.07	**0.19**	**0.14**	**0.31**	0.05	0.11	0.06	0.15	0.14	0.16	0.15	0.12	0.08	0.13	0.09
$\sigma = 0.4$	Proposed algorithm using Tikhonov regularization	d_H	**0.68**	**0.74**	**1.16**	**0.91**	2.71	**1.50**	**0.74**	**0.98**	**0.83**	**1.02**	**1.02**	**0.98**	**0.63**	**0.93**	**1.27**	**0.89**
		d_m	< **0.01**	**0.01**	**0.06**	**0.03**	0.37	< **0.01**	**0.02**	< **0.01**	**0.02**	**0.02**	**0.04**	**0.03**	**0.03**	< **0.01**	**0.02**	< **0.01**
		ζ	**0.05**	**0.08**	**0.21**	**0.17**	0.68	**0.05**	**0.10**	**0.04**	**0.14**	**0.13**	**0.15**	**0.13**	**0.11**	**0.05**	**0.14**	**0.06**
	TV regularization	d_H	0.96	0.87	1.43	1.29	**1.61**	1.54	1.06	1.08	1.18	1.02	1.18	0.98	1.01	1.32	1.80	0.90
		d_m	0.02	0.02	0.06	0.04	**0.07**	0.01	0.05	0.01	0.05	0.04	0.08	0.06	0.05	0.02	0.03	0.02
		ζ	0.12	0.10	0.22	0.19	**0.33**	0.09	0.16	0.10	0.20	0.19	0.21	0.20	0.14	0.12	0.19	0.13
$\sigma = 0.5$	Proposed algorithm using Tikhonov regularization	d_H	**0.96**	**0.96**	**1.43**	**0.94**	3.50	**1.53**	**1.05**	**0.99**	**1.17**	**1.02**	**1.02**	**0.98**	**0.89**	**0.93**	**1.29**	**0.91**
		d_m	< **0.01**	**0.03**	**0.09**	**0.06**	0.61	< **0.01**	**0.04**	< **0.01**	**0.04**	**0.03**	**0.06**	**0.04**	**0.04**	< **0.01**	**0.03**	< **0.01**
		ζ	**0.08**	**0.12**	**0.26**	**0.23**	0.83	**0.07**	**0.13**	**0.05**	**0.18**	**0.17**	**0.18**	**0.16**	**0.12**	**0.06**	**0.19**	**0.06**
	TV regularization	d_H	1.37	1.07	1.65	1.30	**1.69**	1.50	1.30	1.33	1.18	1.45	1.45	1.56	1.11	1.32	1.81	1.27
		d_m	0.03	0.04	0.10	0.07	**0.08**	0.01	0.08	0.02	0.09	0.06	0.11	0.09	0.06	0.03	0.05	0.03
		ζ	0.14	0.14	0.26	0.26	**0.34**	0.13	0.20	0.14	0.26	0.25	0.24	0.24	0.16	0.16	0.25	0.16

perform very well for point cloud denoising, outperforming state-of-the-art techniques.



(a) (b)

Fig. 21: Asterix_model (noise level $\mu = 0$ and $\sigma = 0.5$) (a): C2M metric of denoised point cloud by proposed algorithm using Tikhonov regularization (b): C2M metric of denoised point cloud using TV.

Table 10: MSE comparison of combined geometry and color denoising algorithm for *Greyc* dataset with three noise levels in color attribute.

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
$\sigma = 20$	Noisy	396.81	361.15	374.29	393.14	339.42	397.91	364.35	385.17	368.52	386.45	321.67	376.17	309.85	333.85	378.42	396.90
	Tikhonov	**107.03**	**164.03**	**172.73**	**116.33**	564.41	**97.96**	**125.90**	**173.33**	**127.41**	**105.68**	**130.65**	**121.52**	**269.01**	**119.79**	**219.17**	**119.52**
	TV	360.47	333.08	344.70	358.37	**320.00**	361.15	334.29	354.34	338.20	352.53	294.67	344.65	285.11	303.66	350.36	361.43
$\sigma = 30$	Noisy	876.66	782.13	810.77	863.43	728.82	872.76	793.86	841.43	801.55	848.09	689.96	813.16	652.34	716.01	814.97	882.11
	Tikhonov	**199.35**	**269.90**	**275.74**	**202.90**	**647.76**	**174.94**	**235.39**	**267.61**	**242.29**	**200.79**	**252.53**	**251.17**	**385.45**	**216.69**	**316.62**	**215.98**
	TV	820.72	737.84	764.08	809.03	693.63	815.83	748.37	792.41	755.33	795.76	649.93	766.46	613.50	670.00	769.39	826.59
$\sigma = 40$	Noisy	1505.90	1337.20	1384.60	1459.90	1243.80	1476.30	1353.50	1467.20	1357.00	1458.90	1167.20	1368.70	1088.30	1209.70	1380.00	1523.40
	Tikhonov	**241.45**	**386.44**	**395.22**	**246.54**	**884.03**	**202.77**	**321.99**	**375.98**	**340.35**	**254.89**	**372.51**	**350.60**	**658.49**	**308.36**	**445.18**	**275.12**
	TV	1390.60	1244.00	1286.80	1347.20	1162.90	1358.90	1258.70	1362.30	1262.80	1350.50	1087.20	1275.20	1009.70	1116.70	1283.10	1408.20

Table 11: PSNR comparison of combined geometry and color denoising algorithm for *Greyc* dataset with three noise levels in color attribute.

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
$\sigma = 20$	Noisy	22.15	22.55	22.40	22.19	22.82	22.13	22.52	22.27	22.47	22.26	23.06	22.38	23.22	22.90	22.35	22.14
	Tikhonov	**27.84**	**25.99**	**25.76**	**27.47**	20.61	**28.22**	**27.13**	**25.74**	**27.08**	**27.89**	**26.97**	**27.28**	**23.83**	**27.35**	**24.72**	**27.36**
	TV	22.56	22.91	22.76	22.59	**23.08**	22.55	22.89	22.64	22.84	22.66	23.44	22.76	23.58	23.31	22.69	22.55
$\sigma = 30$	Noisy	18.70	19.20	19.04	18.77	19.50	18.72	19.13	18.88	19.09	18.85	19.74	19.03	19.99	19.58	19.02	18.68
	Tikhonov	**25.13**	**23.82**	**23.73**	**25.06**	**20.02**	**25.70**	**24.41**	**23.86**	**24.29**	**25.10**	**24.11**	**24.13**	**22.27**	**24.77**	**23.13**	**24.79**
	TV	18.99	19.45	19.30	19.05	19.72	19.01	19.39	19.14	19.35	19.12	20.00	19.29	20.25	19.87	19.27	18.96
$\sigma= 40$	Noisy	16.35	16.87	16.72	16.49	17.18	16.44	16.82	16.47	16.81	16.49	17.46	16.77	17.76	17.30	16.73	16.30
	Tikhonov	**24.30**	**22.26**	**22.16**	**24.21**	**18.67**	**25.06**	**23.05**	**22.38**	**22.37**	**22.81**	**24.07**	**22.42**	**19.95**	**23.24**	**21.65**	**23.74**
	TV	16.70	17.18	17.046	16.84	17.48	16.80	17.13	16.79	17.12	16.83	17.77	17.08	18.09	17.65	17.05	16.64

References

[1] R. B. Rusu, 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, P. Frossard, Graph-based compression of dynamic 3D point cloud sequences, IEEE Transactions on Image Processing 25 (2016) 1765–1778.

[3] S. Chen, D. Tian, C. Feng, A. Vetro, J. Kovačević, Fast resampling of three-dimensional point clouds via graphs, IEEE Transactions on Signal Processing 66 (2017) 666–681.

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

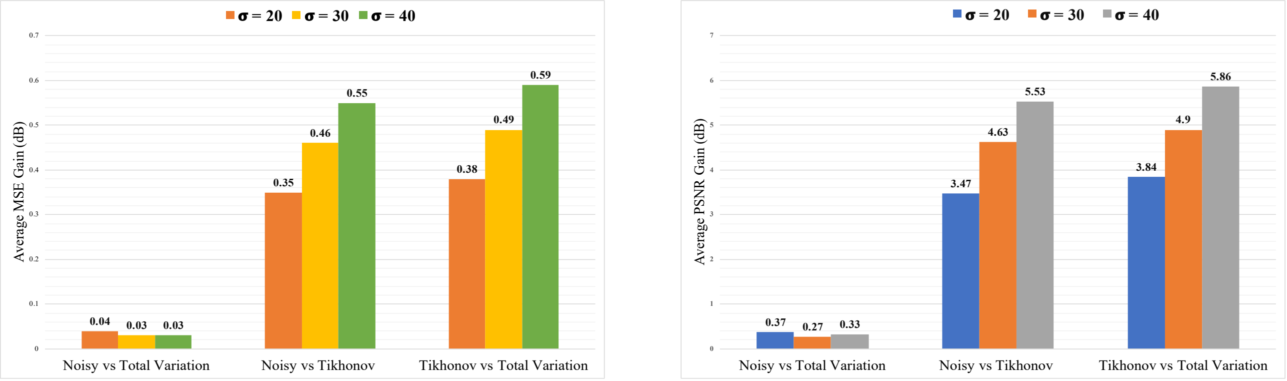
[5] M. Ji, J. Gall, H. Zheng, Y. Liu, L. Fang, Surfacenet: An end-to-end 3D neural network for multiview stereopsis, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2307–2315.

[6] G. Rosman, A. Dubrovina, R. Kimmel, Patch-collaborative spectral point-cloud denoising, in: Computer Graphics Forum, volume 32, Wiley Online Library, 2013, pp. 1–12.

[7] E. Mattei, A. Castrodad, Point cloud denoising via moving RPCA, in: Computer Graphics Forum, volume 36, Wiley Online Library, 2017, pp. 123–137.

[8] Y. Sun, S. Schaefer, W. Wang, Denoising point sets via L0 minimization, Computer Aided Geometric Design 35 (2015) 2–15.

[9] Y. Zheng, G. Li, S. Wu, Y. Liu, Y. Gao, Guided point cloud denoising via sharp feature skeletons, The Visual Computer 33 (2017) 857–867.



(a)

(b)

Fig. 22: (a): Average gain in MSE (dB) for the color denoising in combined geometry and color denoising algorithm (b): Average gain in PSNR (dB) for the color denoising in combined geometry and color denoising algorithm.

[10] G. Guennebaud, M. Gross, Algebraic point set surfaces, in: ACM Transactions on Graphics, volume 26, ACM, 2007, p. 23.

[11] A. C. Öztireli, G. Guennebaud, M. Gross, Feature preserving point set surfaces based on non-linear kernel regression, in: Computer Graphics Forum, volume 28, Wiley Online Library, 2009, pp. 493–501.

[12] Y. Lipman, D. Cohen-Or, D. Levin, H. Tal-Ezer, Parameterization-free projection for geometry reconstruction, in: ACM Transactions on Graphics, volume 26, ACM, 2007, p. 22.

[13] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, H. R. Zhang, Edge-aware point set resampling, ACM Transactions on Graphics 32 (2013) 9.

[14] H. Avron, A. Sharf, C. Greif, D. Cohen-Or, ℓ 1-sparse reconstruction of sharp point set surfaces, ACM Transactions on Graphics 29 (2010) 135.

[15] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao, L. Xiao, A review of algorithms for filtering the 3D point cloud, Signal Processing: Image Communication 57 (2017) 103–112.

[16] F. L. Siena, B. Byrom, P. Watts, P. Breedon, Utilising the Intel RealSense camera for measuring health outcomes in clinical research, Journal of Medical Systems 42 (2018) 53.

[17] X. Gong, M. Chen, X. Yang, Point cloud segmentation of 3D scattered parts sampled by RealSense, in: 2017 IEEE International Conference on Information and Automation (ICIA), IEEE, 2017, pp. 1–6.

[18] R. Das, K. S. Kumar, Gerosim: A simulation framework for gesture driven robotic arm control using Intel RealSense, in: 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), IEEE, 2016, pp. 1–5.

[19] Q. Li, Y. Wang, A. Sharf, Y. Cao, C. Tu, B. Chen, S. Yu, Classification of gait anomalies from Kinect, The Visual Computer 34 (2018) 229–241.

[20] K. Kamal, S. Mathavan, T. Zafar, I. Moazzam, A. Ali, S. U. Ahmad, M. Rahman, Performance assessment of Kinect as a sensor for pothole imaging and metrology, International Journal of Pavement Engineering 19 (2018) 565–576.

[21] D. J. Tan, F. Tombari, N. Navab, Real-time accurate 3D head tracking and pose estimation with consumer RGB-D cameras, International Journal of Computer Vision 126 (2018) 158–183.

[22] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, A. Kolb, State of the art on 3D reconstruction with RGB-D cameras, in: Computer Graphics Forum, volume 37, Wiley Online Library, 2018, pp. 625–652.

[23] F. Verdoja, D. Thomas, 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.

[24] Q. Zhan, Y. Liang, Y. Xiao, Color-based segmentation of point clouds, Laser Scanning 38 (2009) 155–161.

[25] M. C. Dal, P. Zanuttigh, G. M. Cortelazzo, Fusion of geometry and color information for scene segmentation, IEEE Journal of Selected Topics in Signal Processing 6 (2012) 505–521.

[26] Y.-J. Liu, Y.-F. Zheng, L. Lv, Y.-M. Xuan, X.-L. Fu, 3D model retrieval based on color + geometry signatures, The Visual Computer 28 (2012) 75–86.

[27] D. Slater, G. Healey, Combining color and geometric information for the illumination invariant recognition of 3-D objects, in: Proceedings of IEEE International Conference on Computer Vision, IEEE, 1995, pp. 563–568.

[28] C. Dinesh, G. Cheung, 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.

[29] Y. Schoenenberger, J. Paratte, 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.

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

[31] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, M. Beetz, Towards 3D point cloud based object maps for household environments, Robotics and Autonomous Systems 56 (2008) 927–941.

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

[33] W. Huang, Y. Li, P. Wen, X. Wu, Algorithm for 3D point cloud denoising, in: 2009 Third International Conference on Genetic and Evolutionary Computing, IEEE, 2009, pp. 574–577.

[34] F. Yan, Z. Jinlei, Research on scattered points cloud denoising algorithm, in: 2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), IEEE, 2015, pp. 1–5.

[35] D. Levin, Mesh-independent surface interpolation, in: Geometric Modeling for Scientific Visualization, Springer, 2004, pp. 37–49.

[36] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C. T. Silva, Computing and rendering point set surfaces, IEEE Transactions on Visualization and Computer Graphics 9 (2003) 3–15.

[37] G. Guennebaud, M. Germann, M. Gross, Dynamic sampling and rendering of algebraic point set surfaces, in: Computer Graphics Forum, volume 27, Wiley Online Library, 2008, pp. 653–662.

[38] R. B. Rusu, N. Blodow, Z. Marton, A. Soos, M. Beetz, Towards 3D object maps for autonomous household robots, in: 2007 IEEE/RSJ International

Conference on Intelligent Robots and Systems, IEEE, 2007, pp. 3191–3198.

[39] H. Huang, D. Li, H. Zhang, U. Ascher, D. Cohen-Or, Consolidation of unorganized point clouds for surface reconstruction, ACM Transactions on Graphics 28 (2009) 176.

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

[41] C. Dinesh, G. Cheung, I. V. Bajić, C. Yang, Local 3D point cloud denoising via bipartite graph approximation & total variation, in: 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP), IEEE, 2018, pp. 1–6.

[42] S. Deutsch, A. Ortega, 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.

[43] S. Deutsch, A. Ortega, G. Medioni, Robust denoising of piece-wise smooth manifolds, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 2786–2790.

[44] G. Cheung, E. Magli, Y. Tanaka, M. K. Ng, Graph spectral image processing, Proceedings of the IEEE 106 (2018) 907–930.

[45] F. R. Chung, F. C. Graham, Spectral graph theory, 92, American Mathematical Soc., 1997.

[46] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, 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 30 (2013) 83–98.

[47] 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 3 (2011) 1–122.

[48] P. Cignoni, C. Rocchini, R. Scopigno, Metro: measuring error on simplified surfaces, in: Computer Graphics Forum, volume 17, Wiley Online Library, 1998, pp. 167–174.

[49] N. Aspert, D. Santa-Cruz, T. Ebrahimi, Mesh: Measuring errors between surfaces using the Hausdorff distance, in: Proceedings. IEEE International Conference on Multimedia and Expo, volume 1, IEEE, 2002, pp. 705–708.

[50] D. Eberly, Distance between point and triangle in 3D, Magic Software, http://www. magic-software. com/Documentation/pt3tri3. pdf (1999).

[51] V. Oniga, C. Chirilă, Hausdorff distance for the differences calculation between 3D surfaces, Journal of Geodesy and Cadastre RevCAD 15 (2013) 193–202.

[52] D. Girardeau-Montaut, CloudCompare, 2016.

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

[54] J. Zeng, G. Cheung, M. Ng, J. Pang, C. Yang, 3D point cloud denoising using graph laplacian regularization of a low dimensional manifold model, IEEE Transactions on Image Processing (2019).

[55] L. Jun, L. Wei, D. Donglai, S. Qiang, Point cloud registration algorithm based on NDT with variable size voxel, in: 2015 34th Chinese Control Conference (CCC), IEEE, 2015, pp. 3707–3712.

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

[57] N. Perraudin, V. Kalofolias, D. Shuman, P. Vandergheynst, Unlocbox: A MATLAB convex optimization toolbox for proximal-splitting methods, arXiv preprint arXiv:1402.0779 (2014).

[58] K. Fliegel, F. Battisti, M. Carli, M. Gelautz, L. Krasula, P. Le Callet, V. Zlokolica, 3D visual content datasets, in: 3D Visual Content Creation, Coding and Delivery, Springer, 2019, pp. 299–325.

[59] C. Couprie, L. Grady, L. Najman, J.-C. Pesquet, H. Talbot, Dual constrained TV-based regularization on graphs, SIAM Journal on Imaging Sciences 6 (2013) 1246–1273.

[60] J. Pang, G. Cheung, Graph Laplacian regularization for image denoising: Analysis in the continuous domain, IEEE Transactions on Image Processing 26 (2017) 1770–1785.