Weighted Multi-projection: 3D Point Cloud Denoising with Estimated Tangent Planes

Chaojing Duan
Electrical & Computer Engineering
Carnegie Mellon University
Pittsburgh, USA
chaojind@andrew.cmu.edu

Siheng Chen
Uber ATG
Pittsburgh, USA
sihengc@uber.com

Jelena Kovačević
Electrical & Computer Engineering
Carnegie Mellon University
Pittsburgh, USA
jelenak@cmu.edu

Abstract—As a collection of 3D points sampled from surfaces of objects, a 3D point cloud is widely used in robotics, autonomous driving and augmented reality. Due to the physical limitations of 3D sensing devices, 3D point clouds are usually noisy, which influences subsequent computations, such as surface reconstruction, recognition and many others. To denoise a 3D point cloud, we present a novel algorithm, called weighted multi-projection. Compared to many previous works on denoising, instead of directly smoothing the coordinates of 3D points, we use a two-fold smoothing: We first estimate a local tangent plane at each 3D point and then reconstruct each 3D point by weighted averaging of its projections on multiple tangent planes. We also provide the theoretical analysis for the surface normal estimation and achieve a tighter bound than in a previous work. We validate the empirical performance on the dataset of ShapeNetCore and show that weighted multi-projection outperforms its competitors in all nine classes.

Index Terms—3D point cloud, manifold, graph, normal vector, projection

I. INTRODUCTION

With the rapid development of 3D sensing techniques and image-based 3D reconstruction techniques, 3D points are increasingly used in robotics, autonomous driving, augmented reality, computer-aided shape design and many other practical tasks involving objects or environment reconstruction [1]–[3].

3D point clouds are usually obtained from two approaches: 3D scanning devices and image-based 3D reconstruction [4], [5]. Both approaches introduce noise. For example, 3D scanners suffer from measurement noise, especially around edges or corners of objects; image-based 3D reconstruction often fails to manage matching ambiguities or image imperfection [6]. Noise not only makes the reconstructed surface lose fine details and deforms the underlying manifold structure but it also impairs subsequent geometry processing and computation, such as compression, segmentation, registration and recognition [7]. However, 3D point cloud denoising is challenging; 1D time series and 2D images support on grids, a 3D point cloud is usually a set of unordered points scattered in the 3D space without any connectivity or topology information.

To tackle the problem of 3D point cloud denoising, we propose a novel algorithm, called weighted multi-projection (WMP), which includes three modules, graph construction, tangent plane estimation, and reconstruction from multiple projections. We first construct a graph based on the 3D coordinates of points; this graph captures the local and global geometric structures of the underlying manifold. Based on the graph, we then estimate the local tangent planes for all the points, which approximates the surface from which the points are sampled. After that, a multi-projection procedure is followed by weighted averaging. The intuition behind this algorithm is to use the tangent planes to remove the orthogonal component of noise. We further show a theoretical bound for the estimation error of surface normals in the 2D case, which is proved to be a tighter bound

(a) Noise-free point cloud.

(b) Noisy point cloud.

(c) Bilateral filter.

(d) Graph-based filter.
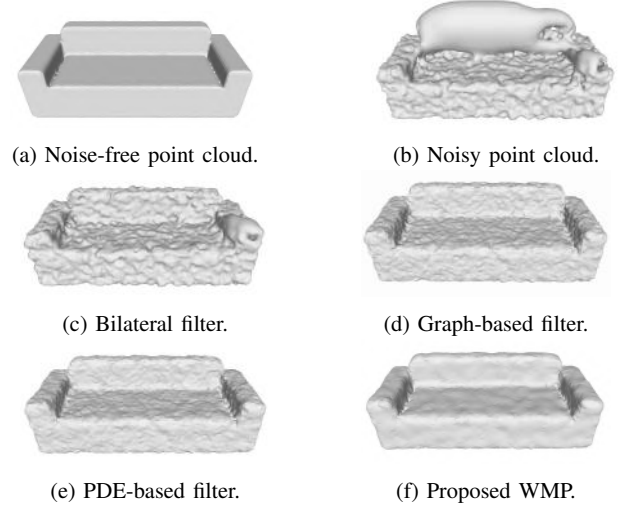
(e) PDE-based filter.

(f) Proposed WMP.

Fig. 1: The proposed WMP produces a cleaner and smoother mesh reconstruction compared to its competitors. Plots (a) and (b) show the mesh reconstructed from noiseless point cloud and noisy point cloud, respectively. Plots (c)-(f) show the meshes reconstructed from the denoised point cloud produced by different algorithms.

comparing with [8] and show that the multiple-projection strategy is better than the one-time projection strategy. To validate the empirical performance, we also test the proposed algorithm on the real dataset of ShapeNetCore. The experimental results show that our algorithm outperforms other algorithms in all nine categories.

Related Work. 3D point cloud denoising has been tackled by various approaches: mesh-based denoising, bilateral-filter-based denoising, partial-differential-equation-based denoising and graph-based denoising. Most mesh-based denoising algorithms use meshes constructed from 3D points as the input [9], [10]. Since mesh construction algorithms are time-consuming and are sensitive to noise [11], researchers have explored algorithms to directly deal with 3D point clouds. The authors in [11] adapt the bilateral filter (BF) from mesh denoising [12], [13] and implement it to denoise point clouds. However, mesh-based algorithms are not fully transferable and causes unavoidable shrinkage and deformation [14]. Partial differential equations (PDE) are widely used in image and mesh denoising [15]; they have been extended to 3D point cloud denoising [16], [17]; however, PDE-based denoising algorithm cause local over-smoothing [15]. Graph-based denoising (GBD) algorithms have received increasing attention because the Laplace-Beltrami operator of manifolds can be approximated by the graph Laplacian [18]; however, a constructed graph is also noisy and cannot reflect the

true manifold. In practice, graph signal denoising algorithms cause clustering and deformation issues [19].

Note that we also use the graph structure to capture the local geometric structure. Instead of using the graph to denoise directly, like those classical graph-based algorithms, we use the graph structure to estimate the tangent planes and use the estimated tangent planes to denoise. This two-fold smoothing process ease the clustering and deformation issues.

II. 3D POINT CLOUD DENOISING ALGORITHM

The proposed point cloud denoising algorithm consists of three modules: graph construction, tangent plane estimation and weighted projection. (1) We first construct a graph based on 3D coordinates of a point cloud. The graph is used to capture the geometric structure of the point cloud. (2) Based on this graph structure, we estimate a tangent plane at each point to locally approximate the underlying manifold. (3) We then project each point onto its own tangent plane and its neighboring points' tangent planes, called the multi-projection. We finally obtain the denoised points by weighted averaging of the coordinates of all the projections. The key component of the proposed algorithm is the tangent plane.

Problem Setting. Let $\mathcal{S} = \{\widetilde{\mathbf{p}}_i \in \mathbb{R}^3 \mid i = 1, ..., N\}$ be a noisy 3D point cloud, where N is the total number of points in $\mathcal{S}$, and each element $\widetilde{\mathbf{p}}_i = [x_i, y_i, z_i]^T$ is the noisy coordinate vector of the ith point. Note that $\widetilde{\mathbf{p}}_i = \mathbf{p}_i + \mathbf{n}_i$, where $\mathbf{p}_i$ is the noiseless coordinate vector of the ith point and $\mathbf{n}_i$ is the noise vector attached to the ith point. Since 3D point clouds are sampled from smooth surfaces, they are essentially 2D manifolds embedded in the 3D space. These smooth surfaces can be locally approximated by tangent planes. We thus use the tangent planes to remove noise.

Theorem 1: Let T be the tangent plane associated with the noiseless point $\mathbf{p}$; $\mathbf{t}$ be the projection of a noisy point $\widetilde{\mathbf{p}}$ on T. Then, $\|\mathbf{t} - \mathbf{p}\|_2^2 \leq \|\widetilde{\mathbf{p}} - \mathbf{p}\|_2^2$.

The proof is obtained from the projection theorem [20]. The projection on the tangent plane removes the orthogonal component of noise. Inspired by this, we aim to estimate tangent planes from noisy points and project noisy points on the estimated tangent planes to obtain denoised points.

Graph Construction. To capture the local and global geometric structure of a point cloud, we construct a graph $\mathcal{G}$ whose nodes are 3D points and edges are proximities of points. Here we build an ϵ-nearest neighbor (ϵ-NN) graph. The ith and jth points are connected by an edge when the Euclidean distance between $\widetilde{\mathbf{p}}_i$ and $\widetilde{\mathbf{p}}_j$ is smaller than the pre-defined ϵ. We then construct a weighted adjacent matrix $W \in \mathbb{R}^{N \times N}$, whose elements are defined as

$$W_{ij} = \begin{cases} \frac{1}{Z_i} \exp(-\frac{\|\widetilde{\mathbf{p}}_i - \widetilde{\mathbf{p}}_j\|_2^2}{2\sigma^2}) & \text{if } \|\widetilde{\mathbf{p}}_i - \widetilde{\mathbf{p}}_j\|_2^2 \leq \epsilon, \\ 0 & \text{otherwise}, \end{cases}$$

where the normalization term is $Z_i = \sum_{j=1}^N \exp(-\|\widetilde{\mathbf{p}}_i - \widetilde{\mathbf{p}}_j\|_2^2/(2\sigma^2))$, with hyperparameters σ and ϵ controlling the decay rate. Let $\mathcal{N}_i$ be the neighboring set for the ith point, containing all the neighboring points that connect to the ith point, $\mathcal{N}_i = \{j \mid W_{ij} \neq 0\}$.

Tangent Plane Estimation. Based on the constructed graph, we estimate the local tangent plane at each 3D point. Let $T_i = \{\mathbf{t} \in \mathbb{R}^3 \mid \mathbf{a}_i^T \mathbf{t} = c_i\}$ be the tangent plane at the ith point, where $\mathbf{a}_i \in \mathbb{R}^3$ is the normal vector, $c_i \in \mathbb{R}$ is the intercept of plane T_i and $\mathbf{t}$ is the point on the plane T_i. Following the work in [21], we can obtain $\mathbf{a}_i$ and c_i by minimizing the sum of weighted square distances from all the points in $\mathcal{N}_i$ to T_i,

$$\mathbf{a}_i , \ c_i = \arg\min_{\mathbf{a}, c} \sum_{j \in \mathcal{N}_i} W_{ij} (\mathbf{a}^T \widetilde{\mathbf{p}}_j - c)^2,$$

$$\text{s.t.} \quad \|\mathbf{a}_i\|_2^2 = 1.$$

We can show that the normal vector $\mathbf{a}_i$ is simply the normalized eigenvector corresponding to the smallest eigenvalue of the weighted covariance matrix $\widetilde{M}_i \in \mathbb{R}^{3 \times 3}$ defined as follows,

$$\widetilde{M}_i = \sum_{j \in \mathcal{N}_i} W_{ij} \widetilde{\mathbf{p}}_j \widetilde{\mathbf{p}}_j^T - \overline{\mathbf{p}}_i \overline{\mathbf{p}}_i^T, \tag{1}$$

with $\overline{\mathbf{p}}_i = \sum_{j \in \mathcal{N}_i} W_{ij} \widetilde{\mathbf{p}}_j$. Once we obtain the normal vector $\mathbf{a}_i$, the intercept is calculated as $c_i = \mathbf{a}_i^T \overline{\mathbf{p}}_i$.

Weighted Multi-Projection. As shown earlier, we can use the projection to remove noise. Let $\widetilde{T}_i$ be the estimated tangent plane associated with the ith point $\widetilde{\mathbf{p}}_i$ and $\mathbf{t}$ be the point on the plane. We define the denoised point $\widehat{\mathbf{p}}_i$ as follows:

$$\widehat{\mathbf{p}}_i = \arg\min_{\mathbf{t} \in \widetilde{T}_i} \|\widetilde{\mathbf{p}}_i - \mathbf{t}\|_2^2, \tag{2}$$

which is called the one-time projection. Based on the geometry, we know that $\widehat{\mathbf{p}}_i$ is the projection of point $\widetilde{\mathbf{p}}_i$ onto the estimated tangent plane $\widetilde{T}_i$ and we have

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

Note that the tangent plane $\widetilde{T}_i$ used in (2) is not the ground-truth tangent plane T_i and the neighboring points are not guaranteed to be on a same plane due to the sampling density and local curvature, which could influence the denoising performance. Since the neighboring points share similar geometry information [14], the tangent planes of neighboring points are inherently similar; thus, we can project each point to its neighbors' tangent planes and average the multiple projections to obtain the denoised point. For the ith point $\widetilde{\mathbf{p}}_i$, we project it onto all of its neighbors' tangent planes,

$$\mathbf{t}_j^{(i)} = \arg\min_{\mathbf{t} \in \widetilde{T}_j} \|\widetilde{\mathbf{p}}_i - \mathbf{t}\|_2^2, \quad j \in \mathcal{N}_i.$$

We then obtain the denoised point $\widehat{\mathbf{p}}_i$ by minimizing the weighted sum of distance from $\widehat{\mathbf{p}}_i$ to $\mathbf{t}_j$ as,

$$\widehat{\mathbf{p}}_i = \arg\min_{\mathbf{p}} \sum_{j \in \mathcal{N}_i} W_{ij} \|\mathbf{p} - \mathbf{t}_j^{(i)}\|_2^2. \tag{3}$$

The solutions is simply $\widehat{\mathbf{p}}_i = \sum_{j \in \mathcal{N}_i} W_{ij} \mathbf{t}_j^{(i)}$. The proposed WMP algorithm for a given noisy point $\widetilde{\mathbf{p}}_i \in \mathcal{S}$ is summarized up in Algorithm 1.

Computational Cost. For a point cloud with N points, the computational complexities for graph construction and tangent plane estimation are $O(N \log N)$. The projection and weighted summation take $O(kN)$ time, where k is the average neighbors of each point decided by ϵ in the ϵ-NN graph. In total, our computational complexity is $O(N(\log N + k))$.

III. THEORETICAL ANALYSIS

In this section, we show the theoretical performance of the proposed WMP. For the simplicity, We just show a 2D case, where 2D points lies along a semicircle. We could generalize the results to 3D point clouds by replacing a semicircle to a sphere, which is left for future works.

Let a set of N 2D points be uniformly sampled from a semicircle centered at the center $[0 , r]$ with radius r. The ith point in a noiseless point cloud is

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} r \sin\theta_i \\ r - r\cos\theta_i \end{bmatrix},$$

Algorithm 1 Weighed multi-projection (WMP)

Input: A noisy point cloud $\mathcal{S} = \{\widetilde{\mathbf{p}}_i\}$,

       A hyperparameter ϵ-NN graph construction in ϵ

Output: A denoised point cloud $\widehat{\mathcal{S}}$

  Construct ϵ-NN graph and calculate weight matrix W

  for $\widetilde{\mathbf{p}}_i \in \mathcal{S}$ **do**

    $\mathcal{N}_i \leftarrow$ neighbors of $\widetilde{\mathbf{p}}_i$ defined by ϵ

    $\overline{\mathbf{p}}_i \leftarrow \sum_{j \in \mathcal{N}_i} W_{ij} \widetilde{\mathbf{p}}_j$

    $\overline{M}_i \leftarrow \sum_{j \in \mathcal{N}_i} W_{ij} \widetilde{\mathbf{p}}_i \widetilde{\mathbf{p}}_j^T - \overline{\mathbf{p}}_i \overline{\mathbf{p}}_i^T$

    $\mathbf{a}_i \leftarrow \arg \min \mathbf{a}^T \overline{M}_j \mathbf{a}, \quad \|\mathbf{a}\| = 1$

    $c_i \leftarrow \mathbf{a}_j^T \overline{\mathbf{p}}_j$,

  end for

  for $\widetilde{\mathbf{p}}_i \in \mathcal{S}$ **do**

    $\mathbf{t}_j^{(i)} \leftarrow \widetilde{\mathbf{p}}_i - \mathbf{a}_j^T \widetilde{\mathbf{p}}_i \mathbf{a}_j + c_j \mathbf{a}_j$

    $\widehat{\mathbf{p}}_i \leftarrow \sum_{j \in \mathcal{N}_i} W_{ij} \mathbf{t}_j^{(i)}$
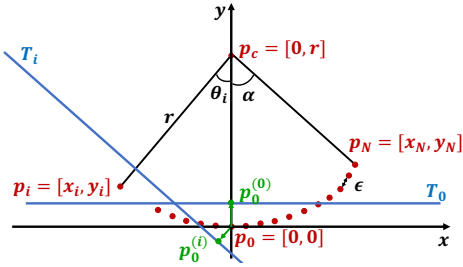
  end for

Fig. 2: The red dots represent the points sampled from a semicircle centered at $\mathbf{p}_c = [0, r]$ with radius r. The coordinate of point p_i is represented as $[x_i, y_i]$, $-N \leq i \leq N$; $\mathbf{p}_0 = [0, 0]$ is the origin point. The blue lines T_0 and T_i denote the tangent lines for point $\mathbf{p}_0$ and $\mathbf{p}_i$, respectively. The tangent lines are reconstructed based on their neighboring points. The length of the double arrows line is ϵ, representing the distance between neighboring points. The green point on the tangent lines $\mathbf{p}_0^{(i)}$ represents the projection of point $\mathbf{p}_0$ onto tangent line T_i. We use the angel θ_i to locate the position of points.

and the corresponding noisy point is

$$\widetilde{\mathbf{p}}_i = \begin{bmatrix} \widetilde{x}_i \\ \widetilde{y}_i \end{bmatrix} = \begin{bmatrix} r \sin \theta_i + n_{x,i} \\ r - r \cos \theta_i + n_{y,i} \end{bmatrix},$$

where $\theta_i = \epsilon i / r$ is the interval angle with ϵ a constant representing the interval distance between two points, $n_{x,i}$ and $n_{y,i}$ are the x and y components of noise and $i \in [-N, N]$. Following this, we have two properties:

1) The derivative at origin point $\mathbf{p}_0 = [0, 0]$ is 0 and the normal vector at point P is $[0, 1]^T$.
2) The curvature of all the sampled points is $\kappa = 1/r$.

Let $M_0, \widetilde{M}_0 \in \mathbb{R}^{2 \times 2}$ be the covariance matrices of the noiseless point $\mathbf{p}_0$, and the noisy point $\widetilde{\mathbf{p}}_0$ respectively, Following the definition in (1), we have

$$M_0 = \frac{1}{(2N+1)} \begin{bmatrix} \sum_{j=-N}^N x_j^2 & \sum_{j=-N}^N x_j y_j \\ \sum_{j=-N}^N x_j y_j & \sum_{j=-N}^N y_j^2 \end{bmatrix}$$

$$= \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix},$$

$$\widetilde{M}_0 = \frac{1}{(2N+1)} \begin{bmatrix} \sum_{j=-N}^N \widetilde{x}_j^2 & \sum_{j=-N}^N \widetilde{x}_j \widetilde{y}_j, \\ \sum_{j=-N}^N \widetilde{x}_j \widetilde{y}_j & \sum_{j=-N}^N \widetilde{y}_j^2 \end{bmatrix}$$

$$= \begin{bmatrix} \widetilde{m}_{11} & \widetilde{m}_{12} \\ \widetilde{m}_{12} & \widetilde{m}_{22} \end{bmatrix}.$$

Let $\mathbf{a}, \widehat{\mathbf{a}}$ be the ground-truth normal vector at point $\mathbf{p}_0$, and the estimated normal vector at point $\widetilde{\mathbf{p}}_0$, respectively. We can bound the error of the surface normal estimation based on Davis-Kahan Theorem [22].

Theorem 2: Let θ be the angle difference between the ground-truth normal vector $\widetilde{\mathbf{a}}$ and the estimated normal vector $\mathbf{a}$. Then,

$$|\sin \theta| \leq \frac{|\widetilde{m}_{12}| + |\widetilde{m}_{22} - m_{22}|}{m_{11}}.$$

We omit the proof here due to space constraints. The previous work on surface normal estimation [21] considers noise is added to y axis and the error bound θ is

$$|\theta| \leq \frac{|\widetilde{m}_{12}| + \widetilde{m}_{22}}{m_{11}}.$$

Compared to this bound, we assume that noise is added to both x and y axes and our bound is tighter when the SNR of a noisy signal is higher than 0.

We now show that the multi-projection (3) is theoretically better than the one-time projection (2).

Theorem 3: Let $\mathbf{p}_0 = [0, 0]$ be the origin point. Let $\widehat{\mathbf{p}}_o$ and $\widehat{\mathbf{p}}_m$ be the reconstructions of $\mathbf{p}_0$ based on the one-time projection (2) and the multi-projection (3), respectively. Then, we have

$$e_o(L, \kappa) = \|\widehat{\mathbf{p}}_o - \mathbf{p}_0\|_2 = \frac{1}{\kappa}(1 - \frac{\sin(L\kappa)}{L\kappa}),$$

$$e_m(L, \kappa) = \|\widehat{\mathbf{p}}_m - \mathbf{p}_0\|_2 = \frac{1}{\kappa}(\frac{1}{2}\frac{\sin 2(L\kappa)}{2(L\kappa)} + \frac{1}{2} - \frac{\sin^2(L\kappa)}{(L\kappa)^2}),$$

where $L = \epsilon N > 0$ and $\kappa = 1/r > 0$, representing the curvature at point $\mathbf{p}_0$.

Fig. 3 compares errors of one-time-projection $e_o(L, \kappa)$ and multi-projection $e_m(L, \kappa)$ for a fixed $\kappa = 1$ and a fixed $L = 1$, respectively. Both projection strategies produce zero error when $L \to 0$ or $\kappa \to 0$, but the proposed multi-projection achieves lower bound for both cases; the multi-projection is more robust than the one-time-projection because the value of $e_m(L, \kappa)$ varies much smoother than $e_o(L, \kappa)$ for $L \to 0$ or $\kappa \to 0$.
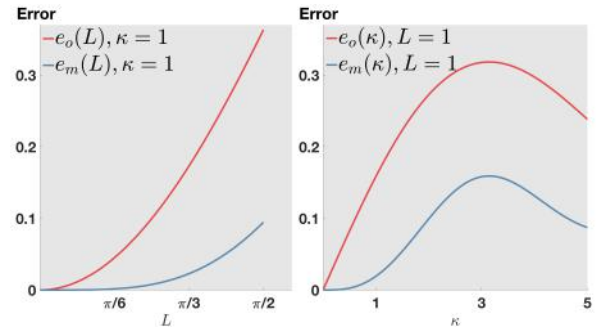

Fig. 3: The multi-projection is better than the one-time projection. Left figure shows the errors of two strategies with a fixed curvature $\kappa = 1$. Right figure shows the errors of two strategies with a fixed $L = 1$. In both, multi-projection produces smaller error and changes more slowly near to zero.

IV. EXPERIMENTAL RESULTS

Dataset. We evaluate our algorithm on the dataset in ShapeNet-

Core, which contains 55 manually verified common object categories with more than 50,000 clean 3D mesh models [23]. We first choose nine categories and randomly select 50 items in each category. Then we sample points from the surfaces of these 3D models with Poisson-disk sampling algorithm [24] and rescale the points into a unit cube centered at the origin. We add Gaussian noise to construct noisy point clouds. We compare the proposed WMP with the state-of-the-art denoising algorithms, including BF algorithm [11], GDB algorithm [25], PDE algorithm [16] and Non-local Denoising (NLD) algorithm [14]. For each of these algorithms, we tune the parameters to produce its own best performance.

Results and Analysis. To quantify the performance of different algorithms, we the following metrics: signal-to-noise ratio (SNR) defined as,

$$\mathrm{SNR}(\mathcal{S}_1, \mathcal{S}_2) = 20 \log \Big(\frac{\sum_{\mathbf{p}_i \in \mathcal{S}_1} \|\mathbf{p_i}\|_2^2}{\sum_{\mathbf{p}_i \in \mathcal{S}_1, \mathbf{q}_i \in \mathcal{S}_2} \|\mathbf{p}_i - \mathbf{q}_i\|_2^2} \Big);$$

mean squared error (MSE) defined as,

$$\mathrm{MSE}(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{N} \sum_{\mathbf{p}_i \in \mathcal{S}_1, \mathbf{q}_i \in \mathcal{S}_2} \|\mathbf{p}_i - \mathbf{q}_i\|_2^2;$$

Chamfer distance (CD) defined as,

$$\mathrm{CD}(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{N} \Big(\sum_{\mathbf{p}_i \in \mathcal{S}_1} \min_{\mathbf{q}_j \in \mathcal{S}_2} \|\mathbf{p}_i - \mathbf{q}_j\|_2^2 + \sum_{\mathbf{q}_i \in \mathcal{S}_2} \min_{\mathbf{p}_j \in \mathcal{S}_1} \|\mathbf{q}_j - \mathbf{p}_i\|_2^2 \Big),$$

where $\mathcal{S}_1$ and $\mathcal{S}_2$ represent the denoised point cloud and original point cloud, respectively, $\mathbf{p}_i$ and $\mathbf{q}_i$ represent the coordinates of points and N is the total number of points.

Figs. 4, 5 and 6 show the denoising performances in nine categories evaluated by SNR, MSE and CD (mean and standard deviation for 50 items in each category), respectively. We see that the proposed WMP outperforms its competitors in all nine categories; all algorithms perform slightly worse on chair and sofa datasets, because chair and sofa contain more sharp edges compared with other categories. Fig. 1 visualizes the denoising performances of a noisy sofa from different algorithms. We construct meshes from a noisy point cloud and its denoised versions by Poisson surface reconstruction algorithm with same parameters in MeshLab [26]. We see that the proposed algorithm outperforms its competitors in all categories of point clouds by all evaluation metrics (SNR, MSE and CD).

Fig. 5: The proposed WMP (in red) outperforms its competitors in terms of MSE. The denoised point clouds produced by WMP stay point-wise closer to the noiseless point clouds compared to the

Fig. 6: The proposed WMP (in red) outperforms its competitors in terms of CD. The denoised point clouds produced by WMP are closer to the original point clouds or the manifolds point clouds are lying on compared to the denoised point clouds produced by other algorithms.

V. CONCLUSIONS

We proposed a novel algorithm for 3D point cloud denoising by averaging the projections from multiple tangent planes. We analyzed the error of estimated normal vector and achieve a tighter bound compared to a previous work. We also showed that the multi-projection is more robust than the one-time projection. We validate the proposed algorithm on real datasets and compared with other 3D point cloud denoising algorithms and show that our algorithm outperforms its competitors in all nine categories for all three evaluation metrics.
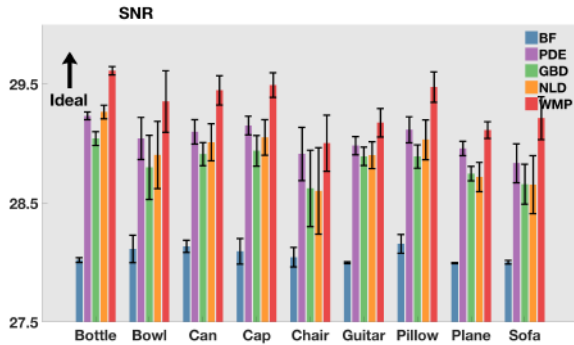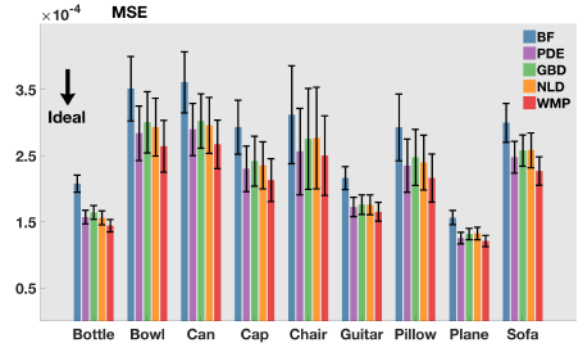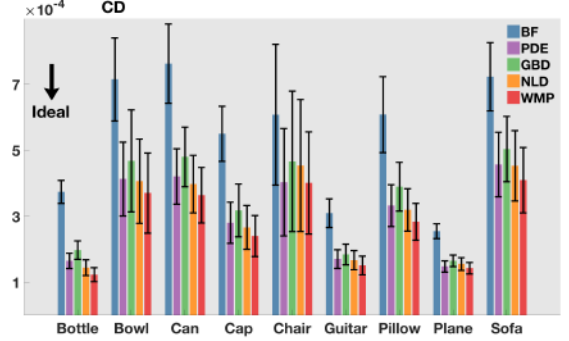

Fig. 4: The proposed WMP (in red) outperforms its competitors in terms of SNR. The denoised point clouds produced by WMP contain higher original signal intensity compared to the denoised point clouds produced by other algorithms.

APPENDIX

A. Davis Kahan theorem

Recall that if $V, \hat{V} \in R^{p \times d}$ both have orthonormal columns, then the vector of d principal angels between their column spaces is give by

$$(\cos^{-1}(\sigma_1), \cdots, \cos^{-1}(\sigma_d))^T, \qquad (4)$$

where $\sigma_1 \geq \cdots \geq \sigma_d$ are the singular values of $\hat{V}^T V$. Let $\Theta(\hat{V}, V)$ denote the $d \times d$ diagonal matrix whose jth diagonal entry is the jth principal angle, and let $\sin \Theta(\hat{V}, V)$ be defined entrywise.

Davis Kahan theorem: Let $\Sigma, \hat{\Sigma} \in R^{p \times p}$ be symmetric, with eigenvalus $\lambda_1 \geq \cdots \geq \lambda_p$ and $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_q$ respectively. Fix $1 \leq r \leq s \leq p$, let $d := s - r + 1$, and let $V = (\mathbf{v}_r, \mathbf{v}_{r+1}, \cdots, \mathbf{v}_s) \in \mathbb{R}^{p \times d}$ and $\hat{V} = (\hat{\mathbf{v}}_r, \hat{\mathbf{v}}_{r+1}, \cdots, \hat{\mathbf{v}}_s) \in \mathbb{R}^{p \times d}$ have orthonormal columns satisfying $\Sigma v_j = \lambda_j v_j$ and $\hat{\Sigma} \hat{v}_j = \hat{\lambda}_j \hat{v}_j$ for $j = r, r + 1, \cdots, s$. If $\delta := \inf\{|\hat{\lambda} - \lambda| : \lambda \in [\lambda_s, \lambda_r], \hat{\lambda} \in (-\infty, \hat{\lambda}_{s-1}] \cup [\hat{\lambda}_{r+1}, \infty)\} > 0$, where $\hat{\lambda}_0 := -\infty$ and $\hat{\lambda}_{p+1} := \infty$, then

$$\| \sin \Theta(\hat{V}, V) \|_F \leq \frac{\| \hat{\Sigma} - \Sigma \|_F}{\delta}. \qquad (5)$$

Frequently in applications, we have $r = s = j$, in which case we can conclude that

$$|\sin \Theta(\hat{\mathbf{v}}_j, \mathbf{v}_j)| \leq \frac{\| \hat{\Sigma} - \Sigma \|_{op}}{\min(|\hat{\lambda}_{j-1} - \lambda_j|, |\hat{\lambda}_{j+1} - \lambda_j|)} \qquad (6)$$

B. Covariance Matrix For Noise-free Point Cloud

For the noise-free point cloud based on the model in Sec. III, we have

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} r \sin \theta_i \\ r - r \cos \theta_i \end{bmatrix},$$

and the mean values of coordinates of the sampled points are as follows,

$$\bar{x} = 0,$$
$$\bar{y} = r(1 - \frac{\sin \alpha}{\alpha}), \qquad (7)$$
$$\alpha = \frac{\epsilon N}{r}.$$

To simplify the analysis, let

$$y(i) = y(i) - \bar{y}, \quad y_n(i) = y_n(i) - \bar{y}.$$

The covariance matrix for origin point $\mathbf{p}_0$ is

$$M_o = \sum_{j=-N}^{N} (\begin{bmatrix} x(j) \\ y(j) \end{bmatrix} - \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix})(\begin{bmatrix} x(j) \\ y(j) \end{bmatrix} - \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix})^T$$
$$= \sum_{j=-N}^{N} \begin{bmatrix} x(j) \\ y_r(j) \end{bmatrix} \begin{bmatrix} x(j) & y(j) \end{bmatrix} \qquad (8)$$
$$= \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} = \begin{bmatrix} m_{11} & 0 \\ 0 & m_{22} \end{bmatrix}.$$

Since we know $0 < \alpha < \frac{\pi}{2}$, we know $m_{11} > m_{12}$. The eigenvalues of this matrix is $\lambda_1 = m_{11}, \lambda_2 = m_{22}$. The corresponding eigenvectors are $v_1 = [1 \quad 0], v_2 = [0 \quad 1]$.

C. Covariance Matrix For Noisy Point Cloud

For the noisy point cloud, we have

$$\tilde{\mathbf{p}}_i = \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \end{bmatrix} = \begin{bmatrix} r \sin \theta_i + n_{x,i} \\ r - r \cos \theta_i + n_{y,i} \end{bmatrix}.$$

The covariance matrix for point $\tilde{\mathbf{p}}_0$ is

$$\widetilde{M}_o = \sum_{j=-N}^{N} (\begin{bmatrix} x_n(j) \\ y_n(j) \end{bmatrix} - \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix})(\begin{bmatrix} x_n(j) \\ y_n(j) \end{bmatrix} - \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix})^T$$
$$= \sum_{j=-N}^{N} \begin{bmatrix} x_n(j) \\ y_n(j) \end{bmatrix} \begin{bmatrix} x_n(j) & y_n(j) \end{bmatrix} \qquad (9)$$
$$= \begin{bmatrix} \tilde{m}_{11} & \tilde{m}_{12} \\ \tilde{m}_{12} & \tilde{m}_{22} \end{bmatrix}.$$

D. Error Estimation

Based on Davis Kahan Theorem, we still have

$$|\sin \Theta(\hat{\mathbf{v}}_j, \mathbf{v}_j)| \leq \frac{\| \hat{\Sigma} - \Sigma \|_{op}}{m_{11}}, \qquad (10)$$

where $\hat{\Sigma} = \hat{M}_c$ and $\Sigma = M_c$. We know

$$\hat{\Sigma} - \Sigma = \begin{bmatrix} 0 & \hat{m}_{12} \\ \tilde{m}_{12} & \tilde{m}_{22} - m_{22} \end{bmatrix},$$
$$\therefore \quad |\sin(\theta)| \leq \frac{|\tilde{m}_{12}| + |\tilde{m}_{22} - m_{22}|}{m_{11}} \qquad (11)$$

REFERENCES

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

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

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

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

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

[6] H. Chen and J. Shen, "Denoising of point cloud data for computer-aided design, engineering, and manufacturing," *Engineering with Computers*, Dec 2017. [Online]. Available: https://doi.org/10.1007/s00366-017-0556-4

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

[8] N. J. Mitra, A. Nguyen, and L. Guibas, "Estimating surface normals in noisy point cloud data," in *special issue of International Journal of Computational Geometry and Applications*, vol. 14, no. 4–5, 2004, pp. 261–276.

[9] 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*, 2002, pp. 124–131.

[10] M. Wei, J. Yu, W. M. Pang, J. Wang, J. Qin, L. Liu, and P. A. Heng, "Bi-normal filtering for mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 1, pp. 43–55, Jan 2015.

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

[12] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Jan 1998, pp. 839–846.

[13] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03. New York, NY, USA: ACM, 2003, pp. 950–953.

[14] J.-E. Deschaud and F. Goulette, "Point cloud non local denoising using local surface descriptor similarity," 2010.

[15] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao, and L. Xiao, "A review of algorithms for filtering the 3d point cloud," *Sig. Proc.: Image Comm.*, vol. 57, pp. 103–112, 2017.

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

[17] T. V. Thong, A. Elmoataz, and O. Lézoray, "Nonlocal pdes-based morphology on weighted graphs for image and data processing," *IEEE Transactions on Image Processing*, vol. 20, pp. 1504–1516, 2011.

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

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

[20] M. Vetterli, J. Kovaevi, and V. K. Goyal, *Foundations of Signal Processing*. Cambridge University Press, 2014.

[21] N. J. Mitra, A. Nguyen, and L. Guibas, "Estimating surface normals in noisy point cloud data," in *special issue of International Journal of Computational Geometry and Applications*, vol. 14, no. 4–5, 2004, pp. 261–276.

[22] Y. Yu, T. Wang, and R. J. Samworth, "A useful variant of the daviskahan theorem for statisticians," *Biometrika*, vol. 102, no. 2, pp. 315–323, 2015. [Online]. Available: http://dx.doi.org/10.1093/biomet/asv008

[23] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q.-X. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015.

[24] M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE Transaction on Visualization and Computer Graphics*, vol. 18, no. 6, pp. 914–924, 2012, http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.34. [Online]. Available: http://vcg.isti.cnr.it/Publications/2012/CCS12

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

[26] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, V. Scarano, R. D. Chiara, and U. Erra, Eds. The Eurographics Association, 2008.