Normal Correction Towards Smoothing Point-based Surfaces

Paola Valdivia*, Douglas Cedrim*, Fabiano Petronetto[†], Afonso Paiva*, Luis Gustavo Nonato*

* ICMC, USP, São Carlos - Brazil
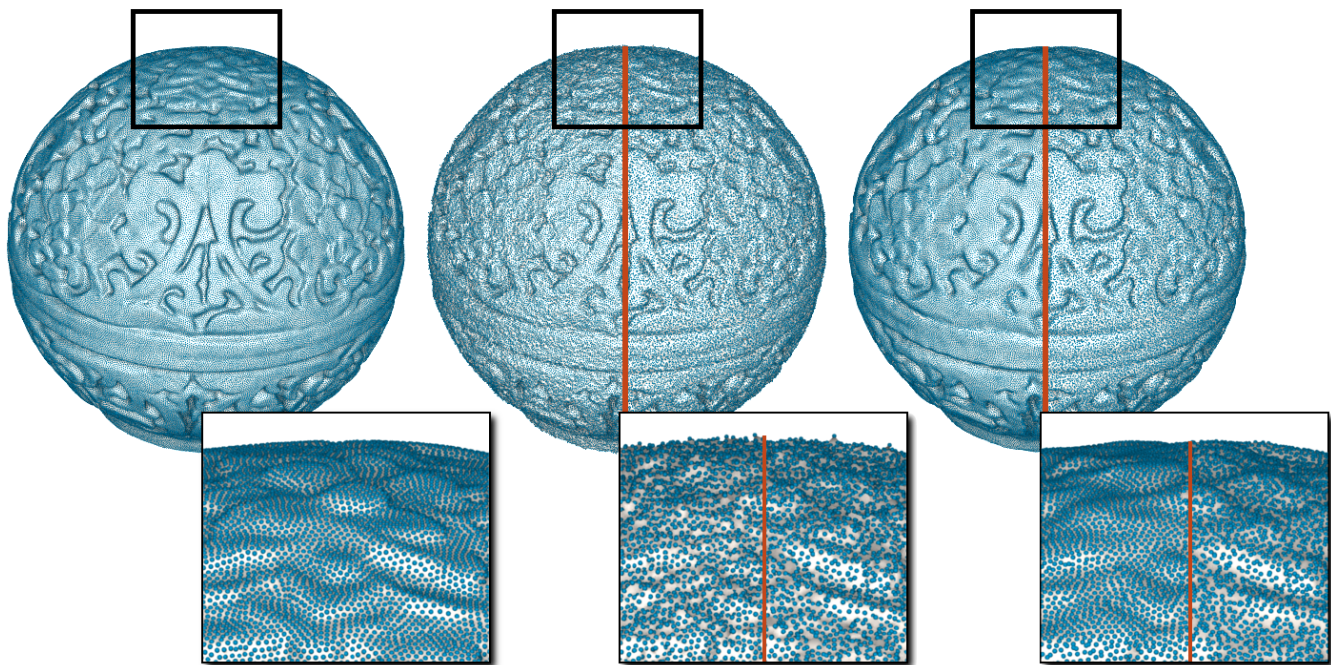[†] Universidade Federal do Espírito Santo - Brazil

Fig. 1. From left to right: original point-based surface, positions perturbed by Gaussian noise on normal (left) and random directions (right), their denoised counterparts.

Abstract—In recent years, surface denoising has been a subject of intensive research in geometry processing. Most of the recent approaches for mesh denoising use a two-step scheme: normal filtering followed by a point updating step to match the corrected normals. In this paper we propose an adaptation of such two-step approaches for point-based surfaces, exploring three different weighting schemes for filtering normals. Moreover, we also investigate three techniques for normal estimation, analyzing the impact of each normal estimation method in the whole point-set smoothing process. Towards a quantitative analysis, in addition to conventional visual comparison, we evaluate the effectiveness of different choices of implementation using two measures, comparing our results against state-of-art point-based denoising techniques.

Keywords-point-based surface; normal estimation; surface smoothing;

I. INTRODUCTION

Surface smoothing is a well established field in the context of geometry processing, where many methods have been proposed towards removing noise while preserving surface features as much as possible. Existing techniques vary considerably as to mathematical foundation, encompassing methodologies derived from spectral theory, diffusion map, projection operators, and bilateral filters. Most of those methods have been originally developed for mesh-based surface representation, inspiring variants in the context of surfaces represented by point clouds, which is the focus of this work.

Specifically in the context of point-based surfaces (a comprehensive discussion about mesh-based surface smoothing is beyond the scope of this work), smoothing techniques have experienced a substantial progress in the last decade. The development of robust methods for discretizing the Laplace-Beltrami operator on point-based surfaces has leveraged most

of such development. Pauly et al. [1] were one of the pioneers in using Laplace operators to perform point-based surface smoothing. Their approach uses an umbrella operator as discretization mechanism, carrying out the smoothing as a diffusion process. Lange and Polthier [2] make use of an anisotropic version of the Laplace operator so as to detect and preserve surface features such as edges and corners during the smoothing process. More recently, Petronetto et al. [3] have exploited spectral properties of SPH-based discretization of the Laplace-Beltrami operator to smooth point set surfaces, but without feature preservation.

Projection operators comprise another important class of smoothing technique for point set surfaces. Firstly proposed by Alexa et al. [4], projection operators typically accomplish the smoothing in two steps, normal estimation and polynomial fitting. Distinct approaches have been proposed to handle each step, ranging from surface fitting other than polynomial [5] to robust statistics [6], [7], [8], which allows for better preserving features during smoothing.

A common characteristic of Laplace-based methods and projection operators discussed above is their sensitivity to normals, that is, the quality of the normals (given or computed) affects the performance of those methods considerably. The dependence of normals is mitigated by techniques such as locally optimal projection (LOP) [9] and weighted locally optimal projection (WLOP) [10], which rely on a global minimization problem that does not make use of normal information to provide a second order approximation to smooth surfaces. Representing sharp features is an issue that can hardly be addressed by LOP and WLOP. Moreover, the high computational cost of these methods impairs their use in big data sets. Liao et al. [11] combine the mathematical formulation of LOP with normal estimation, bilateral weighting, and a sampling mechanism so as to better capture sharp features while lessening computational times during smoothing. Liao's approach, however, leads to an intricate algorithm made up of several complex steps.

One can see from discussion above that normal vectors are fundamental when sharp features and details have to be preserved during the smoothing process. However, in the presence of noise data, normals can hardly be estimated accurately, hampering existing point-based surface smoothing algorithms to work properly. This problem is also known in the context of mesh-based surface smoothing, where alternatives have been proposed to compute a fair normal field. A typical approach for denoising surface meshes while preserving features is first to filter normal vectors towards removing noise and then update mesh vertex position such that the mesh cope with the filtered normal field [12], [13], [14], [15]. Although most authors claim that normal filtering methods can be adapted to the context of point set surfaces, as far as we know, no such extension/adaptation has been presented in the literature, raising doubts on the effectiveness and pitfalls of normal filtering to smooth surface of points. More specifically, the literature presents approaches that filter normals for rendering purposes but without point update [16] or update points

considering the original normals [17]. Assessing the efficacy of combining both normal filtering and point update is still an issue to be investigated.

The goal of this paper is to investigate the two-step filtering methodology, that is, normal filtering and point update, in the context of point set surfaces. The proposed methodology investigates different methods for estimating normals, filtering the normal field, and updating the point set to match the filtered normals. The provided comprehensive study allows for identifying the best set of tools and how to implement them in each step of the smoothing process in order to reach a robust and effective normal filtering-based denoising technique for point set surface. Comparisons with state-of-art techniques show that, when properly implemented, normal filtering denoise methods is quite effective in the context of point set surfaces, outperforming most existing methods as to feature preservation and noise reduction.

In summary, the contributions of this paper are:

- A thorough investigation of alternatives to extend normal filtering denoise and point update to the context of point set surface. More specifically, we investigate different ways of implementing each step of the smoothing pipeline, analyzing the effectiveness of each combination.
- A comprehensive set of comparisons against state-of-art point-based denoising techniques.
- The definition of quantitative measures to analyze the effectiveness of smoothing techniques, which allows for clearly assessing the quality of point-based surface denoise methods.

II. Smoothing Mechanism and Its Alternatives

The proposed methodology for point set surface smoothing is made up of three main steps: normal estimation, normal filtering, and point update. Different techniques can be used to perform each of these steps and in the current work we investigate three possible methods for normal estimation, three distinct approaches for normal filtering and an up-to-date method for point update. Fig. 2 illustrates the pipeline of our approach together with the different methods used to implement each step. The mathematical and computational tools used to implement each step of the proposed point-based surface smoothing process are described below.

A. Normal Estimation

The first step of our pipeline is normal estimation. Given a (noisy) point cloud $S = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$, we investigate three different mechanisms to estimate a normal $\mathbf{n}_i$ for each point $\mathbf{p}_i$, namely, PCA, Weighted PCA, and Randomized Hough Transform.

PCA-based (Principal Component Analysis) normal estimation method computes the normal for each point $\mathbf{p}_i$ by constructing a covariance matrix from points $\mathbf{p}_j$ in the neighborhood of $\mathbf{p}_i$, setting the normal vector $\mathbf{n}_i$ as the principal component of the covariance matrix with smallest covariance (the eigenvector associated to the smallest eigenvalue) [18]. The nearest neighbors of $\mathbf{p}_i$ defines the neighborhood used
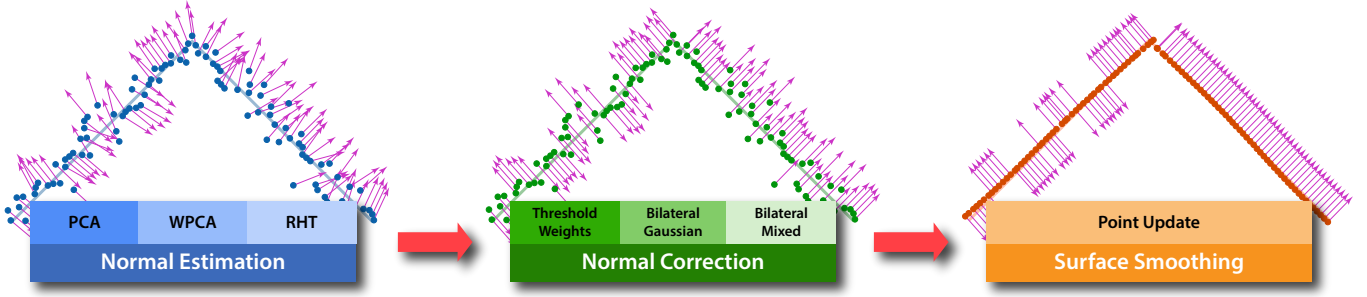
Fig. 2. Point set surface smoothing pipeline and its alternative implementations.

to build the covariance matrix (we have performed tests with different number of neighbors, as discussed in Section III).

Weighted PCA (WPCA) is an extension of PCA that can also be used to estimate normals. WPCA works quite similarly to PCA, except that a weighting function is used to define the contribution of each point to the covariance matrix. More specifically, the covariance matrix $\mathbf{C}_i$ associated with the point $\mathbf{p}_i$ is defined by

$$\mathbf{C}_i = \mathbf{X}\mathbf{W}\mathbf{X}^\top \qquad (1)$$

where $\mathbf{X}$ is the matrix with columns formed by the coordinates of points in the neighborhood of $\mathbf{p}_i$ and $\mathbf{W}$ is a diagonal matrix with nonzero entry w_{jj} corresponding to the weight associated with the point $\mathbf{p}_j$. In our implementation the weight associated with each point $\mathbf{p}_j$ in the neighborhood of $\mathbf{p}_i$ is given by the inverse of the euclidean distance between $\mathbf{p}_i$ and $\mathbf{p}_j$. As far as we know WPCA has never been used to estimate normals, despite its reported effectiveness in applications such as data clustering and classification.

Randomized Hough Transform has been proposed by Boulch and Marlet [19] as a robust mechanism for normal estimation. The idea is to randomly choose three points in the neighborhood of $\mathbf{p}_i$ so as to define a plane. The normal of each plane votes to a bin using a Hough accumulator. The number of random triples need to reach a reliable normal estimation is defined as $T = \frac{1}{2\delta^2} \log\left(\frac{2M}{1-\alpha}\right)$, where α is the minimum tolerated probability, such that the distance between the theoretical distribution and the observed distribution is at most δ. This number can be reduced by stop picking triples as soon as the confidence intervals of two bins do not intersect, i.e., the confidence is greater than $2\sqrt{1/T}$. The normals of the most voted bin are averaged to define the normal $\mathbf{n}_i$.

B. Normal Correction

Normals estimated with any of the methods described in the previous subsection are prone to not vary smoothly due to noise in the point set. The typical approach to denoise normals is to iteratively average normals from neighbor points using specific weights, more specifically,

$$\mathbf{n}_i^l = \frac{\left(\sum_{j \in \mathcal{N}_i} w_j \mathbf{n}_j^{l-1}\right)}{\left\| \left(\sum_{j \in \mathcal{N}_i} w_j\right) \right\|}, \qquad (2)$$

where $\mathcal{N}_i$ accounts for the k-nearest neighbors points of $\mathbf{p}_i$, $\mathbf{n}_j^l$ is the normal obtained at iteration l, setting $\mathbf{n}_i^0$ as the initial estimated normal, and w_j is the weight used to define the contribution of each normal.

Normal filtering methods differ mainly as to the choice of the weights w_j. We have investigated three different methods to compute the weights, one based on a simple threshold scheme and two based on bilateral filtering.

Thresholded weights Sun et al. [15] proposed a simple threshold-based scheme to define the weights w_i used in the iterative filtering mechanism 2. Such threshold-based scheme can be stated as follows:

$$w_j = \begin{cases} 0 & , \text{if } \mathbf{n}_i \cdot \mathbf{n}_j \leq T \\ (\mathbf{n}_i \cdot \mathbf{n}_j - T)^2 & , \text{otherwise} \end{cases} \qquad (3)$$

where $0 \leq T \leq 1$ is a threshold defined by the user. In our implementation we use $T = 0.65$.

Bilateral-Gaussian Zheng et al. [14] presented, in the context of meshes, a weighting scheme inspired on bilateral filtering commonly used in image denoising. Zheng's weighting scheme is given by:

$$w_j = W_c(||\mathbf{p}_i - \mathbf{p}_j||)W_s(||\mathbf{n}_i - \mathbf{n}_j||), \qquad (4)$$

where W_c and W_s are Gaussian functions as below:

$$W_c(x) = \exp(-x^2/2\sigma_c^2), \quad W_s(x) = \exp(-x^2/2\sigma_s^2), \quad (5)$$

and σ_c, σ_s are the standard deviation of the Gaussians. In the context of meshes, σ_c has been set as the average length of edges incident to $\mathbf{p}_i$. Following the same reasoning, we have set σ_c as the average distance from $\mathbf{p}_i$ to $\mathbf{p}_j$ in the neighborhood of $\mathbf{p}_i$. There is no well established mechanism to tune σ_s, even in the context of mesh-based denoising, so it is determined manually.

Bilateral-mixed Wang et al. [20] proposed a weighting scheme that combines both thresholded-weight and bilateral-Gaussian. We have adapted Wang's formulation to the context of point sets as follows:

$$w_j = W_c(||\mathbf{p}_i - \mathbf{p}_j||)\Phi_s(\mathbf{n}_i, \mathbf{n}_j), \qquad (6)$$

where W_c is the same as in 5 and Φ_s is function given by:

$$\Phi_s(\mathbf{n}_i, \mathbf{n}_j) = \begin{cases} 0 & \text{, if } (\mathbf{n}_i - \mathbf{n}_j) \cdot \mathbf{n}_i \geq T \\ ((\mathbf{n}_i - \mathbf{n}_j) \cdot \mathbf{n}_i - T)^2 & \text{, otherwise} \end{cases}$$
$$(7)$$

where $T = \sqrt{\sum_{j \in \mathcal{N}_i} ((\mathbf{n}_i - \mathbf{n}_j) \cdot \mathbf{n}_i)^2 / k}$, and k is the number of points in $\mathcal{N}_i$.

The methods described above do not work properly when normals are not consistently oriented, as illustrated in Fig. 3. However, since those methods rely on a local procedure to filter the normals, we only need to guarantee local consistence regarding orientation. In other words, we can make the filter work properly by simply flipping normals $\mathbf{n}_j$ if $\mathbf{n}_i \cdot \mathbf{n}_j \leq 0$. This procedure preserves the orientation of normals, thus it does not guarantee a consistently oriented normal field. However, this is not an issue in our formulation, since the point update mechanism we adapted from [15] (described below) does not require a consistently oriented normal field.
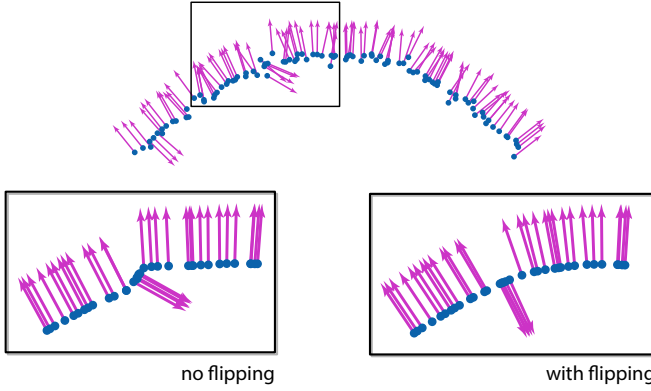

Fig. 3. Estimated normals with different orientation (top row). Filtering without flipping and with flipping procedure, respectively (bottom row).

C. Surface Smoothing

We build upon the ideas of Sun et al. [15] to formulate a point update scheme that operates on point set (Sun's approach relies on meshes). The advantages of Sun's approach is twofold, namely, it does not demand surface area computations, which is difficult to accurately estimate using point sets, and it is oblivious to normal orientation, that is, it does not require normals to be consistently oriented.

Given the filtered normals, denote by $\mathbf{n}'_i$, point positions have to be updated to match the normals $\mathbf{n}'_i$. The points are updated according to the iterative scheme:

$$\mathbf{p}_i^l = \mathbf{p}_i^{l-1} + \frac{1}{\sum_{j \in \mathcal{N}_i} w_j} \sum_{j \in \mathcal{N}_i} \mathbf{n}'_j (w_j \mathbf{n}'_j \cdot (\mathbf{p}_j^{l-1} - \mathbf{p}_i^{l-1})), \quad (8)$$

where $\mathbf{p}_i^l$ is the point position in the lth iteration step.

Notice that the updating process (8) can be seen as a displacement based on weighted average of normals $\mathbf{n}'_j \in \mathcal{N}_i$, where weights are given by $w_j \mathbf{n}'_j \cdot (\mathbf{p}_j^{l-1} - \mathbf{p}_i^{l-1})$, and w_j is defined using the bilateral mechanism:

$$w_j = W_c(\|\mathbf{p}_i - \mathbf{p}_j\|) W_s(1 - (\mathbf{n}_i . \mathbf{n}_j)), \quad (9)$$

where W_c and W_s are the same Gaussian functions as in 5, with $\sigma_c = \max_{j \in \mathcal{N}_i}(\|\mathbf{p}_i - \mathbf{p}_j\|)$ and $\sigma_s = 1/3$.

Next section shows the effectiveness of the proposed point update scheme when combined with different mechanism to estimate and filter normals.

III. RESULTS AND COMPARISONS

In this section we analyze the performance of each possible variation of the proposed surface smoothing method. More precisely, we have tested the following combinations: PCA + Threshold Weights + Point Update (PTU), PCA + Bilateral Gaussian + Point Update (PGU), PCA + Bilateral Mixed + Point Update (PMU), WPCA + Thresholded Weights + Point Update (WPTU), WPCA + Bilateral Gaussian + Point Update (WPGU), WPCA + Bilateral Mixed + Point Update (WPMU), Randomized Hough Transform + Thresholded Weights + Point Update (HTU), Randomized Hough Transform + Bilateral Gaussian + Point Update (HGU), Randomized Hough Transform + Bilateral Mixed + Point Update (HMU). Moreover, we have also tested the impact of varying the size of the neighborhood involved in each step of the pipeline, considering neighborhoods with size 7, 15, and 21 nearest points for each point p_i. The number on the right of each acronym indicates the size of the neighborhood, for instance, PTU7 means that the 7 nearest neighbor of each point is considered in the computations.

We have run the distinct implementations in 5 different models with Gaussian noise added to the points. Gaussian noise has been added in two different ways, normal (GN) as well as random directions (GR) with standard deviation σ proportional to 0.2 of the mean edge length of the underlying mesh (the original models have a meshed counterpart). Fig. 4 and Fig. 5 illustrate the models used in our tests, specifically, the models are: Bitorus, Elephant, Femur, Nicolo, and Fandisk.

The accuracy of each alternative implementation has been measured using two different metrics, which are defined as follows. Since we have a one-to-one correspondence between the points in the original and the smoothed surface, we measure the quality of the smoothing process by simply computing, for each point $\mathbf{p}_i$, the difference between the curvature in the point $\mathbf{p}_i$ before and after smoothing, averaging those differences to produce a quality metric. The closer to zero the better is the smoothing mechanism. We use the algorithm proposed by Pauly et al. to estimate the curvature in point-set surface [21].

Comparing the original surface area against the smoothed surface area is a measure of smoothness widely used by the geometry processing community which we also employ in our comparisons. The models used in our experiments are endowed with triangular mesh, thus allowing to compute areas before and after smoothing, although, the whole smoothing process relies only on the points (the coordinates of the vertices on the original mesh) to denoise the model.

Another important aspect to be considered when implementing any of the alternatives of the proposed pipeline is the
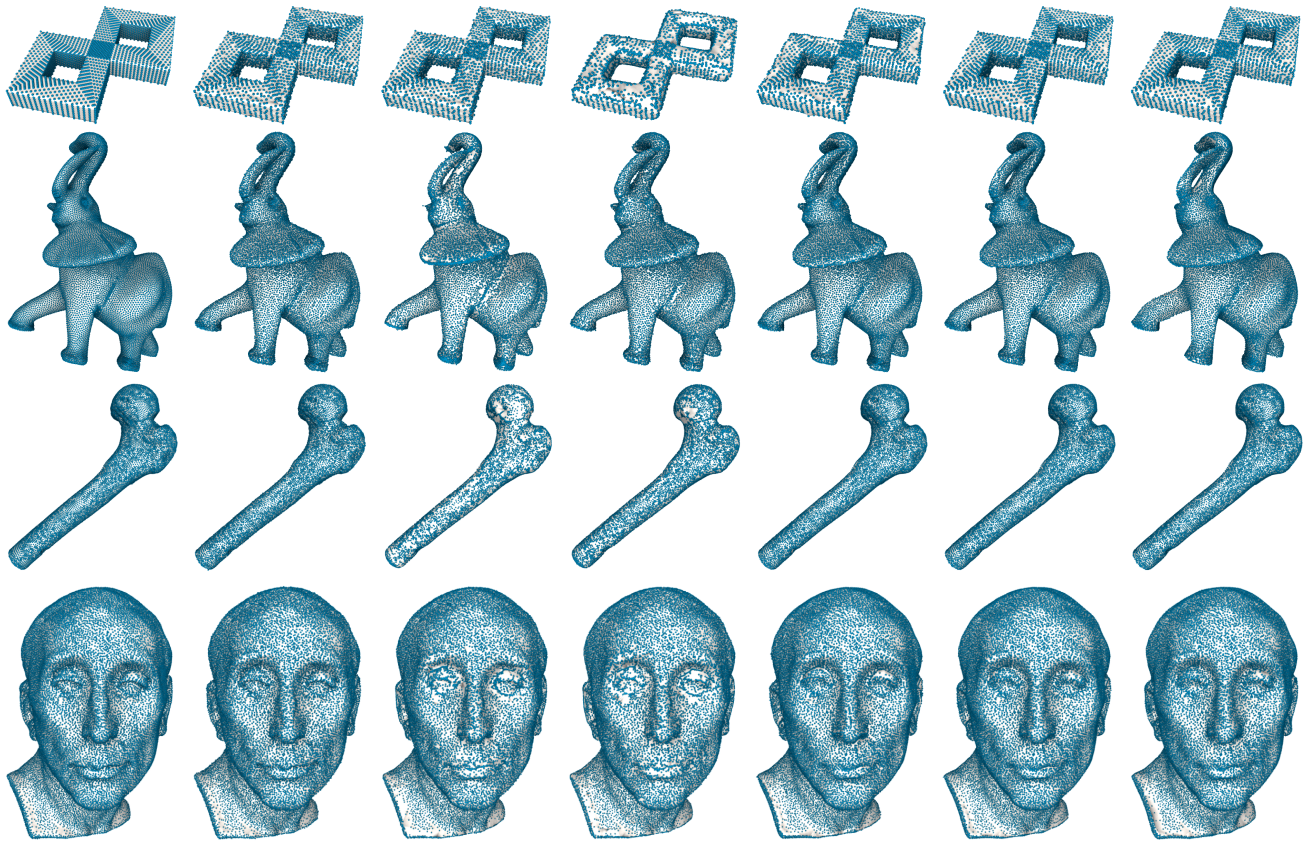
Fig. 4. Mesh denoising on Bitorus with 4350 points, Elephant with 24955 points, Femur with 15185 points and Nicolo with 25239 points. From left to right: original meshes, their noisy counterparts (random direction), and the results from LOP, WLOP, APSS, RIMLS and our methods.
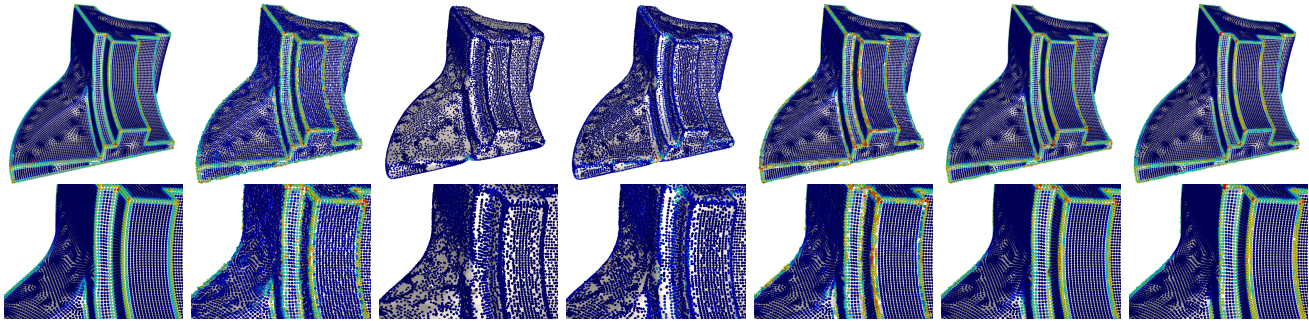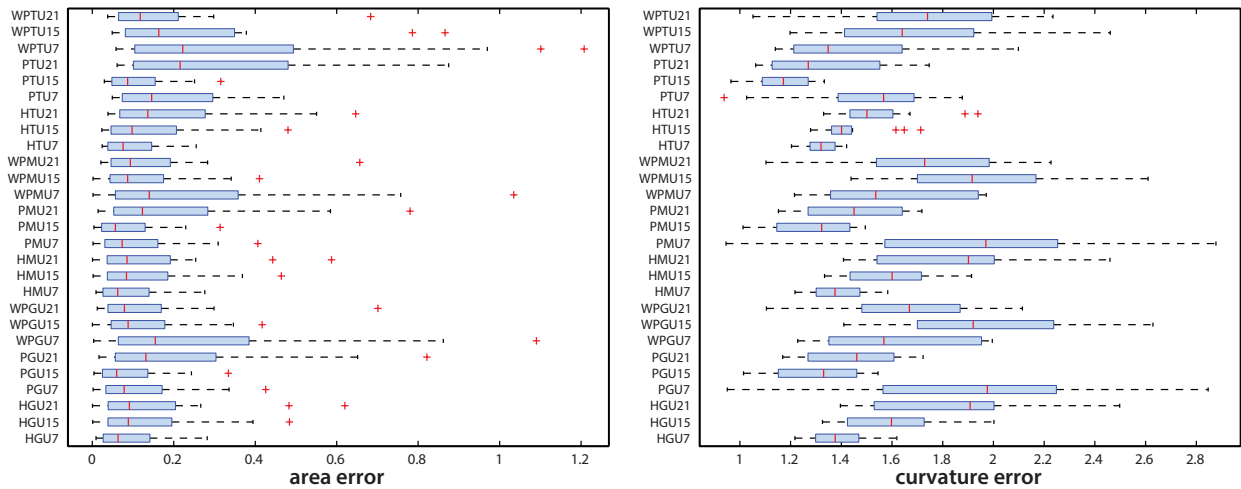

Fig. 5. Smoothing point-based approach is capable of preserving sharp features while smoothing the meshes. From left to right: original and noisy surfaces; and results with LOP, WLOP, APSS, RIMLS and our method.

number of iterations to be employed during the normal filtering and during the point update steps, as the convergence of the iterative methods described in (2) and (8) are, respectively, not guaranteed or only ensured under restrictive hypothesis (in the context of surface meshes). Therefore, in practice, the number of iterations has to be set a priori. We have tested the normal filtering with five different number of iteration steps, namely, 2, 4, 8, 16 e 32, and the point update with four different number of iteration steps, namely, 5, 10, 20 e 40, resulting in twenty combinations.
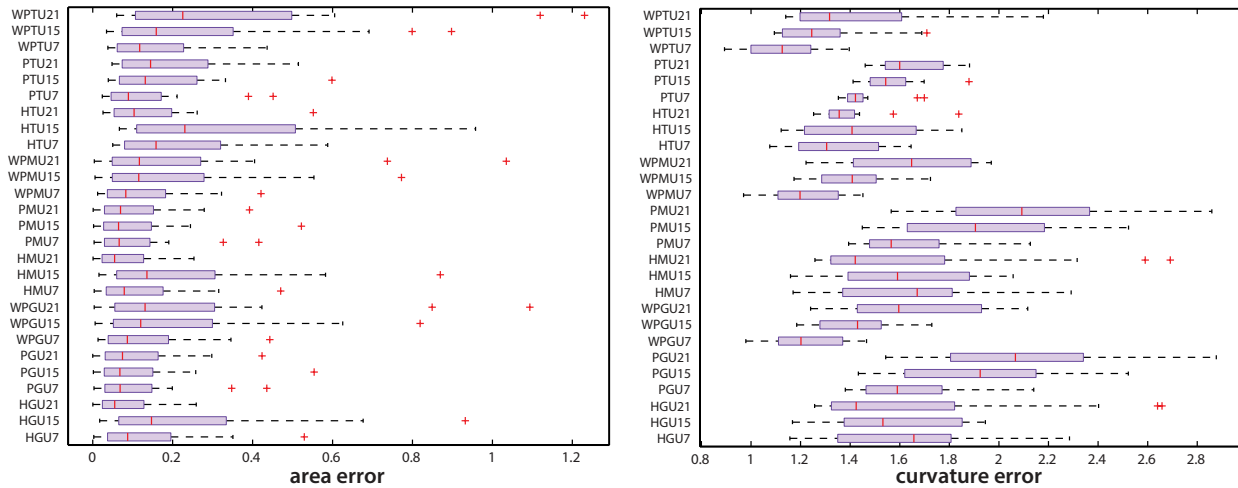
Box plots in Fig. 6 show the performance of each alternative implementation when facing each of the twenty possible choices of interaction steps when smoothing the Elephant data set. Notice that the error varies considerably for all methods, showing that the number of iteration steps affects considerably their performance. Moreover, one can see that methods using PCA and Hough with neighborhood 7 and 15 tend to generate smaller errors than the other alternatives for the area error for both normal and random Gaussian noise. For curvature error, PCA normal estimation with neighborhood 15 has a better performance for normal Gaussian noise while WPCA with neighborhood 7 is the best choice for random Gaussian noise.

Considering the 60 variations of each implementation (3 different neighborhood size and 20 choices of iteration steps),

(a) Gaussian noise on normal direction.

(b) Gaussian noise on random direction.

Fig. 6. Quantitative evaluation of the metrics using different combinations of smoothing methods on elephant model ($\times 10^{-4}$).

Fig. 7 shows the average curvature and area error of each method for all surface models. It is clear that Thresholded Weights approach performs better regarding area error while it is not possible to clearly point out the best method as to curvature error.

Fig. 8 depicts the $area \times curvature$ error scatter plot for all 540 possible implementation choices (9 techniques with 3 different neighborhood size and 20 choices of iteration steps) applied to smooth the Bitorus data set. Highlighted dots correspond to the best (bottom left) and worst (top right) combined result as well as the best result as to area (bottom right) and the worst in terms of curvature (top left). The model highlighted on left middle is an average case.

Our method is compared with LOP (Locally Optimal Projection), WLOP (Weighted Locally Optimal Projection), Algebraic Point Set Surfaces (APSS) and Robust Implicit Moving Least Squares (RIMLS). For each of these methods, we have tested three parameter combinations: the default parameter set and two other combinations that produced visually pleasing

results. From the three results we chose the one that gives the best results as to area and curvature metrics. The parameter sets for the methods are: LOP and WLOP (support radius h, μ
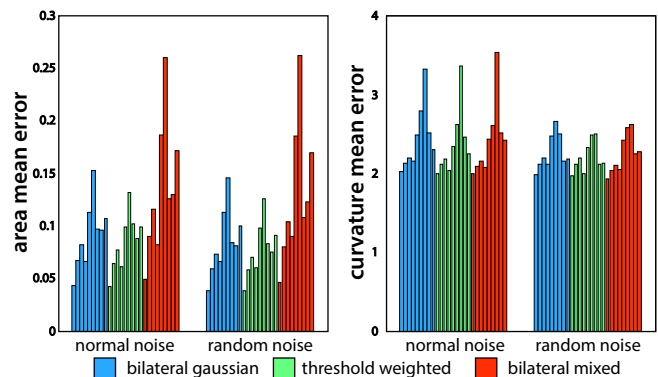

Fig. 7. Average error of each alternative implementation for all models varying among 27 choices of neighborhood size and iteration steps.
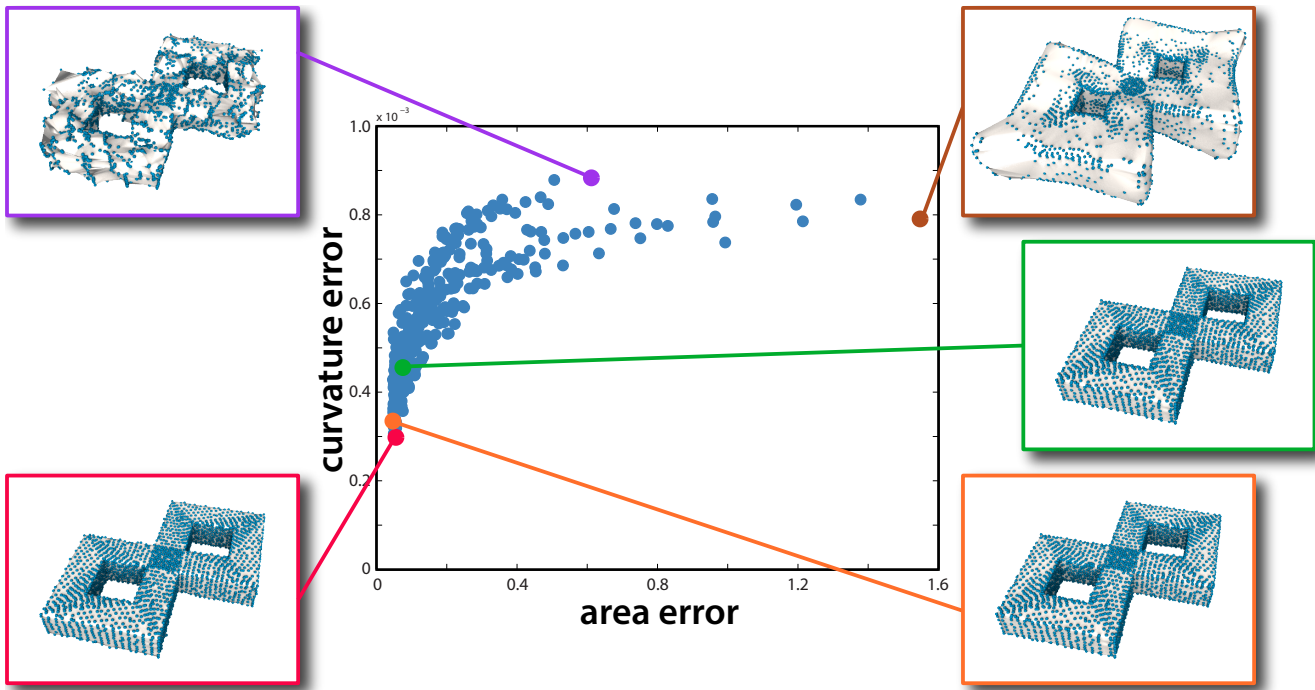
Fig. 8. Scatter metrics (area,curvature) in the bitorus model (random noise).

controlling the repulsion force), APSS (filter scale, spherical parameter), RIMLS (filter scale, sharpness) and ours (acronym described above, number of normal filtering iterations, number of point update iterations). Table I shows the result of such comparison, where E_A and E_k account for area and curvature error respectively. Notice that our method is quite competitive, outperforming in some cases the state-of-art RIMLS method.

We conclude this section with a qualitative result showing the effectiveness of our methodology in preserving sharp features. Fig. 5 shows denoising results on the fandisk model resulting from LOP, WLOP, APSS, RIMLS and ours. Color code shows the curvature distribution. One can notice that our method yields good results, comparable again to RIMLS, showing that normal filtering combined with point update is an alternative for point-based surface smoothing.

We have also tested our two-step method with a real scanned model with raw noise. Fig. 9 shows the denoised result of the scanned bust model. We have used the following parameters PGU7 with 5 iterations in normal updating step and 7 iterations for perform the point updating step. We can observe that our method preserves feature to some extent.

IV. CONCLUSION

In this paper we show that the normal-filtering/point-update smoothing procedure commonly employed in the context of surface meshes can be extended to the context of point-based surface with good results. The comprehensive set of tests we performed showed that parameters such as the number of neighbors and iteration steps used in the different phases of the pipeline can impact substantially on the quality of the smoothing process. Moreover, the provided comparisons show that normal-filtering/point-update can outperform existing methods, being quite competitive and a good alternative for point-based surface smoothing.

Like most previous denoising methods based on distance neighborhoods, our method does not perform well in the presence of two parallels close sheets surfaces, as shown in Fig. 10. A possible solution to this problem could be to consider orienting normals before normal estimation. In that way, normals could be more accurately filtered and furthermore, normals with opposite orientation would not have a big influence in the point updating step.

As a future work, one could use the theoretical framework proposed by Mitra et al. [22] to ensure bounds on the normal estimation step. Moreover, the analysis could be improved by different noise distributions.

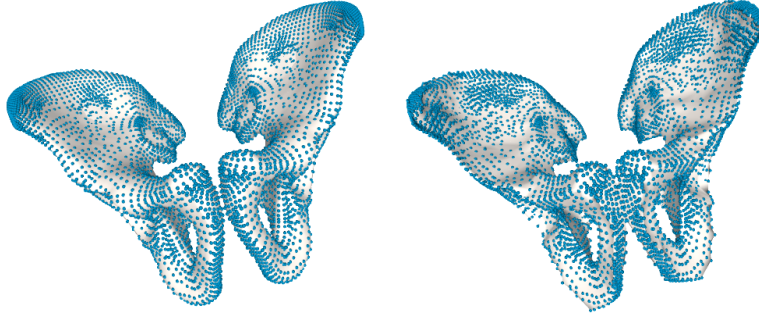Fig. 9. Denoised a real scanned model with raw noise.

Fig. 10. A drawback with point set surfaces is to determine the neighborhood relation of points on surfaces with close sheets.

TABLE I
BEST PARAMETER SETS AND ERRORS

Model	Noise	Method	Parameters	$E_A \times 10^{-3}$	$E_k \times 10^{-3}$
Double Torus ($\|S\| = 4350$)	GN	LOP	(0.14, 0.3)	53.62	0.78
		WLOP	(0.12, 0.3)	26.45	0.72
		APSS	(3, 0.5)	5.59	0.49
		RIMLS	(5, 0.5)	4.24	**0.24**
		Ours	(PTU7, 32, 5)	**0.59**	0.31
	GR	LOP	(0.9, 0.4)	95.68	0.45
		WLOP	(0.12, 0.3)	59.42	0.72
		APSS	(3, 0.5)	23.16	0.54
		RIMLS	(3, 0.5)	12.39	**0.31**
		Ours	(WPTU21, 4, 5)	**0.22**	0.34
Elephant ($\|S\| = 24955$)	GN	LOP	(0.07, 0.4)	11.86	0.18
		WLOP	(0.05, 0.5)	5.64	0.24
		APSS	(3, 0.75)	0.70	**0.12**
		RIMLS	(4, 1)	6.59	0.14
		Ours	(PGU21, 32, 5)	**0.24**	0.24
	GR	LOP	(0.05, 0.4)	42.14	0.15
		WLOP	(0.03, 0.4)	38.87	0.13
		APSS	(2, 1)	1.57	0.09
		RIMLS	(2, 0.75)	1.58	**0.08**
		Ours	(HG21, 32, 5)	**0.04**	0.22
Fandisk ($\|S\| = 25894$)	GN	LOP	(0.08, 0.3)	1.22	0.21
		WLOP	(0.07, 0.35)	37.36	0.20
		APSS	(2, 1)	5.76	0.10
		RIMLS	(4, 0.5)	0.59	**0.05**
		Ours	(HTU21, 16, 5)	**0.02**	0.08
	GR	LOP	(0.07, 0.35)	21.82	0.20
		WLOP	(0.07, 0.35)	35.21	0.20
		APSS	(2, 1)	0.58	**0.08**
		RIMLS	(4, 0.75)	1.13	**0.08**
		Ours	(HGU7, 32, 5)	**0.04**	0.09
Nicolo ($\|S\| = 25239$)	GN	LOP	(0.06, 0.3)	141.30	0.10
		WLOP	(0.07, 0.25)	10.52	0.10
		APSS	(4, 0.5)	9.74	0.10
		RIMLS	(4, 1)	0.18	**0.08**
		Ours	(WPTU15, 8, 5)	**0.05**	0.09
	GR	LOP	(0.04, 0.3)	71.31	0.11
		WLOP	(0.04, 0.25)	25.52	0.10
		APSS	(4, 0.75)	2.67	0.11
		RIMLS	(4, 1)	0.36	0.09
		Ours	(PGU7, 16, 5)	**0.22**	**0.09**
Femur ($\|S\| = 8168$)	GN	LOP	(0.07, 0.4)	13.92	0.26
		WLOP	(0.06, 0.4)	**0.17**	0.26
		APSS	(3, 0.75)	1.55	**0.22**
		RIMLS	(4, 0.75)	0.40	0.23
		Ours	(HTU7, 32, 5)	9.07	**0.22**
	GR	LOP	(0.06, 0.3)	33.19	0.26
		WLOP	(0.06, 0.4)	7.84	0.26
		APSS	(3, 1)	0.98	0.23
		RIMLS	(3, 0.75)	**0.60**	**0.21**
		Ours	(HTU7, 32, 5)	10.20	0.22

REFERENCES

[1] M. Pauly, L. Kobbelt, and M. Gross, "Multiresolution modeling of point-sampled geometry." ETH Zurich, Tech. Rep., 2002.

[2] C. Lange and K. Polthier, "Anisotropic smoothing of point sets," *Comput. Aided Geom. Design*, vol. 22, no. 7, pp. 680–692, 2005.

[3] F. Petronetto, A. Paiva, E. S. Helou, D. E. Stewart, and L. G. Nonato, "Meshfree discrete Laplace-Beltrami operator," *Comput. Graph. Forum*, 2013, (*to appear*).

[4] M. Alexa, J. Behr, D. Cohen-or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, pp. 3–15, 2003.

[5] G. Guennebaud and M. Gross, "Algebraic point set surfaces," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[6] B. Mederos, L. Velho, and L. H. de Figueiredo, "Robust smoothing of noisy point clouds," in *Proc. SIAM Conf. Geom. Des. Comput.*, 2003.

[7] S. Fleishman, D. Cohen-Or, and C. T. Silva, "Robust moving least-squares fitting with sharp features," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 544–552, 2005.

[8] A. C. Öztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 493–501, 2009.

[9] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[10] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 176:1–176:7, 2009.

[11] B. Liao, C. Xiao, L. Jin, and H. Fu, "Efficient feature-preserving local projection operator for geometry reconstruction," *Comput. Aided Design*, vol. 45, pp. 861–874, 2013.

[12] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *ACM Trans. Graph.*, vol. 22, no. 3. ACM, 2003, pp. 950–953.

[13] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," in *ACM Trans. Graph.*, vol. 22, no. 3. ACM, 2003, pp. 943–949.

[14] Y. Zheng, H. Fu, O.-C. Au, and C.-L. Tai, "Bilateral normal filtering for mesh denoising," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 10, pp. 1521–1530, 2011.

[15] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Fast and effective feature-preserving mesh denoising," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 5, pp. 925–938, 2007.

[16] T. R. Jones, F. Durand, and M. Zwicker, "Normal improvement for point rendering," *IEEE Comput. Graph. Appl.*, vol. 24, no. 4, pp. 53–56, 2004.

[17] R. Pajarola, "Stream-processing points," in *VIS '05*, 2005, pp. 239–246.

[18] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *SIGGRAPH '92*, 1992, pp. 71–78.

[19] A. Boulch and R. Marlet, "Fast and robust normal estimation for point clouds with sharp features," *Comp. Graph. Forum*, vol. 31, no. 5, pp. 1765–1774, 2012.

[20] J. Wang, X. Zhang, and Z. Yu, "A cascaded approach for feature-preserving surface mesh denoising," *Comput. Aided Design*, pp. 597–610, 2012.

[21] M. Pauly, R. Keiser, and M. Gross, "Multi-scale feature extraction on point-sampled surfaces," in *Computer graphics forum*, vol. 22, no. 3. Wiley Online Library, 2003, pp. 281–289.

[22] N. J. Mitra, A. Nguyen, and L. Guibas, "Estimating surface normals in noisy point cloud data," *Inter. Journal of Comput. Geo. & App.*, vol. 14, no. 04, pp. 261–276, 2004.