

# GPF: GMM-inspired Feature-preserving Point Set Filtering

Xuequan Lu, Shihao Wu, Honghua Chen, Sai-Kit Yeung, *Member, IEEE*, Wenzhi Chen, *Member, IEEE*, and Matthias Zwicker, *Member, IEEE*

**Abstract**—Point set filtering, which aims at reconstructing noise-free point sets from their corresponding noisy inputs, is a fundamental problem in 3D geometry processing. The main challenge of point set filtering is to preserve geometric features of the underlying geometry while at the same time removing the noise. State-of-the-art point set filtering methods still struggle with this issue: some are not designed to recover sharp features, and others cannot well preserve geometric features, especially fine-scale features. In this paper, we propose a novel approach for robust feature-preserving point set filtering, inspired by the Gaussian Mixture Model (GMM). Taking a noisy point set and its filtered normals as input, our method can robustly reconstruct a high-quality point set which is both noise-free and feature-preserving. Various experiments show that our approach can soundly outperform the selected state-of-the-art methods, in terms of both filtering quality and reconstruction accuracy.

**Index Terms**—GPF, Gaussian Mixture Model, point set filtering, feature preserving.

## 1 INTRODUCTION

POINT set filtering, which aims at reconstructing noise-free point sets from their corresponding noisy inputs, is a fundamental problem in 3D geometry processing. The filtered point set models can serve as a basis for a variety of further geometry processing, animation, and rendering applications, such as 3D reconstruction and point-based rendering. In fact, the design of robust point set filtering methods has become increasingly important with the growing availability of various scanning equipments, especially the consumer-grade depth sensors (e.g., Microsoft Kinect [1]). The main technical challenge of point set filtering is to robustly preserve geometric features while effectively removing noise.

State-of-the-art point set filtering methods, for example, LOP (locally optimal projection) [2], RIMLS (robust implicit moving least squares) [3], WLOP (weighted LOP) [4], EAR (edge-aware resampling) [5], and CLOP (continuous LOP) [6], have achieved noticeable successes recently. However, these point set filtering methods are either not designed for sharp feature preservation, or are less robust in removing noise while preserving features. Specifically, LOP [2], WLOP [4], and CLOP [6] are all LOP-based methods, which have been shown to remove noise and outliers

very robustly. But none of them is designed to preserve sharp features, due to their inherent isotropic nature. In other words, they do not take geometric features into account. EAR [5] is an extended LOP-based method that does take geometric features into account. It may smear fine-scale geometric features, however, since it needs to utilize a rather big neighborhood size, which acts like a smoothing kernel, to push points away from prominent edges. RIMLS [3] also considers feature preservation, but due to the strong dependence on the initial normal estimation it is generally more sensitive to outliers and noise compared to LOP-based methods. The above issues substantially limit the robustness and effectiveness of these methods in point set filtering.

Motivated by the above challenges, in this paper we present a GMM-inspired feature-preserving point set filtering (abbreviated as *GPF*) method which is a projection technique. The Gaussian Mixture Model, or GMM, is known to be a powerful statistical probability model [7], [8], [9]. By introducing normal information, our proposed anisotropic projection method is robust and effective in filtering point sets. Given a noisy point set and its filtered normals, our projection method can robustly generate a corresponding noise-free point set, with geometric features automatically preserved. The main contributions of our work are:

- a robust and effective method that preserves geometric features;
- a GMM-inspired formulation leading to an algorithm that is simple to implement.

Through a variety of experiments, we show that our method can significantly outperform the state-of-the-art techniques [3], [4], [5], [6], in terms of both filtering quality and reconstruction accuracy (Section 5), as illustrated in Figure 1. Furthermore, we discuss the key differences between our method and other selected techniques, as well as the limitations of our approach (Section 6).

## 2 RELATED WORK

In this section, we review previous research that is most related with our work. We first review point set filtering methods, fol-

- X. Lu and W. Chen\* are with the College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang Province, P. R. China. W. Chen is the corresponding author.  
E-mails: xuequanlu@zju.edu.cn, chenwz@zju.edu.cn
- S. Wu is with the Institute of Computer Science, University of Bern, Bern, Switzerland.  
E-mail: wu@inf.unibe.ch
- H. Chen is with Nanjing Normal University, Nanjing, Jiangsu Province, P. R. China.  
E-mail: chenhonghuacn@gmail.com
- S.K. Yeung is with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore.  
E-mail: saikit@sutd.edu.sg
- M. Zwicker is with the Department of Computer Science, University of Maryland, College Park, Maryland, USA.  
E-mail: zwicker@cs.umd.edu

Manuscript received September 15th, 2016; revised on November 26th, 2016 and March 14th, 2017; accepted on July 4th, 2017.

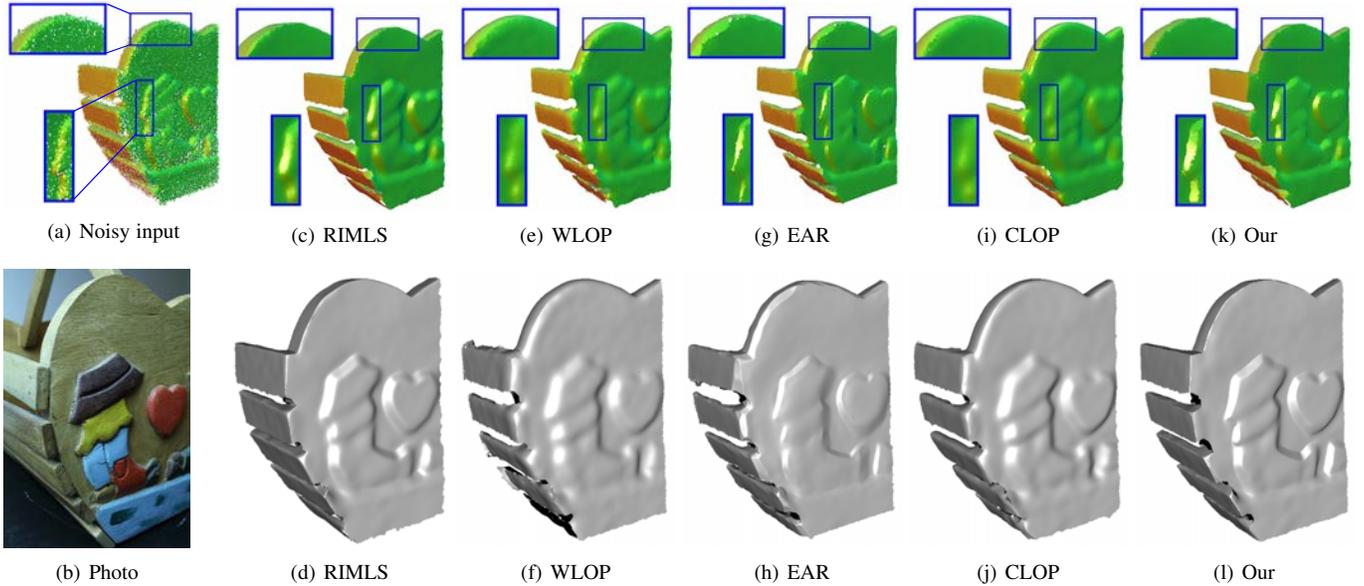


Fig. 1. Filtered results of the raw Basket point set. The zoomed regions highlight how our approach better preserves geometric features while robustly removing noise compared to previous techniques. We show corresponding surface reconstruction results in the bottom row.

lowed by surface reconstruction techniques. Finally, we review some related work that also employed the Gaussian Mixture Model (GMM).

## 2.1 Point Set Filtering

Given a point set, a core problem that has been studied widely in the context of point set surfaces is how to define a projection of the input points onto the underlying surface. Levin [10], [11] introduced the seminal moving least squares (MLS) formulations. To smooth noisy point clouds, pioneering work [12], [13], [14] defined moving least squares (MLS) and extremal surfaces. Later, Lange et al. [15] put forward an anisotropic smoothing method for point sets. Fleishman et al. presented a statistics-based moving least squares technique to reconstruct piecewise smooth point set surfaces [16]. Öztireli et al. [3] proposed robust implicit moving least squares (RIMLS) for feature-preserving point set projection.

More recently, promising point set projection methods [4], [6] based on the LOP (locally optimal projection) operator [2] have emerged. LOP [2] is parameterization free, i.e., it does not rely on estimating a local normal, fitting a local plane, or using any other local parametric representation. However, LOP may fail to converge, and cannot distribute points uniformly under significant non-uniformity of the input points. As a consequence, a weighted LOP was proposed [4]. Avron et al. introduced a  $L_1$ -optimization framework for reconstruction of sharp point set surfaces [17]. Later, a Kernel LOP was presented to decrease the computational cost of LOP [18], but the result quality deteriorates quickly as the number of kernels is reduced. Preiner et al. [6] developed an accelerated variant of LOP, called CLOP, which reformulates the data term to be a continuous representation of the input point set. Sun et al. proposed a method for filtering point sets via  $L_0$  minimization [19], but a post-processing step is required to solve the cross artifact problem around regions with sharp edges. Wu et al. formulated point set consolidation, skeletonization, and completion into a unified framework [20].

## 2.2 Surface Reconstruction

Surface reconstruction has been extensively studied for more than twenty years. As a seminal work [21], Hoppe et al. proposed an implicit model via estimating tangent planes. In 1998, a new Voronoi-based surface reconstruction algorithm was proposed by Amenta et al. [22]. In the same year, RBF [23] and Delaunay-based Cocone [24] were both proposed. More recently, popular surface reconstruction methods are Poisson Reconstruction [25], [26], APSS [27] and RIMLS [3]. Most of the methods, such as Cocone, RBF, Poisson, and APSS [23], [24], [25], [26], [27], are edge-oblivious, while RIMLS [3] is edge-aware. Recently, a scale space meshing method for raw data point sets was developed by Digne et al. [28]. More recently, Xiong et al. introduced a unified framework for surface reconstruction via dictionary learning [29]. Interested readers are referred to [30], [31] for a comprehensive review on this topic. Most of the surface reconstruction methods rely on correct normal information and point locations. Hence, a point set filtering step is commonly applied before the reconstruction.

## 2.3 Gaussian Mixture Model

The Gaussian Mixture Model (GMM) has been broadly used in a variety of research fields, for example, speaker identification/verification [32], [33], background subtraction [34], rigid and non-rigid point set registration [7], [35], [36], pose estimation [8], detection of compound structures [37], compressive sensing [9], and so on. Inspired mainly by Myronenko et al. [7], we extend GMM to point set filtering. The key idea is to relate the input points to the projected points via the GMM. The Expectation-Maximization (EM) algorithm [38] is utilized to solve this projection problem, in an iterative way.

## 3 METHOD OVERVIEW

The main component of our work is the GMM-inspired feature-preserving point set filtering (*GPF*). Before applying *GPF*, we smooth normals of the input point set with the bilateral filter [5].

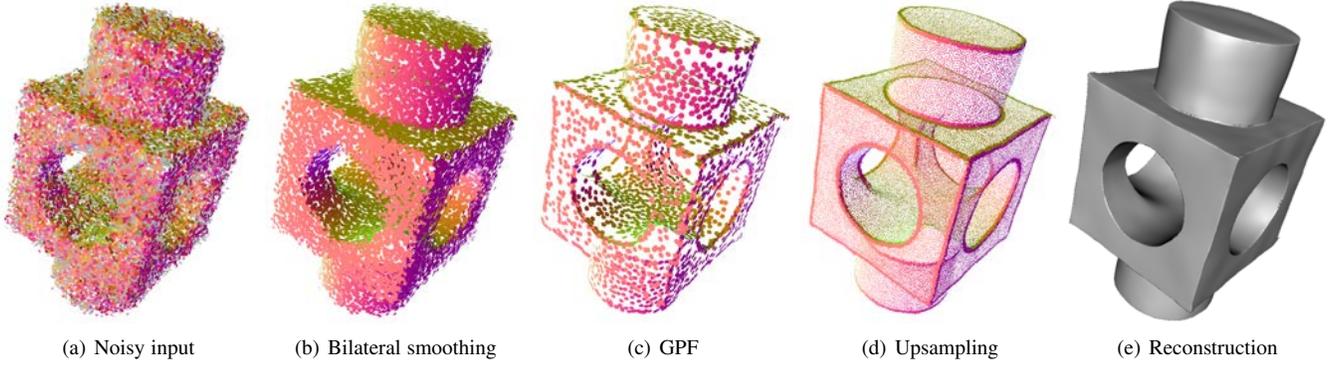


Fig. 2. An overview of our pipeline involving bilateral filtering (b), our proposed filtering approach GPF (c), upsampling (d) and surface reconstruction (e). We utilize bilateral normal filtering [5] before the proposed GPF, and further upsample the projected point set defined by GPF and reconstruct surfaces [3].

To further enhance point set rendering and surface reconstruction, we utilize a point set upsampling algorithm [5] in a post-process to increase the point density of the output of *GPF*. Interested readers are referred to the work by Huang et al. [5] for bilateral smoothing (Section 4) and upsampling (Section 5). Figure 2 shows an example of our pipeline involving bilateral filtering, *GPF*, upsampling and reconstruction.

Many state-of-the-art point set projection methods [2], [4], [6] are not designed for preserving sharp features by nature. We present *GPF* that accounts for point set filtering and feature preservation simultaneously. Given a noisy point set and its filtered normals, *GPF* projects points onto the underlying surface. Features in the projected point set can be automatically preserved (e.g., Figure 2), by considering the filtered normal information during projection.

## 4 GMM-INSPIRED FEATURE-PRESERVING PROJECTION

In this section we introduce our *GPF* approach by formulating the projection problem using a GMM. Then we explain how to perform the projection using an EM algorithm, and we introduce energy terms that allow us to preserve geometric features and obtain a uniform point distribution on the surface.

### 4.1 The Probabilistic Framework

Let  $V = \{v_1, v_2, \dots, v_N\}$  be the input, which is an unorganized, noisy point set. Given  $V$ ,  $X = \{x_1, x_2, \dots, x_M\}$  is a set of points projected onto the underlying surface, which is defined by our *GPF* model. We initialize  $X$  by randomly downsampling points from  $V$ . Inspired by Myronenko et al. [7], we assume the distribution of the noisy input points  $V$  follows a GMM, which is defined by a set of centroids and covariances, and we view  $X$  as the centroids of the GMM. The goal is now to determine the parameters of the GMM, i.e., the centroids and covariances, that best explain the noisy input data  $V$ . In other words, we need to maximize the likelihood that  $V$  is generated by the GMM by optimizing over the parameters of the GMM, i.e., the centroids  $X$  and covariances. Under the GMM model, we formulate the probability density for an input point  $v_n \in V$  as

$$p(v_n) = (1 - \omega) \sum_{m=1}^M \frac{1}{M} p(v_n | x_m) + \omega \frac{1}{N}, \quad (1)$$

where  $p(v_n | x_m) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{\|v_n - x_m\|^2}{2\sigma^2}}$  is the  $m$ -th Gaussian component, and  $d$  is the dimensionality of the point set ( $d = 3$  in this work).  $v_n$  and  $x_m$  are row vectors ( $1 \times 3$ ). Similar to Myronenko et al. [7], we use equal isotropic covariances  $\sigma^2$  for all components and equal membership probabilities, represented by the factor  $\frac{1}{M}$  above. Finally, we include a uniform distribution in the model (i.e.,  $p = \frac{1}{N}$ ), which accounts for noise and outliers, and weight it with  $\omega \in [0, 1]$ . The likelihood function of our model is  $\mathcal{L}(X, \sigma^2) = \prod_{n=1}^N p(v_n)$ .

The GMM centroid locations  $X$  and the covariance  $\sigma^2$  can be estimated by maximizing the likelihood, or by minimizing the negative log-likelihood function,

$$E(X, \sigma^2) = - \sum_{n=1}^N \log \left( \frac{1 - \omega}{M} \sum_{m=1}^M p(v_n | x_m) + \frac{\omega}{N} \right). \quad (2)$$

### 4.2 EM Optimization

In order to find  $X$  and  $\sigma^2$ , we use the EM (i.e., Expectation-Maximization) algorithm [38] to minimize Eq.2. Given the  $X$  and  $\sigma^2$  values, the E-step is to compute the posterior probability of the mixture components using Bayes' theorem. Based on the posterior probability, the M-step is to find the new  $X$  and  $\sigma^2$  values by minimizing the expectation of the complete negative log-likelihood function [39]. The E-step and M-step are alternately performed until convergence, or a maximum number of iterations.

**E-step.** In this step, the posterior probabilities  $p^{old}(x_m | v_n)$  are computed using "old values" estimated from the last iteration according to Bayes' theorem,

$$p^{old}(x_m | v_n) = \frac{e^{-\frac{\|v_n - x_m^{old}\|^2}{2\sigma^2}}}{\sum_{m'=1}^M e^{-\frac{\|v_n - x_{m'}^{old}\|^2}{2\sigma^2}} + \frac{(2\pi\sigma^2)^{d/2} \omega M}{(1-\omega)N}}. \quad (3)$$

**M-step.** At the M-step, we estimate the parameters ( $X$  and  $\sigma^2$ ) by minimizing the upper bound of Eq. (2), shown as follows.

$$Q(X, \sigma^2) = D(X, \sigma^2) + \frac{dN_p}{2} \log \sigma^2, \quad (4)$$

where  $N_p = \sum_{n=1}^N \sum_{m=1}^M p_{mn}$ ,  $p_{mn} = p^{old}(x_m | v_n)$ , and  $D(X, \sigma^2) = \frac{1}{2\sigma^2} \sum_{n=1}^N \sum_{m=1}^M p_{mn} \|v_n - x_m\|^2$ . Generally,  $D(X, \sigma^2)$  is called a *data term*, which pushes the projected points  $x_m$  to be close to the observed data  $v_n$ , and hence to approximate the underlying geometry.

### 4.3 Feature-preserving Data Term and Repulsion Term

**Data term.** Similar to [2], [4], [6], the data term  $\|v_n - x_m\|^2$  cannot preserve sharp features during projection. To address this problem, we introduce a modified data term that includes the filtered normal information as a key component of our approach. That is, we replace  $\|v_n - x_m\|^2$  with  $\|(v_n - x_m)b_n^T\|^2$ , where  $b_n$  is the normal vector ( $1 \times 3$ ) of the  $n$ -th point in  $V$ , and  $T$  is its transpose,

$$D(X, \sigma^2) = \frac{1}{2\sigma^2} \sum_{n=1}^N \sum_{m=1}^M p_{mn} \|(v_n - x_m)b_n^T\|^2. \quad (5)$$

Intuitively, this means we replace the point-to-point distance with a point-to-plane distance, where the plane is given by the filtered normal at each point. This encourages projected points  $x_m$  to stay close to local tangent planes, and it avoids the smoothing out of geometric features.

**Repulsion term.** Similar to existing projection operators [2], [4], [6], we also introduce a repulsive force to Eq. 4, striving for an even distribution of the projected points (i.e.,  $X$ ). Like in our data term, we also introduce normal information to the repulsion term to keep the projected points on the local tangent planes. We achieve this by projecting the repulsive forces onto the tangent planes. Hence, our repulsion forces are based on the difference vector between close-by points  $x_m$  and  $x_{m'}$ , projected onto the tangent plane given by the normal  $b'_m$  of  $x'_m$ . This leads to the repulsion term

$$R(X) = \sum_{m=1}^M \lambda_m \sum_{m'=1, m' \neq m}^M \eta(r_{mm'}) \theta(r_{mm'}), \quad (6)$$

where  $r_{mm'} = \|(x_m - x_{m'}) - (x_m - x_{m'})b_{m'}^T b_{m'}\|$  is the distance vector  $x_m - x_{m'}$  projected onto the tangent plane defined by the point-normal pair  $(x_{m'}, b_{m'})$ . Similar to Huang et al. [4], our final repulsion force consists of a term  $\eta(r) = -r$  and a smoothly decaying weight function  $\theta(r) = e^{(-r^2)/(h/2)^2}$ .  $h$  is the support radius which should be adjusted according to different point set models.  $h$  is normally increased with the growing noise level.  $h$  needs to be decreased when there are close-by surfaces in the point set, for example the block model in Figure 8. In our experiments,  $h$  is set to be in the range  $[3h_0, 12h_0]$  where  $h_0 = d_{bb}/\sqrt{N}$  and  $d_{bb}$  is the diagonal length of the bounding box of the input point set.

Thus, our final optimization objective for the M-step is based on Eq. 4, but with our modified data and novel repulsion terms. By slight abuse of notation, we redefine

$$Q(X, \sigma^2) = D(X, \sigma^2) + R(X) + \frac{dN_p}{2} \log \sigma^2. \quad (7)$$

Note that the point-to-plane distance is not used in the E-step, because we found using it would not make a difference visually.

### 4.4 Minimization

We now describe how to find the projected points  $X$  and covariance  $\sigma$  in the M-step as  $\arg \min_{X, \sigma^2} Q(X, \sigma^2)$ . The partial derivative of Eq. 7 with respect to  $x_m$  is

$$\frac{\partial Q}{\partial x_m} = \frac{1}{\sigma^2} \sum_{n=1}^N p_{mn} (x_m - v_n) b_n^T b_n - \lambda_m \sum_{m'=1, m' \neq m}^M \beta_{mm'} (x_m - x_{m'}) C_{mm'}^T C_{mm'}, \quad (8)$$

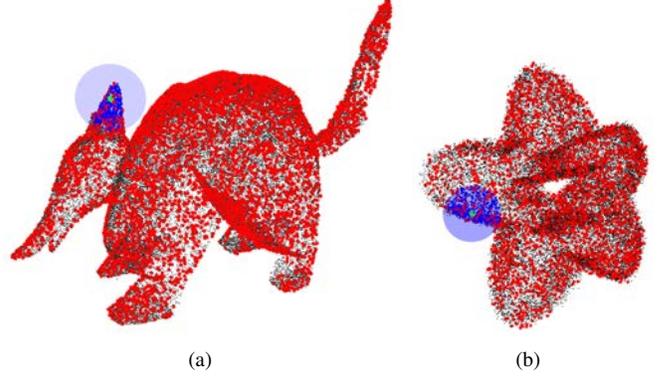


Fig. 3. Two examples illustrating the local notion in GPF. The input noisy points and the projected points are colored with black and red, respectively. The green point indicates a sample point with its local neighbors in the original point set, which we take into account during our optimization steps, colored in blue.

where  $\beta_{mm'} = \frac{\theta(r_{mm'})}{r_{mm'}} \left| \frac{\partial \eta(r_{mm'})}{\partial r} \right|$ ,  $C_{mm'} = I - b_{m'}^T b_{m'}$ , and  $I$  is the  $3 \times 3$  identity matrix. It is not difficult to prove that  $C_{mm'}^T = C_{mm'}$  and further  $C_{mm'} C_{mm'}^T = C_{mm'}$ .

As shown in Eq. 8, the coefficients of  $x_m$  in the derivatives of the data and repulsion terms are matrices rather than scalars. This is quite different from LOP and WLOP [2], [4]. Therefore, we apply gradient descent to find  $X$ , which is iterative and fast. In each gradient descent step we update  $x_m$  using

$$x_m^{k+1} = x_m^k + \frac{\sum_{n=1}^N p_{mn}^k (v_n - x_m^k) b_n^T b_n}{\sum_{n=1}^N p_{mn}^k} + \mu (\sigma^2)^k \frac{\sum_{m'=1, m' \neq m}^M \beta_{mm'}^k (x_m^k - x_{m'}^k) C_{mm'}^k}{\sum_{m'=1, m' \neq m}^M \beta_{mm'}^k}, \quad (9)$$

where  $k+1$  and  $k$  represents the  $(k+1)$ -th and  $k$ -th gradient descent iteration, respectively,  $(\sigma^2)^k$  means  $\sigma^2$  at the  $k$ -th iteration, and we scaled the gradient from Eq. 8 by  $(\sigma^2)^k / \sum_{n=1}^N p_{mn}^k$  to adjust the gradient descent step size. In addition, we introduced the notation  $\mu = \lambda_m \frac{\sum_{m'=1, m' \neq m}^M \beta_{mm'}}{\sum_{n=1}^N p_{mn}}$ . In practice, we replace  $\mu (\sigma^2)^k$  with a user parameter  $\hat{\mu}$ , which allows us to manually control the relative strength of the repulsion term. We set  $\hat{\mu}$  in the range  $[0.05, 0.30]$  in our experiments.

In addition, we also take density weights into account, as suggested by Huang et al. [4]. The fixed point iteration for  $x_m^{k+1}$  finally becomes

$$x_m^{k+1} = x_m^k + \frac{\sum_{n=1}^N p_{mn}^k / \rho_n (v_n - x_m^k) b_n^T b_n}{\sum_{n=1}^N p_{mn}^k / \rho_n} + \hat{\mu} \frac{\sum_{m'=1, m' \neq m}^M w_{m'}^k \beta_{mm'}^k (x_m^k - x_{m'}^k) C_{mm'}^k}{\sum_{m'=1, m' \neq m}^M w_{m'}^k \beta_{mm'}^k}, \quad (10)$$

where the density weights are  $\rho_n = 1 + \sum_{n'=1, n' \neq n}^N \theta(\|v_n - v_{n'}\|)$ , and  $w_{m'}^k = 1 + \sum_{m''=1, m'' \neq m'}^M \theta(\|x_m^k - x_{m''}^k\|)$ .

To compute  $\sigma^2$ , we take the partial derivative of Eq. 7 with respect to  $\sigma^2$ . Hence, we obtain  $\sigma^2$  at the  $(k+1)$ -th iteration as

$$(\sigma^2)^{k+1} = \frac{1}{dN_p} \sum_{n=1}^N \sum_{m=1}^M p_{mn}^k \|(v_n - x_m^{k+1}) b_n^T\|^2, \quad (11)$$

Finally, we summarize the proposed GMM-inspired feature-preserving point set filtering (GPF) method in Algorithm 1.

**Algorithm 1** GMM-inspired feature-preserving point set filtering**Input:** noisy point set  $V$ , filtered normals  $\{b_n\}$ **Output:** projected point set  $X$ 

- 1: Initialize:  $\omega = 0.02$ ,  $\sigma^2 = 0.004$ ,  $h \in [3h_0, 12h_0]$ ,  $\hat{\mu} \in [0.05, 0.30]$ , downsample  $V$  to  $X^0$
- 2: **repeat**
- 3:   E-step: update  $\{p_{mn}\}$  via Eq. 3
- 4:   M-step: update  $X$  via Eq.10 and  $\sigma^2$  via Eq.11
- 5: **until** convergent OR maximum iterations are reached
- 6: **return**  $X$

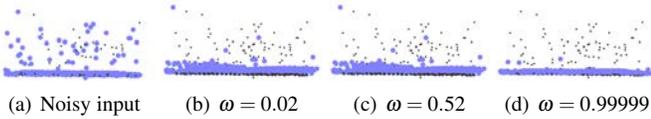


Fig. 4. A test of different outlier weights  $\omega$  on a plane (side view) corrupted with large outliers. Larger  $\omega$  better suppress outliers.

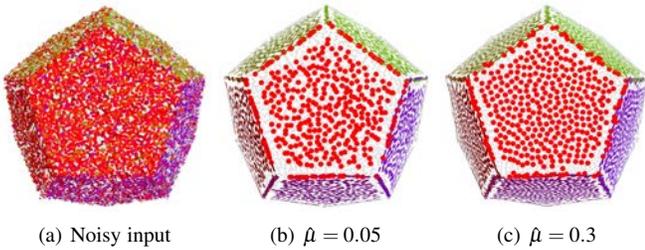


Fig. 5. A comparison for different repulsion strengths  $\hat{\mu}$ . A relatively larger  $\hat{\mu}$  leads to a more even distribution of projected points.

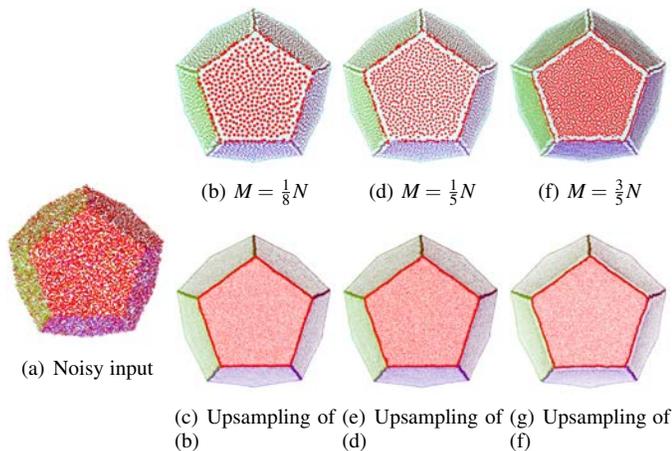


Fig. 6. A comparison for different numbers of downsampled points  $M$ . The corresponding upsampling results are shown in the bottom.

Similar to the strategy used in [8], we used a fixed initial value for  $\sigma^2$  (0.004) since we found no obvious artifacts compared with the initialization using the input points [7].

In practice, GPF is implemented in a local sense as illustrated in Figure 3. A local implementation is necessary because the original method involves massive memory and computational cost (e.g., computation for the  $M \times N$  matrix  $\{p_{mn}\}$ ). Specifically, for the data term, we set a ball radius ( $h/2$ ) to obtain the neighbors in  $V$  of each point in  $X$  as well as the neighbors in  $X$  of each point in  $V$ . For the repulsion term, we set a larger ball radius ( $h$  in Eq. (6)) to find the neighbors in  $X$  of each point in  $X$ . A larger

radius for the repulsion term is to alleviate the point aggregation issue and generate a more uniform points distribution. This local implementation scheme significantly reduces the computational and memory cost.

We illustrate the influence of the GPF parameters (outlier weight  $\omega$ , repulsion strength  $\hat{\mu}$ , number of downsampled points  $M$ ) in Figures 4, 5, and 6. We tested the effect of using different  $\omega$  on a plane model corrupted with strong outliers (Figure 4), and found a great  $\omega$  value (approaching 1) is helpful in dealing with significant outliers. This is mainly because the update variations of  $\sigma^2$  are not large, and a large  $\omega$  is thus needed to account for a significant value of  $\frac{(2\pi\sigma^2)^{d/2}\omega M}{(1-\omega)^N}$  (Eq. (3)). In general, we found  $\omega = 0.02$  works well in our experiments (Section 5) where outliers are less prominent compared to the particular example in Figure 4. The parameter  $\hat{\mu}$  controls the relative power of repulsion forces. Figure 5 shows a comparison of using different  $\hat{\mu}$ . Figure 6 demonstrates that different numbers (typically  $< N$  as suggested in [2], [4]) of the downsampled points would generate very similar upsampling results in the end. In the experiments (Section 5), we randomly select about half of the original points as the downsampled points.

## 5 RESULTS

We compared both our approach and the selected state-of-the-art methods (i.e., RIMLS [3], WLOP [4], EAR [5], CLOP [6]) on a wide range of synthetic and raw scans. Both visual and quantitative comparison results are presented. Please refer to the supplemental material for more results.

**Experiment setup.** For rendering purposes, the output of all five methods are upsampled [5] and further reconstructed via the feature preserving surface reconstruction method (i.e., RIMLS [3] in Meshlab). For the sake of fair comparison and visualization, we adopt the following principles. For each comparative experiment, (1) we carefully tuned the main parameters of each method to achieve best visual results; (2) all five methods used approximately the same numbers of downsampled (upsampled) points; (3) identical surface reconstruction parameters are employed; (4) we use PCA normal estimation to acquire the same initial normals for all five methods. Regarding normal-related methods, RIMLS [3], EAR [5] and GPF employ the bilateral filter [5] by setting the same parameters. WLOP and CLOP [4], [6] are independent of normals; (5) back-face culling is used for point set rendering.

**Synthetic point scans.** Synthetic point scans are corrupted with a certain amount of noise, for example, 1.0% of the diagonal length of the bounding box. As shown in Figures 7, 8 and 9, our approach yields substantially better results than the selected state-of-the-art methods, in terms of feature preservation. Note that RIMLS [3] and EAR [5] can also preserve features to some extent, but they produce less pleasing results, for example, obvious flaws in feature regions (e.g., 8(b)), remarkably curved edges (e.g., 8(f)), uncleanness (i.e., unrobustness to outliers, e.g., 9(g)). Comparing to the selected methods, the proposed GPF produces the best result even when handling a high level of noise (Figure 7).

**Raw point scans.** Besides the synthetic point clouds, we also tested the five methods on raw scans data. We observed, from Figures 1, 10 and 11, that our method can produce nicer results than the state-of-the-art methods, in terms of fine-features preservation. Figure 12 shows that the final result of our method is generally more desirable than the other methods, regarding to

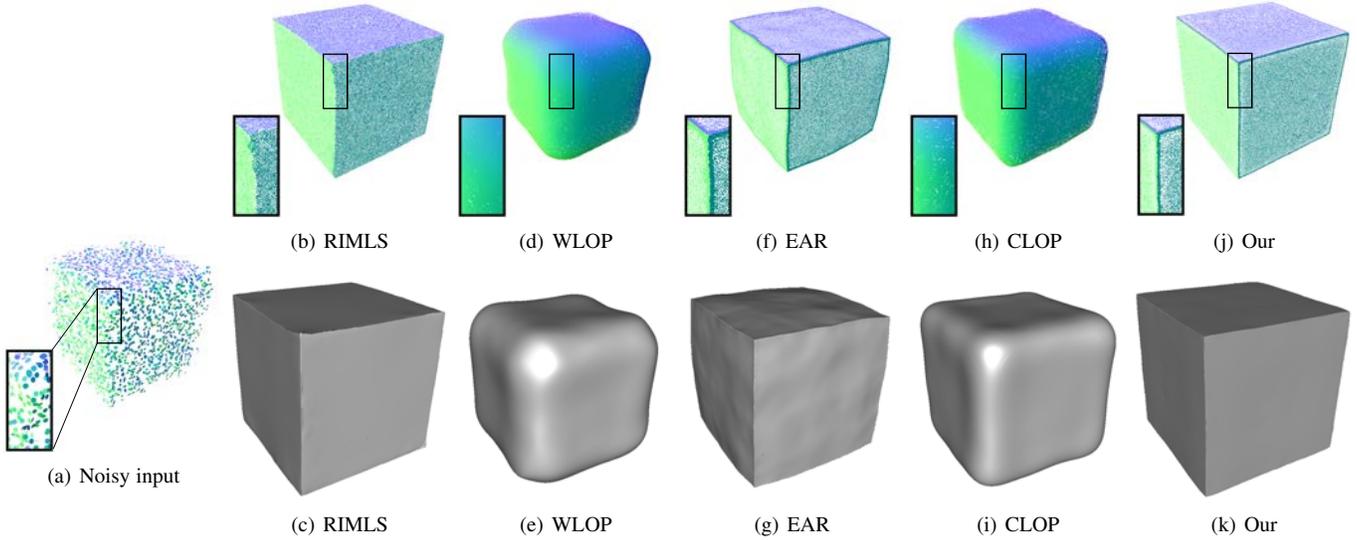


Fig. 7. Filtered results of the Cube point set (3.0% noise).

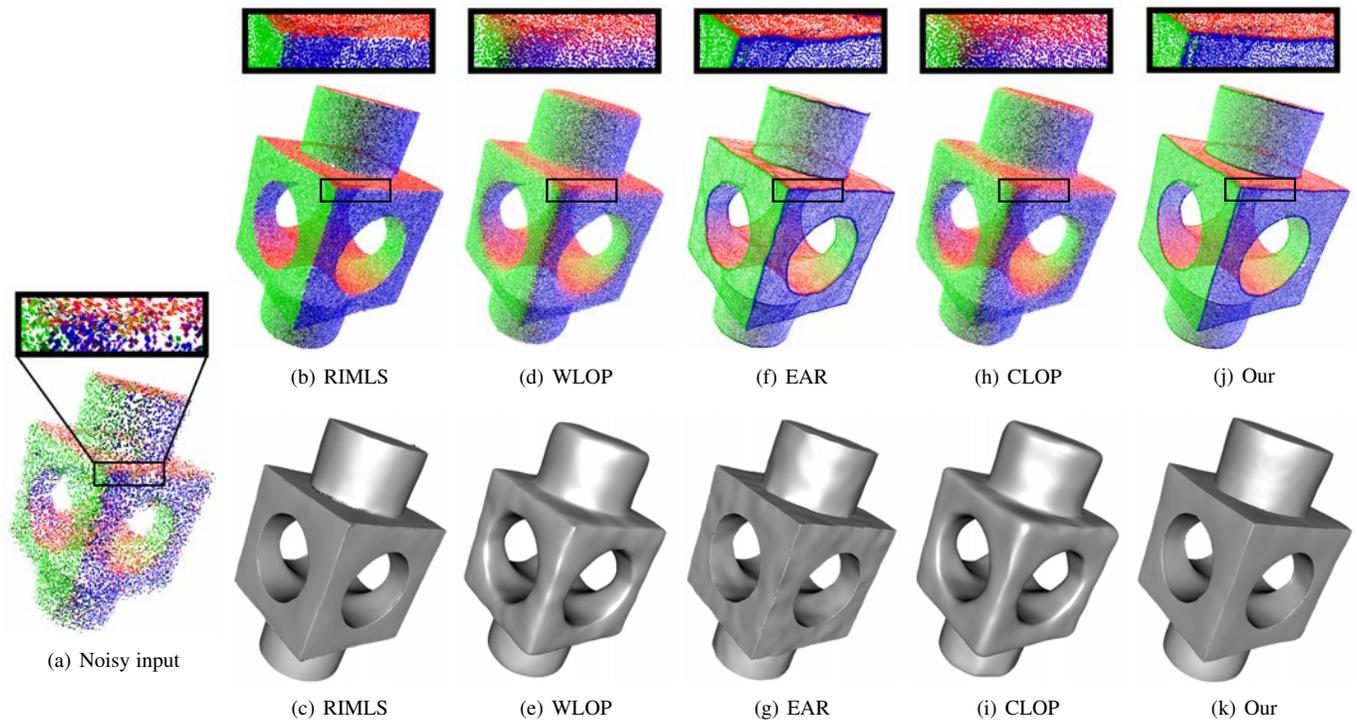


Fig. 8. Filtered results of the Block point set (1.5% noise). The corresponding surface reconstruction results are shown in the bottom. Please refer to the zoomed regions.

the sharpness of the geometric features and the smoothness of the planar regions.

**Quantitative statistics.** In addition to the above visual comparisons, two quantitative evaluations are carried out.

**Accuracy.** We compare the reconstruction accuracy of all five methods on two point sets. Specifically, we use digital virtual scans of synthetic models (ground truth) to evaluate the reconstruction accuracy [20]. The error is measured using the average distance between points of the ground truth and their closest points of the reconstructed point set. As shown in Figures 13, 14 and 15, the reconstructions based on the output of GPF is the most accurate one among the five methods.

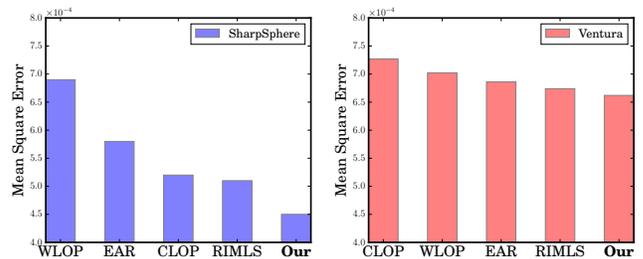


Fig. 15. The mean square errors of the reconstructed models (Figures 13 and 14) for all five methods.

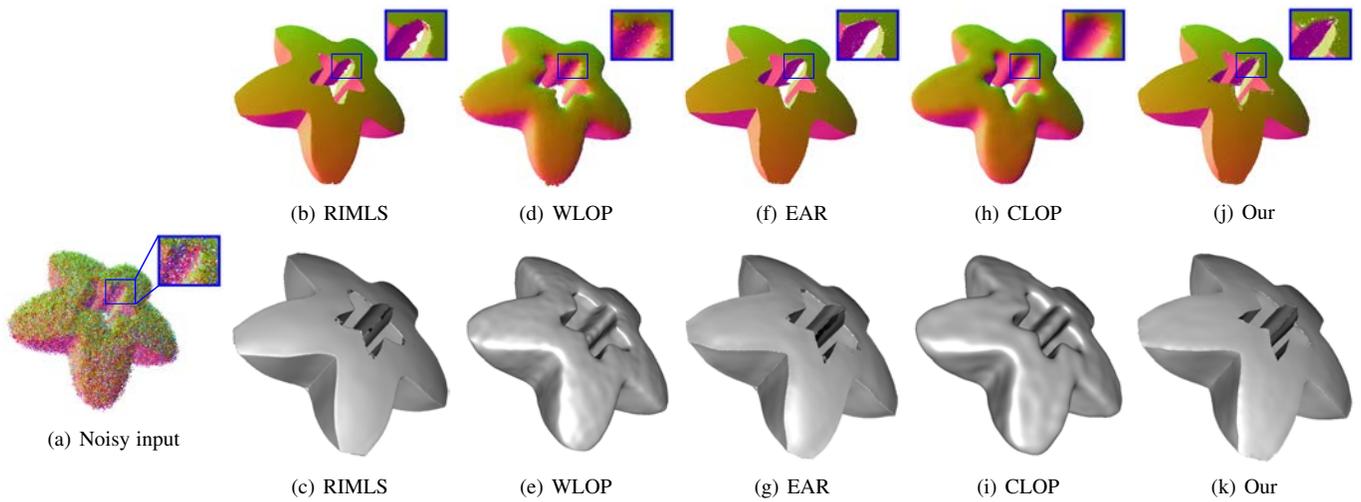


Fig. 9. Filtered results of the Trim-star point set (1.0% noise). The corresponding surface reconstruction results are shown in the bottom. Please refer to the zoomed regions to see the improvements provided by GPF more clearly.

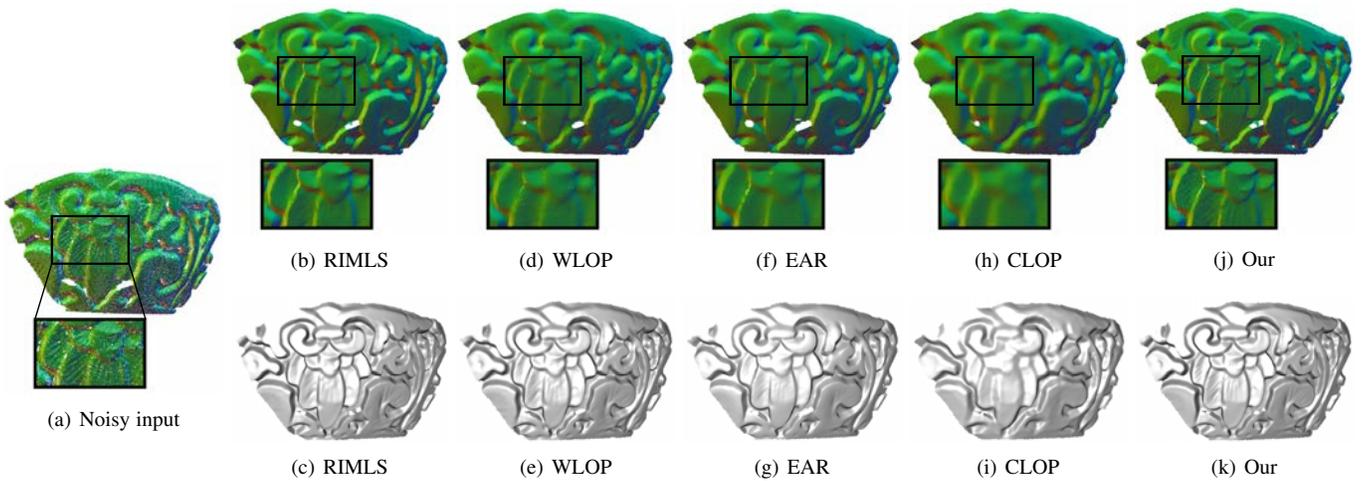


Fig. 10. Filtered results of the raw embossed point set. The corresponding surface reconstruction results are shown in the bottom.

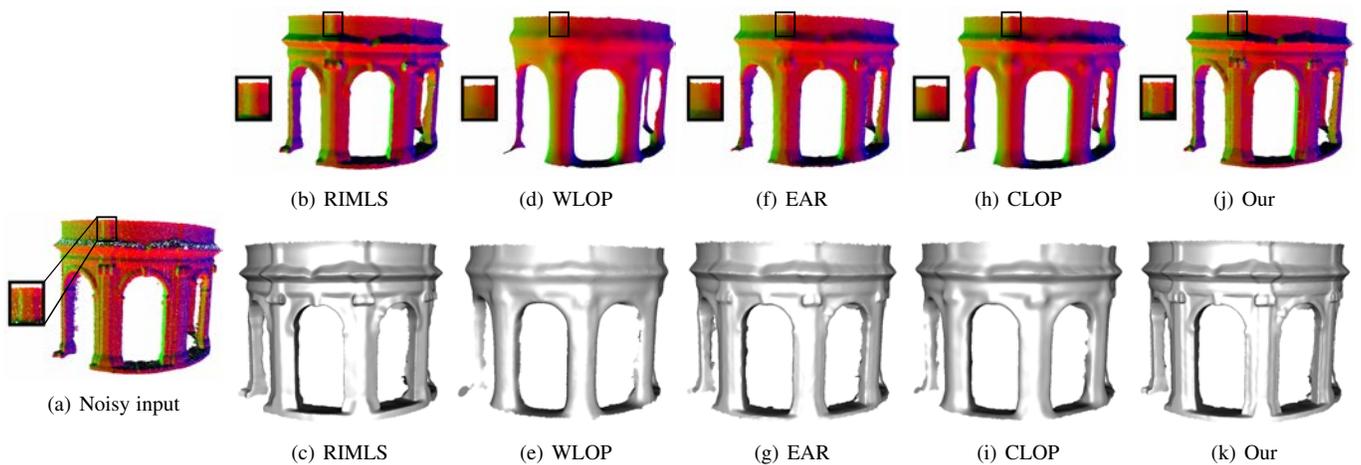


Fig. 11. Filtered results of the raw Saint-jean point set. The corresponding surface reconstruction results are shown in the bottom. Please refer to the zoomed regions.

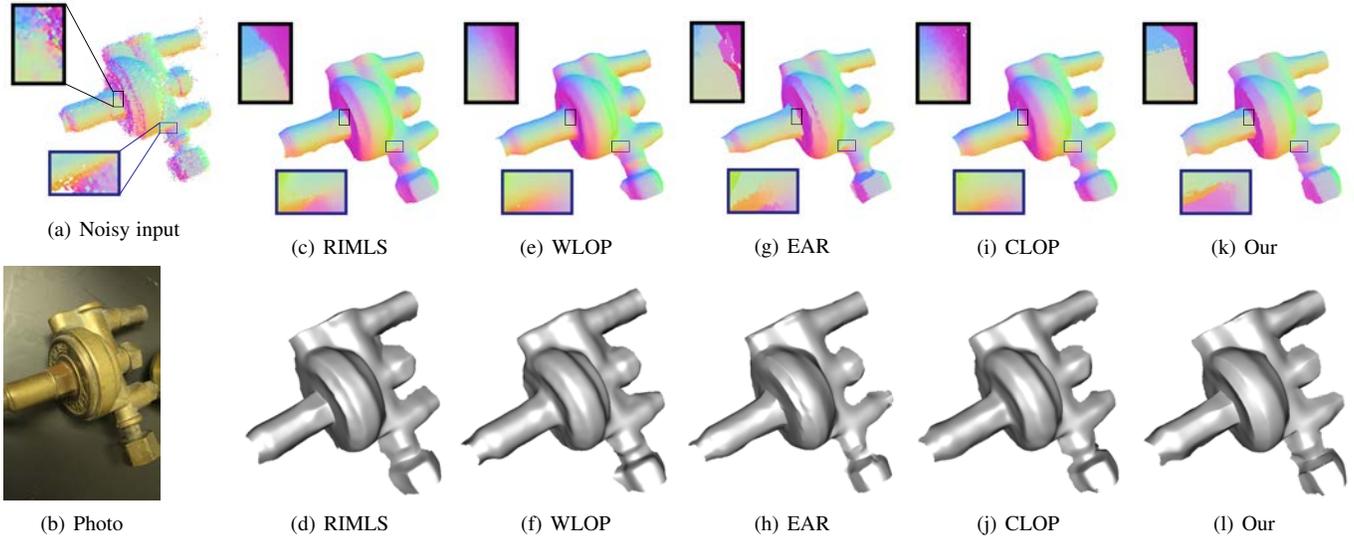


Fig. 12. Filtered results of the raw Water-Meter point set. The corresponding surface reconstruction results are shown in the bottom.

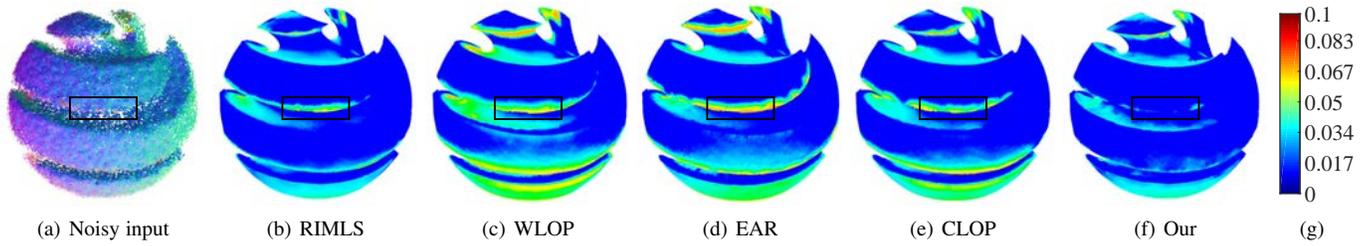


Fig. 13. Reconstruction accuracies of the raw Sphere point set. (g) is the error bar.

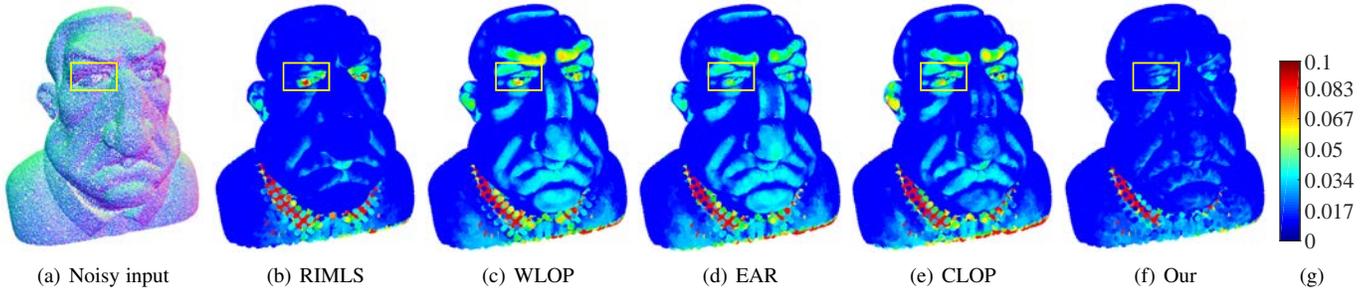


Fig. 14. Reconstruction accuracies of the raw Ventura point set. (g) is the error bar.

TABLE 1

Timing statistics (in seconds) for different approaches.  $N$  and  $M$  are the numbers of input points and downsampled points, respectively (Section 4). All examples were run on the same computer that equipped with an Intel(R) Core(TM) Q8400 2.66-GHz CPU.

Models	Figure 8 $N$ : 33490 $M$ : 17K	Figure 12 $N$ : 50629 $M$ : 23K	Figure 9 $N$ : 75053 $M$ : 35K	Figure 10 $N$ : 48305 $M$ : 27k
RIMLS	18.22	161.73	60.54	16.87
WLOP	52.56	260.08	100.87	17.14
EAR	54.9	198.34	148.34	24.22
CLOP	7.72	13.15	25.13	3.78
Our	16.69	150.37	65.38	17.34

*Timing.* For fair comparisons, we accumulate the running time of PCA normal estimation, bilateral normal smoothing and the

*GPF* as the total running time of our work. We run RIMLS, WLOP, EAR and CLOP on the same machine and report their running time. We can see from Table 1 that our method is generally faster than WLOP and EAR, but slower than CLOP.

## 6 DISCUSSION

Besides the above comparative results (Section 5), it is essential to discuss the key differences between our method and the four selected state-of-the-art methods in more detail.

**GPF vs RIMLS.** RIMLS [3], which considers feature preservation, is an advantageous combination of MLS and local kernel regression. It encodes a low pass filter and can naturally deal with noise. Our GPF is inspired by the GMM and incorporates normal information to automatically preserve geometric features. In fact, RIMLS is mainly designed for surface reconstruction and does not consider the non-uniformity issue of projected points (Figure 16).

**GPF vs EAR.** EAR [5] is a resampling algorithm, which consists of two steps: resampling away from edges and edge-aware upsampling. Excluding normal smoothing and upsampling, there exist significant differences between GPF and ALOP (i.e., anisotropic LOP) which plays a key role in the first step of EAR. Regarding to the formulation differences, ALOP modified the spatial weight function of the data term by using the filtered normals; while our GPF is GMM-inspired and incorporates filtered normal information in a very different manner. We use point-to-plane distance in our data term and project the repulsive forces onto the local tangent plane. In terms of results, ALOP leads to gaps around sharp edges and then fills the gaps via upsampling; the proposed GPF attracts points to edges, leading to high sampling density around the sharp feature areas. Furthermore, EAR requires running WLOP as a pre-process, and needs to use a rather big neighborhood size for ALOP to push points away from edges. As a result, EAR may smear fine-scale features.

**GPF vs LOP/WLOP.** WLOP [4] is a weighted version of LOP [2], which augments the regularity of points distribution. Similar to LOP, it contains two terms: data and repulsion term. The data term locally approximates the underlying geometry, and the repulsion term strives for an even distribution of the projected points. The main differences between our method and LOP/WLOP [2], [4] are: (1) LOP and WLOP is based on the  $L_1$ -median concept while we assume the input point cloud  $V$  follows a GMM and derive a different iterative projection process from the GMM-inspired formulation; (2) LOP and WLOP only utilize positional information while our technique introduces both positional and normal information; (3) LOP and WLOP are isotropic so that they are not designed to preserve sharp features, while our method can automatically preserve geometric features.

**GPF vs CLOP.** CLOP [6], which reformulates the data term to be a continuous representation of the input point cloud, is technically a variant of LOP [2]. Even though both CLOP [6] and GPF are inspired by GMM, there still exist major distinctions between them: (1) CLOP uses a hierarchical expectation-maximization (HEM) algorithm to compute the Gaussians which are used to represent the density of the input points (the filtering quality and accuracy would possibly deteriorate with increasing hierarchy levels due to the agglomerative nature of HEM), while we directly apply EM on the input points; (2) CLOP only tries to adopt a geometric regularization when merging clusters in HEM and does not consider the geometric features in the input points, while GPF preserves features by introducing normal information into both the data and repulsion terms; (3) CLOP reformulates the attraction force by fitting the generated Gaussians into the data term form of WLOP [4], while the proposed GPF uses a

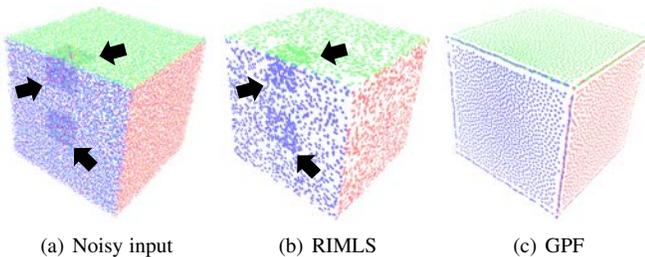


Fig. 16. The non-uniformity issue. Black arrows indicate denser regions. Note that the gap in our result can be alleviated in the limitation discussion (Figure 21).

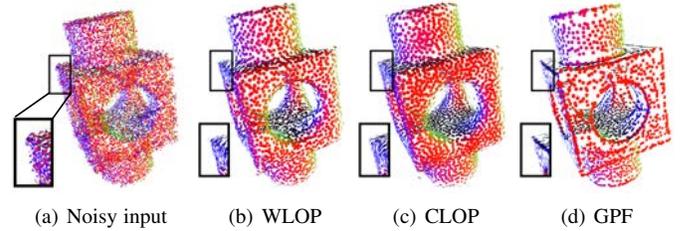


Fig. 17. Projected results of WLOP, CLOP and our GPF.

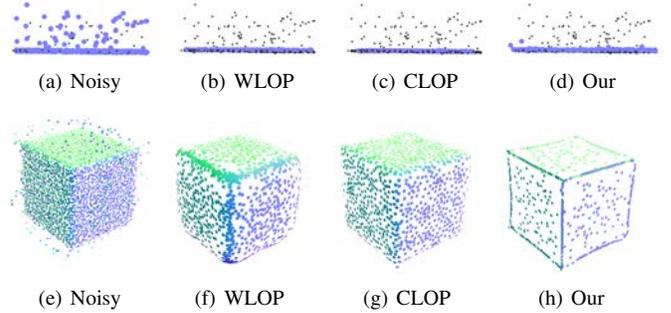


Fig. 18. Sensitivity to strong outliers of LOP-based methods (e.g., WLOP and CLOP) and the proposed GPF. The first row: the original and sampled points are colored in black and blue, respectively. The second row: a shape with sharp edges. For all three methods, big radii are used to deal with strong outliers.

GMM-inspired formulation and ends up with a different iterative projection operator by assuming the input point set follows a GMM.

Figure 17, which compares the projection results of WLOP [4], CLOP [6] and the proposed GPF, shows that our method can well preserve geometric features. Note that the LOP-based techniques [2], [4], [6] are insensitive to outliers in the data due to the  $L_1$ -norm formulation. We tested the outlier sensitivity of WLOP, CLOP and our approach, shown in Figure 18. The proposed GPF can also deal with heavy outliers, mainly due to the weighted uniform distribution term  $\frac{\rho}{N}$  (Eq. (1)) accounting for outliers and noise. Regarding the plane example (the first row in Figure 18), our method is a bit more sensitive than WLOP and CLOP, mainly since there are no sharp edges which can automatically attract points (including outliers).

Note that our method aims at filtering: removing noise while preserving geometric features including discontinuities (e.g., sharp edges in the cube model). Some sampling techniques (e.g., spectral sampling [40]) are specifically designed for resampling rather than filtering. It is similar to the upsampling step in EAR [5], and can be potentially used as the upsampling step after our GPF step (Figure 2). We found the upsampling technique in EAR is adequate for our experiments. In this work, we also compared our method with a number of state-of-the-art approaches which are designed for filtering. The surface reconstruction results by RIMLS could be possibly improved by sampling techniques like spectral sampling.

Despite the demonstrated performance improvements of our approach over the state-of-the-art methods, it still has several limitations.

(i) Similar to EAR [5], we employ the bilateral normal filter [5] to obtain smoothed normals because of its effectiveness and efficiency. However, GPF may produce less than satisfactory results when handling point scans that are contaminated by severe

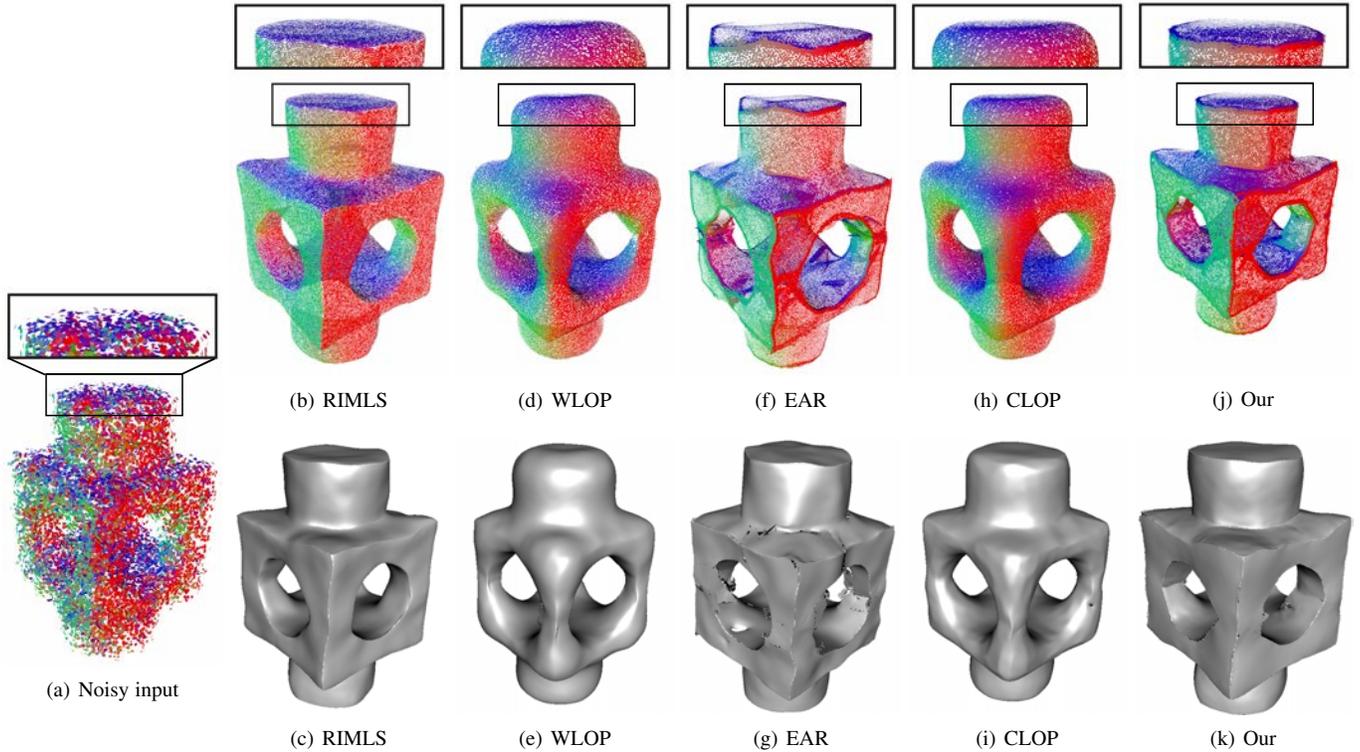


Fig. 19. Filtered results of the point set corrupted with heavy noise (3.0% noise). The corresponding surface reconstruction results are shown in the bottom. Please refer to the zoomed regions. The mean square errors of these five methods (from left to right) are  $2.1 \times 10^{-4}$ ,  $8 \times 10^{-4}$ ,  $4.1 \times 10^{-4}$ ,  $5.7 \times 10^{-4}$  and  $1.8 \times 10^{-4}$ , respectively.

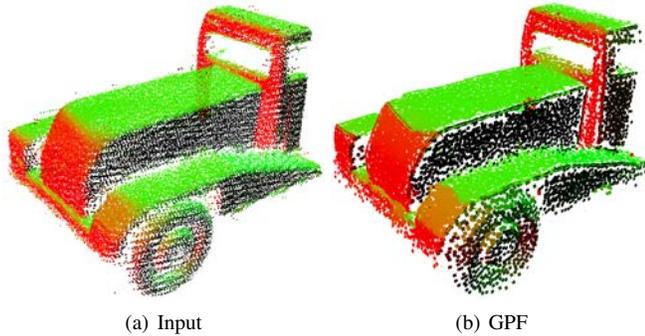


Fig. 20. A failure example of our approach when handling a point scan with a number of open boundaries.

noise, as a result of highly inaccurate normals estimation. An example is shown in Figure 19. Despite the fact that the reconstructed result by GPF is far from perfect, GPF still visually and quantitatively performs the best among all five methods when the filtered normal field is quite unreliable. Some promising normal estimation methods [41], [42] have been proposed recently. A more effective and accurate normal estimation technique could certainly improve the filtering quality of GPF, which we consider to be beyond the scope of this work.

(ii) Similar to WLOP [4], EAR [5] and CLOP [6], GPF is not designed for handling open boundaries and inputs with large missing data, and it may generate undesired results in these situations. Such a failure example is shown in Figure 20. Similar to many existing methods (e.g., [5]), automatically finding optimal parameters for different inputs still remains a challenge. In the

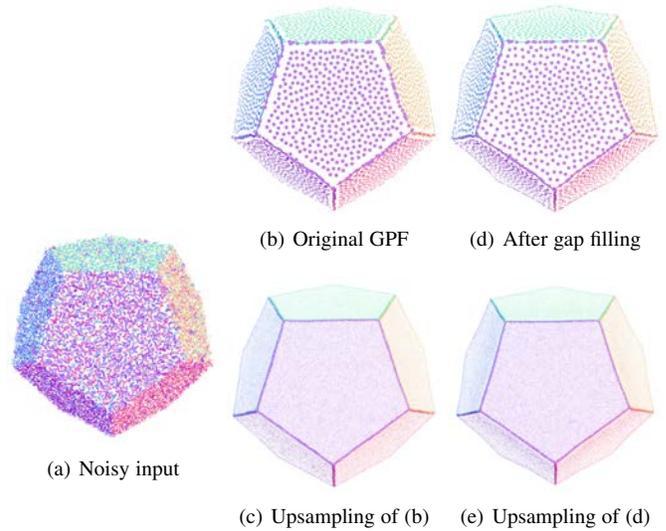


Fig. 21. Alleviating the gap issue near edges. The corresponding upsampling results shown at the bottom, however, are very similar with and without the extra step to fill gaps near edges.

proposed GPF, the main parameters that need tuning are the support radius  $h$  and the repulsion weight  $\hat{\mu}$ .

(iii) The proposed GPF may produce gaps near the edges, after points are attracted to and accumulated around edges. In our experiments, however, we found that such “gap regions” do not affect the final upsampling and surface reconstruction results. Still, one way to alleviate the gap issue is to run several iterations of GPF with a rather big neighborhood after the original GPF, which

encourages points to move to the gap regions along the underlying surface. Figure 21 shows such an example which demonstrates very similar upsampling results of GPF points with or without gaps.

## 7 CONCLUSIONS

In this paper, we presented a GMM-inspired locally optimal projection approach for feature-preserving point set filtering (i.e., *GPF*). Taking a noisy and outlier-ridden point cloud and its filtered normals as input, our approach automatically and robustly preserves geometric features while removing noise. Various experimental results show the robustness of our method in filtering both synthetic and raw point scans. In this work, we focus on point set filtering, which strives at obtaining a projected point set that can approximate the underlying geometry, preserve sharp features and details, and produce a relatively uniform points distribution. For future work, it would be interesting to incorporate hole filling and boundary handling techniques into our point set filtering framework. Feature-preserving mesh denoising techniques (e.g., [43], [44]) can be further combined with our point set filtering framework, to achieve desired mesh representations.

## ACKNOWLEDGMENTS

This work is supported in part by Singapore MOE Academic Research Fund MOE2013-T2-1-159 and SUTD-MIT International Design Center Grant IDG31300106. We acknowledge the in-part support of the SUTD Digital Manufacturing and Design (DManD) Centre which is supported by the National Research Foundation (NRF) of Singapore. This research is also supported in part by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative.

## REFERENCES

- [1] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultiMedia*, vol. 19, no. 2, pp. 4–10, 2012.
- [2] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007. [Online]. Available: <http://doi.acm.org/10.1145/1276377.1276405>
- [3] A. C. Öztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," *Computer Graphics Forum*, vol. 28, no. 2, pp. 493–501, 2009. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2009.01388.x>
- [4] 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, Dec. 2009. [Online]. Available: <http://doi.acm.org/10.1145/1618452.1618522>
- [5] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. R. Zhang, "Edge-aware point set resampling," *ACM Trans. Graph.*, vol. 32, no. 1, pp. 9:1–9:12, Feb. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2421636.2421645>
- [6] R. Preiner, O. Mattausch, M. Arikian, R. Pajarola, and M. Wimmer, "Continuous projection for fast I1 reconstruction," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 47:1–47:13, Jul. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2601097.2601172>
- [7] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, pp. 2262–2275, Dec 2010.
- [8] M. Ye and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 2353–2360.
- [9] J. Yang, X. Liao, X. Yuan, P. Llull, D. J. Brady, G. Sapiro, and L. Carin, "Compressive sensing by learning a gaussian mixture model from measurements," *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 106–119, 2015.
- [10] D. Levin, "The approximation power of moving least-squares," *Math. Comput.*, vol. 67, no. 224, pp. 1517–1531, Oct. 1998. [Online]. Available: <http://dx.doi.org/10.1090/S0025-5718-98-00974-0>
- [11] D. Levin, *Mesh-Independent Surface Interpolation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 37–49. [Online]. Available: [http://dx.doi.org/10.1007/978-3-662-07443-5\\_3](http://dx.doi.org/10.1007/978-3-662-07443-5_3)
- [12] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Point set surfaces," in *Proceedings of the Conference on Visualization '01*, ser. VIS '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 21–28. [Online]. Available: <http://dl.acm.org/citation.cfm?id=601671.601673>
- [13] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 9, no. 1, pp. 3–15, 2003.
- [14] N. Amenta and Y. J. Kil, "Defining point-set surfaces," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 264–270, Aug. 2004. [Online]. Available: <http://doi.acm.org/10.1145/1015706.1015713>
- [15] C. Lange and K. Polthier, "Anisotropic smoothing of point sets," *Computer Aided Geometric Design*, vol. 22, no. 7, pp. 680 – 692, 2005, geometric Modelling and Differential Geometry. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167839605000750>
- [16] 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, Jul. 2005. [Online]. Available: <http://doi.acm.org/10.1145/1073204.1073227>
- [17] H. Avron, A. Sharf, C. Greif, and D. Cohen-Or, "L1-sparse reconstruction of sharp point set surfaces," *ACM Trans. Graph.*, vol. 29, no. 5, pp. 135:1–135:12, Nov. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1857907.1857911>
- [18] B. Liao, C. Xiao, L. Jin, and H. Fu, "Efficient feature-preserving local projection operator for geometry reconstruction," *Computer-Aided Design*, vol. 45, no. 5, pp. 861 – 874, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0010448513000341>
- [19] Y. Sun, S. Schaefer, and W. Wang, "Denoising point sets via  $l_0$  minimization," *Computer Aided Geometric Design*, vol. 35-36, pp. 2 – 15, 2015, geometric Modeling and Processing 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167839615000345>
- [20] S. Wu, H. Huang, M. Gong, M. Zwicker, and D. Cohen-Or, "Deep points consolidation," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 176:1–176:13, Oct. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2816795.2818073>
- [21] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, Jul. 1992. [Online]. Available: <http://doi.acm.org/10.1145/142920.134011>
- [22] N. Amenta and M. Bern, "Surface reconstruction by voronoi filtering," in *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, ser. SCG '98. New York, NY, USA: ACM, 1998, pp. 39–48. [Online]. Available: <http://doi.acm.org/10.1145/276884.276889>
- [23] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 67–76. [Online]. Available: <http://doi.acm.org/10.1145/383259.383266>
- [24] T. K. Dey and J. Giesen, "Detecting undersampling in surface reconstruction," in *Proceedings of the Seventeenth Annual Symposium on Computational Geometry*, ser. SCG '01. New York, NY, USA: ACM, 2001, pp. 257–263. [Online]. Available: <http://doi.acm.org/10.1145/378583.378682>
- [25] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, ser. SGP '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 61–70. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1281957.1281965>
- [26] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, Jul. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2487228.2487237>
- [27] G. Guennebaud and M. Gross, "Algebraic point set surfaces," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007. [Online]. Available: <http://doi.acm.org/10.1145/1276377.1276406>
- [28] J. Digne, J.-M. Morel, C.-M. Souzani, and C. Lartigue, "Scale space meshing of raw data point sets," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1630–1642, 2011. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2011.01848.x>
- [29] S. Xiong, J. Zhang, J. Zheng, J. Cai, and L. Liu, "Robust surface reconstruction via dictionary learning," *ACM Trans. Graph.*,

vol. 33, no. 6, pp. 201:1–201:12, Nov. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2661229.2661263>

- [30] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva, “A benchmark for surface reconstruction,” *ACM Trans. Graph.*, vol. 32, no. 2, pp. 20:1–20:17, Apr. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2451236.2451246>
- [31] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, and C. T. Silva, “State of the Art in Surface Reconstruction from Point Clouds,” in *Eurographics 2014 - State of the Art Reports*, S. Lefebvre and M. Spagnuolo, Eds. The Eurographics Association, 2014.
- [32] D. Reynolds, R. C. Rose *et al.*, “Robust text-independent speaker identification using gaussian mixture speaker models,” *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.
- [33] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.
- [34] Z. Zivkovic, “Improved adaptive gaussian mixture model for background subtraction,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 28–31.
- [35] A. Myronenko, X. Song, and M. A. Carreira-Perpinán, “Non-rigid point set registration: Coherent point drift,” in *Advances in Neural Information Processing Systems*, 2006, pp. 1009–1016.
- [36] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt, “Algorithms for 3d shape scanning with a depth camera,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 5, pp. 1039–1050, May 2013.
- [37] C. Ari and S. Aksoy, “Detection of compound structures using a gaussian mixture model with spectral and spatial constraints,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 10, pp. 6627–6638, 2014.
- [38] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.
- [39] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.
- [40] A. C. Öztireli, M. Alexa, and M. Gross, “Spectral sampling of manifolds,” *ACM Trans. Graph.*, vol. 29, no. 6, pp. 168:1–168:8, Dec. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1882261.1866190>
- [41] X. Liu, J. Zhang, J. Cao, B. Li, and L. Liu, “Quality point cloud normal estimation by guided least squares representation,” *Computers & Graphics*, vol. 51, pp. 106 – 116, 2015, international Conference Shape Modeling International. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0097849315000710>
- [42] A. Boulch and R. Marlet, “Deep learning for robust normal estimation in unstructured point clouds,” *Computer Graphics Forum*, vol. 35, no. 5, pp. 281–290, 2016. [Online]. Available: <http://dx.doi.org/10.1111/cgf.12983>
- [43] X. Lu, Z. Deng, and W. Chen, “A robust scheme for feature-preserving mesh denoising,” *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 3, pp. 1181–1194, 2016.
- [44] L. He and S. Schaefer, “Mesh denoising via l0 minimization,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 64:1–64:8, Jul. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2461912.2461965>



**Shihao Wu** is a Ph.D. candidate in the Computer Graphics Group in the University of Bern. He received his M.S. degree in South China University of Technology, and B.S. degree in the South China Normal University. His research interests include computer graphics, geometric modeling, and point set processing.



**Honghua Chen** is currently a M.S. candidate in the College of Education Science at Nanjing Normal University, China. His research interests include geometry processing and motion tracking.



**Sai-Kit Yeung** is currently an Assistant Professor at the Singapore University of Technology and Design (SUTD), where he leads the Vision, Graphics and Computational Design (VGD) Group. He received his PhD in Electronic and Computer Engineering from the Hong Kong University of Science and Technology (HKUST) in 2009. He also received his BEng degree (First Class Honors) in Computer Engineering and MPhil degree in Bioengineering from HKUST in 2003 and 2005 respectively. His current research focus is primarily on 3D content reconstruction, understanding, creation, redesign and fabrication.



**Wenzhi Chen** was born in 1969, received his Ph.D. degree from Zhejiang University, Hangzhou, China. He is now a Professor and Ph.D. supervisor of college of Computer Science and Technology of Zhejiang University. His areas of research include computer graphics, computer architecture, system software, embedded system and security.



**Xuequan Lu** earned his Ph.D. in the College of Computer Science and Technology at Zhejiang University (China) in June 2016. He earned his B.S. degree in Computer Science from Northwest A&F University (China). His research interests include computer graphics and vision, mainly focused on vision-based geometry processing, motion tracking and physics-based simulation/animation.



**Matthias Zwicker** is the Reginald Allan Hahne Endowed E-Nnovate Professor in Computer Science at the University of Maryland. He obtained his PhD from ETH in Zurich, Switzerland, in 2003. From 2003 to 2006 he was a post-doctoral associate with the computer graphics group at the Massachusetts Institute of Technology, and then held a position as an Assistant Professor at the University of California in San Diego from 2006 to 2008. From 2008-2017, he was a professor in Computer Science at the University of Bern, Switzerland, and the head of the Computer Graphics Group. His research focus is on efficient high-quality rendering, signal processing techniques for computer graphics, data-driven modeling and animation, and point-based methods.