This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2827998, IEEE Transactions on Visualization and Computer Graphics

Multi-Normal Estimation via Pair Consistency Voting

Jie Zhang§, Junjie Cao§, Xiuping Liu, He Chen, Bo Li, Ligang Liu.

Abstract—The normals of feature points, *i.e.*, the intersection points of multiple smooth surfaces, are ambiguous and undefined. This paper presents a unified definition for point cloud normals of feature and non-feature points, which allows feature points to possess multiple normals. This definition facilitates several succeeding operations, such as feature points extraction and point cloud filtering. We also develop a feature preserving normal estimation method which outputs multiple normals per feature point. The core of the method is a pair consistency voting scheme. All neighbor point pairs vote for the local tangent plane. Each vote takes the fitting residuals of the pair of points and their preliminary normal consistency into consideration. Thus the pairs from the same subspace and relatively far off features dominate the voting. An adaptive strategy is designed to overcome sampling anisotropy. In addition, we introduce an error measure compatible with traditional normal estimators, and present the first benchmark for normal estimation, composed of 152 synthesized data with various features and sampling densities, and 288 real scans with different noise levels. Comprehensive and quantitative experiments show that our method generates faithful feature preserving normals and outperforms previous cutting edge normal estimation methods, including the latest deep learning based method.

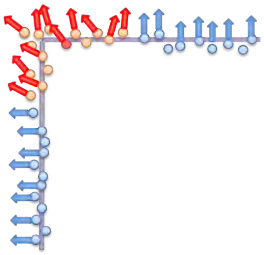
Index Terms—Normal estimation, Point clouds, Sharp feature, Subspace, Sampling anisotropy.

1 Introduction

Normals are important geometric properties of a point cloud. They are heavily used in many analyzing and processing works, such as point based rendering [1], [2], surface reconstruction [3], anisotropic smoothing [4], segmentation [5], and feature detection and extraction [6], [7], which renders normal estimation a long term research topic. But fast and reliable normal estimation is still an intriguing challenge since sharp features, noise, and sampling anisotropy are inevitably contained in acquired point clouds. Moreover, the absence of the definition of point cloud normals is a long existent issue of normal estimation.

It is well known that the points are sampled from a piecewise manifold and the normal of a point lying on the interior of a smooth surface patch is the normal of the surface naturally. But the normals of points on the intersections of different surface patches are ambiguous and undefined. In this paper, we present a unified definition of point cloud normals for these two cases. We claim that a point may have multiple normals, each of which is determined by one surface patch where the point is lying. A multi-normal estimation algorithm (PCV-MN) is also proposed. The definition provides at least two benefits. First, extracting multiple normals for feature points may enhance some succeeding applications as exhibited in Section 8. Moreover, according to the definition, we improve the Root Mean Square measure (RMS_τ) introduced in [8]. Thus the errors of estimated normals, especially for the feature points, could be analyzed quantitatively.

For each edge/corner point, our PCV-MN outputs a set of normals describing all the underlying surfaces at the point. It is implemented by iterating a pair consistency voting (PCV) method. PCV is designed to extract the normal of the surface patch which takes the largest area among the patches sharing the point in the neighborhood. For a point, we collect a set of candidate planes, each of which is determined by three randomly picked neighbors. The current point's neighbors vote for each plane and the plane with the most votes wins.



The challenge is how to design a robust and accurate voting strategy since geometry properties of scanned points around sharp edges are unreliable as shown in the wrapped figure. As specified by [11], [12], in the neighborhood of a feature point, the neighbors relatively far from features are more stable and reliable despite being far from the current point. We solve the challenge by accumulating the votes from all the neighbor point pairs, instead of neighbor points. The vote of a point pair takes the fitting residuals and the consistency of two points' preliminary normals into consideration. The planes are encouraged to fit the blue neighbors which share similar preliminary normals as shown in the wrapped figure. We also adapt the voting weights and the amounts of candidate planes according to local sampling density to extract the tangent plane robustly for anisotropic sampling. As illustrated in Fig. 1, the proposed method is capable of preserving sharp

- *J. Zhang and J. Cao contributed equally.*
- *J. Zhang is with the School of Mathematics, Liaoning Normal University, Dalian, China, 116029. E-mail: jzhang@lnnu.edu.cn*
- *J. Cao, X. Liu and H. Chen are with School of Mathematical Sciences, Dalian University of Technology, Dalian, China, 116024. E-mail: jjcao1231@gmail.com, xpLiu@dlut.edu.cn, and ankachan@126.com*
- *B. Li is with College of Mathematics and Information Science, Nanchang Hangkong University, Nanchang, China, 330063. E-mail: libo@nchu.edu.cn*
- *L. Liu is with School of Mathematical Sciences, University of Science and Technology of China, Hefei, China, 220026. E-mail: lgliu@ustc.edu.cn*

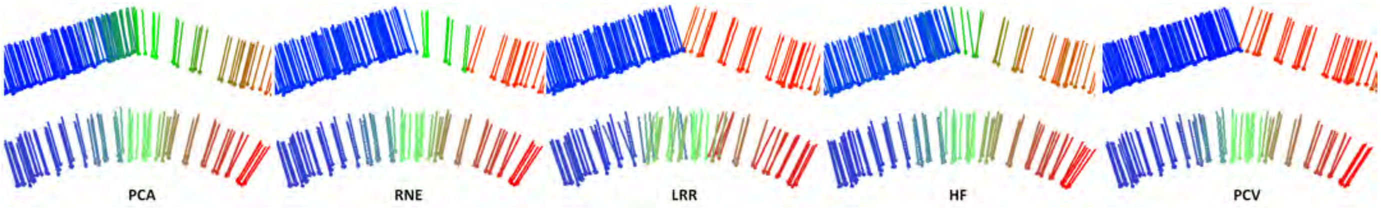


Fig. 1: Estimated normals of two planes with a shallow angle (top) and a smooth curved region (bottom). The results of Hoppe et al. [9] (PCA), Li et al. [10] (RNE), zhang et al. [11] (LRR), Boulch and Marlet [8] (HF) and our method (PCV) are shown from the first column to the last. The points are sampled non-uniformly in the top row. The points and normals are colored according to the normal directions.

features accurately, while generating correct normals for the points in the smooth curved regions.

Another factual issue of normal estimation is the absence of a benchmark for comparing existing methods fairly and thoroughly. We present a benchmark for normal estimation which assigns multiple normals per point. The benchmark includes 152 synthesized models and 288 real models. The synthesized models include point clouds with various sharp features, details, non-uniform sampling, small and big noise. The real models are sampled from reconstructed meshes scanned by Microsoft Kinect v1 and v2. They provide two different noise levels and resolutions. Comprehensive experiments on the benchmark show that our algorithm outperforms the other cutting edge methods.

The contributions of our work are summarized as follows:

1) A definition of point cloud normals is given, which benefits some succeeding applications. A novel error measure compatible with the definition is also formulated to evaluate various normal estimators quantitatively.
2) A voting approach based on pair consistency of preliminary normals is presented for normal estimation. By taking all neighbor pairs into consideration, abundant reliable pairs dominate the voting and guarantee the robustness and effectiveness of the algorithm.
3) We construct the first benchmark for normal estimation. The data, with various features, details, sampling densities, noise levels and resolutions, provides a sound and empirical basis for research on normal estimation.

2 Related work

Normal estimation is a challenging task for point clouds with noise, sharp features, and sampling anisotropy. There has been a considerable amount of works on the topic. They can be categorized into three types: Delaunay/Voronoi based, mollification based and regression based [10]. Delaunay/Voronoi based methods [13], [14], [15] only perform well for noise free data or point cloud with small noise. None of them can handle point clouds with outliers and sharp features [10]. Since the preliminary normals are likely to be noisy or smoothed, normal mollification methods [16], [17], [18], [19] are studied to improve the initial normals. However, these methods fail to recover sharp features, if the initial normals do not preserve sharp features sufficiently, as demonstrated by Boulch and Marlet [8]. The rest of this section focuses on the relevant regression based methods.

The most famous regression based normal estimation method is proposed by Hoppe *et al.* [9] (PCA). It estimates the normal of a point by fitting a local plane to all neighbors of it. The estimated normals tend to smooth sharp features, see Fig.1. To lower the contribution of distant neighbors in plane fitting, Mitra *et al.* [20] adapt the size of neighborhood by investigating the effect of local noise, curvature and sampling density. Pauly *et al.* [21] assign a distance weight to each neighbor. In addition to plane, quadric [22] and spherical surface [23] are fitted on the isotropic neighborhood. However the smooth effects are only alleviated rather than eliminated, since the neighborhood is isotropic and the neighbors from different surface patches still affect the fitting.

Ideally, the neighbors belonging to different surface patches should be discarded. For this, robust statistics approaches are employed to select a plane approximating the neighbors from the same surface patch as the current point [10], [24], [25], [26], [27]. Based on the observation that the neighbors with large fitting residuals are often from the wrong side of an edge, sharp features are handled by ignoring such neighbors during the plane fitting [10], [24]. Some weight functions are also introduced to favor the neighbors sharing similar properties with the current point, such as locations [26] or preliminary normals [27]. However, these methods do not address the variation of density at edges and would be biased towards the surface patches with higher point density, as shown in the top row of Fig.1. Wang *et al.* [28] choose the neighbors based on the observation that two points coming from the same substructure share similar residuals to the hypotheses planes. This method is robust to the non-uniform sampling. But all these methods suggest that the neighbors sharing similar properties with the current point, such as locations [26] or support hyperplanes [28], should offer higher credibility. However, when a point is extremely near a sharp feature, these properties are unreliable. To overcome this problem, we consider all the neighbor pairs.

By utilizing the subspace structures of the underlying piecewise surfaces, Zhang *et al.* [11] (LRR) segment the neighborhood of a point into several parts and a consistent one is selected for normal estimation, *i.e.*, only the neighbors located on the same surface patch with the current point

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2827998, IEEE Transactions on Visualization and Computer Graphics

will be considered. This method is capable of preserving sharp features even in the presence of noise and anisotropic samplings, but at the price of intolerable runtime which may take hours. To make it more practical, a fast version of LRR is proposed by Liu *et al.* [29] which generates quality normals as faithful as LRR, but with far less runtime. Unfortunately, LRR and its variant may lead to some erroneous normals in smooth curved regions because they compulsively split the whole neighborhood into several subneighborhoods, as illustrated in the bottom row of Fig.1. The proposed PCV can explore the neighborhood's subspace structure without explicit segmentation.

Recently, Boulch and Marlet [8] adopt a robust randomized Hough transform method with an uniform sampling strategy. However, its selecting process is subject to space discretization. To overcome this effect, they need to estimate a normal repeatedly at each point, randomly rotate the accumulator to change discretization boundaries. This process significantly increases the computation time and introduces additional parameters. Convolutional neural networks are introduced to overcome this issue [30]. However, the users have to retrain the network if they want to adapt specifically to the input data. Moreover, in the vicinity of sharp features, Hough transform based methods still generate some erroneous normals, see Fig.1.

3 Overview

The input of point cloud normal estimation is a point cloud $\mathcal{P} = \{p_i\}_{i=1}^{n} \subseteq \mathbb{R}^3$ sampled from a piecewise manifold composed of multiple surfaces $\{S_i\}_{i=1}^{m}$. Previous methods compute one normal for each point p_i. The normals of feature points, such as edge points and corner points, are undefined or ambiguous. Here we give a clear definition for point cloud normals.

Definition 1. The normals of each point p_i is a set of normals of the underlying surfaces at the point:

$$N_{p_i} = \{n_i^j \mid \textbf{if exit } r_j, \ \textbf{s.t. } p_i \in S_{r_j}\},$$

where n_i^j is the normal of S_{r_j} at p_i.

It means that if a point lies in the interior of one smooth surface it has one normal and if it belongs to multiple surfaces, *i.e.*, the intersection of these surfaces, it has multiple normals. In other words, edge points have two normals and corner points have multiple normals.

The definition can be applied to mesh models too. Each vertex is shared by multiple faces, *i.e.*, planes. Hence the normals of these planes are taken directly as the normals of each vertex. It facilitates the preparation of ground truth data when building the normal estimation benchmark since precise feature points labeling is tedious and error prone, especially for weak edges. Using triangular meshes as ground truth data, the labeling is avoided and novel error measures can be defined to evaluate existing normal estimators on the benchmark, which will be detailed in Section 6.1.

Following the new definition of point cloud normals, we propose a multi-normal estimation algorithm. For each point, it outputs one or multiple normals. First, we classify all points into the points near sharp features (*i.e.*, candidate points) and the points far from the features (*i.e.*, smooth points), see Section 4. Then we choose the classical PCA method to compute a preliminary normal for each point. The preliminary normals of the smooth points are reliable and it is only necessary to estimate the candidate points' normals. Given the neighborhood of one candidate point p_i, we design the pair consistency voting (PCV) method to estimate one element of N_{p_i}, which is detailed in Section 5. The normal set N_{p_i} is computed by iterating PCV, see Section 5.3.

4 Candidate points selection

We take the same operations as [11], [29] to estimate a preliminary normal for each point and classify all the points into smooth points and candidate points. To make our paper self-contained, we give a brief introduction as follows and details are referred to [11].

We first compute the coefficient w at every point via eigenvalue analysis of the local neighborhood. For each point p_i, the covariance matrix is defined as

$$\mathbf{T}_i = \frac{1}{S}\begin{bmatrix} p_i^1 - \hat{p}_i \\ \cdots \\ p_i^S - \hat{p}_i \end{bmatrix} \cdot \begin{bmatrix} p_i^1 - \hat{p}_i \\ \cdots \\ p_i^S - \hat{p}_i \end{bmatrix}^T, \ p_i^j \in \mathcal{N}_i, \qquad (1)$$

where $\mathcal{N}_i$ is the neighborhood of point p_i, S is the size of $\mathcal{N}_i$, $\hat{p}_i$ is the mean position of its neighbors. The initial normal n_i^0 computed by PCA is the eigenvector corresponding to the smallest eigenvalue λ_0 of $\mathbf{T}_i$. For the point p_i, the feature coefficient w_i is computed as

$$w_i = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2},$$

where $0 \le \lambda_0 \le \lambda_1 \le \lambda_2$ are the singular values of the covariance matrix. Then, we normalize w to $[0, 1]$.

The coefficient w_i measures how close point p_i is to sharp features. The coefficient of a point at\near sharp features is larger than that of a point in smooth regions. Therefore, we regard p_i as a candidate point, if w_i is larger than a given threshold w_t. The percentage of candidate points in a point cloud is relatively rare. We treat the histogram of coefficient w as a 1D signal and smooth it by the following minimization:

$$\hat{f}_w = \arg\min_{\hat{f}_w} \|\hat{f}_w - f_w\|_F + \lambda\|D\hat{f}_w\|_1,$$

where λ is a tradeoff parameter, f_w is the histogram of $\{w_i\}_{i=1}^{n}$, D is the first order difference matrix, $\|\cdot\|_F$ and $\|\cdot\|_1$ represent F-norm and l_1-norm, respectively. The gentle decrease after the first peak of the smoothed histogram is used to determine the threshold w_t. Specifically, the point satisfies that the difference between its previous and next bins is smaller than $0.05 \times \max(f_w)$. The value of λ is fixed at 1, in our implementation.

5 Normal estimation for the candidate points

5.1 Previous methods

For each candidate point, the initial normal computed by PCA does not respect sharp features, since its neighbors may be from several surface patches intersecting at the features

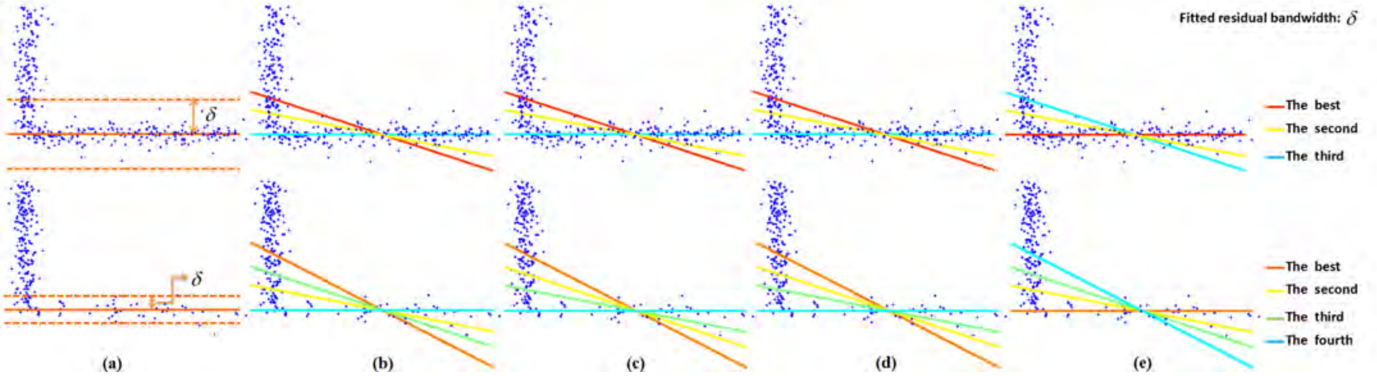


Fig. 2: The ranking results of three (top row) or four (bottom row) candidate plans given by RNE (b), IRPF (c), NIRPF (d), and our method (e). (a) is the fitting residual bandwidth which is big enough to cover all the points belonging to each plane. Our ranking results are correct even for large residual bandwidth (top row) and non-uniform sampling (bottom row).

and all of them contribute equally. To estimate feature preserving normals, the common strategy is to reduce the influence of the neighbors from different surface patches.

Based on the observation that the neighbors sampled from different surface patches often result in large fitting residuals, Li *et al.* [10] (RNE) design the following objective function to reduce the influence of such neighbors:

$$\theta_i^* = \underset{\theta}{\operatorname{argmax}} \frac{1}{\Omega} \sum_{p_i^j \in \mathcal{N}_i} \rho_1\left(\frac{r_{j,\theta}^2}{\sigma_{r,i}^2}\right), \tag{2}$$

where θ represents a plane, $\mathcal{N}_i$ is the neighborhood of p_i, Ω is a normalization factor, ρ_1 is a decreasing function, $r_{j,\theta}$ is the distance (fitting residual) from the point p_i^j to the plane θ, and $\sigma_{r,i}$ is the fitting residual bandwidth for p_i. Mederos *et al.* [26] (IRPF) further suggest that the neighbors near to the p_i should contribute more and minimize the following objective function:

$$\theta_i^* = \underset{\theta}{\operatorname{argmin}} \sum_{p_i^j \in \mathcal{N}_i} \rho_2\left(\frac{r_{j,\theta}^2}{\sigma_{r,i}^2}\right) w_d(p_i, p_i^j), \tag{3}$$

where ρ_2 is a monotonic increasing function and $w_d(p_i, p_i^j)$ is a Gaussian weight function related to the distance between p_i and its neighbor p_i^j. Inspired by the fact that the normal of a point is in general closer to the normals of its neighbors located on the same surface patch, Wang *et al.* [27] (NIRPF) add another weight function to penalize the neighbors with normals departing from p_i's normal:

$$\theta_i^* = \underset{\theta}{\operatorname{argmin}} \sum_{p_i^j \in \mathcal{N}_i} \rho_2\left(\frac{r_{j,\theta}^2}{\sigma_{r,i}^2}\right) w_d(p_i^j) w_n(n_i, n_{i,j}), \tag{4}$$

where n_i and $n_{i,j}$ represent the normals of points p_i and p_i^j, $w_n(n_i, n_{i,j})$ is a Gaussian weight function related to the difference between the normals n_i and $n_{i,j}$.

These methods effectively reduce the influence from scanned outliers and neighboring points from the wrong side of the edge. However, they have some shortcomings as well. First, several bandwidth-related parameters need to be estimated, especially the fitted residual bandwidth $\sigma_{r,i}$ which has an important impact on the results. Although RNE and NIRPF devise an automatic procedure to evaluate the bandwidth according to the varying local geometry, calculating the exact value is still a very difficult problem, especially when the data contains noise and outliers, which is common in the raw point clouds. As shown in the first row of Fig. 2 (a), when the data is occupied by large noise and outliers, a relatively large fitting residual bandwidth may be given to include all the points belonging to each plane. Under the effect of the given bandwidth, the chosen planes tend to smooth out the sharp features. Secondly, these methods do not address the variation of density near edges and the estimated normals would be biased toward the surface patch with higher point density even if the bandwidth-related parameters are proper, as shown in the second row of Fig. 2.

5.2 Pair consistency voting (PCV)

In this section we will introduce a new assessment which is insensitive to the fitting residual bandwidth and robust to the variation of density.

When an outsize fitting residual bandwidth is given, the incorrect votes cast by neighbors across sharp features are trusted. In order to reduce the influence of such votes, IRPF and NIRPF suggest that the neighbors share similar properties with the current point p_i, such as locations or initially estimated normals, are worthy of trust. However, when a point is extremely near some sharp features, its properties are unreliable, as shown in the wrapped figure of Section 1. Given a point p_i marked by the red point, the neighbors far from the sharp features (marked by blue points) should play more important role in the voting process than the neighbors marked by orange points. However, choosing the blue neighbors directly is not a wise option.

Inspired by the idea of neighborhood segmentation, we introduce a new objective function which reduces the effect of neighbors near sharp features and uses the reliable regions (blue neighbors) to dominate the voting process. Our method optimizes for the plane maximizing the following

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2827998, IEEE Transactions on Visualization and Computer Graphics

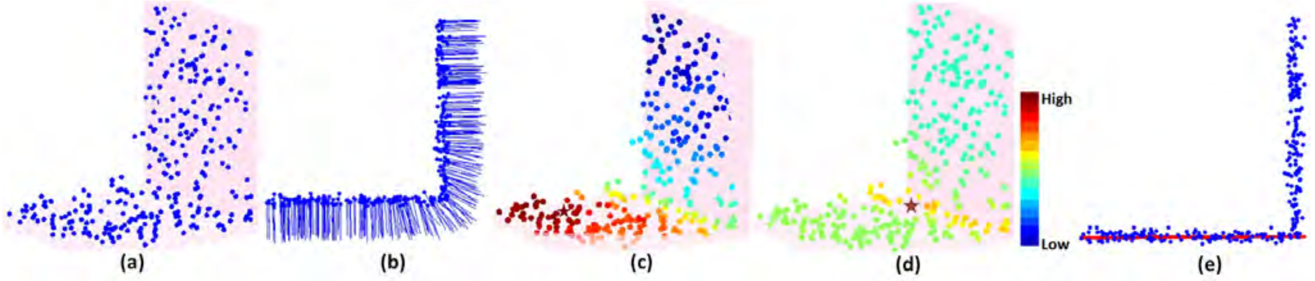


Fig. 3: The visual rendering of weight $w_a(p_i^j, p_i^k)$ in the neighborhood of p_i (a), and the final selected plane (e). (b) is the initial normals given by PCA. p_i^j, represented by a red star, may be relatively far from the edge (c) or very close to the edge (d). The pairs with both points lying in the same plane have larger weights and contribute more in the voting.

objective function:

$$\begin{aligned} \theta_i^* &= \operatorname*{argmax}_{\theta} E(\theta) \\ &= \operatorname*{argmax}_{\theta} \sum_{p_i^j, p_i^k \in \mathcal{N}_i} \rho(p_i^j, \theta)\rho(p_i^k, \theta) w_a(p_i^j, p_i^k), \end{aligned} \tag{5}$$

where

$$\rho(p_i^j, \theta) = exp(-r_{j,\theta}^2/\sigma_{r,i}^2),$$

$$w_a(p_i^j, p_i^k) = exp(cos^{\alpha}(\widehat{n_j^0 n_k^0})/\sigma_{a,i}^{\alpha}),$$

$\widehat{n_j^0 n_k^0}$ is the angle between n_j^0 and n_k^0, $\sigma_{a,i}$ and $\sigma_{r,i}$ are respectively the normal difference bandwidth and fitting residual bandwidth for p_i. $\sigma_{a,i}$ has similar roles with the variance in the Gaussian distribution. If the value of $\sigma_{a,i}$ is very large, all the pairs play the same role. If it is very small, two pairs with similar angles share quite different weights. We fix its value at $cos(45^o)$ empirically. For each neighbor p_i^j, we compute the residual from it to the plane with normal n_i^0. The average of these residuals is regarded as p_i's median residual. $\sigma_{r,i}$ is defined as $\sigma_{r,i} = \beta * \delta_i$, where δ_i is detected automatically by the average of smooth points' median residuals in the neighborhood $\mathcal{N}_i$. The α and β are user-definable parameters that will be evaluated later.

From the definition $E(\theta)$, we see that actually it encourages the neighbors with similar initial normals to lie in the same plane. Specifically, if the normals of p_i^j and p_i^k are similar, the weight $w_a(p_i^j, p_i^k)$ is large, as shown in Fig. 3 (c). Therefore, the fitting residuals of these two points should both be small, *i.e.*, points p_i^j and p_i^k fall on the same plane. Moreover, if a point pair includes at least a point near features, its weight w_a is small, such as sharp edges in Fig. 3 (c) and (d), because the initial normals near sharp features change rapidly (see Fig. 3 (b)). Benefiting from this property, the points near sharp features play a minor role in the voting process and $E(\theta)$ is insensitive to the bandwidth adopted, as illustrated in Fig. 2 and Fig. 11.

We solve the optimization in (5) by discrete strategy, *i.e.*, building a set of candidate planes via random sampling and selecting the one with the most votes. Specifically, for a candidate point p_i, a large enough neighborhood $\mathcal{N}_i$ with S^* neighbors is determined. The value of S^* should be larger than S, since only partial neighbors are used to estimate the normals for the feature points. Three non-collinear points are chosen randomly to instantiate a candidate plane θ. This process is repeated M_i times to generate M_i candidate planes until there is a high probability that at least one triple used for plane instantiation is clean, *i.e.*, the triple only consists of neighbors from the same surface patch as p_i. The determination of M_i will be explained later. For each candidate plane θ, we compute $E(\theta)$. The plane maximizing the objective function (5) and containing p_i is selected as the tangent plane of p_i.

5.3 The iteration of PCV (PCV-MN)

By iterating the above procedure, we can compute multiple normals for edge and corner points satisfying the definition in Section 3. Denote the first chosen tangent plane θ_i^* as θ_1. We define a new neighborhood $\mathcal{N}_i^2$ by removing all the points in θ_1:

$$\mathcal{N}_i^2 = \{p_i^j \in \mathcal{N}_i | \rho(p_i^j, \theta_1) > \sigma_{r,i}\}.$$

Then we pursue a voting in $\mathcal{N}_i^2$ and choose the plane θ_2 with the maximal $E(\theta)$ over all the candidate planes instantiated by three non-collinear points randomly chosen from the $\mathcal{N}_i^2$. Following the same procedure, we obtain a series of planes: θ_1, θ_2, $\theta_3, \ldots$. The iterations stop after the k-th iteration if the sum of squared density weight corresponding to the points in $\mathcal{N}_i^{k+1}$ is less than a threshold $T(k)$, which is defined as:

$$T(k) = \sum_{p_i^j \in \mathcal{N}_i} \gamma_j^2 (\frac{1}{k+1} - 0.03k).$$

where γ_j is the p_i^j' density weight defined in the next subsection, $0.03k$ takes the points on edges or corner into consideration, which may be contained in several planes and deleted in previous steps.

Now we have a series of planes to describe the structure of the neighborhood and each plane approximates one surface patch. But these surface patches do not necessarily contain p_i. So we choose those planes whose distances from p_i are less than $\sigma_{r,i}$ as p_i's final tangent planes, and use their normals as p_i's multiple normals. We denote the output normal set of p_i as $\widetilde{N}_{p_i} = \{\widetilde{n}_i^j\}_{j=1}^{d_i}$, where d_i stands for the number of normal vectors assigned to point p_i. $d_i = 1$ means that p_i is a non-feature point.

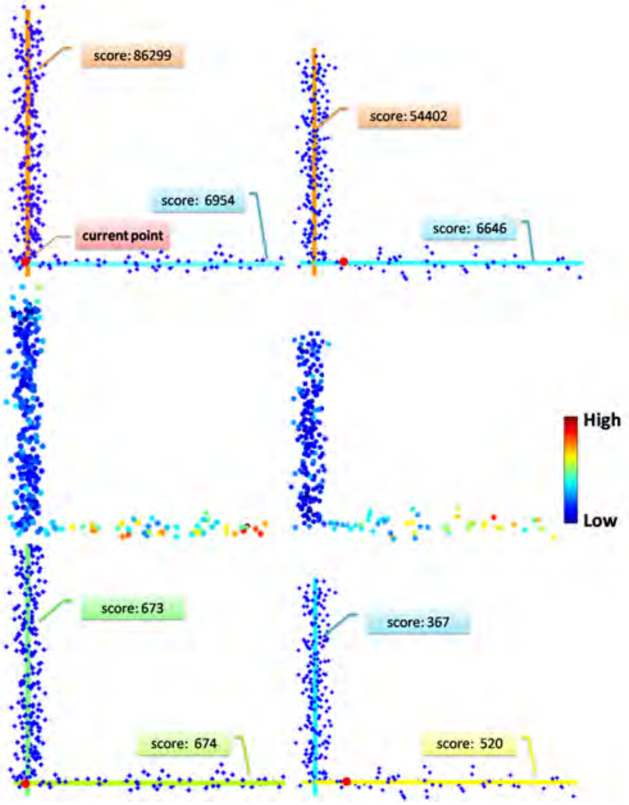


Fig. 4: The scores will be sensitive to non-uniform sampling (top row) without considering the density weight γ (middle row). With γ, the scores are robust to non-uniform sampling (bottom row).

5.4 Dealing with sampling anisotropy

If the sampling is uniform, the normal estimation process discussed above is capable of estimating normals accurately. However, when the sampling density is anisotropic, the plane obtained by Eq. (5) tends to bias toward the regions with high density. In this section, we introduce two improvements which contribute to make our algorithm robust to the density variation.

1) Density weight. First, we design a density weight to put more confidence to the neighbors from the surface patches with low sampling density. The improved objective function becomes:

$$\begin{aligned}\theta_i^* &= \underset{\theta}{\operatorname{argmax}}\, E(\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \sum_{p_i^j, p_i^k \in \mathcal{N}_i} \rho(p_i^j, \theta)\rho(p_i^k, \theta) w_a(p_i^j, p_i^k)\gamma_j^2\gamma_k^2, \end{aligned} \tag{6}$$

where γ_j is the density weight defined as the average distance from p_i^j to its k nearest neighbors. From the definition of density weight, we see that it is larger for the point with low sampling density. The improvement of density weight in the plane selection is illustrated in Fig. 4.

2) Adaptive size of the candidate plane set. When point p_i lies in a surface patch with sparse sampling, the probability that three non-collinear points are all from the same surface as p_i is lower. Therefore, we need a larger candidate plane set. We introduce the ratios of sampling to determine the size of the candidate plane set automatically. Let ε_i be the fraction of neighbors which belong to the same patch with point p_i. After M_i selections, the probability P_i that one clean triple has been chosen is computed as [10]

$$P_i = 1 - (1 - \varepsilon_i^3)^{M_i}.$$

Therefore we determine M_i by

$$M_i = \frac{\log(1 - P_i)}{\log(1 - \varepsilon_i^3)}.$$

In our implementation, we set $P = 0.9$ and ε_i is defined as $\varepsilon = (\gamma_i^{min})^2/2(\gamma_i^{max})^2$, where γ_i^{min} and γ_i^{max} are respectively the minimal and maximal density weights in the neighborhood.

6 Benchmark for Feature Preserving Normal Estimation

6.1 The error measure

Various normal estimators outputting a normal per point can be compared with the Root Mean Square with threshold (RMS$_\tau$) [8]. We generalize it to handle the case that each point has multiple ground truth normals. The Root Mean Square measure with threshold for multiple normals ($RMSM_\tau$) is defined as:

$$RMSM_\tau = \sqrt{\frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\big(\frac{1}{|\widetilde{N}_p|}\sum_{\widetilde{n}_p\in\widetilde{N}_p}\min_{n_p\in N_p} f(\widehat{n_p\widetilde{n}_p})^2\big)},$$

where N_p and $\widetilde{N}_p$ are the ground truth and estimated normal sets of p, respectively, $\widehat{n_p\widetilde{n}_p}$ is the angle between n_p and $\widetilde{n}_p$, $f(\widehat{n_p\widetilde{n}_p})$ is defined as

$$f(\widehat{n_p\widetilde{n}_p}) = \begin{cases}\widehat{n_p\widetilde{n}_p}, \text{if } \widehat{n_p\widetilde{n}_p} < \tau \\ \pi/2, \text{otherwise}\end{cases}.$$

For the algorithms outputting one normal per point, the size of $\widetilde{N}_p$ is one. As in [8], We set $\tau = 10°$.

6.2 Ground truth dataset

Our ground truth data are points with multiple normals per point. The points are sampled from triangular meshes, which are synthesized or reconstructed from real scanning data. To generate feature points precisely, we use the mesh's vertices as the samples and the normals of its surrounding faces as the ground truth normals of the point:

$$N_{p_i} = \{n_f | f \text{ is a face containing } p_i\}.$$

The variation of N_{p_i} is very small when the point lies in the interior of some smooth surface patch, while that would be large when the point is sampled from feature edges or corners.

Two groups of data are collected: 152 synthesized point clouds and 288 real scans. The synthesized data are point clouds with various properties which are divided into five categories. Each captures one kind of challenges in normal estimation. There are thirty, twenty one and twenty one point clouds for SharpFeature, Details and NonUniform categories respectively, which are sampled from ten, seven and seven models using three different resolutions. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2827998, IEEE Transactions on Visualization and Computer Graphics

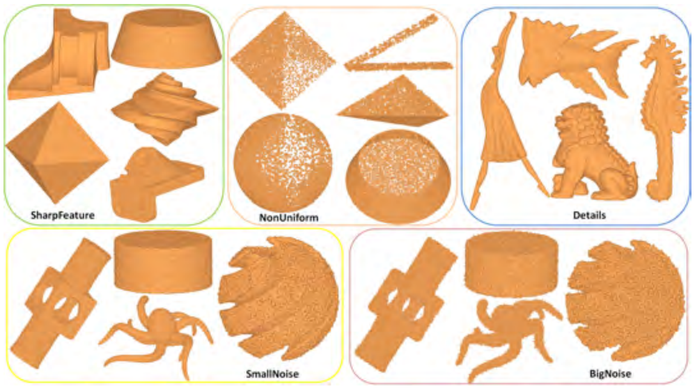


Fig. 5: Partial synthesized point clouds.

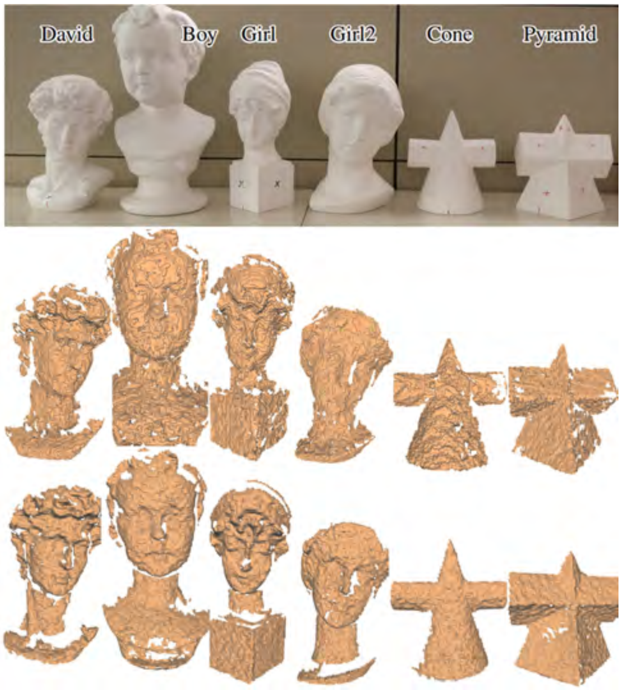


Fig. 6: The real models and partial scans. From top to bottom are the models made by plaster and resin (the figure comes from [31]), the scans of Kinect v1 and Kinect v2.

SmallNoise category includes forty models with sharp features perturbed by centered Gaussian noise with the deviation defined as 30%, 40%, 50%, and 60% average distance between points. The last category is BigNoise. It includes forty models with sharp features perturbed by centered Gaussian noise with deviation defined as 0.2%, 0.3%, 0.4%, and 0.5% of the diagonal of the axis-aligned bounding box. Partial synthesized data are illustrated in Fig. 5.

The second group of data are sampled from triangular meshes scanned by Microsoft Kinect v1 and Kinect v2, provided by [31]. Each mesh is extracted from a single depth frame. There are 144 Kinect v1 scans and 144 Kinect v2 scans for seven models made by plaster and resin. The total number of facets is about 2.6M and 930k, respectively. The height of models is about 35 cm. The real models and some scans are shown in Fig. 6. For each scan, a high-resolution one scanned by an Artec Spider^{TM} scanner (accuracy 0.5 mm) is also provided for computing ground truth normals. Please refer to [31] for more details of register and ground truth normals computing.

7 Results

7.1 Competitors

To evaluate the performance of our approach thoroughly, we compare our method, including PCV and PCV-MN versions, with some classic and state-of-the-art methods: PCA [9], RNE [10], HF [8], LRR [11], LRRfast [29], and HF_CNN [30]. We use the trained models of HF_CNN provided by the authors. According to the scales, HF_CNN has two versions: HF_CNN1s and HF_CNN5s. Using different sampling strategy, HF has three versions: HF_points, HF_cubes, and HF_unify. RMSMτ is used to quantitatively analyze the results.

We compare the above methods by setting two kinds of neighborhood size S^*: 100 and 500. The smaller one is recommended by HF_CNN1s, and the bigger one is suggested by HF and RNE. HF_CNN5s simultaneously considers five neighborhood sizes (32, 64, 128, 256 and 512). Hence, we compare it with other methods in both cases. All other parameters, if any, are set to default values. With limited space available, we only give the quantitative and some visual comparisons. More visualization results can be found in the supplemental material.

7.2 Comparison on synthesized data

The average RMSMτ of each category and all the 152 synthesized models are reported in Fig. 7. We do not compare with LRR and HF_unify for $S^* = 500$ as their computational cost is too high, which need more than 20 hours for one model with 50K points. The visualization of computed normals near sharp features are shown in Fig. 9.

Overall, normals estimated by PCA are smooth near sharp features, see Fig. 9, and their RMSM$_\tau$ are higher than other methods with small or large neighborhood, except for estimating normals of big noise models with 100 neighbors. This is mainly because the feature preserving methods fail to analyze neighborhood structures correctly with small neighborhood size, when the point clouds suffer from such heavy noise. Larger neighborhoods are necessary for generating quality results as shown in the right side of Fig. 7.

For RNE and HF, the results are also a bit smooth, especially when the neighborhood structures are complex or with anisotropic sampling, see Fig. 9. More comparison can be found in the supplemental material. HF_points achieves the lowest RMSM$_\tau$ for models with details when $S^* = 100$. But in general, among the three HF methods, HF_unify performs best and is more computational expensive.

HF_CNN1s and HF_CNN5s fail to defeat some traditional methods with $S^* = 100$. For larger neighborhood $S^* = 500$, HF_CNN5s, using multiple neighborhood sizes simultaneously, is more successful than these methods. It achieves the best results for the ShapeFeature and Details categories, and our method is more suitable for the BigNoise and NonUniform categories. In general, our method copes with all kinds of challenges of all the 152 models, and the two versions' average RMSMτ are both lower than the two HF_CNN methods.

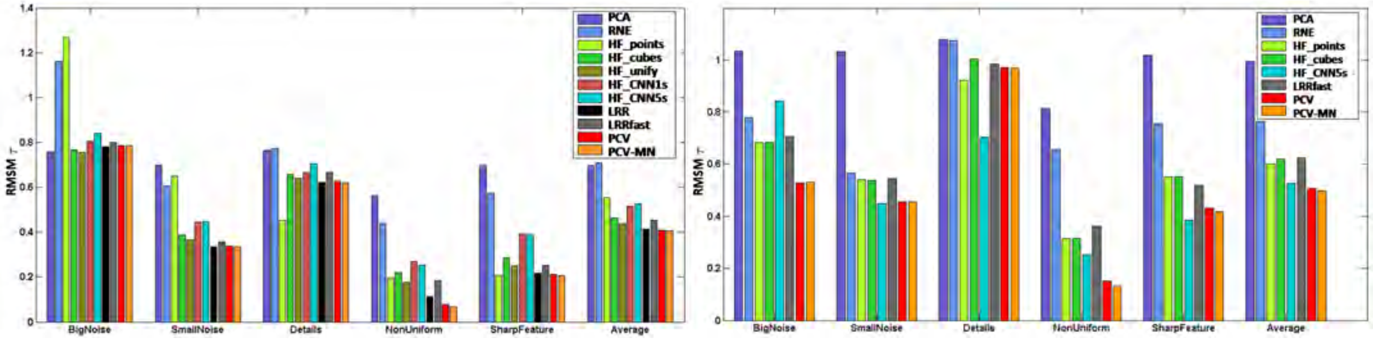


Fig. 7: The average RMSM$_\tau$ of various methods with neighborhood size of 100 (left) and 500 (right) on the 152 synthesized data.

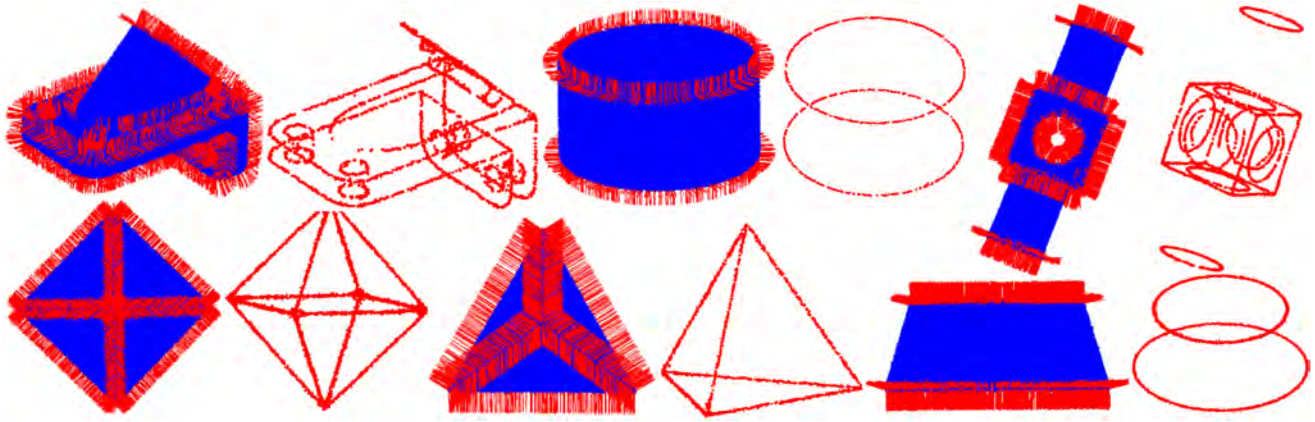


Fig. 8: Visualization of PCV-MN. The point clouds are represented by blue. The points with multiple normals are represented by red and the output multiple normals are shown in red lines.

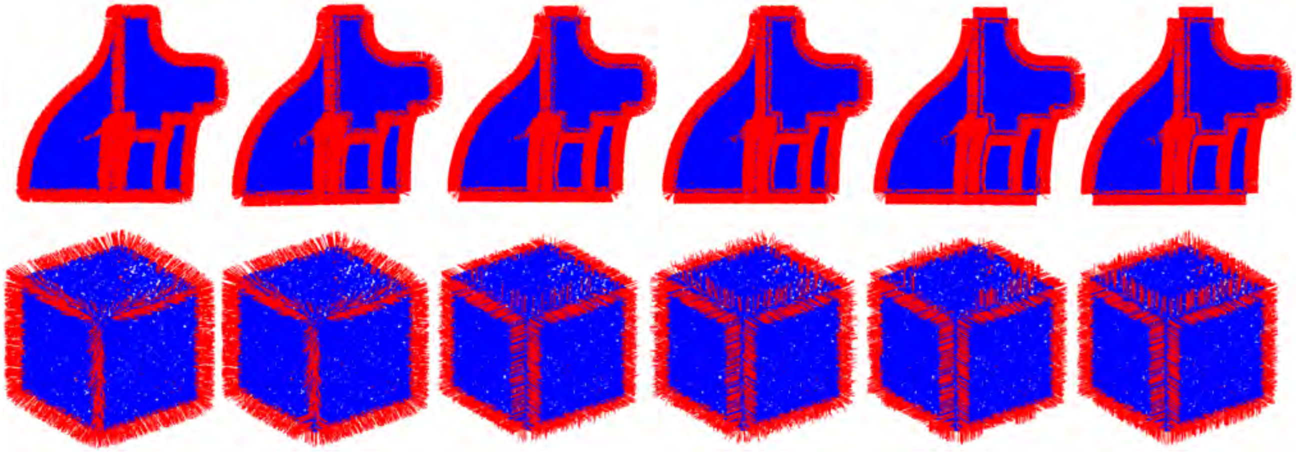


Fig. 9: Visualization of estimated normals near sharp features via various methods. The results of PCA, RNE, HF_cubes, HF_CNN5s, LRRfast and PCV are shown from the left to the right. In the top row, neighborhood size is 100 and the model is Fandisk (77K) perturbed by Gaussian noise with deviation defined as 30% average distance between points. In the bottom row, neighborhood size is 500 and the model is Cube (20K) sampled with face-specific variations of density. The ratios of sampling on the six faces are 1:2:3:4:5:6.

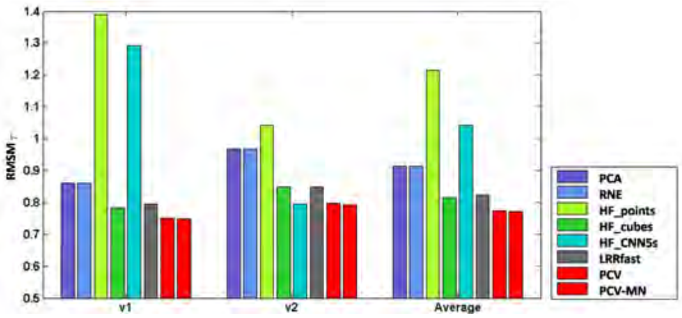


Fig. 10: The average RMSM$_\tau$ of various methods with neighborhood size of 500 on the 288 point clouds scanned by Microsoft Kinect v1, Kinect v2.

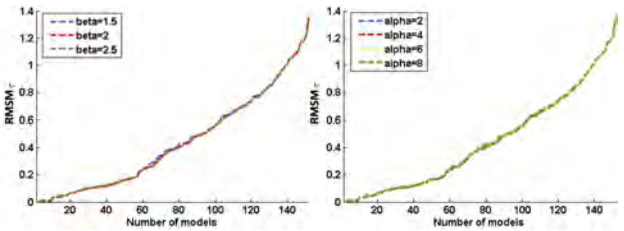


Fig. 11: RMSM$\mathcal{T}$ of the 152 synthesized data with a varying α or β. In the left, we fix the α at 4 and change β between 1.5 and 2.5. In the right, we fix the β at 2 and change α between 2 and 8. The x-axes represent the indexes of the models which are sorted in ascending order of the RMSM$\mathcal{T}$ with $\alpha = 4$ and $\beta = 2$.

LRR and LRRfast preserve sharp features better. However, they are not good at handling non-uniform sampling (the bottom row in Fig. 9). Moreover, they are relatively time-consuming, see Tab. 1.

In summary, our method, PCV and PCV-MN, handles all the problems and achieves the lowest average RMSM$\mathcal{T}$ of all the synthesized models, using either small or large neighborhoods. Moreover, the visualization in Fig. 9 also shows that PCV is capable of estimating normals accurately even in the presence of noise and anisotropic samplings, while preserving sharp features of the original models. The results of PCV-MN are illustrated in Fig. 8. We see that the points with multiple normals are positioned on the edges and corners. Moreover, the multiple output normals exactly approximate the normals of underlying surfaces which intersect at the edge/corner points.

Computation time. We report the speed of all the compared methods in Tab. 1. The experiments are all performed on a 3.5GHz Intel i7-2700M laptop with 4 cores. PCA is the fastest, even with the MATLAB implementation (denoted as ”M”), at the expense of feature preserving. The methods with C/C++ implementation (including RNE, HF, and HF_CNN, denoted as ”C”) are fast. The speed of our methods are acceptable and they are the fastest among the methods implemented by MATLAB with or without a C/C++ library (including LRR and LRRfast, denoted as ”M&C”). A parallel C++ implementation could improve their performance remarkably.

7.3 Comparison on real data

We only compare the methods with 500 neighbors, see Fig. 10, since the 288 scans have large noise. Our method achieves the lowest average RMSM$_\tau$ of all the scans. Specifically, our method outperforms all the other methods for the scans with larger noise by Microsoft Kinect v1, and it is comparable with HF_CNN5s for those with relatively smaller noise by Microsoft Kinect v2. In summary, our method is more suitable for the real scan data.

7.4 Parameters

The parameters of our algorithm are summarized below:

- S: the number of neighbors used for smooth points.
- S^*: the number of neighbors used for candidate points.
- k: the size of k-distance neighborhood.
- α: the power in w_a.
- β: the parameter used to compute σ_r.

The values of S and S^* are affected by the noise scale. Larger values of S and S^* are necessary for big noise. Moreover, the value of S^* should be larger than S, since only part neighbors are used to estimate the normals for the feature points. In our implementation, the $S^* = 2 * S$. Our method is insensitive to the variation of α and β. But the estimated normals may be a bit smooth with a too small α and too large β. The values of k, α, and β are selected as $k = 10$, $\alpha = 4$, and $\beta = 2$.

The main parameters in our algorithm are α and β. To observe the influence of these two parameters on the estimation, we test PCV with a varying α or β on the 152 synthesized data. Fig. 11 shows the results. As it turns out, PCV could work well under a wide range of parameter setting. While the parameter α ranges from 2 to 8 or the fitted residual bandwidth β ranges from 1.5 to 2.5, the RMSM$\mathcal{T}$ varies slightly.

7.5 More results

We further demonstrate the effectiveness of our algorithm on more real scanned point clouds. Fig. 12 shows the results of PCV on the raw scans of real objects: House, Genus2, Nyon, and Armadillon. The edges of these models are recovered faithfully. Models with nearby surface sheets always challenge the normal estimation. Detail 2 of Genus2 shows such a case. By introducing reliable neighborhood structure, PCV handles the nearby surface sheets well. Moreover, PCV handles the non-uniform sampling well, see detail 3 of Nyon and detail 1 of Genus2 and House. The raw scan of the house contains many tiny structures, as shown in detail 2, which challenge the normal estimation persistently. PCV copes with such case and generates faithful normals.

8 Advantage of Multi-Normal

The advantage of multi-normal is verified by some classic tasks: filtering, feature extraction, point cloud rendering, and visualization of segmentation.

TABLE 1: The average computation time (in seconds) on each category and all the 152 models.

		SharpFeature	SmallNoise	BigNoise	NonUniform	Details	Average
Average points for each category		71228.9	67055	67055	25557	196690	80056
PCA	$S^* = 100$	12.54	10.45	10.05	3.50	62.10	19.73
code: M	$S^* = 500$	17.33	15.68	15,99	8.87	90.66	22.74
RNE	$S^* = 100$	26.06	24.87	30.40	8.29	102.02	37.73
code: C	$S^* = 500$	67.15	66.87	65.40	23.29	226.02	89.89
HF_points	$S^* = 100$	6.84	24.81	30.48	2.09	6.84	29.22
code: C	$S^* = 500$	34.58	39.94	45.77	15.80	120.36	47.13782
HF_cubes	$S^* = 100$	23.00	34.28	134.64	7.27	88.75	57.63
code: C	$S^* = 500$	35.50	39.27	145.92	11.33	126.36	74.76
HF_unif	$S^* = 100$	1395.9	1028.4	997.5	523	3915.9	1615.3
code: C	$S^* = 500$	-	-	-	-	-	-
LRR	$S^* = 100$	222.97	1955.38	1221.51	641.55	4549.72	1992.00
code: M&C	$S^* = 500$	-	-	-	-	-	-
LRRfast	$S^* = 100$	101.41	108.17	82.10	19.08	450.64	90.90
code: M&C	$S^* = 500$	7735.8	4481.6	4098.7	3364.4	7756.0	5321.1
HF_CNN1s	$S^* = 100$	10.30	9.98	10.15	5.24	24.07	11.38
code: C	$S^* = 500$	-	-	-	-	-	-
HF_CNN5s code: C	Multiscale	29.92	27.58	29.28	13.06	77.01	33.31
PCV	$S^* = 100$	81.77	72.91	40.82	26.14	169.55	73.10
code: M	$S^* = 500$	497.94	469.25	467.88	163.60	674.72	460.71
PCV-MN	$S^* = 100$	97.76	82.16	41.60	46.34	200.34	85.94
code: M	$S^* = 500$	634.94	569.32	522.77	227.84	823.73	557.97

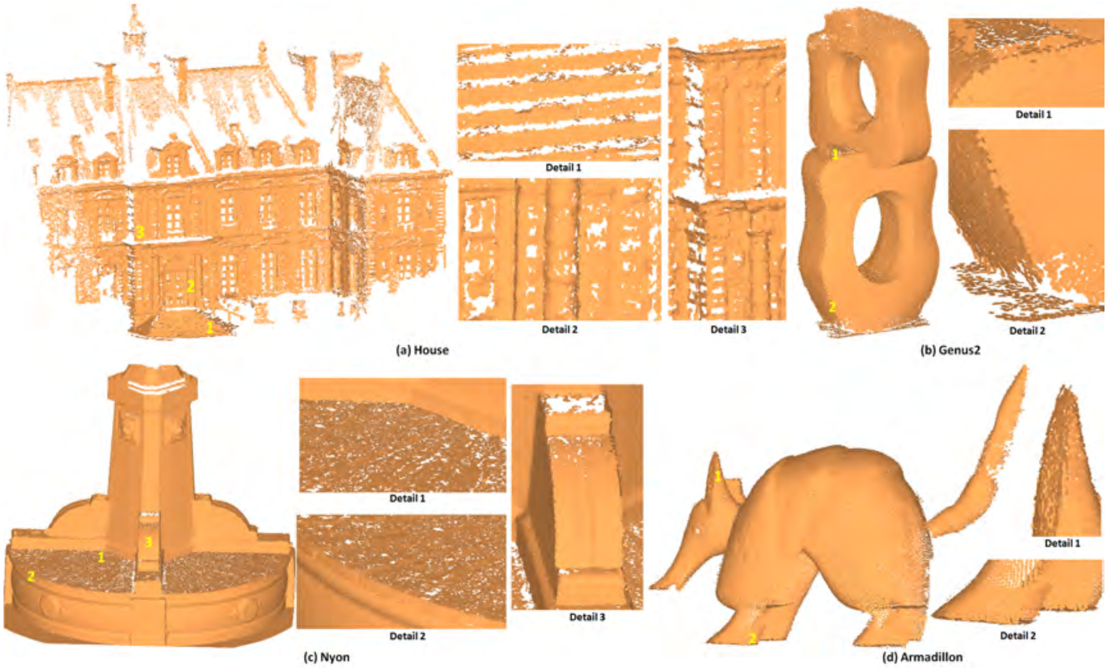


Fig. 12: The normal estimation results on the raw scans of House, Genus2, Nyon, and Armadillon.

Filtering: Filtering plays a fundamental role in point cloud processing. Conventional single normal based approaches may produce cross artifacts, especially when the point cloud contains excessive noises [32]. To reduce the artifacts and preserve the sharp features more effectively, the idea that the points near sharp features should be projected onto feature lines is used in [32], [33]. For the point p_i close to some sharp features, Sun *et al.* [32] compute a neighbor point p_j with large normal difference and reposition p_i such that it minimizes the distances to both the tangent planes defined by p_i and p_j. Zheng *et al.* [33] address this issue by assigning feature points with multiple normals and projecting them onto the intersection of multiple surfaces. We follow the framework of [33] and replace their sharp feature detection and multiple normals initialization by our method. Specifically, the joint bilateral normal filtering [34] is performed on the normal field $\{\widetilde{n}_i^j\}$ to get a piecewise smooth one $\{\widehat{n}_i^j\}$. The normals of point p_i should be orthogonal to its incident edges as much as possible [35], so the point position is updated by the following iterative formula:

$$\widehat{p}_i^{t+1} = \widehat{p}_i^t + \frac{1}{d_i}\sum_{j=1}^{d_i}\sum_{k\in\mathcal{N}_i}\sum_{l=1}^{d_k}\frac{w(i,j,k,l)}{d_k W_{ij}}[\widehat{n}_k^l\cdot(\widehat{p}_i^t-\widehat{p}_k^t)]\widehat{n}_k^l.$$

where $\widehat{p}_i^t$ is the position of point p_i after t-th iteration, $w(i,j,k,l) = w_d(p_i,p_k)w_n(\widehat{n}_i^j,\widehat{n}_k^l)$, and $W_{ij} =$

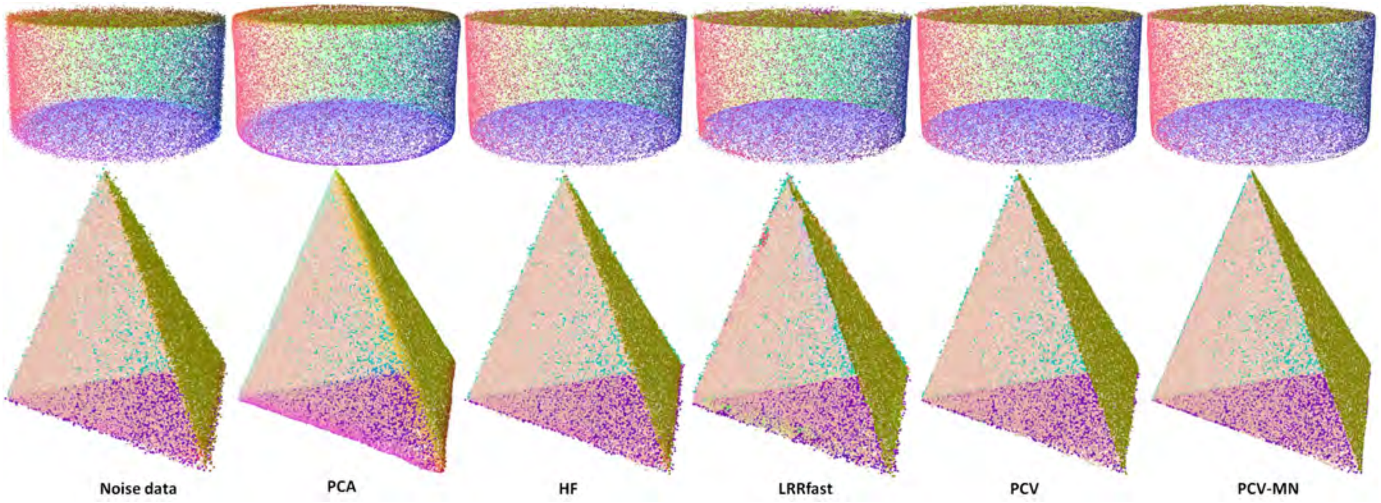


Fig. 13: Comparison of filtering results with normals obtained by different methods. Clearer sharp features and smoother feature curves are obtained with the support of multi-normal (PCV-MN).

$\sum_{k\in\mathcal{N}_i}\sum_{l=1}^{d_k}(w(i,j,k,l)/d_k)$ is employed to normalize local weights. $w_n(g_i^j, g_k^l)$ and $w_d(p_i, p_k)$ have been defined in Eqs (3) and (4), respectively. Please refer to [33] for more details. Within the framework, we also compute the filtering results with single normal obtained by PCA, HF, and LRRfast. The results are presented in Fig. 13. It can be seen that sharper and clearer feature edges are obtained with the support of multi-normal.

Feature extraction: As we know, the sharp features are usually detected as the intersection of multiple piecewise smooth surfaces. For each candidate feature point p_i, we obtain a series of planes to describe the structure of the neighborhood. Then we choose those planes whose distances from p_i are less than $\sigma_{r,i}$ as p_i's final tangent planes, and use their normals as p_i's multiple normals. If p_i has more than one normal, it is included by multiple surfaces and should be treated as a feature point.

Our method can distinguish between weak features and pseudo features better, since the neighborhood structures are used. Moreover, we explore neighborhood structures with the guidance of more stable neighbors which are relatively far from features. The results will be more accurate and robust. The visual comparison between the classical method MSTV [36] and our method is given in Fig. 14. The results of MSTV are provided by the authors. MSTV loses some real weak feature points, as shown in the top row of Fig. 14. In comparison, results from the bottom row of Fig. 14 demonstrate that our method is capable of recovering more weak feature points with almost no added pseudo feature points. More results enhanced by neighbor supporting [37] are shown in Fig. 15. Our method achieves almost perfect detection results for these models containing different types of complexities. For instance, the Octaflower model contains thin surface sheets, weak features, and corners with four twisty surfaces. The Bearing models contain close features and various corners. And the Twist model has nontrivial topology and the size of bend is various.

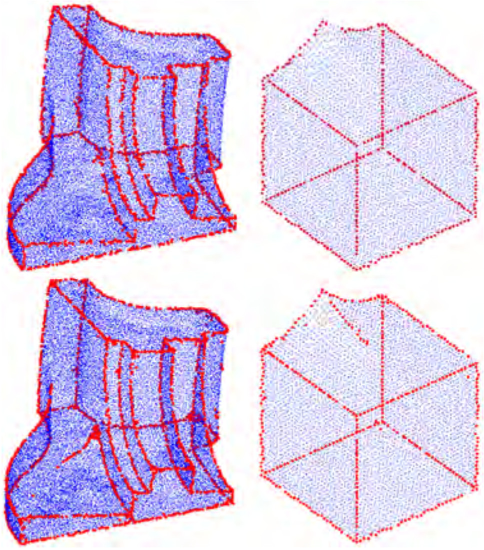


Fig. 14: The sharp feature detect results by MSTV (top) and our method (bottom). Left: Fandisk model with 20% Gaussian noise. Right: Smooth-feature model with 20% Gaussian noise.

Rendering: The definition benefits the rendering of point cloud models as illustrated in Fig. 16, where we render several copies of each feature point, assign each a normal from its normal set, and all points are rendered in surfels [38]. Previous methods extract one normal per point. The feature lines boundaries are either blurred (if the normal is not feature preserved) or irregularly notched (since the normals of adjacent points may face different directions randomly). With the notion of multi-normal, we render them in a superior way.

Visualization of segmentation: In Fig. 17, visualization of a segmentation result using multiple normals per feature point is compared to that with one normal per point. The visualization using multiple normals computed by PCV-MN has a more tidier, clear layout for the feature lines.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2827998, IEEE Transactions on Visualization and Computer Graphics

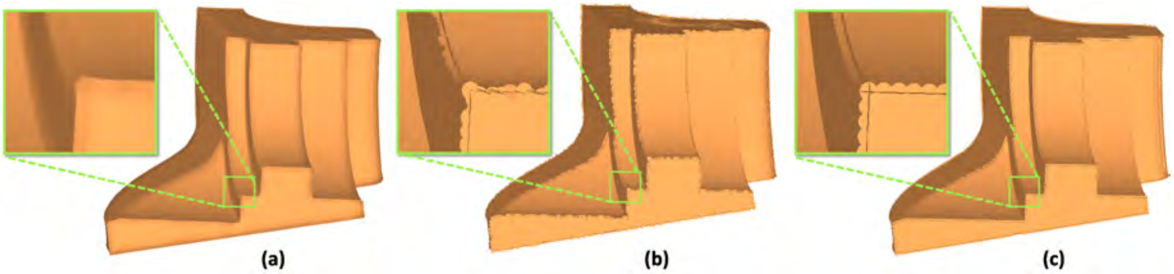


Fig. 16: Rendering of a point cloud model. (a), (b), and (c) are the results rendered using normals computed by PCA, PCV, and PCV-MN, respectively. Note that the feature lines are blurred by PCA and irregularly notched by PCV. But PCV-MN render them in a superior way.

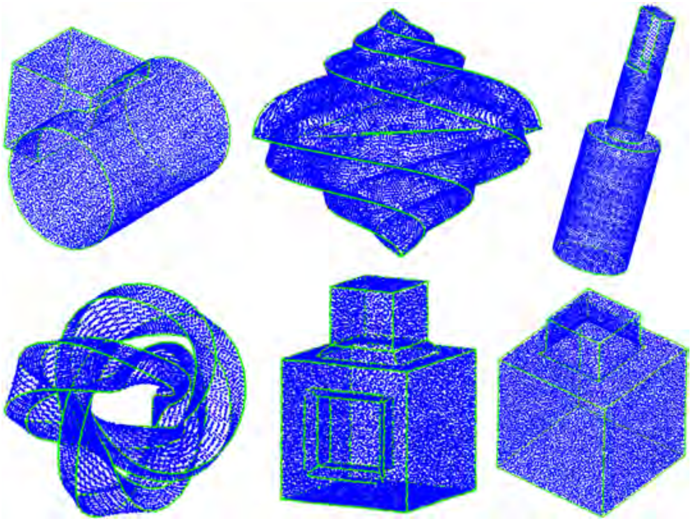
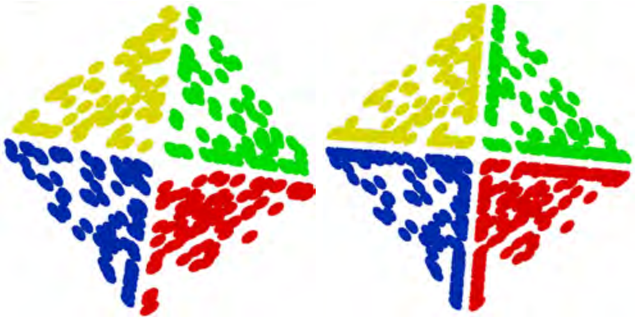


Fig. 15: Results of sharp feature detection.



(a) Single Normal (b) Multi Normals

Fig. 17: Visualization of a segmentation result using multiple normals computed by PCV-MN (right) is superior to that with the normals computed by PCV (left).

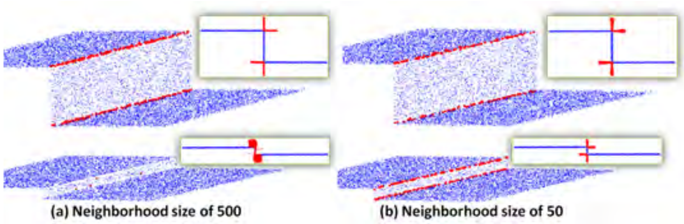


Fig. 18: Comparison of PCV-MN with different neighborhood sizes. With large neighborhood, PCV-MN obtains more faithful normals and more complete features for the typical CAD features in the top row, but loses tiny details in the bottom row. The red lines in the box show the output normals near sharp features.

9 Conclusion

We define point cloud normals in a unified way and a point may possess one or multiple normals to describe its neighborhood structure - interior of some surface or intersection of multiple surfaces. A multi-normal algorithm is designed based on the definition. For each point, it extracts multiple normals by iterating our pair consistency voting scheme which incorporates the neighborhood's subspace structure into the tangent plane assessment. Faithful normals are estimated without resorting to time consuming subspace segmentation techniques. Based on the definition, the Root Mean Square measure RMSτ is also generalized to evaluate the estimated normals quantitatively. We further present the first benchmark for normal estimation. The experiments on the benchmark show that our approach outperforms previous cutting edge methods.

Some limitations and future research directions remain.

Multi-scale features: The size of neighborhood decides the initial normals, candidate planes and their evaluation. In our implementation, it is fixed. However, a large neighborhood will be inevitably smoothed out very small details of point clouds, see the bottom row of Fig. 18. But the performance of a small neighborhood will drop for large noise. Therefore determining the neighborhood size adaptively will further improve the performance for handling multi-scale features.

Potential gap in the filtering: We choose the planes whose distances from p_i are less than $\sigma_{r,i}$ as p_i's final tangent planes. $\sigma_{r,i} = \beta \times \delta_i$ depends linearly on δ_i (the average of median residuals of smooth points in p_i's neighborhood). Larger noise usually means larger δ_i, which leads to larger $\sigma_{r,i}$ and more feature points (points with multiple normals). Please refer to Fig. 19 of the paper and Fig. 5 of the supplementary material. The gaps appear in the filtering results when the noise level is high, since we use multiple normals and project the feature points onto feature lines. But the estimated multiple normals are still faithful to constrain the filtering results, *i.e.*, restore a feature preserving struc-

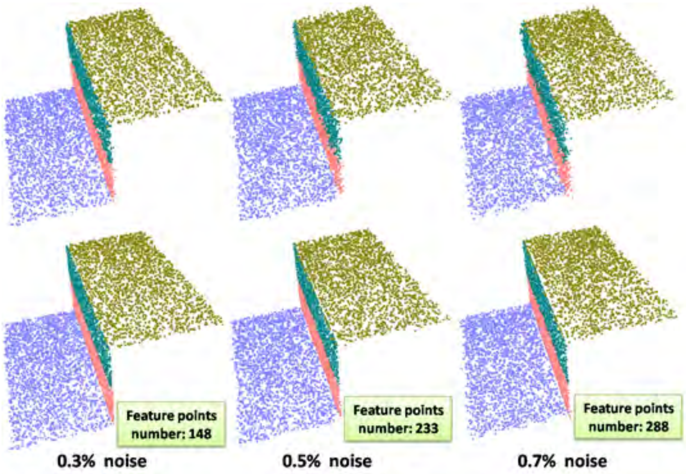

Feature points number: 148
Feature points number: 233
Feature points number: 288
0.3% noise
0.5% noise
0.7% noise

Fig. 19: Filtering results on the Stair model (including two horizontal planes and one horizontal plane) with different noise levels. The noise inputs and filtering results are shown in the top and bottom rows, respectively. The synthetic noise is a centered Gaussian noise with deviation defined as a percentage of the diagonal of the axis-aligned bounding box. The gaps appear when the noise level is high.

ture. Furthermore, some consolidation process, such as [12], could be applied to fill the gaps.

More applications: The multi-normal notion provides a viable way to achieve cleaner sharp features and jaggy-free feature curves, which has been demonstrated by four applications. It would be interesting to apply it to more applications, such as segmentation and surface reconstruction.

Acknowledgments

The authors would like to thank Bao Li and Alexandre Boulch for providing the code used for comparison, Pengshuai Wang, Yang Liu, and Xin Tong for providing the real scans of the benchmark, Yinglong Zheng and Guiqing Li for providing the results of filtering. Jie zhang is supported by the NSFC Fund (No. 61702245) and the Foundation of Liaoning Educational Committee (No. L201683663). Junjie Cao is supported by the NSFC Fund (Nos. 61572099 and 61772104). Xiuping Liu is supported by the NSFC Fund (No. 61370143). Bo Li is supported by the NSFC Fund (Nos. 61562062 and 61762064). Ligang Liu is supported by the NSFC Fund (Nos. 61672482, 61672481, 11626253) and the One Hundred Talent Project of the Chinese Academy of Sciences.

References

[1] S. Rusinkiewicz and M. Levoy, "Qsplat: a multiresolution point rendering system for large meshes," in *SIGGRAPH*, 2000, pp. 343–352.

[2] M. Zwicker, H. Pfister, J. van Baar, and M. H. Gross, "Surface splatting," in *SIGGRAPH*, 2001, pp. 371–378.

[3] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, and C. Silva, "State of the art in surface reconstruction from point clouds," *Eurographics Star Reports*, vol. 1, no. 1, 2014.

[4] C. Lange and K. Polthier, "Anisotropic smoothing of point sets′ ," *Computer Aided Geometric Design*, vol. 22, no. 7, pp. 680–692, 2005.

[5] E. Grilli, F. Menna, and F. Remondino, "a review of point clouds segmentation and classification algorithms," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 339–344, 2017.

[6] M. Pauly, R. Keiser, and M. Gross, "Multi-scale feature extraction on point-sampled surfaces," vol. 22, no. 3, pp. 281–289, 2003.

[7] K. Demarsin, D. Vanderstraeten, T. Volodine, and D. Roose, "Detection of closed sharp edges in point clouds using normal estimation and graph theory," *Computer-Aided Design*, vol. 39, no. 4, pp. 276–283, 2007.

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

[9] H. Hoppe, T. DeRose, T. Duchamp, J. A. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1992*, 1992, pp. 71–78.

[10] B. Li, R. Schnabel, R. Klein, Z. Cheng, G. Dang, and S. Jin, "Robust normal estimation for point clouds with sharp features," *Computers & Graphics*, vol. 34, no. 2, pp. 94–106, 2010.

[11] J. Zhang, J. Cao, X. Liu, J. Wang, J. Liu, and X. Shi, "Point cloud normal estimation via low-rank subspace clustering," *Computers & Graphics*, vol. 37, no. 6, pp. 697–706, 2013.

[12] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang, "Edge-aware point set resampling," *ACM Transactions on Graphics*, vol. 32, pp. 9:1–9:12, 2013.

[13] N. Amenta and M. W. Bern, "Surface reconstruction by voronoi filtering," *Discrete & Computational Geometry*, vol. 22, no. 4, pp. 481–504, 1999.

[14] D. Ouyang and H. Y. Feng, "On the normal vector estimation for point cloud data from smooth surfaces," *Computer-Aided Design*, vol. 37, no. 10, pp. 1071–1079, 2005.

[15] T. K. Dey and S. Goswami, "Provable surface reconstruction from noisy samples," *Comput. Geom.*, vol. 35, no. 1-2, pp. 124–141, 2006.

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

[17] T. R. Jones, F. Durand, and M. Zwicker, "Normal improvement for point rendering," *IEEE Computer Graphics and Applications*, vol. 24, no. 4, pp. 53–56, 2004.

[18] F. Calderón, U. Ruiz, and M. Rivera, "Surface-normal estimation with neighborhood reorganization for 3d reconstruction," in *Progress in Pattern Recognition, Image Analysis and Applications*, 2007, pp. 321–330.

[19] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Point set surfaces," in *IEEE Visualization 2001, October 24-26, 2001, San Diego, CA, USA, Proceedings*, 2001, pp. 21–28.

[20] N. J. Mitra, A. Nguyen, and L. J. Guibas, "Estimating surface normals in noisy point cloud data," *Int. J. Comput. Geometry Appl.*, vol. 14, no. 4-5, pp. 261–276, 2004.

[21] M. Pauly, R. Keiser, L. Kobbelt, and M. H. Gross, "Shape modeling with point-sampled geometry," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 641–650, 2003.

[22] F. Cazals and M. Pouget, "Estimating differential quantities using polynomial fitting of osculating jets," in *Eurographics/acm Siggraph Symposium on Geometry Processing*, 2003, pp. 177–187.

[23] G. Guennebaud and M. Gross, "Algebraic point set surfaces," in *ACM SIGGRAPH*, 2007, p. 23.

[24] 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.

[25] M. Yoon, Y. Lee, S. Lee, I. P. Ivrissimtzis, and H. Seidel, "Surface and normal ensembles for surface reconstruction," *Computer-Aided Design*, vol. 39, no. 5, pp. 408–420, 2007.

[26] B. Mederos, L. Velho, and L. H. de Figueiredo, "Robust smoothing of noisy point clouds," in *PROC. SIAM CONFERENCE ON GEOMETRIC DESIGN AND COMPUTING*, 2003.

[27] Y. Wang, H.-Y. Feng, F.-É. Delorme, and S. Engin, "An adaptive normal estimation method for scanned point clouds with sharp features," *Computer-Aided Design*, vol. 45, no. 11, pp. 1333 – 1348, 2013.

[28] J. Wang, K. Xu, L. Liu, J. Cao, S. Liu, Z. Yu, and X. D. Gu, "Consolidation of low-quality point clouds from outdoor scenes," *Computer Graphics Forum*, vol. 32, no. 5, p. 207?216, 2013.

[29] 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, no. C, pp. 106–116, 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2827998, IEEE Transactions on Visualization and Computer Graphics

[30] A. Boulch and R. Marlet, "Deep learning for robust normal estimation in unstructured point clouds," vol. 35, no. 5, pp. 281–290, 2016.

[31] P. S. Wang, Y. Liu, and X. Tong, "Mesh denoising via cascaded normal regression," *Acm Transactions on Graphics*, vol. 35, no. 6, p. 232, 2016.

[32] Y. Sun, S. Schaefer, and W. Wang, "Denoising point sets via l 0 minimization," *Computer Aided Geometric Design*, vol. 35, pp. 2–15, 2015.

[33] Y. Zheng, G. Li, S. Wu, Y. Liu, and Y. Gao, "Guided point cloud denoising via sharp feature skeletons," *Visual Computer*, pp. 1–11, 2017.

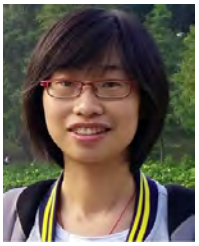
[34] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *Acm Transactions on Graphics*, vol. 26, no. 3, p. 96, 2007.

[35] G. Taubin, "Linear anisotropic mesh filters," 2001, pp. 110–51.

[36] K. P. Min, S. J. Lee, and K. H. Lee, "Multi-scale tensor voting for feature extraction from unstructured point clouds," *Graphical Models*, vol. 74, no. 4, pp. 197–208, 2012.

[37] X. C. Wang, J. J. Cao, X. P. Liu, B. J. Li, X. Q. Shi, and Y. Z. Sun, "Feature detection of triangular meshes via neighbor supporting," *Journal of Zhejiang University-SCIENCE C (Computers & Electronics)*, vol. 13, no. 6, pp. 440–451, 2012.

[38] J. Zhang, "Surfels: Surface elements as rendering primitives," pp. 335–342, 2000.



Jie Zhang received the PhD degree in 2015 from the Dalian University of Technology, China. She is currently a lecturer with the School of Mathematics, Liaoning Normal University, China. Her current research Interests include geometric processing and machine learning.

Junjie Cao received the BSc degree in 2003 and the PhD degree in 2010 from Dalian University of Technology, china. He is a lecturer with the School of Mathematical Sciences, Dalian University of Technology, China. Between 2014 and 2015, he paid an academic visit to Simon Fraser University, Canada. His research interests include geometric processing and image processing.

Xiuping Liu received the BSc degree in 1990 and the PhD degree in 1999 from Jilin University, China and Dalian University of Technology, China, respectively. She is a professor at the Dalian University of Technology. Between 1999 and 2001, she conducted research as a postdoctoral scholar in the School of Mathematics, Sun Yat-sen University, China. Her research interests include shape modeling and analyzing.

He Chen is a graduate student in computational mathematics, Dalian University of Technology, china. He received bachelorł degree of electrical engineering from Hunan University of China, and turns to study computational mathematics in his postgraduate education. His research interests include point cloud normal estimation and registration.

Bo Li received the Ph.D. degree in 2008 from Dalian University of Technology, China. Now he is an associate professor with the School of Mathematics and Informatic Science, Nanchang Hangkong University, China. His current research interests include the areas of image processing and computer graphics.

Ligang Liu received the BSc degree in 1996 and the PhD degree in 2001 from Zhejiang University, China. He is a professor at the University of Science and Technology, China. Between 2001 and 2004, he was at Microsoft Research Asia. Then he was at Zhejiang University during 2004 and 2012. He paid an academic visit to Harvard University during 2009 and 2011. His research interests include geometric processing and image processing. His research works could be found at his research website: http://staff.ustc.edu.cn/ lgliu