CGI2018 manuscript No.
(will be inserted by the editor)

Robust Normal Estimation in Unstructured 3D Point Clouds by Selective Normal Space Exploration

Claudio Mura · **Gregory Wyss** · **Renato Pajarola**

Abstract We present a fast and practical approach for estimating robust normal vectors in unorganized point clouds. Our proposed technique is robust to noise and outliers and can preserve sharp features in the input model while being significantly faster than the current state-of-the-art alternatives. The key idea to this is a novel strategy for the exploration of the normal space: first, an initial candidate normal vector, optimal under a robust least median norm, is selected from a discrete sub-region of this space, chosen conservatively to include the correct normal; then, the final robust normal is computed, using a simple, robust procedure that iteratively refines the candidate normal initially selected. This strategy allows us to reduce the computation time significantly with respect to other methods based on sampling consensus and yet produces very reliable normals even in the presence of noise and outliers as well as along sharp features. The validity of our approach is confirmed by an extensive testing on both synthetic and real-world data and by a comparison against the most relevant state-of-the-art approaches.

Keywords Normal estimation · Point cloud processing · Robust statistics

1 Introduction

Recent advances in 3D acquisition technologies have lead to scanning devices capable of capturing point-based representations of complex objects and environments in a matter of minutes. The resulting models typically consist of many thousands or even millions of individual 3D points and are commonly known as *point clouds*. Thanks to the research efforts in the domain of point-based graphics [12], it is nowadays possible to directly use point cloud data in rendering [27,19] as well as in modeling pipelines [2,25,6], without requiring an explicit reconstruction step to extract a surface mesh from the 3D points.

Often, scanning devices only output a set of raw and unstructured 3D samples, which lack any kind of connectivity or higher level information about the underlying surface. Even for pipelines that work directly on point data, some basic additional information is needed besides the position of the scanned samples – first and foremost, per-point normal vectors. Since such vectors represent useful first-order information on the real-world object represented by the point

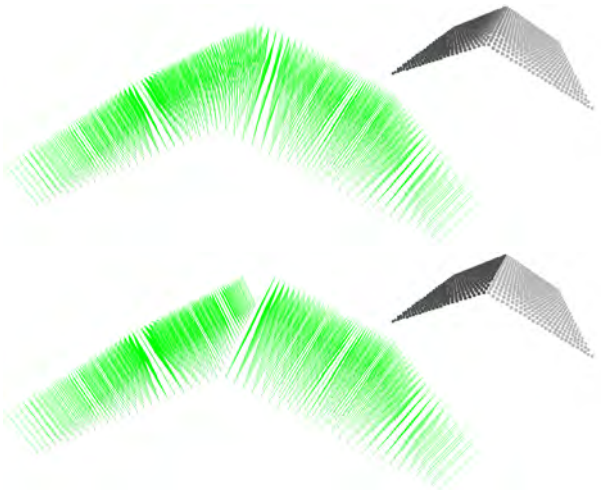


Fig. 1: Along sharp edges, PCA-based techniques erroneously estimate smooth normal vectors (TOP), which differ significantly from the correct ones (BOTTOM). The effect is particularly evident when computing lighting information based on the estimated normals (see top-right corners).

C. Mura, G. Wyss, R. Pajarola
Department of Informatics, University of Zurich, Switzerland
Binzmühlestrasse 14, 8050 Zurich, Switzerland
E-mail: {claudio | wyss | pajarola}@ifi.uzh.ch

set, they are of fundamental importance not only in point-based pipelines, but also for more general geometry processing tasks such as surface reconstruction.

However, the computation of normals in real-world datasets is hindered by many issues. Besides the unstructured nature of the input point cloud, imperfections in the sensing technologies and in the acquisition process can result in noisy measurements, in the presence of scattered outliers and in an irregular distribution of samples on the scanned entities. Moreover, many man-made objects exhibit sharp features like corners or edges, which are easily lost when applying standard low-pass noise filtering methods (see Fig. 1).

For all these reasons, normal estimation in point clouds is a well-studied and yet still challenging problem in the computer graphics and computer vision domains. Over the last decades, researchers have proposed a number of different approaches, each focusing on some specific aspects of this multi-faceted task. In their seminal work [15], Hoppe et al. propose estimating the normal vector at a point by performing a *Principal Component Analysis* (PCA) over its set of neighbors. While effective against Gaussian measurement noise, this basic approach is not robust against other real-world issues (e.g. outliers) and does not preserve sharp features. Many approaches build on top of this method by improving its theoretical properties [23] or by using techniques from robust statistics [16] to preserve sharp features and cope with outliers and irregular sampling [20,7]. However, these method are often based on complex random sampling processes that generate a very high number of candidate solutions; this can results in a high number of input parameters to be tweaked and – more importantly – leads to very high computation times when processing real-world data composed of millions of points.

In this paper, we propose a novel approach for normal estimation that not only is robust to noise and outliers in the input data but also preserves sharp features while being significantly faster as well as simpler to implement than other state-of-the-art alternatives. Our method combines and extends in a non-trivial way several techniques proposed by previous researchers. In particular, we observe that the methods that sample the solution space typically explore an overly large and unnecessary region of this parameter space (often, its entirety), with the goal of computing the minimum number of samples that need to be drawn to ensure correct results. Instead, we discretize the solution space and use the estimated PCA normal and the local properties of each point to define a sub-region of this discretized space in which the correct normal can lie. We then focus on exploring this constrained region systematically, extracting the optimal discrete normal from it, and eventually use its support points (inliers) to robustly compute the final normal vector. This idea allows us to extract accurate normals even in the presence of sharp features (e.g. edges or corners), minimizing the influence of noise and outliers, and at a fraction of the computational complexity of previous robust solutions.

A thorough testing on both synthetic and real-world datasets, as well as a comparison against other state-of-the-art pipelines, demonstrate that our method works well in different settings and represents a valid practical alternative to other more elaborate yet also more expensive approaches.

2 Related Work

The estimation of normal vectors in 3D point clouds is a fundamental problem that has been widely studied in the graphics and vision communities over the last decades.

One of the first normal estimation techniques for unorganized point clouds was presented in the context of surface reconstruction by Hoppe et al. [15]. In their work, the normal of a point is computed as the eigenvector with the smallest eigenvalue of the covariance matrix constructed from the k-nearest neighbors. This approach, which uses PCA for the computation of the eigenvectors, is simple to implement and works well in the presence of purely Gaussian noise; however, due to its inherent low-pass filtering effect, it has the major drawback of smoothing the normals around sharp edges. Researchers have tried to solve this issue in several ways, for instance by assigning Gaussian weights to the neighboring points [25] and by adapting the radius of the local neighborhood based on local estimates of curvature, noise-scale and sampling density [23]. Yoon et al. [30] improve the robustness of the PCA by using ensemble techniques from statistics, while other approaches [2,9,13] fit higher-level surfaces like algebraic spheres and quadrics instead of planes, as done in the other regression-based methods. These improvements only manage to mitigate the undesired smoothing effect and do not address another major issue of real-world point clouds, i.e. the presence of outliers. In contrast to this, our method only uses PCA as a last step to calculate the final normal, applying it in an iterative manner and only to a set of inlier points selected with a criterion inspired by robust statistics.

In fact, robust statistics is at the basis of many different approaches to estimate normals in point clouds [22,11,28]. Such methods are generally robust to outliers and can preserve sharp features, but result in very costly computational pipelines. This is the case for the method by Li et al. [20], who propose a robust noise-scale estimator as well as an objective function based on *Kernel Density Estimation*(KDE), and for more recent approaches based on the extraction of anisotropic neighborhoods along sharp features [29,31]. Although efficient optimizations have been proposed [21], the complexity of these methods remains high in practical settings. The approach that is most similar in spirit to ours is the one by Boulch and Marlet [7], who draw triplets from

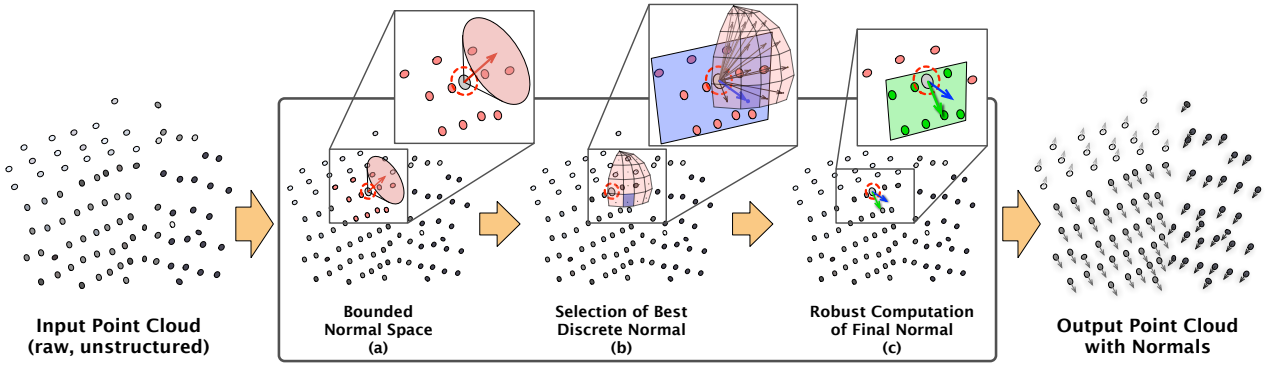


Fig. 2: Visual overview of our pipeline. For each point of the input cloud, we first compute its PCA normal using the points in its local neighborhood and define a sub-region of the normal space in which the correct normal can lie (a). We then discretize such sub-region and consider all the discrete normals as candidates for a locally tangent plane, selecting the one with minimal median distance to the points in the neighborhood (b). As a last step (c), we estimate the final normal by iteratively performing a PCA on a set of inliers that is refined robustly at each iteration.

the neighborhood of a point and accumulate the corresponding normals in a uniformly discretized unit hemisphere, representing the space of all possible normals. This method achieves robustness by generating a high number of candidate normals for each point (similar to a plain RANSAC procedure [10]), which significantly increases the computation time. In contrast, we first limit the normal search space to a cone in which the correct normal should lie, and then use a robust iterative procedure to compute the final normal.

At the other end of the spectrum compared to the expensive pipelines based on robust statistics are *mollification* methods, which use robust filtering techniques to iteratively refine an initial field of noisy normals [17,13,32]. These methods require good input normals to produce consistent results, but they have the advantage of being fast and easy to implement – a trait that is shared by our own method.

With the recent breakthroughs in the field of deep learning, several data-driven approaches have been proposed also for normal estimation. In particular, Boulch and Marlet [8] revisited their method based on the Hough transform [7] by applying a *Convolutional Neural Network* to an adapted version of the original accumulator. Very recently, Guerrero et al. [14] have presented a patch-based learning pipeline that estimates local properties in point clouds, including normals and curvature. These approaches achieve superior results, at the cost of a non-trivial learning infrastructure, which is in contrast to our practical and easy-to-implement solution.

It is worth noting that all general surface reconstruction approaches which generate a surface from a point cloud can be used to derive point normals [3,18]. Given that the reconstruction is not limited to smooth surfaces, the normals will be faithful near sharp edges and corners too. However, feature-preserving surface reconstruction methods typically involve more complicated geometry processing steps and expensive numerical computations. For an extensive review on these methods, we refer to the survey of Berger et al. [4].

3 Robust Normal Estimation

Our approach computes the normal of each point of the input cloud independently, through a local analysis of its neighboring points. The key idea is to compute from the properties of the neighborhood the bounds of the sub-region of the normal space in which the correct normal must lie. We uniformly subdivide this region (which corresponds to a solid angle) into discrete patches and select the patch corresponding to the direction that provides the tangent plane with the least median distance to the points in the neighborhood. From this plane, we robustly select the inlier points that are input to the PCA calculation, using an iterative procedure that yields the final estimated normal.

The main steps of our method, also shown in Fig. 2, are:

1. definition of a limited solution space, expressed as a diverging angle from an initial PCA normal (Sec. 3.1);
2. discretization of the limited space and selection of the candidate normal providing the tangent plane with the least median distance to the point's neighbors (Sec. 3.2);
3. computation of the final normal based on iterating a robust selection of the inliers of the chosen tangent plane and a PCA-based normal update (Sec. 3.3).

3.1 Limited Sample Space

The goal of this step is to limit the space of the solutions in which we have to search for the correct normal. In particular, we use the normal obtained from the PCA of the k nearest neighbors $\mathcal{N}_{\boldsymbol{p}}$ of a point $\boldsymbol{p}$ as an initial mean normal vector $\bar{\boldsymbol{n}}$ and then compute an upper bound to the maximum angular deviation α between this normal and the correct one. To find this angular bound, we move from the key results obtained by Mitra and Nguyen [23]: their study introduces the formula 1, which expresses the maximum angular error α

between the PCA normal $\bar{\boldsymbol{n}}$ and the correct one in terms of the curvature κ, the search radius r, the noise-scale σ_n and the sampling density ρ. The formula further depends on the small constant values c_1, c_2 and c_3, linked to the properties of the specific point cloud, and on the error tolerance ε.

$$\alpha \leq c_1 \kappa r + c_2 \frac{\sigma_n}{\sqrt{\varepsilon \rho}\, r^2} + c_3 \frac{\sigma_n^2}{r^2} \tag{1}$$

In their paper, Eq. 1 is used to compute a varying search radius r that defines the size of the local neighborhood for the PCA-based normal estimation and that is larger in planar non-noisy areas, and smaller in noisy high-curvature regions. In our approach, this formula is used to limit the solution space to a solid angle around the initial PCA normal. This of course relies on the assumption that the quantities used in the formula are estimated correctly.

In an extension of their work [24], Mitra and colleagues propose the values $c_1 = 1$, $c_2 = 4$ and $c_3 = 1$ for the constants in the formula, and define the curvature and the sampling density, respectively, as $\kappa = \frac{2d}{\mu^2}$ and $\rho = \frac{k}{\pi s^2}$. In these formulas, d is the distance from the investigated point $\boldsymbol{p}$ to the least squares fitted plane in the k-nearest neighborhood, μ is the average distance from $\boldsymbol{p}$ to all its neighbors and s is the distance from $\boldsymbol{p}$ to the k-th neighbor, with k being the number of points in the local neighborhood. The estimate of the noise level σ_n has to be provided as input to the formula.

These estimates suffer from a number of drawbacks. In particular, the estimate for the curvature does not depend on the local noise level, which can result in overly high curvature values in noisy, planar neighborhoods; moreover, defining μ in terms of the mean of the distances makes the estimation not robust to outliers. With respect to the sampling density, its estimate is based on two points only, i.e. $\boldsymbol{p}$ and its k-th neighbor, which makes it unreliable in case one of them is affected by noise or is an outlier.

We propose instead to use more robust estimators for the curvature κ and the sampling density ρ, using information already available from the computation of the initial PCA normal. In particular, we define κ using the ratio between the smallest eigenvalue λ_1 and all three eigenvalues λ_1, λ_2 and λ_3 combined:

$$\kappa = \max\left(\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} - \sigma_n,\, 0\right), \tag{2}$$

Note that subtracting the estimated noise compensates for the fact that the noise is encoded in the eigenvalues, making this estimator more robust to noise than the original one due to the inherent low-pass filtering. We estimate the noise scale σ_n as the median distance of the points in the neighborhood $\mathcal{N}_{\boldsymbol{p}}$ to the least squares fitted plane.

As for our sampling density estimator, we define it as $\rho = 2 \cdot \frac{k}{\pi d_{med}^2}$, where d_{med} denotes the median distance from $\boldsymbol{p}$ to its k neighbors $\mathcal{N}_{\boldsymbol{p}}$. With respect to the original formulation, the key difference lies in using d_{med} instead of the distance from $\boldsymbol{p}$ to the k-th neighbor: this increases the robustness of the estimate, as the k-th neighbor can be an outlier located at any arbitrary distance from $\boldsymbol{p}$.

Our improved definitions of κ and ρ are less sensitive to defects in the input data than the ones of Mitra et al. [24]. This can be noticed in the example of Fig. 3: in particular, due to the presence of noise, the original estimator for κ produces some high values even in flat areas, whereas our formulation results in uniformly low curvature in the entire flat region. This greatly increases the effectiveness of Eq. 1 in practice. Using the corresponding value α, we can define a region in the normal space around the initial PCA normal $\bar{\boldsymbol{n}}$ and ensure that the correct normal lies within it with a probability of $1 - \varepsilon$, where ε is the error tolerance used in Eq. 1 and which we set to 0.005. This results in a probability of 99.5% that the correct normal is included in the sub-region of the normal space analyzed.

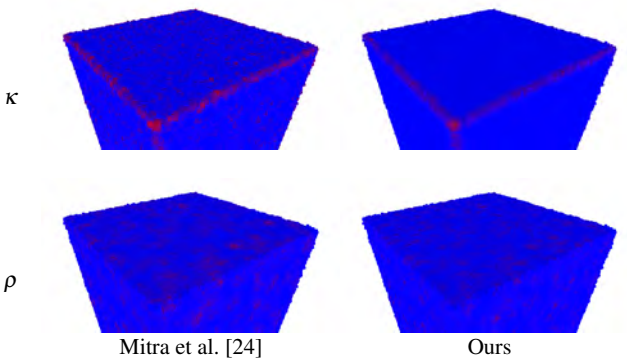
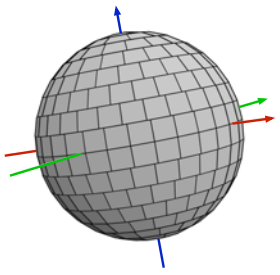


Fig. 3: Comparison between the estimators for curvature κ and sampling density ρ of Mitra et al. [24] (LEFT) and our alternative formulations (RIGHT). The values are visualized in color-coding from blue (low) to red (high).

3.2 Selection of Best Discrete Normal

After restricting the final solution space to the cone around $\bar{\boldsymbol{n}}$ given by the solid angle α, we consider the surface segment spanned by α on the discrete unit sphere. Our discretization (shown in the inset) is inspired by a well-established accumulator design [5], also used by Boulch and Marlet [7]. In particular, we first partition the sphere into N_S slices along the longitudinal axis; each such slice is delimited by two parallels and spans the same angle of $\frac{\pi}{N_S+1}$. We then further subdivide each slice according to the meridians, adapting the number of subdi-



visions depending on the longitudinal angle of the slice (i.e. the angle between the middle of the slice and the plane at the equator), so that the number of subdivisions at the equator is $2 \cdot N_S$ and that the resulting patches on the sphere have similar area. In our implementation, there are two additional longitudinal slices for the two poles, each covering an angle of $\frac{1}{2} \cdot \frac{\pi}{N_S+1}$ and associated to a single patch.

Given this discretization of the sphere, we extract the set of patches that correspond to the solid angle α around the initial normal $\bar{\boldsymbol{n}}$ and consider the vectors that originate in the sphere center and end in the midpoints of these patches. For each of these vectors, which represents a candidate normal $\tilde{\boldsymbol{n}}_j$, we compute the distances from the k-nearest neighbors $\mathscr{N}_{\boldsymbol{p}}$ to the plane defined by the analyzed point $\boldsymbol{p}$ and $\tilde{\boldsymbol{n}}_j$, saving the median distance. We then compare the median distances of all investigated candidate normals $\tilde{\boldsymbol{n}}_j$ and select the normal $\hat{\boldsymbol{n}}$ that corresponds to the minimal median distance as our best discrete normal:

$$\hat{\boldsymbol{n}} = \underset{\tilde{\boldsymbol{n}}_j}{\operatorname{argmin}}\, \mathit{median}(\, \mathit{distances}(\, \mathit{plane}(\tilde{\boldsymbol{n}}_j, \boldsymbol{p}), \mathscr{N}_{\boldsymbol{p}}\,)\,). \qquad (3)$$

This best discrete normal selection process is illustrated in Steps (a) and (b) of Fig. 2.

3.3 Robust Computation of the Final Normal

Having evaluated an initial best (discrete) normal $\hat{\boldsymbol{n}}$ in the restricted sub-region of the true solution space, we apply an iterative procedure to refine it and obtain the final normal $\boldsymbol{n}^*$. Starting from $\boldsymbol{n}_0 = \hat{\boldsymbol{n}}$, we perform the update $\boldsymbol{n}_i \rightarrow \boldsymbol{n}_{i+1}$ according to the following steps:

1. compute the median $d_{med} = \mathit{median}(\{d_j |\, \boldsymbol{p}_j \in \mathscr{N}_{\boldsymbol{p}}\})$ of all distances d_j to the plane given by $\boldsymbol{p}$ and the current normal estimate $\boldsymbol{n}_i$;
2. select all inliers $\mathscr{I} = \{\boldsymbol{p}_j |\, d_j \leq d_{med}\} \subseteq \mathscr{N}_{\boldsymbol{p}}$;
3. compute updated normal $\boldsymbol{n}_{i+1}$ from the PCA of $\mathscr{I}$;
4. if $i = 2$ or if $|\boldsymbol{n}_i \cdot \boldsymbol{n}_{i+1}| > 1 - \varepsilon_{\boldsymbol{n}}$, let $\boldsymbol{n}^* = \boldsymbol{n}_{i+1}$ and terminate the refinement;
5. let $i \rightarrow i+1$ and restart at 1.

Here $\varepsilon_{\boldsymbol{n}}$ is a small constant that we set to 1^{-4}. Note that we stop the refinement if the updated normal does not differ significantly from the previous one (i.e. if the dot product of $\boldsymbol{n}_i$ and $\boldsymbol{n}_{i+1}$ is almost 1) and that we perform a maximum of 3 iterations, as our tests revealed that this is sufficient to achieve good results. This procedure is simple, yet very effective and results in a robust final normal estimation $\boldsymbol{n}^*$ for the analyzed point $\boldsymbol{p}$, corresponding to Step (c) in Fig. 2.

4 Results

We validated our approach under a variety of different settings, using a set of test models that includes both synthetic and real-world point clouds. The synthetic models (Figs. 4(a)-(d)) were obtained by sampling a set of mesh models with an open-source tool [1] and then corrupting each sample with additive gaussian noise along the direction of its normal; to this purpose, we used a normal distribution with standard deviation $\sigma = 0.1\% \cdot d_{BB}$, where d_{BB} is the diagonal of the bounding box of the model. The real-world model (Fig. 4(e)) was acquired using a phase-shift static laser scanner.

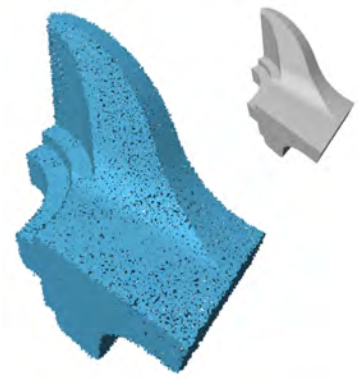
We implemented a prototype of our method in C++, using the PCL library [26]; the computation is single-threaded, although the code is trivially parallelizable, since each normal is computed independently. All timings were taken on a *MacBook Pro* equipped with a *Intel Core i7* processor clocked at *2.5 GHz* and with *16 GB DDR3 RAM*.

Our pipeline depends on two main parameters: the number of nearest neighbors k used to estimate the initial normal and the resolution of the discretization of the normal space, expressed as the number of vertical slices N_S of the accumulator (see Sec. 3.2). In all our experiments, we used $k = 64$ and $N_S = 16$. We set all constants c_1, c_2, c_3 in Eq. 1 to 1.

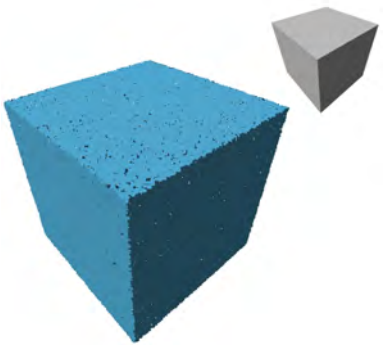
The tests performed are focused on the following aspects: a qualitative evaluation, based on the analysis of the point clouds rendered with shading using the estimated normals; a quantitative evaluation (performed on the synthetic datasets, for which ground-truth normal vectors are available); a comparison with the most related alternative approaches, which considers both the quality of the results and the efficiency of the computation; an analysis of the robustness of the approach with respect to increasing levels of noise and outliers and to variations in the parameters.

4.1 Qualitative Analysis

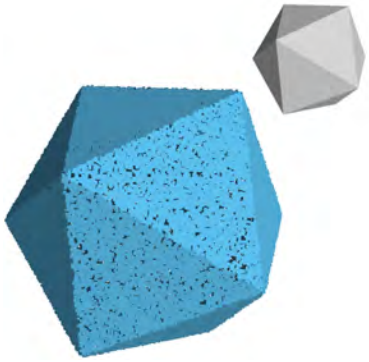
An immediate way of assessing the quality of the estimated normals is to examine the visual variations of shades in the rendered point clouds. In particular, curved regions should exhibit a smooth transition between different shades, whereas a sudden variation in shading should appear along sharp features. These properties are generally valid for our estimated normals and are evident in the synthetic models 'Double Torus' and 'Box' (Fig. 5); nevertheless, the edge-preseving effect of our method can also be noticed in the real-world dataset 'SoundLab' (Fig. 6), in particular along adjacent wall surfaces and in the room corner. As expected, a minor smoothing can still be noticed for the points that lie in the immediate proximity of actual sharp edges (see Fig. 5). However, this case is handled coherently for all points that are equally close to the edge, achieving a very similar overall result as Boulch and Marlet [7].



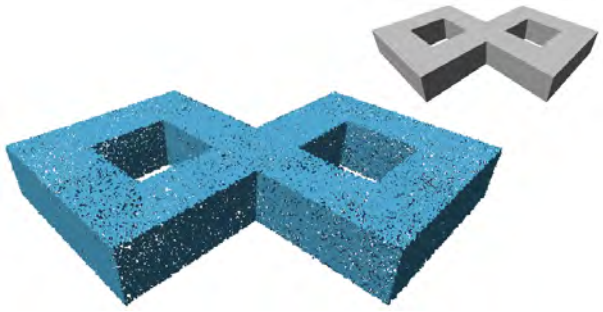
(a) Fandisk (100K points)



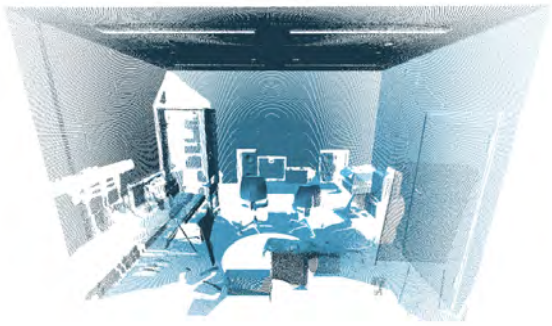
(b) Box (100K points)



(c) Icosahedron (100K points)



(d) Double Torus (100K points)



(e) SoundLab (668K points)

Fig. 4: Overview of the datasets used in the evaluation. Note that point clouds (a)-(d) were obtained by sampling a mesh model (shown in the top right corner) and then adding gaussian noise along the direction of the normal.

4.2 Quantitative Analysis

In the case of synthetic models, the availability of ground-truth surface normals allows to accurately measure the angular error for each estimated normal. We show such errors in the color-coded visualization of Fig. 7. Especially in the case of 'Box' and 'Double Torus' it is easy to see that most of the errors appear exactly on the edges, which shows that the smoothing effect is only limited to the regions in which distinguishing between two adjacent surfaces is inherently ambiguous due to the presence of noise. Note that this behavior is not fully verified in the case of 'Fandisk': in addition to sharp features, this model also contains large smooth regions, in which a significant fraction of the higher errors are located. As discussed in the next section (Sec. 4.3), these cases are best addressed with a more standard PCA-based approach, which however behaves significantly worse in the presence of sharp features.

This analysis is confirmed by the statistics of Tab. 1, which show that the quality of the estimated normals is generally very high, with a percentage of errors $< 10^\circ$ that ranges from over 90% for 'Double Torus' to about 97% for 'Box'.

4.3 Comparison with Other Methods

Fig. 7 and Tab. 1 also include information on three state-of-the art techniques that we selected for comparison. Since our focus is on *practical* and *fast* computation, we do not consider more sophisticated yet computationally expensive methods like the work of Zhang et al. [31] and restrict our scope to the approaches of Hoppe et al. [15], to the improved version by Mitra et al. [23] and to the robust yet efficient approach of Boulch and Marlet [7]. To allow for a fair comparison, we used the default values for all the parameters of these pipelines; the only exception is the number of nearest neighbors k, which we fixed to 64 to ensure that the estimation is performed at the same spatial scale by all methods.

The color-coded error visualization of Fig. 7 intuitively describes the general behaviour of the methods considered. In particular, it is clear that the plain PCA estimation of Hoppe et al. [15] causes the most significant smoothing of the sharp features, as shown by the wide non-green areas around the edges; the problem persists also when using the parameters proposed by Mitra and colleagues [23]. Interestingly enough, in our tests this latter approach failed to deliver significantly better results and often performed worse

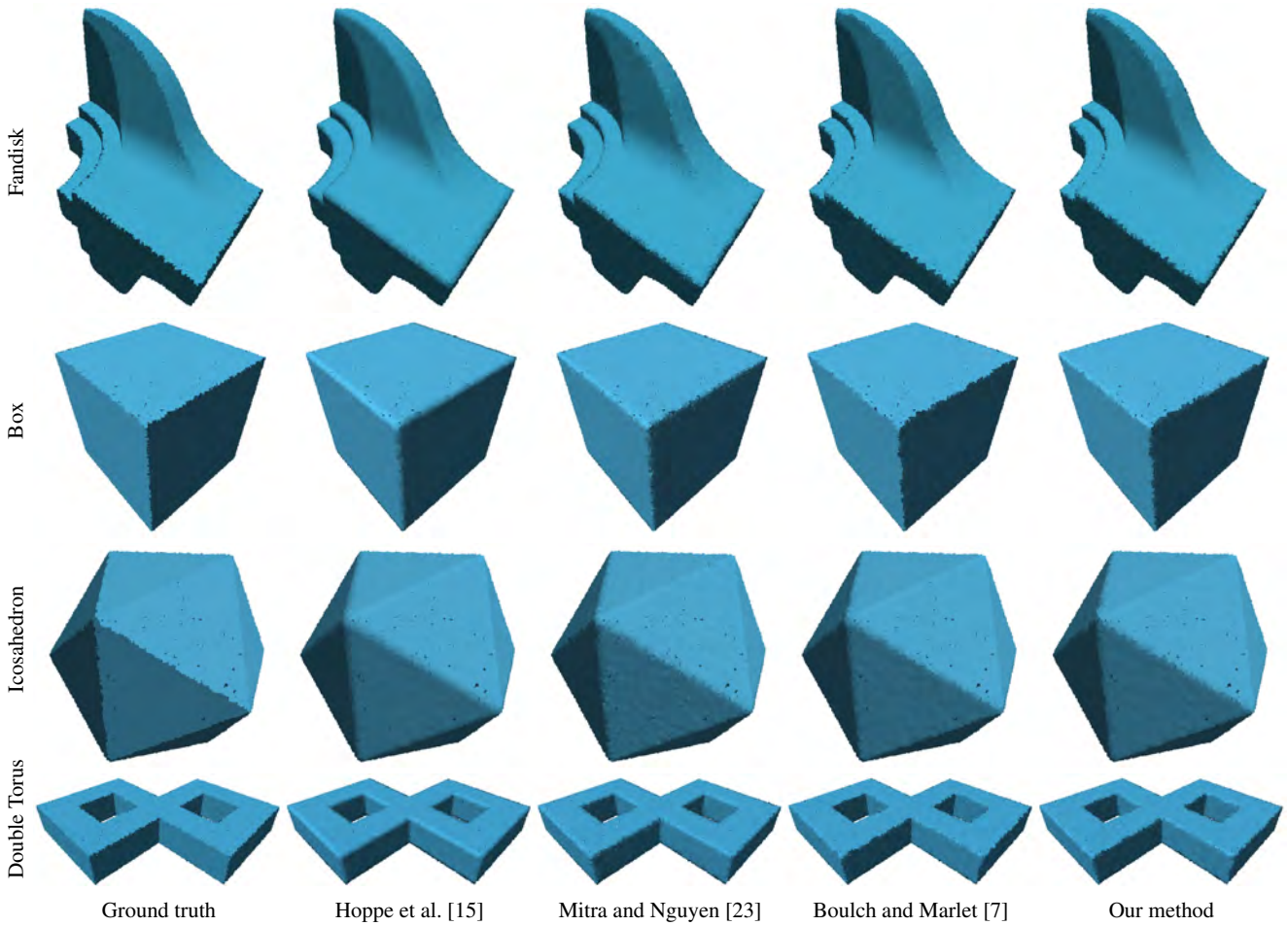


Fig. 5: Qualitative analysis of the normal estimation on synthetic data.

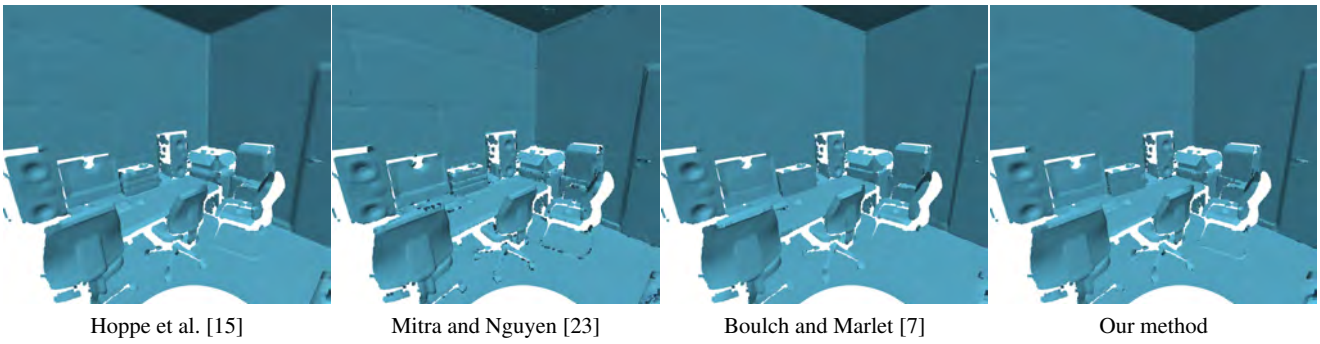


Fig. 6: Qualitative analysis of the normal estimation on real-world data ('SoundLab').

than the original method (see, in particular, 'Fandisk' and 'Double Torus', or the noisy shadings in the real-world results of Fig. 6). We attribute this to the difficulty of adapting their estimate of the search radius to the specific properties of our test models. Our approach achieves consistently better results than the two PCA-based pipelines, and attains accuracy levels that are only slightly lower than those of Boulch and Marlet [7]; still, these two pipelines exhibit similar behaviours, as made particularly evident by the results on the scanned model 'SoundLab' (Fig. 6), for which both methods manage to preserve the main sharp edges of the scene.

It is important to notice that our pipeline is consistently faster than the one proposed by Boulch and Marlet. This is clearly shown in Tab. 2, in which we compare the running times of the methods on upscaled versions of our synthetic point clouds. The speed-up is particularly evident in the real-world model 'SoundLab', which has a more irregular distribution of samples and in which our pipeline is over 7 times

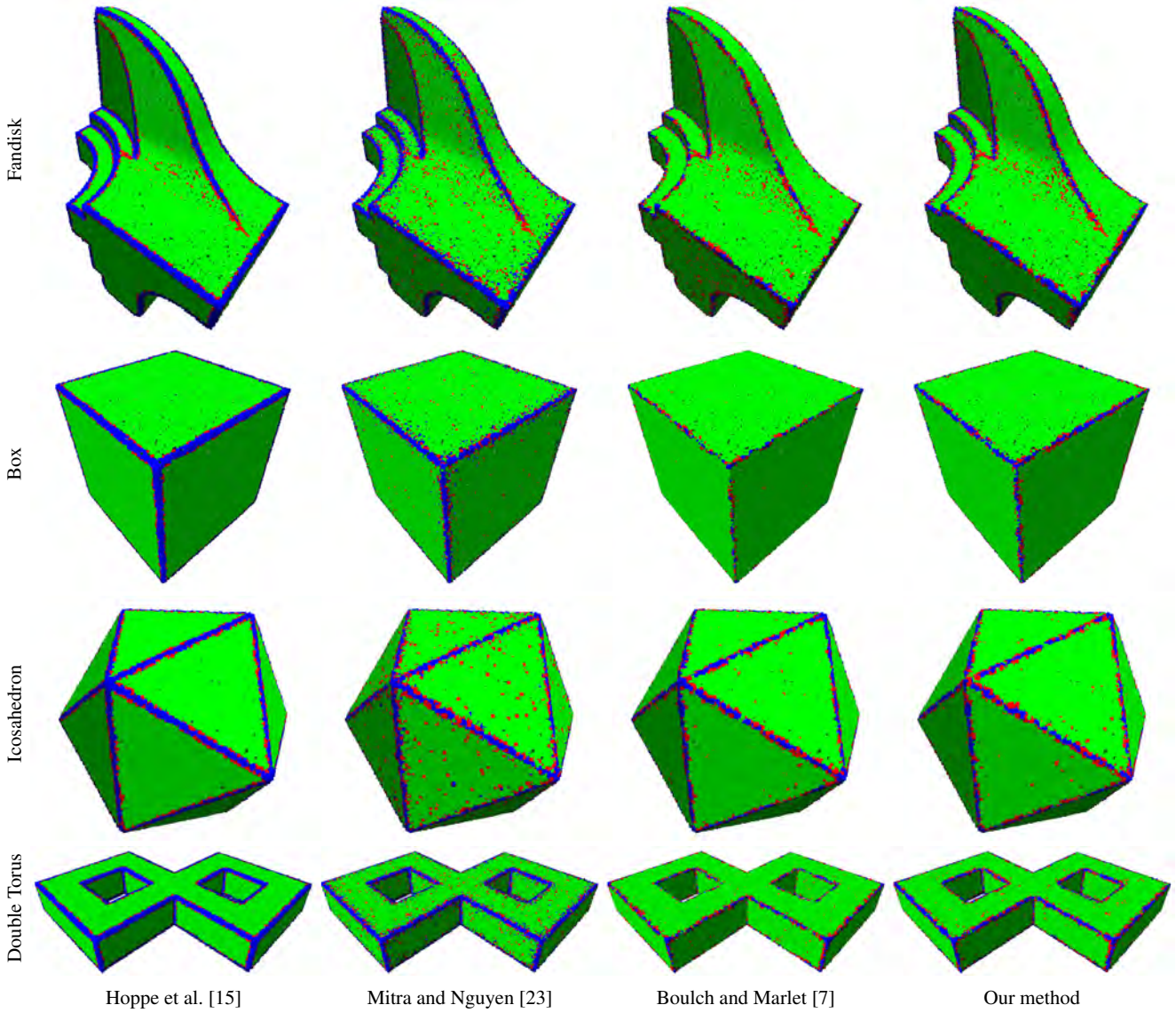


Fig. 7: Quantitative analysis of the normal estimation on synthetic data. Green denotes points with an estimated normal that deviates by less than 5° from the ground-truth; blue encodes angular deviations between 5° and 10°; red marks errors > 10°.

faster than its competitor. Interestingly enough, while the two PCA-based approaches are simpler and therefore generally faster than ours, the one by Mitra and Nguyen [23] is over 2 times slower than ours on our real-world model. We consider this as a further sign of the difficulty to adapt to the properties of different point clouds. Note that, since their method requires an estimate of the noise scale as input, we set it to $\sigma = 1mm$ based on the specifications of the scanner.

4.4 Robustness

To study how our method is affected by defects in the input point set, we corrupted the synthetic model 'Box' with increasing levels of noise and outliers. In particular, we considered the values $\sigma = 0.2\% \cdot d_{BB}$ and $\sigma = 0.4\% \cdot d_{BB}$ for the noise distribution used (in addition to $\sigma = 0.1\% \cdot d_{BB}$, already used for the other tests) and introduced a number of outliers (i.e. points offset along an arbitrary direction with respect to their original position) equal to 2.5% and 5% of the size of the point cloud. The offsets of the outliers were drawn from a uniform distribution in the range $[5 \cdot \sigma, d_{BB}/4]$. The results, listed in Tab. 3 and visualized in color-coding in Fig. 8, highlight that outliers only have a minor effect on the accuracy of the estimation; in particular, it is interesting to notice that increasing the amount of outliers for a given level of noise does not produce meaningful changes in the quality of the results. On the other hand, increasing the noise scale results in significantly higher errors in the estimated

Model	Our method			Hoppe et al. [15]			Mitra and Nguyen [23]			Boulch and Marlet [7]		
	Mean	Median	$< 10^\circ$	Mean	Median	$< 10^\circ$	Mean	Median	$< 10^\circ$	Mean	Median	$< 10^\circ$
Fandisk	4.56	2.10	92.06%	5.69	1.66	84.11%	5.86	2.08	83.59%	4.49	2.73	94.97%
Box	2.55	1.20	96.86%	3.33	0.96	91.22%	3.30	1.10	91.87%	2.40	1.49	98.78%
Icosahedron	3.16	1.85	93.58%	3.10	1.42	90.34%	3.44	1.71	90.30%	3.18	2.07	94.13%
Double Torus	5.04	2.06	90.90%	6.31	1.58	81.62%	6.53	2.03	80.94%	4.40	2.17	95.09%

Table 1: Relevant statistics on the estimation error. For each synthetic model, we show the mean and median angular error (in degrees) between estimated and ground-truth normals, as well as the percentage of errors lower than 10 degrees, attained by our method and by the state-of-the-art alternatives considered.

Model	Our method	Hoppe et al. [15]	Mitra and Nguyen [23]	Boulch and Marlet [7]
Box (250K pts)	4.36	1.82	6.71	13.11
Box (500K pts)	9.69	3.67	13.87	19.74
Double Torus (250K pts)	4.99	1.77	6.55	13.15
Double Torus (500K pts)	13.06	3.49	13.89	23.23
SoundLab (668K pts)	9.86	4.26	21.80	74.24

Table 2: Computation times of our approach and of the state-of-the-art alternatives considered. All timings are in seconds.

normals, as clearly indicated by Fig. 8. Note, however, that the results shown were obtained without changing the parameter k, which can be increased to account for the higher noise level in the data.

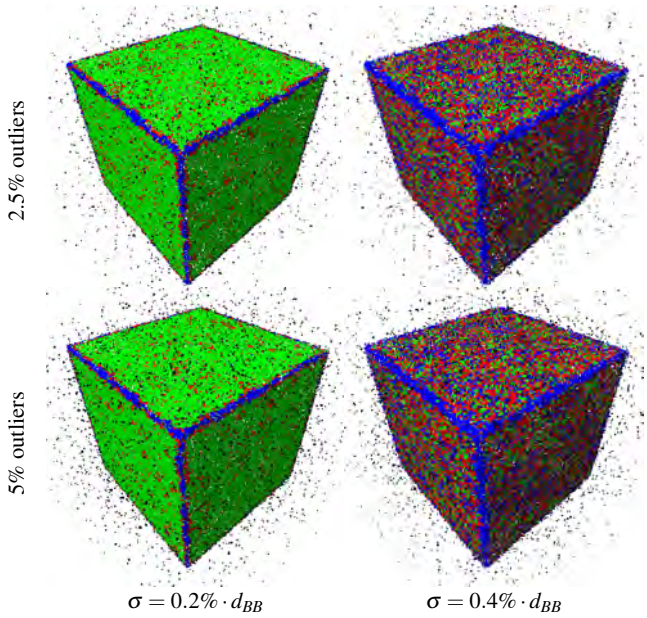


Fig. 8: Quantitative analysis of the estimation on synthetic model 'Box' for different combinations of noise levels (horizontal axis) and percentage of outliers (vertical axis). Here d_{BB} denotes the diagonal of the bounding box of the model. The color-coding scheme is the same as in Fig. 7.

We also examined the sensitivity of the pipeline to variations in the values of its parameters. As noted at the beginning of this section, the method depends only on two parameters: the number N_S of subdivisions of the accumulator (see Sec. 3.2); the number k of nearest neighbors used. We increased the value of N_S from its default 16 to the value 32 with steps of 4, noticing no meaningful changes in the mean and median errors and only marginal improvements in the percentage of angular errors $< 10^\circ$, at the cost of an increased computation time. With respect to k, we considered the values $k = 8, 16, 32, 64, 128$ and analyzed how the accuracy of the estimation for 'Box' changed, under the default noise level $\sigma = 0.1\% \cdot d_{BB}$ and in absence of outliers. For the aforementioned values of k, the fractions of errors smaller than 10° were 30.25%, 72.82%, 96.00%, 96.86%, 97.37%, respectively. This suggests that no significant improvements in accuracy are achieved for $k > 32$.

Model	Mean	Median	$< 10^\circ$
Box ($\sigma = 0.2\% \cdot d_{BB}$, 2.5% outl.)	4.53	2.75	93.75%
Box ($\sigma = 0.2\% \cdot d_{BB}$, 5% outl.)	4.52	2.72	93.61%
Box ($\sigma = 0.4\% \cdot d_{BB}$, 2.5% outl.)	9.94	8.02	65.40%
Box ($\sigma = 0.4\% \cdot d_{BB}$, 5% outl.)	9.94	7.98	65.80%

Table 3: Statistics on the estimation error on the synthetic model 'Box' for different combinations of noise level σ and percentage of outliers. Here d_{BB} denotes the diagonal of the bounding box of the model. Note that outliers did not contribute to the computation of these measures.

5 Conclusions

We have presented a novel practical approach to estimate robust normal vectors in unorganized point clouds. Our work moves from several key results from robust statistics and combines them in an original manner, resulting in a pipeline

that is at the same time robust to defects in the input data, simple to implement, computationally efficient and edge-preserving. We have demonstrated the validity of our work by extensive testing on both synthetic and scanned point clouds, showing that our estimated normals are accurate with respect to the ground-truth and at the same time provide visually consistent results when used to render the corresponding point cloud. With respect to the state-of-the-art, we showed that our method is a good compromise between quality of the estimation and computational efficiency, potentially making it the solution of choice in many real-world application scenarios.

Being fast and practical by design, our method cannot achieve the accuracy of more sophisticated approaches, like e.g. the recent data-driven pipelines based on Deep Learning [8, 14]. These should be used when the quality of the results obtained has precedence over ease of implementation and efficiency. Moreover, our approach does not completely eliminate the undesired smoothing effect for the points that lie exactly on sharp features. Coping with this issue, as well as improving the estimation on smooth regions and devising an adaptive scheme for the selection of the number of nearest neighbors k, is left for future work.

Acknowledgments. The authors would like to express their gratitude to Rafael Ballester for his help and his valuable inputs.

Compliance with Ethical Standards. Funding: This work was partially funded by the Swiss National Science Foundation (grant number 159225). Conflict of Interest: The authors declare that they have no conflict of interest.

References

1. CloudCompare (version 2.9.1). [GPL software] (2017). URL http://www.cloudcompare.org/
2. Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C.T.: Point set surfaces. In: Proceedings IEEE Visualization, pp. 21–28 (2001)
3. Avron, H., Sharf, A., Greif, C., Cohen-Or, D.: l_1-Sparse reconstruction of sharp point set surfaces. ACM Transactions on Graphics **29**(5), 135:1–12 (2010)
4. Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A., Silva, C.T.: A survey of surface reconstruction from point clouds. Computer Graphics Forum **36**(1) (2017)
5. Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A.: The 3D hough transform for plane detection in point clouds: A review and a new accumulator design. 3D Research **2**(2), 32:1–32:13 (2011)
6. Botsch, M., Kobbelt, L.: Real-time shape editing using radial basis functions. Computer Graphics Forum **24**(3), 611–621 (2005)
7. Boulch, A., Marlet, R.: Fast and robust normal estimation for point clouds with sharp features. Computer Graphics Forum **31**(5), 1765–1774 (2012)
8. Boulch, A., Marlet, R.: Deep learning for robust normal estimation in unstructured point clouds. Computer Graphics Forum **35**(5), 281–290 (2016)
9. Cazals, F., Pouget, M.: Estimating differential quantities using polynomial fitting of osculating jets. Computer Aided Geometric Design **22**(2), 121 – 146 (2005)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
11. Fleishman, S., Cohen-Or, D., Silva, C.T.: Robust moving least-squares fitting with sharp features. ACM Transactions on Graphics **24**(3), 544–552 (2005)
12. Gross, M.H., Pfister, H. (eds.): Point-Based Graphics. Series in Computer Graphics. Morgan Kaufmann Publishers (2007)
13. Guennebaud, G., Gross, M.: Algebraic point set surfaces. ACM Transactions on Graphics **26**(3), 23 (2007)
14. Guerrero, P., Kleiman, Y., Ovsjanikov, M., Mitra, N.J.: Pcpnet: Learning local shape properties from raw point clouds. Computer Graphics Forum (2018). To appear
15. Hoppe, H., DeRose, T., Duchampt, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: Proceedings ACM SIGGRAPH, pp. 71–78 (1992)
16. Huber, P.J., Ronchetti, E.M.: Robust Statistics. Wiley Series in Probability and Statistics. Wiley (2009)
17. Jones, T.R., Durand, F., Zwicker, M.: Normal improvement for point rendering. IEEE Computer Graphics and Applications **24**(4), 53–56 (2004)
18. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Transactions on Graphics **32**(3), 29:1–13 (2013)
19. Kobbelt, L., Botsch, M.: A survey of point-based techniques in computer graphics. Computers & Graphics **28**(6), 801–814 (2004)
20. Li, B., Schnabel, R., Klein, R., Cheng, Z., Dang, G., Shiyao, J.: Robust normal estimation for point clouds with sharp features. Computers & Graphics **34**(2), 94–106 (2010)
21. Liu, X., Zhang, J., Cao, J., Li, B., Liu, L.: Quality point cloud normal estimation by guided least squares representation. Computers & Graphics **51**(Supplement C), 106 – 116 (2015)
22. Miller, J.V., Stewart, C.V.: MUSE: Robust surface fitting using unbiased scale estimates. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp. 300–306 (1996)
23. Mitra, N.J., Nguyen, A.: Estimating surface normals in noisy point cloud data. In: Proceedings ACM Symposium on Computational Geometry, pp. 322–328 (2003)
24. Mitra, N.J., Nguyen, A., Guibas, L.: Estimating surface normals in noisy point cloud data. International Journal of Computational Geometry & Applications **14**(4–5), 261–276 (2004)
25. Pauly, M., Keiser, R., Kobbelt, L., Gross, M.: Shape modeling with point-sampled geometry. ACM Transactions on Graphics **22**(3), 641–650 (2003)
26. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: International Conference on Robotics and Automation (ICRA), pp. 1–4 (2011)
27. Sainz, M., Pajarola, R., Lario, R.: Points reloaded: Point-based rendering revisited. In: Proceedings Eurographics/IEEE VGTC Symposium on Point-Based Graphics, pp. 121–128 (2004)
28. Schnabel, R., Wahl, R., Klein, R.: Efficient RANSAC for point-cloud shape detection. Computer Graphics Forum **26**(2), 214–226 (2007)
29. Wang, Y., Feng, H.Y., Delorme, F.E., Engin, S.: An adaptive normal estimation method for scanned point clouds with sharp features. Computer-Aided Design **45**(11), 1333 – 1348 (2013)
30. Yoon, M., Lee, Y., Lee, S., Ivrissimtzis, I., Seidel, H.P.: Surface and normal ensembles for surface reconstruction. Computer-Aided Design **39**(5), 408 – 420 (2007)
31. Zhang, J., Cao, J., Liu, X., Wang, J., Liu, J., Shi, X.: Point cloud normal estimation via low-rank subspace clustering. Computers & Graphics **37**(6), 697–706 (2013)
32. Zheng, Y., Fu, H., Au, O.K.C., Tai, C.L.: Bilateral normal filtering for mesh denoising. IEEE Transactions on Visualization and Computer Graphics **17**(10), 1521–5130 (2011)