

Shape Detection from Raw LiDAR Data with Subspace Modeling

Jun Wang and Kai Xu

Abstract—LiDAR scanning has become a prevalent technique for digitalizing large-scale outdoor scenes. However, the raw LiDAR data often contain imperfections, e.g., missing large regions, anisotropy of sampling density, and contamination of noise and outliers, which are the major obstacles that hinder its more ambitious and higher level applications in digital city modeling. Observing that 3D urban scenes can be locally described with several low dimensional subspaces, we propose to locally classify the neighborhoods of the scans to model the substructures of the scenes. The key enabler is the adaptive kernel-scale scoring, filtering and clustering of substructures, making it possible to recover the local structures at all points simultaneously, even in the presence of severe data imperfections. Integrating the local analyses leads to robust shape detection from raw LiDAR data. On this basis, we develop several urban scene applications and verify them on a number of LiDAR scans with various complexities and styles, which demonstrates the effectiveness and robustness of our methods.

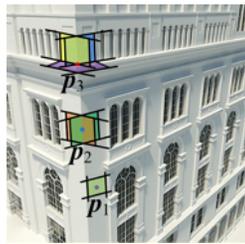
Index Terms—Urban Building, Raw LiDAR Scan, Modeling, Reconstruction, Substructure Modeling

1 INTRODUCTION

RECENT advances in light detection and ranging (LiDAR) technology greatly facilitate the acquisition of 3D point data of large-scale environments, benefiting a variety of applications such as digital city modeling [16]. However, the major drawback of LiDAR scanning is that the captured point data often suffer from low quality due to occlusion, motion, multiple reflections, etc. (see Figure 1 on the left), hindering its applications in many high level processing tasks.

For example, structure-aware modeling of 3D buildings [20] is preconditioned on structural analysis of the input, e.g., the identification of meaningful components and their mutual relations. The latter is extremely challenging with raw LiDAR data input. Some works rely on structural priors, e.g., symmetry [37] or grid structure [28] to recover the underlying structures. Nevertheless, the success of these approaches still relies on the robust detection of repetitive structures, which is also quite challenging on low-quality point cloud data. Therefore, robust processing and analysis of raw LiDAR data is a crucial problem.

We are interested in LiDAR scans of urban scenes which are mainly comprised of buildings/architectures. The surface geometry of a 3D building is generally composed of planes, cylinders and other primitive surfaces [2]. Given a point on the surface of a building, its local structure is composed of



either a single surface patch (p_1 in the inserted figure) or multiple surface patches (e.g., p_2 is at the intersection of two patches and p_3 is at the intersection of multiple).

In this paper, we propose to model the local structure around a point as a *union of substructures*, each of which corresponds to a surface patch. The local structure recovery for the point is then formulated as *substructure classification* of its neighborhood. Robustly estimating the local structures at all points naturally leads to the robust recovery of meaningful decomposition of the input scans, which will in turn facilitate the accurate processing of the point cloud, such as shape detection and reconstruction.

However, local substructure classification on raw LiDAR data is exceedingly challenging due to the following special characteristics of these data.

Density anisotropy. The density of LiDAR data depends on the measuring distance and the slope angle between the scanning direction and the scanned surface of the object [35]. The longer the distance or the bigger the slope, the sparser the scan data (see Figure 1 on the left: ①). However, most existing data classification or clustering methods are built upon the assumption that the data are distributed *uniformly* on subspaces (i.e., substructures) [9]. This is because the evaluation of a substructure is based on counting the inliers of the substructure, making it heavily dependent on the density of the point cloud [10]. Our method takes into account not only the composition of the substructure (i.e., the inliers of the substructure), but also the *size of the underlying manifold surface measured in a sampling-independent fashion*. This combination makes our clustering method particularly insensitive to the variation of density distribution (see Figure 1 on the right).

Noise and outliers. The raw LiDAR data are inevitably contaminated with severe noise and outliers (see Figure 1 on the left: ②). Most current substructure estimation methods are developed based upon singular value decomposition (SVD), principle component analysis (PCA), or their

• J. Wang is with the College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China. E-mail: junwang@outlook.com.

• K. Xu is with the School of Computer Science, National University of Defense Technology, Changsha 410073, China. E-mail: kevin.kai.xu@gmail.com.

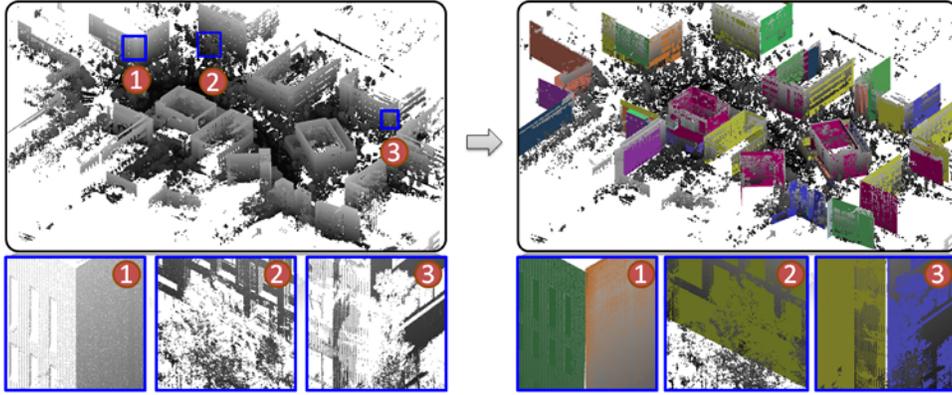


Fig. 1. A large-scale urban scene is automatically and efficiently reconstructed with patches from defect-laden, raw LiDAR point clouds using our method without user intervention. Zoom-in views demonstrate the typical characteristics of raw LiDAR data, i.e., density anisotropy, noise and outliers, and occlusions, while our approach is still capable of modeling them with good geometric fidelity to the input scene.

variants [40], [41]. These techniques are based on the least squares approach, and are, therefore, sensitive to outliers and noise [7]. We, for the first time, apply the kernel density technique to estimate the noise scale for the 3D point cloud, based on which the inliers belonging to each substructure are distinguished from outliers. Furthermore, the noise within inliers to the substructure is truncated with the residual ranking technique. As a result, *only the inliers, excluding noise, are used to characterize the substructures*, making the clustering of the substructure, and hence the discovery of local structures, especially resilient to noise (see Figure 1 on the right: ②).

Unknown number of substructures. For LiDAR data processing applications, the number of local substructures is required to determine accurately for each point. However, for a random point of a LiDAR scan, the number of substructures constituting its local structure is unknown in advance (see Figure 1 on the left: ③). Previous methods (e.g., RANSAC and related approaches) usually adopt a sequential “finding-and-removing” procedure (i.e., interleaving between parameters fitting for finding a substructure and inliers removal for consequent substructure discovery), where the number of substructures has to be prescribed. To solve this problem, we propose an effective mechanism by combining data-driven entropy thresholding and medoid shift clustering, which *automatically removes weak substructures and clusters similar significant substructures to obtain unique representative substructures*.

Contributions. We propose the first subspace modeling framework for shape detection and reconstruction from raw LiDAR data. Our substructure classification approach is inspired by the recent work on multi-structure fitting in computer vision [29]. However, we make two substantial changes against their method in order to adapt it for meeting the specific challenges in handling raw LiDAR data:

- To deal with *anisotropic density*, we devise a new density-invariant weighting metric to measure the significance of substructures in handling *non-uniform* sparse data.
- To extract geometrically similar yet distinct substructures from raw LiDAR scans, we need not only discriminate outliers but also differentiate inliers from

different substructures. To this end, we propose a new metric based on the permutation of the residual preferences of surrounding points to a substructure, which can robustly detect different substructures, regardless of how geometrically close they are, from severely corrupted point data. As a consequence, the *number of substructures* can be precisely determined, even in the presence of *heavy noise and outliers* from our method.

Overall, our main contributions are three-fold:

- 1) We propose a robust density-invariant scoring technique to measure the significance of substructures, which is insensitive to the variation of density distribution.
- 2) We design an effective substructure similarity metric based on the discriminative preference feature, empowering our method to discriminate the unknown number of subtle yet distinct substructures.
- 3) Based on the recovered local substructures, we develop robust primitive modeling, surface reconstruction applications and achieve favorable results from defect-laden, raw LiDAR data.

1.1 Related work

Shape detection and reconstruction. Surface reconstruction and modeling of indoor objects has made substantial progress in the past two decades. We refer the reader to the comprehensive survey on recent surface reconstruction techniques [3]. In this paper, we mainly focus on outdoor scenes. Recently, urban scene reconstruction has attracted increasing attention in computer graphics [21]. For instance, based on the well-known fitting technique, RANSAC, Schnabel et al. [25] proposed an automatic method to detect basic shapes from unorganized point data of outdoor buildings by decomposing the point cloud into several primitive shapes. Similarly, Lin et al. [19] designed a complete system for scene reconstruction from 3D point clouds of residential scenes captured by mobile scanners. These RANSAC-based approaches are robust to a certain level of outliers and noise. However, They have difficulty in handling sparse and incomplete LiDAR data.

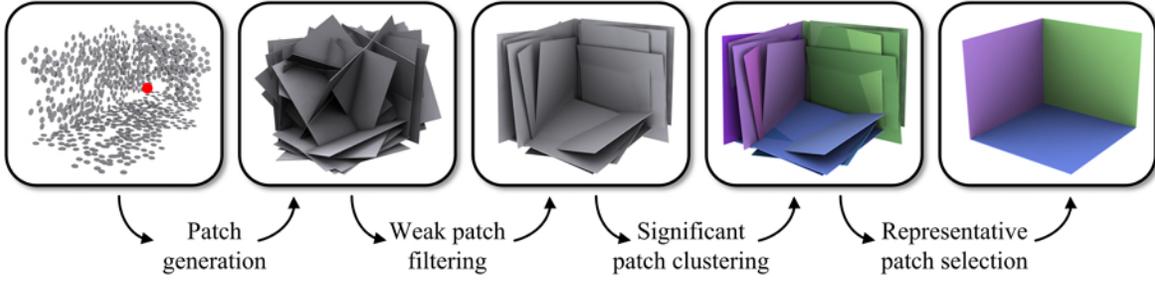


Fig. 2. An overview of our local modeling technique for local structure recovery at a given point marked in red. The extracted substructures are represented with planar patches.

Recently, Oesau et al. [39] proposed a method for planar shape detection and regularization from raw point sets. A number of seeds are sampled uniformly on the input point set, and shape detection is performed through region growing, interleaved with regularization through detection and reinforcement of regular relationships, such as coplanar, parallel and orthogonal. This method achieves favorable results from the point data with moderate corruption, which, however, fails to deal with a relatively high level of outliers and noise.

Substructure modeling. The purpose of substructure modeling (also referred to as subspace modeling) is to find a multi-substructure representation that best fits a set of data sampled from a high-dimensional space. In recent years, several related methods have been proposed and adopted to solve various problems for computer vision applications [24]. Elhamifar et al. [9] presented a method based on sparse representation to cluster data drawn from multiple low-dimensional substructures embedded in a high-dimensional space using ℓ_1 optimization, which is fairly sensitive to outliers and noise.

In computer graphics, Decoret et al. [8] presented a plane identification method for billboard clouds using a modified Hough transform scheme. It is computationally expensive due to the nature of Hough Transform. Cohen-Steiner et al. [5] proposed a variational framework to approximate surfaces with planes, extended to more elaborate shape elements by Wu and Kobbelt [15]. Their goal is to seek a number of primitives to approximately represent the input geometry. These two methods are designed based on the connectivity relationship of input data; meanwhile, least squares fitting is used extensively so that they are sensitive to outliers. Fleishman et al. [11] utilized a statistics method, namely forward-search paradigm [13], to classify regions of a point-set to multiple outlier-free smooth sub-regions in a “fitting and removing” manner. Since the number of substructures is unknown in advance, the termination condition of the “fitting and removing” procedure has to be ad-hoc pre-defined. In contrast, our method is able to automatically determine the number of substructures.

2 ROBUST SUBSTRUCTURE MODELING

2.1 Algorithm Overview

According to the differential geometry of surfaces, the local geometry of any point on a 3D model is formed by its underlying surface patches. On this basis, we propose a

robust algorithm to classify the local neighborhood of each point into point subsets each of which comes from a single surface patch, as illustrated in Figure 2. For each point and its local neighborhood, we generate a set of initial surface patches through random sampling strategy, and then filter out false positives to retain significant surface patches. We then classify all of the remaining surface patches into groups and extract the representative surface patch from each group. As a result, all surface patches can be obtained for the input LiDAR points.

As observed, the geometries of the urban buildings are generally composed of planes; hence we consider the surface patch here as linear geometry. Note that it can be extended to any parametric surface, if required. We first give the problem formulation w.r.t. local substructure modeling.

Problem formulation. Given a point p and its neighboring point set $N(p) = \{p_1, p_2, \dots, p_n\}$, we assume that $N(p)$ is distributed on a union of unknown surface patches $\theta_1 \cup \theta_2 \cup \dots \cup \theta_L$; that is, there are L surface patches of \mathbb{R}^3 . Discretely, we have the point set $N(p) \in \mathbb{R}^3$ consisting of n points in \mathbb{R}^3 , which may be partitioned as:

$$N(p) = \chi_0 \cup \chi_1 \cup \dots \cup \chi_L \quad (1)$$

where χ_0 stands for possible outliers; and for each $l \geq 1$, χ_l is a set of N_l points chosen from θ_l , which could contain noise. The task is now simply stated as follows: without any prior knowledge regarding the number of surface patches, we are to recover all underlying patches from the local neighborhood.

2.2 Significant Patch Determination

Given a point p and its neighboring point set $N(p) = \{p_1, p_2, \dots, p_n\}$, we are able to generate a set of initial surface patches $\Theta = \{\theta_1, \theta_2, \dots, \theta_m\}$ in the manner of RANSAC [10]. Each patch is fitted over a randomly chosen minimal subset of size κ , where κ is the minimum number of data required to instantiate the geometric primitive of interest. For example, $\kappa = 3$ for estimating the parameters of a plane.

Patch scoring. Having the initial patch set, it is important to effectively score each patch to measure the “goodness” of the patch. Ideally, if a patch only contains inliers coming from a single structure, the score of the patch should be as high as possible; otherwise, it should be close to zero. Given an arbitrary patch θ and a set of n points $P = \{p_1, p_2, \dots, p_n\}$, the residual set $\mathbf{r}(\theta) = \{r_1, r_2, \dots, r_n\}$ can be obtained from all points of P to θ . According to the non-parametric kernel density estimate techniques [26],

the variable bandwidth kernel density estimate at r can be defined as:

$$f_{\theta}(x) := \frac{1}{n} \sum_{i=1}^n \frac{1}{h(\theta)} \mathbf{K} \left(\frac{x - r_i}{h(\theta)} \right) \quad (2)$$

where $h(\theta)$ and $\mathbf{K}(\cdot)$ are the bandwidth and the Epanechnikov kernel, which can be estimated by the technique [26]. On this basis, Wang et al. [29] weighted the patch using the density estimate at the origin (\mathbf{O}), while suppressing patches that produce large scale estimates, i.e.,:

$$w := \frac{f_{\theta}(\mathbf{O})}{\sigma(\theta)} \quad (3)$$

where $\sigma(\theta)$ is the scale estimate [29]. For calculating $f_{\theta}(\mathbf{O})$ and $\sigma(\theta)$, please refer to Wang et al. [29]. We notice that this scoring scheme is basically a summation over the inlier points, which is quite sensitive to the sampling density of points.

Observing that the significance of a patch should link to the size of its underlying surface, and such an area should be measured independently of point sampling, we thereby introduce a *sampling-independent area-based* measure into the kernel density metric, to better reveal significant patches in a density-invariant fashion.

The intuition of our scoring scheme is to favor patches with a larger area. To achieve density-invariance, we approximate area by the summation of edge lengths in the kNN graph constructed from P . Specifically, given a patch θ , we collect its inlier set containing points with the 15% smallest residuals against θ , which forms multiple connected components in the kNN graph. The score of θ is defined over the maximal connected component, denoted by χ , as:

$$w(\theta) := \frac{f_{\theta}(\mathbf{O})}{\sigma(\theta)} \cdot \sqrt{|\chi|} \cdot \sum_{e_i \in E} |e_i| \quad (4)$$

where $|\chi|$ is the number of points in χ , E is the set of edges in χ , $|e_i|$ is the length of edge e_i . From the definition, our score incorporates both the measure of inliers and the size of the underlying surface. The bigger the size of the underlying surface, the higher the score. Apparently, the size of surface is independent on the sampling density of points on the surface. Therefore, our metric is insensitive to the variation of sampling density distribution. Figure 3 presents the insensitivity of our scoring metric to sampling sparsity. 2000 points are randomly sampled from a plane and a certain level of Gaussian noise ($\sigma = 0.02$) is added. By down-sampling the sampled points, we evaluate the insensitivity of our scoring metric to sampling density. For each down-sampled point set, we generate a number of plane hypotheses. Among them, the best hypothesis with the highest score is obtained. On this basis, we plot the highest score in terms of the corresponding sampling ratio. From the result, we can see our scoring metric is insensitive to sampling density.

Patch filtering. Since the initial patch set Θ is generated through random sampling, Θ inevitably contains a number of “weak” surface patches which diverge considerably from the underlying surfaces within the local neighborhood. Therefore, we need to differentiate these weak patches (i.e., false positives) from those fairly close to the underlying

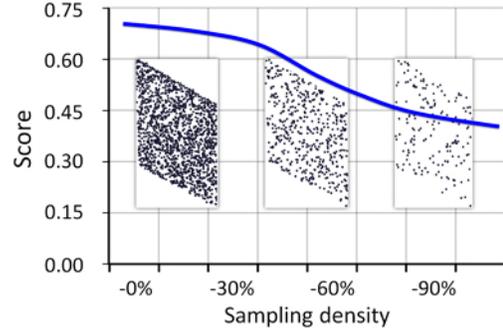


Fig. 3. Insensitivity to sampling sparsity. As the down-sampling ratio increases, the corresponding score degrades gradually. Even though the down-sampling ratio becomes fairly high, the score is still relatively significant.

surfaces, referred to as significant patches. Based on the scores obtained above, which well measure the closeness of the generated patches to the underlying surfaces, we are able to set a cut-off value to filter out the weak patches. However, it is non-trivial to obtain this magic cut-off value. Hence, we employ a *data-driven thresholding* method to automatically discriminate significant patches from weak ones, inspired by the Maximum Entropy Thresholding in image processing [36].

Suppose we have built a histogram of patches with respect to their scores. Our task is to find an optimal threshold to divide the histogram into two parts, corresponding to significant and weak patches, respectively. This can be achieved by maximizing the total amount of information provided in the two parts of the histogram. Please refer to Appendix A for determination of the optimal threshold. Having the optimal threshold \hat{w} , the patch with the score higher than \hat{w} is regarded as a significant patch; otherwise, it is a weak one. Accordingly, we are able to automatically filter out weak patches and retain all of the significant patches.

2.3 Representative Patch Extraction

For each point, its significant patches are retained, which comprise groups representing the local underlying substructures. To extract the underlying substructures, we cluster the significant patches into prominent groups and then select for each group a representative patch. It is observed that the preferences of inliers from the same structure towards a set of patches are correlated. Based on this observation, we define the preferences of inliers to patches as permutations, and seek clusters among the permutations such that the significant patches can be clustered into a series of prominent groups.

Patch preference permutations. Given a point p and its neighboring point set $N(p) = \{p_1, p_2, \dots, p_n\}$, $\Theta_s = \{\theta_1, \theta_2, \dots, \theta_l\}$ is the set of significant patches obtained from Section 2.2. For each patch θ_i , the surface of which is represented by S_i , the residuals of $N(p)$ to S_i are computed and denoted by $\mathbf{r}^i = [r_1^i, r_2^i, \dots, r_n^i]$. We sort \mathbf{r}^i in increasing order to obtain the following residual list:

$$\hat{\mathbf{r}}^i = [r_{\mu_1^i}^i, r_{\mu_2^i}^i, \dots, r_{\mu_n^i}^i] \quad (5)$$

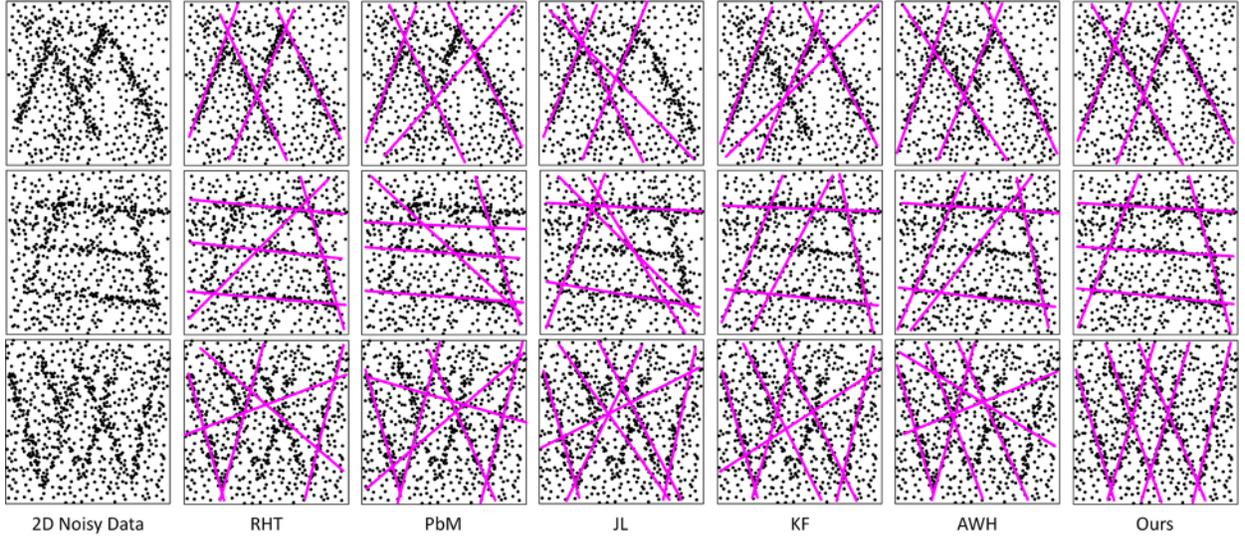


Fig. 4. Line fitting comparisons. There are four, five and six lines from the first to the fourth rows. The number of points is 20,000 for each case. The noise and outlier rates are 8% and 90%, respectively. RHT, PbM and JL do not obtain as good results as KF, AWH and ours. Among them, ours succeeds in fitting all lines correctly.

such that $r_{\mu_1^i}^i \leq r_{\mu_2^i}^i \leq \dots \leq r_{\mu_n^i}^i$. The sorting $\hat{\mathbf{r}}^i$ essentially ranks the n points according to the preference of S_i , i.e. θ_i ; the higher a point is ranked the more likely it is an inlier to θ_i .

Accordingly, the preference permutation of $N(p)$ to patch θ_i is defined as:

$$\pi(\theta_i) = [\mu_1^i, \mu_2^i, \dots, \mu_n^i] \quad (6)$$

For all patches $\Theta_s = \{\theta_1, \theta_2, \dots, \theta_l\}$, we have the permutations:

$$\mathbf{R}(\Theta_s) = \{\pi(\theta_1), \pi(\theta_2), \dots, \pi(\theta_l)\} \quad (7)$$

Significant patch clustering. After obtaining the preference permutations $\mathbf{R}(\Theta_s)$, we are to seek clusters over $\mathbf{R}(\Theta_s)$ based on the rationale that the preferences tend to cluster based on their structure membership. To this end, we model the preference permutations using the mixture model of parts. Since there is no priori regarding the number of parts, we adopt the medoid shift algorithm [27] to conduct non-parametric cluster seeking over the set of permutations. To simplify the exposition, we present the detailed clustering algorithm over permutations in Appendix B. Once the set of clusters are detected, the significant patches are accordingly clustered into a series of prominent groups, each of which consists of similar patches sampled from the same substructure.

Representative patch selection. With the clustered groups $\mathbf{G} = \{\mathbf{g}^1, \mathbf{g}^2, \dots, \mathbf{g}^u\}$, we choose the first patch θ_1^i of each group $\mathbf{g}^i = \{\theta_1^i, \theta_2^i, \dots, \theta_o^i\}$ as a representative patch, i.e., the one with the highest score, to represent the group. As a result, the local neighborhood of a point p is clustered into the set of u representative patches, $\mathfrak{S}(p) = \{\theta_1^1, \theta_1^2, \dots, \theta_1^u\}$.

2.4 Experiments

We evaluate the performance of our proposed modeling method in line, plane fitting with an emphasis on local

multi-structure recovery from a variety of synthetic and real raw data, which is accomplished by comprehensively comparing our method with the related approaches from Toldo et al. [30] (JL), Meer et al. [31] (PbM), Kultanen et al. [32] (RHT), Chin et al. [33] (KF) and Wang et al. [29] (AWH). When running these methods, we adjust their respective parameters carefully to guarantee that the approaches of JL, PbM, RHT, KF and AWH output the correct structure number, which is automatically obtained by our method.

2.4.1 Synthetic data

We test the six approaches on line fitting using four sets of synthetic data in Figure 4, and present the quantitative results in Figure 5 and Table 1. In Figure 4, the lines in the data of each row are arranged to produce challenging configurations for line fitting. The lines in each row contain a certain number I of inliers, contaminated with Gaussian noise of standard deviation σ . A total of G points of gross outliers are randomly added within the range of the data, i.e., $[(0, 1), (0, 1)]$. From the comparisons, RHT, PbM and JL basically do not obtain as good results as KF, AWH and ours. Among them, only ours succeeds in fitting all lines correctly.

Quantitative evaluation. We perform quantitative evaluations on these approaches by comparing their corresponding fitting errors. First, we find the correspondence between the estimated lines and the ground-truth lines. Let $\mathcal{L} = \{\mathbf{l}_1, \dots, \mathbf{l}_N\}$ and $\hat{\mathcal{L}} = \{\hat{\mathbf{l}}_1, \dots, \hat{\mathbf{l}}_N\}$ respectively be the corresponding sets of the ground-truth lines and the estimated lines, and $\mathcal{I} = \{\mathbf{i}\mathbf{l}_1, \dots, \mathbf{i}\mathbf{l}_N\}$ and $\hat{\mathcal{I}} = \{\hat{\mathbf{i}}\mathbf{l}_1, \dots, \hat{\mathbf{i}}\mathbf{l}_N\}$ respectively the inlier sets of the ground-truth lines and the estimated lines, we then compute the fitting error between \mathcal{L} and $\hat{\mathcal{L}}$ as:

$$error = \sqrt{\frac{\sum_{i=1}^N \left(dist^2(\mathbf{i}\mathbf{l}_i, \hat{\mathbf{l}}_i) + dist^2(\hat{\mathbf{i}}\mathbf{l}_i, \mathbf{l}_i) \right)}{N}} \quad (8)$$

where $dist(\cdot, \cdot)$ measures the geometric distance between the inliers to the line. Table 1 shows the fitting errors of

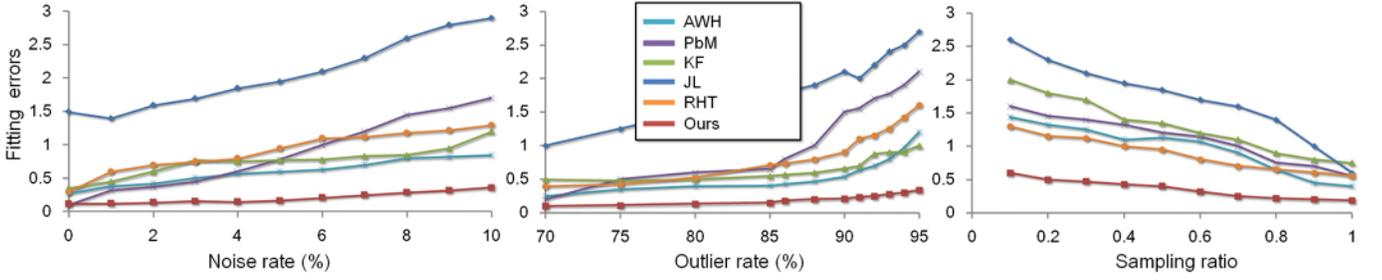


Fig. 5. Fitting errors from six compared approaches on the “six-line” dataset in Figure 4. Different levels of noise, outliers and down-sampling are applied to the point data. From the histograms, our method is capable of tolerating a reasonably high level of noise, outliers and data sparsity.

TABLE 1
The fitting errors of synthetic data in Figure 4.

Data	RHT	PbM	JL	KF	AWH	Ours
Four-line	0.56	0.98	1.84	1.56	0.46	0.38
Five-line	1.23	2.12	4.52	1.62	1.57	0.39
Six-line	4.36	3.03	3.69	1.86	2.84	0.43

TABLE 2
The computational time of plane fitting in Figure 6 (in seconds).

Points	RHT	PbM	JL	KF	AWH	Ours
45,236	3.6	13.6	281.3	2483.1	12.3	11.2
2,954	0.6	2.1	21.2	182.6	1.6	1.8
742,896	63.2	253.2	5372.3	43523.9	156.2	143.9

synthetic data in Figure 4.

Based on the quantitative metric, we test the performance of the methods under the influence of different levels of outliers, noise as well as data sub-sampling, and show the comparison results in Figure 5 with respect to the Six-line data in Figure 4. For the noise case, we fix G at 20000 and vary σ from 0 to 0.1 in steps of 0.01. For the outlier case, we fix G at 20000 and σ at 0.05. For the sparsity case, we fix σ at 0.05 and vary G from 2000 to 20000. For each case, we generate 20000 hypotheses randomly, and compute the respective fitting errors from all compared methods, as shown in Figure 5. From the plots, JL does not obtain as good results as the others, and AWH basically achieves better results than PbM, RHT, KF. Comparatively, our method exhibits robustness to tolerate different levels of outliers, noise and data sparsity, and yields the most accurate fitting results among the competing approaches.

2.4.2 Real data

We further evaluate the effectiveness of these approaches on 3D real scan datasets in Figure 6, which contain a certain level of noise, outliers and missing regions. The scenes consist of many planar structures, and we test how many planes can be detected from each approach. For the corner-box scene, there are 19 planes within the raw point cloud, which is fairly noisy. PbM, KF, AWH and ours generate relatively better results than RHT and JL. The box-desk scene contains 15 planes and the data are quite sparse on some parts. Among all these methods, only ours extracts all of planar structures correctly. In terms of the roof scene, it includes 26 planes on the building roofs, together with several trees around the buildings, which are outliers to the planar structures of our interest. From the results, AWH and KF obtain more correct planes than RHT, PbM and JL. In contrast, ours achieves more accurate results than AWH and KF. The computational time of six approaches on plane fitting experiments in Figure 6 is presented in Table 2.

3 URBAN SCENE APPLICATIONS

Based on the proposed substructure modeling algorithm, we develop several urban scene applications, such as surface reconstruction and primitive modeling, and achieve favorable results from defect-laden, raw LiDAR data.

3.1 Primitive Modeling

Since the neighborhood of any one point can be represented by its underlying substructures, we characterize each point with its representative patches. On this basis, the global clustering is performed by grouping spatially adjacent, similar points in terms of their representative patches. Subsequently, we execute surface fitting over all clusters to generate the 3D model of the building.

Greedy clustering. We group “similar” points to form a set of clusters, so that each cluster can be modeled by surface fitting. To this end, we measure the similarity of points based on their representative surface patches recovered from Section 2.3. Intuitively, two points p and q will share many similar representative surface patches, if they are inliers from the same structure. According to this observation, we define the following function to measure the similarity between p and q :

$$sim(p, q) := \frac{|\mathfrak{S}(p) \cap \mathfrak{S}(q)|}{\max(|\mathfrak{S}(p)|, |\mathfrak{S}(q)|)} \quad (9)$$

where $|\mathfrak{S}(p) \cap \mathfrak{S}(q)|$ seeks the number of similar elements shared by $\mathfrak{S}(p)$ and $\mathfrak{S}(q)$. From the definition, if two points are potentially from the same underlying structure, their representative patch subsets have many similar elements and, therefore, their similarity is high. Otherwise, the similarity is low. Based on the similarity metric, we perform point cloud clustering greedily using the region growing strategy. More advanced techniques, such as graph cuts, can also be employed, with our similarity measure incorporated.

Specifically, we first randomly choose an un-clustered point as a seed. Starting with the seed point, we grow the region by progressively adding un-clustered, adjacent

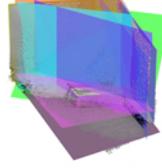
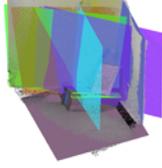
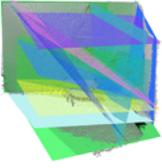
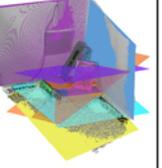
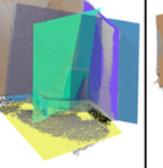
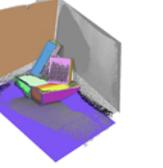
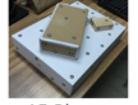
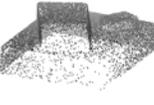
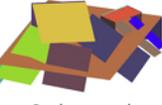
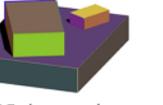
Real scenes	3D raw scans	RHT	PbM	JL	KF	AWH	Ours
 19 Planes		 9 detected	 14 detected	 4 detected	 16 detected	 17 detected	 19 detected
 15 Planes		 7 detected	 9 detected	 3 detected	 12 detected	 13 detected	 15 detected
 26 Planes		 7 detected	 10 detected	 3 detected	 16 detected	 20 detected	 24 detected

Fig. 6. Plane fitting comparisons on 3D raw point data captured from real scenes. The raw point clouds contain some noise, outliers and even missing regions. RHT, PbM, JL and KF do not extract as many planes as AWH and ours. For the first two rows, our method succeeds in fitting all planes, while AWH misses some of them. Ours fails to fit two planes for the last case due to significantly missing data.

points to the seed. The similarity values between the newly added points and the seed point should be greater than a given threshold δ . The region growing process terminates when no more point can be added into the region, which results in a cluster. This procedure repeats iteratively until all points have been clustered.

Surface fitting. After performing greedy clustering, we proceed to model each cluster with plane via least squares fitting. Since all points in one cluster have a similar representative patch, we fit each cluster with a planar surface in the least squares sense. We then project the points to the fitted plane and compute the aligned bounding rectangle of the projection points on the plane. As a result, the bounding polygons of all clusters comprise the final model. Optionally, we can refine the model by adjusting the boundaries of the polygons interactively similar to O-Snap [2].

3.2 Surface Reconstruction

The moving least squares (MLS) technique has been used as a powerful tool to reconstruct surface from point data. It is known that the traditional MLS is incapable of preserving sharp features [1]. To address this issue, we devise a new projection operator which projects each point onto the surface patches obtained from Section 2.3. As a result, surface reconstruction is faithfully achieved from point clouds, while sharp features are well preserved.

Given a point p , let $\mathfrak{S}(p) = \{\theta_1, \theta_2, \dots, \theta_u\}$ be the representative patch set within its neighborhood, obtained from Section 2.3. According to the type of p , we use the following projection scheme to update its position. 1) *Non-feature point*: ($u = 1$): the closest point of p on θ_1 is simply considered as the new position of p . 2) *Feature point*: ($u > 1$): we project p to all representative patches to obtain its new position. For the latter case, we design a comprehensive projection scheme as elaborated below.

For simplicity, let us illustrate our projection scheme in 2D. Given a point p , let θ_1, θ_2 be its two substructures.

Suppose c_1, c_2 are the centroids of the point sets corresponding to θ_1, θ_2 , $\mathbf{v}_1, \mathbf{v}_2$ are the projection vectors of c_1, c_2 onto θ_2, θ_1 , and $\mathbf{v}_3, \mathbf{v}_4$ are the projection vectors of p onto θ_2, θ_1 , respectively. According to the relations between the projection vectors, p can be classified into four quadrants as:

$$p \in \begin{cases} Q-I & \text{if } (\mathbf{v}_1 \cdot \mathbf{v}_3 \leq 0 \text{ and } \mathbf{v}_2 \cdot \mathbf{v}_4 \leq 0) \\ Q-II & \text{if } (\mathbf{v}_1 \cdot \mathbf{v}_3 > 0 \text{ and } \mathbf{v}_2 \cdot \mathbf{v}_4 < 0) \\ Q-III & \text{if } (\mathbf{v}_1 \cdot \mathbf{v}_3 \geq 0 \text{ and } \mathbf{v}_2 \cdot \mathbf{v}_4 \geq 0) \\ Q-IV & \text{otherwise.} \end{cases} \quad (10)$$

On this basis, if p lies in Quadrant I or III, we project it onto the intersection of θ_1 and θ_2 . If it lies in Quadrant II or IV, we directly project it onto θ_1 or θ_2 , respectively.

For each point, we apply the robust projection operator described above to project it onto representative patches. Subsequently, based on the MLS reconstruction technique [1], the piecewise smooth surfaces can be faithfully generated from point data, while original sharp features are well preserved.

4 EXPERIMENTAL RESULTS

This section provides experimental results on a variety of raw LiDAR scans from urban scenes to validate the performance of our proposed algorithms, including urban facade reconstruction and modeling. As shown, raw LiDAR data always suffer from deflection, such as noise corruption, occlusions and non-uniform density distribution, which are directly processed and automatically modeled by our method without intervention.

4.1 Effects of Algorithm Stages

Our reconstruction and modeling algorithms are built upon the proposed subspace modeling technique, which consists of two key stages, i.e., significant patch determination and representative patch extraction. On the one hand, we propose a robust density-invariant patch scoring technique to determinate significant patches, which has a stronger

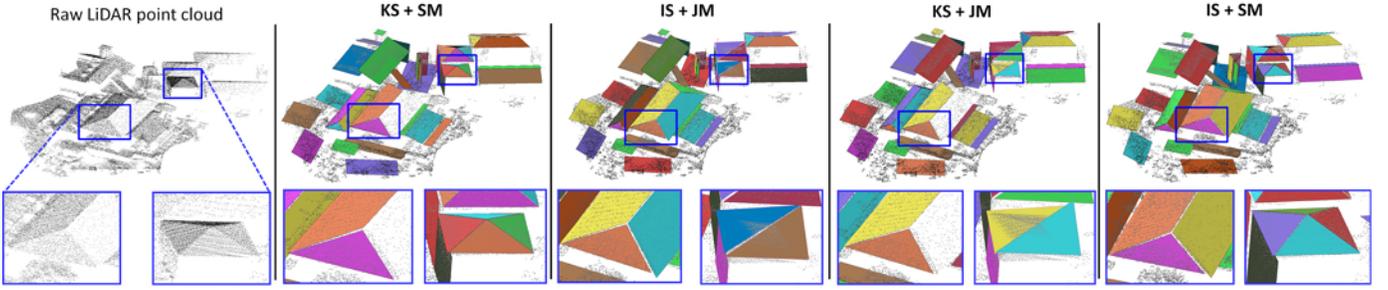


Fig. 7. Illustration of how each kernel algorithm stage affects the result. The method in the paper is cascaded with two stages, which essentially are driven by our proposed techniques, i.e., the density-invariant scoring strategy (denoted by **IS**) and the substructure similarity measuring scheme (**SM**). The corresponding state-of-the-art techniques to ours are the kernel density scoring [29] (**KS**) and the Jaccard distance metric [29], [30] (**JM**), respectively. Only our combination succeeds in handling data sparsity and distinguishing geometrically similar-yet-distinct structures. By comparing different technique combinations, we can see that ours outperforms the others, as highlighted in the zoom-in views.

capability of tolerating the variation of density distribution of point clouds, compared to the most recent algorithm which is the kernel density scoring strategy [29]. On the other hand, we design an effective substructure similarity metric to extract representative patches by leveraging a discriminative permutation-based preference feature. In contrast to a previous, well-known metric, like Jaccard distance [29], [30], ours is more sensitive to the geometrically similar yet distinct substructures. To verify the effectiveness of our method, we compare the modeling results with different technique combinations of ours and the most related work [29], [30]. For ease of description, we denote our density-invariant scoring by **IS**, and the kernel density scoring [29] by **KS**; our substructure similarity metric by **SM**, and the Jaccard distance metric [29], [30] by **JM**.

Figure 7 shows the modeling results on the raw LiDAR data of building roofs from the technique combinations, i.e., **KS+SM**, **IS+JM**, **KS+JM** and **IS+SM**. Due to the special characteristics of roofs, the points on some roofs are fairly sparse. Meanwhile, the dihedral angles of some adjacent roof patches are quite obtuse and the adjacent patches are almost co-planar, as highlighted in Figure 7. As analyzed in Section 2.2, **KS** is sensitive to density distribution. Consequently, the patches of roofs with sparse points fail to be extracted as shown in the middle column of Figure 7. The metric [29], [30] based on **JM** is insensitive to the inconsiderable difference between geometrically similar substructures. As a consequence, some roof patches, the normal vectors of which are quite close, are not detected, and a couple of adjacent roof patches are merged by mistake (see the results from **JM**). Comparatively, by using our proposed techniques, i.e., **IS+SM**, all roof patches are extracted successfully even in the presence of data imperfection.

4.2 Comparisons on Raw LiDAR Data

To further evaluate the performance of our algorithms, we compare our modeling results on raw LiDAR point clouds to an advanced, RANSAC-based method [25], discrete labeling based Pearl [14], constrained data fitting based Globfit [18] and regular arrangements of planes based RAPter [38]. Most of the raw LiDAR point clouds used in our paper are scanned by our long-range, 3D laser scanner (Leica ScanStation P20), and the others are taken directly from [38]. The comparisons are comprehensively performed

in terms of the characteristics of raw LiDAR data, i.e., noise, outliers, density anisotropy, occlusions and large-scale. In particular, we use the available implementations of those modeling algorithms from RAPter [38]. Additionally, we also compare our reconstruction method with the state-of-the-arts, including the algorithms from Ohtake et al. [22], Lafarge et al. [17], Oztireli et al. [23].

4.2.1 Noise

We evaluate the effectiveness on the raw LiDAR scan of a building facade corrupted by severe *noise* in Figure 8. The main structure is extracted correctly from all methods except RANSAC, which over-fits big planar patches over the facade structures. Most of small structures on the facade are severely missing from Pearl and RAPter, while they are successfully extracted by Globfit and ours. As seen, Globfit generates a number planar artifacts due to noise, which do not exist in our result. Our method leverages the kernel density estimation method to calculate the noise scale of the point data, and thus the noise within structure inliers is truncated with the residual ranking strategy. Therefore, our method is able to produce relatively better results in the presence of noisy point data.

4.2.2 Outliers

In Figure 9, we illustrate how our algorithm detects planar primitives from raw LiDAR point data contaminated by heavy *outliers*. There are several trees around the scanned building, and thus the corresponding points are acquired within the raw LiDAR scan. The main structures are basically obtained from RANSAC, Pearl and RAPter, while a number of false positives (patches) are yielded at the same time due to severe outliers. Globfit only extracts several significant planar patches, while most of structures are missing. RAPter obtains many planes; however, a few of them are placed incorrectly around the areas with outliers. By incorporating the contamination scale in our hypothesis weighting metric, the outliers can be distinguished from the inliers of structures and hence only the inliers excluding noise and outliers are used to fit the structures. As a result, our method succeeds in obtaining most of planar structures from the outlier-laden, raw LiDAR data, though several small structures are still missing.

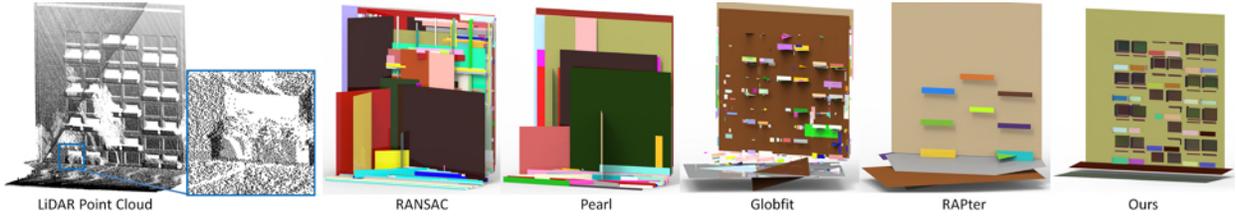


Fig. 8. Comparison to the other methods on raw LiDAR data with severe *noise*. All methods basically extract the main planar structure. In terms of recovering small structures, RANSAC, Pearl and RAPter do not obtain as good results as Globfit and ours. Note that we restrict all fitted planar patches to be either horizontal or vertical.

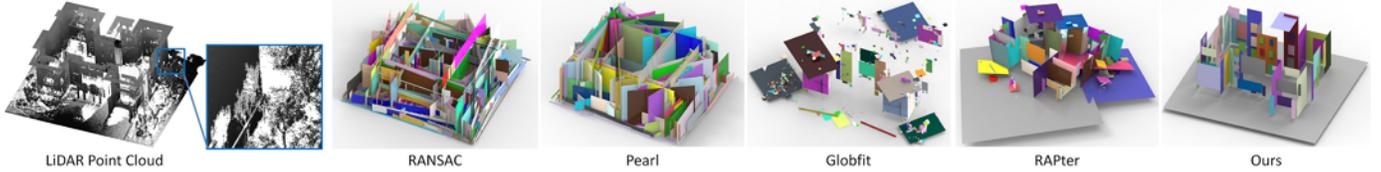


Fig. 9. Comparison to the other methods on raw LiDAR data with heavy *outliers*. Outliers severely ruin RANSAC and Pearl, and consequently a number of incorrect planar patches are generated. Globfit produces numerous fragmental artifacts. RAPter obtains a number of planes, while some of them are placed incorrectly around the areas with outliers. Comparatively, most of the main structures are retained correctly with ours.

4.2.3 Density anisotropy

We assess the capability of our method to deal with *density anisotropy* of raw LiDAR data in Figure 10. Since the experimental data are captured with the long-range laser scanner, the scans always contain many ill-sampled regions, where the density distributions are non-uniform and anisotropic. All methods extract main planar structures except Globfit. Meanwhile, RANSAC produces a large amount of fake planar patches, and Pearl also generates a certain number of false positives. Globfit over-fits small planar structures on the building facades, while significant patches are missing. Most of big planar structures are yielded from the result of RAPter, where a number of small planes are missing. Thanks to our density-invariant substructure selection technique, we are able to extract dominant structures from facades, as well as reasonably small structures.

4.2.4 Occlusions

In Figure 11, we demonstrate how our approach is exploited to process raw LiDAR data with missing regions due to *occlusions* during scanning. The building consists of horizontal and vertical planar structures, which are scanned by using the long-range laser scanner, resulting in large occluded regions. The main directions are detected by all methods. RANSAC fails to extract almost all smaller structures and some dominant vertical planes. Pearl obtains most of the vertical planes, while meanwhile producing many false positives, i.e. slanted planes. Globfit succeeds in generating a certain number of small structures (see the vertical ones on the left-side surface of the facade); however, some big vertical planes are missing. Most of important planes are successfully generated from our result, where small structures are partially obtained. RAPter obtains the dominant planar structures as well as small planes, while a few artifacts are also generated.

4.2.5 Non-planar surfaces

Figure 12 demonstrates that our method is also able to deal with *non-planar surfaces* within raw LiDAR point data. There

are a cylinder and a cone above the church roof. Specifically, we take the “fitting-and-removing” strategy to detect planar patches first, and extract cylindrical and conical surfaces subsequently. Note that we require to set the surface types manually during modeling. Compared to the other methods, only ours can obtain the non-planar shapes from the point data.

4.2.6 Large-scale

We evaluate the computational efficiency of our method against large-scale, raw LiDAR point data in Figure 13. The acquired area is about 1.5 square kilometers and the scanning is performed by a Velodyne LiDAR sensor (HDL-32E) mounted on a drone. The number of scanned points is over 300 million. We could not run the other methods on such a large scale scene due to performance reasons. There are more than 100 planar patches and we are able to extract almost all of them successfully in a relatively efficient manner. From the fusion view in Figure 13(c), we can see that the reconstructed planar patches have good geometric fidelity to the raw LiDAR point cloud. The quantitative details are presented later.

4.2.7 Mesh reconstruction

Figure 14 shows a comparison of reconstruction results on a raw LiDAR scan from four methods: Ohtake et al.’s [22], Lafarge et al.’s [17], Oztireli et al.’s [23] and ours. Our result is generated by utilizing our MLS projection scheme followed by applying the reconstruction method in [1]. The latter three methods outperform Ohtake et al.’s [22] in terms of feature preservation. Our method recovers more details than Lafarge et al.’s [17], and exhibits superiority over Oztireli et al.’s [23] on the reconstruction of sparse points.

4.3 Quantitative Evaluations

The results shown above have visually demonstrated the superiority of our algorithm in terms of effectiveness on large-scale, defect-laden, raw LiDAR point data. We provide



Fig. 10. Comparison to the other methods on raw LiDAR data with *density anisotropy*. All methods reconstruct dominant planar structures, except Globfit. The relatively smaller structures, like windows, are successfully obtained from our method, which, however, are missing from the others.

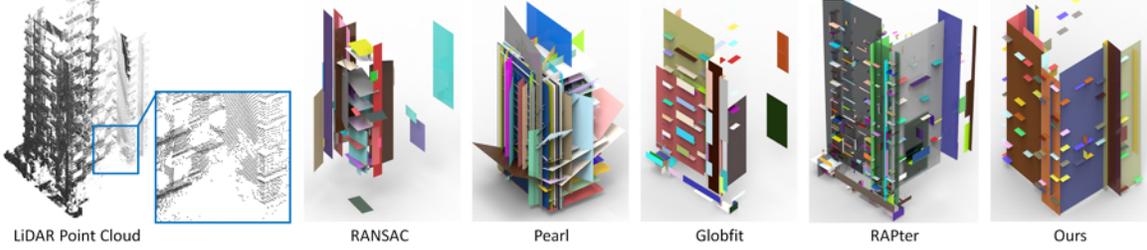


Fig. 11. Comparison to the other methods on raw LiDAR data with *occluded regions*. Pearl yields more false positives than the others. Globfit, RAPter and ours obtain more small structures than RANSAC. A few small planar structures are missing from Globfit and ours, which are obtained from RAPter.

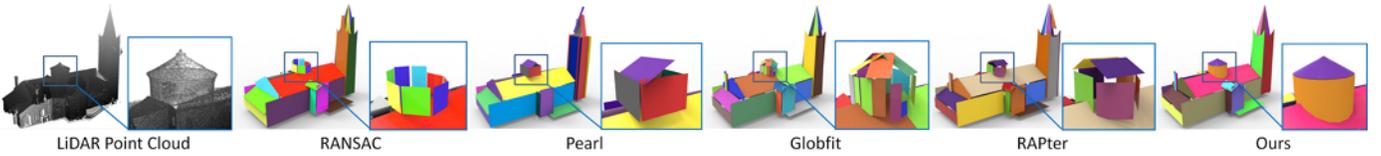


Fig. 12. Comparison to the other methods on raw LiDAR data with *non-planar shapes*. In contrast, only ours obtains the cylinder and the cone successfully from the point cloud.

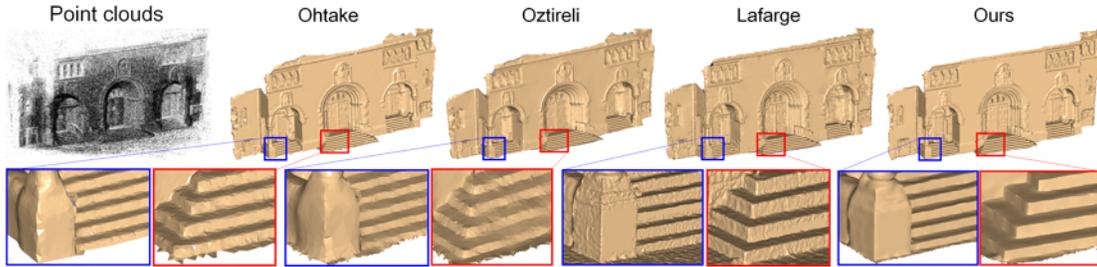


Fig. 14. Comparison of surface reconstruction on raw LiDAR data from Ohtake et al. [22], Oztireli et al. [23], Lafarge et al. [17] and ours. The point clouds are corrupted with a high level of noise and outliers. The methods of Oztireli et al. [23] and Lafarge et al. [17], as well as ours, obtain better results than Ohtake et al.'s [22]. Our method outperforms all of the other three on detail preservation.

some quantitative comparisons in Table 3, and also further investigate its robustness to different levels of noise, outliers and sparsity in Figure 15.

4.3.1 Quantitative metrics

Coverage. We design a coverage metric (i.e., percentage of inlier points associated with modeled structures) to measure the completeness of modeling results against the ground truth. Let \mathcal{P} be the input raw point set and \mathcal{M} the respective modeling result, the modeling coverage metric is defined as:

$$coverage = \frac{|\{p|\forall p \in \mathcal{P}, s.t. \|p - \mathcal{M}\| < \epsilon\}|}{|\mathcal{P}|} \times 100\% \quad (11)$$

where $\|p - \mathcal{M}\|$ is the Euclidean distance between p and \mathcal{M} ; $|\cdot|$ is the set cardinality; ϵ is a small distance threshold which is fixed at 0.036 in our experiments.

RMSE. We choose the root-mean-square error (RMSE) metric to measure the geometric fidelity of modeling results to the input data. Suppose the reconstructed model \mathcal{M} comprises n primitive patches, i.e., $\mathcal{M} = \{\mathcal{E}_i\}_{i=1}^n$. For each patch \mathcal{E}_i , the associated point subset is denoted by \mathcal{P}_i , in which each point is closer to \mathcal{E}_i than any other patches. The modeling error metric is then defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n \sum_{p \in \mathcal{P}_i} \|p - \mathcal{E}_i\|^2}{\sum_{i=1}^n |\mathcal{P}_i|}} \quad (12)$$

where $\|p - \mathcal{E}_i\|$ is the Euclidean distance between p and \mathcal{E}_i ; $|\mathcal{P}_i|$ is the number of points within \mathcal{P}_i .

Complexity. We introduce the complexity metric to measure the degrees of freedom of the reconstructed planar shapes. Specifically, we perform normal vector clustering

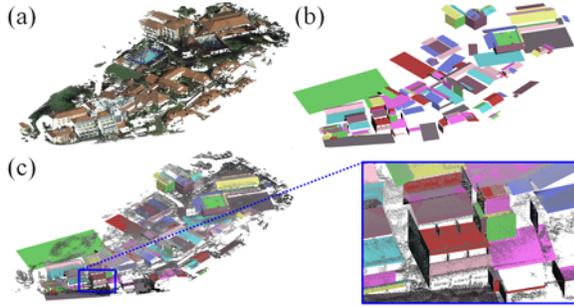


Fig. 13. Comparison to the other methods on *large-scale*, raw LiDAR data. (a) The input raw LiDAR data; (b) The reconstructed planar patches; (c) The fusion view of the point data and the reconstructed model. The number of points is over 300 million and there are more than 100 planar patches generated from our method within several hours.

over all obtained planes by applying 3D Hough Transform on a unit sphere. By uniformly dividing the unit sphere into a certain number of cells, we are able to get all of the clusters each of which contains at least one normal vector vote. In our implementation, the angles along two orthogonal directions are both set at 3 degrees for each cell. As a result, we regard the number of normal clusters as the complexity metric. From the definitions, high complexity refers to low regularity and vice-versa. We scale all experimental data uniformly within an unit sphere before modeling and then compute the above quantitative metrics.

4.3.2 Robustness

Ground truth. We manually design a building model in *Trimble Sketchup* and perform virtual scanning on it to generate 3D points. Note that our virtual scanning simulates the working mechanism of real laser scanning and, therefore, the scanned dataset is distributed non-uniformly. Specifically, a virtual scanner is set in front of the building model at a given distance. The laser ray starting from the scanner casts outwards and the casting angle changes uniformly along the vertical and horizontal directions. Once the ray touches the building model, we record the intersection point. In such a way, we are able to store all scanned point data at the given station. For the whole building, we generally execute scanning at 3 different stations around the building model, and perform registration to align the point data into the same coordinate system. As a result, the entire point data is obtained for the building model. The synthetic noise is made by a zero-mean Gaussian function proportional to the diagonal length of the bounding box of the input data, and the outliers are generated randomly in the bounding box. The synthetic point cloud provides us with a ground truth, because we can distinguish between the original points and the added noise and outliers. As a result, we are able to quantitatively analyze the modeling results against the ground truth.

Our method is capable of handling a high level of noise, outliers and sparsity, as illustrated in Figure 15. As the corruption intensifies, the modeling accuracy degrades gracefully. Even though 50% of the outliers, 6% of the noise are added, the window structures are still recovered correctly and the accuracy remains relatively high. Similar results can be achieved by down-sampling 75% of the original points.

TABLE 3

Quantitative benchmark shows the reconstruction accuracy, coverage and complexity comparisons on the experimental data sets from RANSAC, Pearl, Globfit and our method.

Data sets	Methods	Quantitative metrics		
		RMSE	Coverage	Complexity
Figure 7	RANSAC	0.48	72%	1.00
	Pearl	0.49	74%	0.95
	Globfit	0.56	67%	0.21
	RAPter	0.61	69%	0.22
	Ours	0.42	85%	0.25
Figure 8	RANSAC	0.28	85%	1.00
	Pearl	0.31	75%	0.54
	Globfit	0.29	83%	0.26
	RAPter	0.27	79%	0.25
	Ours	0.25	90%	0.20
Figure 9	RANSAC	1.42	83%	1.00
	Pearl	1.56	77%	0.97
	Globfit	1.82	57%	0.16
	RAPter	1.76	81%	0.18
	Ours	1.66	88%	0.21
Figure 10	RANSAC	0.15	87%	1.00
	Pearl	0.16	80%	0.98
	Globfit	0.22	52%	0.18
	RAPter	0.23	78%	0.21
	Ours	0.18	89%	0.15
Figure 11	RANSAC	0.24	76%	0.31
	Pearl	0.27	78%	1.00
	Globfit	0.31	81%	0.12
	RAPter	0.28	88%	0.16
	Ours	0.25	86%	0.19
Figure 12	RANSAC	0.16	79%	0.75
	Pearl	0.25	76%	0.76
	Globfit	0.19	82%	1.00
	RAPter	0.24	83%	0.80
	Ours	0.12	88%	0.64

While data imperfection increases significantly, our method tends to be unstable.

4.4 Parameters and Performance

Two parameters are required in our substructure modeling method: 1) the size of the nearest neighbor n ; 2) the ratio of outliers η to determine the number of initial patches in Section 2.2. The choice of n depends on the density of the data. If the data are dense, choosing a high value of n would be better; otherwise, some features could be blurred. After increasing n to a relatively high value (e.g. 60), the change of n has a very limited impact on the processing results. Hence, we set $n = 60$ for all experiments. The ratio of outliers η is used to determine the sample number m in Section 2. η is usually unknown, so we set the sample number m instead. A number of experiments with different levels of densities show that the results are insensitive to the choice of m , in which $m = 200$ generally produces satisfactory results. In terms of our applications, the similarity threshold δ between two points needs to be set for greedy clustering. We empirically fix δ at 0.82 for all experiments, which generally produce favorable results.

We have implemented our algorithm in C++ and all experiments are performed on a PC with a 2.4 GHz CPU and 4.0 GB of RAM. Note that our implementation does not take advantage of the very parallelizable nature of some of the stages and doing so could increase the efficiency.

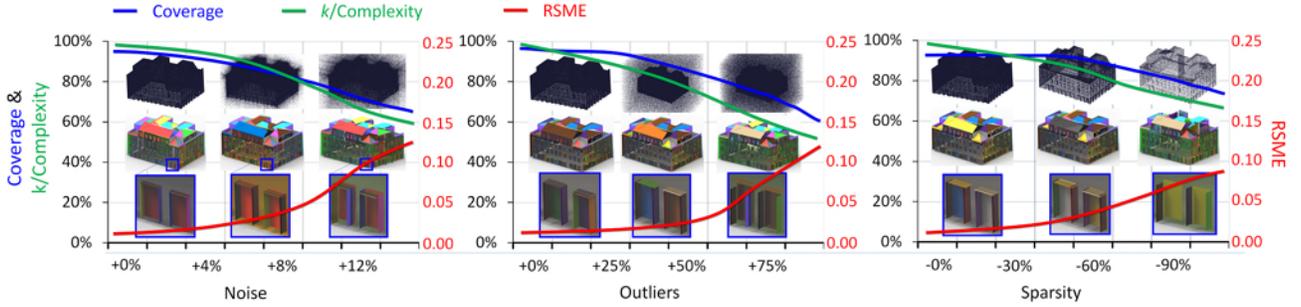


Fig. 15. Robustness to noise, outliers and sampling sparsity. The ground truth is generated by virtual scanning on a manually designed building. We evaluate our method while increasingly adding synthetic noise, outliers, and down-sampling points. With an increasing amount of noise, outliers and down-sampling, the reconstruction accuracy and coverage degrades gradually, and the complexity also increases slowly. When the level of deflection increases significantly high, the method tends to be unstable. Overall, our algorithm is able to handle a reasonably high level of noise, outliers and data sparsity. k is the complexity of the ground truth.

TABLE 4

Timing of our method on experimental LiDAR point clouds (in minutes).

Data sets	Points	Time
Figure 1	120,543,196	210.82
Figure 7	12,483,984	23.72
Figure 8	383,636	0.72
Figure 9	3,532,890	6.14
Figure 10	2,987,382	5.21
Figure 11	739,121	1.38
Figure 12	1,229,966	2.69
Figure 13	310,267,839	541.05
Figure 14	569,626	1.49
Figure 15	1,273,987	2.22

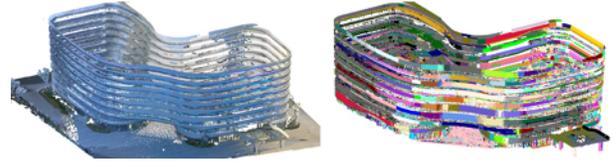


Fig. 17. The building facade with freeform surfaces. The middle part of the building can be represented by a cylindrical surface approximately so the corresponding points are clustered into groups. However, the end parts are not designed by cylindrical surfaces and consequently the associated points can't be clustered into groups.

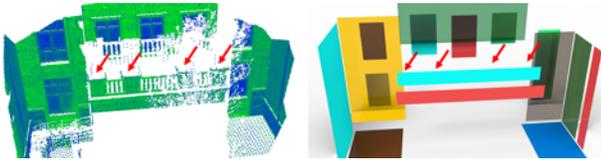


Fig. 16. The points on the top faces of the columns are significantly missing. Consequently, our local classification fails to detect the top patches.

Table 4 gives the computational timing of our method on experimental LiDAR point clouds.

4.5 Limitations

When the missing regions are fairly large, our local recovery could fail and the corresponding patches are missing consequently, as demonstrated in Figure 16. Our method is able to deal with parametric surfaces, while incapable of handling freeform surfaces. As shown in Figure 17, we fail to model the structures formed by freeform surfaces. Additionally, in order to deal with multiple types of surface patches, our framework has to pre-define the surface types in advance.

5 CONCLUSION, DISCUSSION AND FUTURE WORKS

For raw LiDAR point clouds of urban buildings, we propose to model the local neighborhood using surface patches. Our method is based on a robust local classification technique, which is able to automatically determine the number and

the parameters of underlying surface patches. Compared with existing methods, our methods achieve better results on both synthetic and real LiDAR scans.

Priors in point cloud processing. The use of priors, either geometric [12] or structural [37], for point cloud reconstruction has been successfully demonstrated by many studies. The local substructures, being a specific type of parametric surface patch, can also be seen as local shape priors. Although integrating these local substructures can benefit the recovery of a more global structure as we have demonstrated with clustering and modeling applications, the substructures are still limited to the connected surface patch. A more versatile framework should incorporate solid shape primitives, such as sphere, cuboid, etc., to characterize the local structures of a given point with more global information.

Multiscale subspace analysis. If we pursue larger scale substructures, the recovered structures would be able to encompass the delicate structures as shown in Figure 17 and the more global surface patches could be extracted. Therefore, it would be interesting to exploit neighborhoods in multiple scales for substructure extraction and to automatically adapt to the underlying geometry.

Support to high-level analysis. We have demonstrated some results of our method on clustering and feature analysis. Although it is still preliminary, we have shown the potential of the idea of local structure recovery in supporting structural analysis of raw LiDAR scans. An interesting future direction is to integrate the local structural analysis with the high-level analysis, for example, hierarchical analysis of facades, in a single framework.

APPENDIX A

OPTIMAL THRESHOLD DETERMINATION

Suppose that $\Theta = \{\theta_1, \theta_2, \dots, \theta_m\}$ is the initial patch set obtained through random sampling, and $W = \{w_1, w_2, \dots, w_m\}$ is the corresponding score set, the minimum and the maximum scores can be searched and denoted by w_{\min} , w_{\max} , respectively. We scale the scores within the range of $[0, 1]$ by updating $w_i = \frac{w_i - w_{\min}}{w_{\max} - w_{\min}}$, followed by dividing the unit range into a certain number of uniform intervals (e.g., 100). Therefore, we are able to obtain the histogram of the scores. On this basis, we define the probability of w_i as:

$$p(w_i) = \frac{f(w_i)}{\sum_{i=1}^m f(w_i)} \quad (13)$$

where $f(w_i)$ is the frequency of w_i in the histogram. Let w^* be the threshold, the probability mass functions of the weak and significant patches can be defined as:

$$\begin{aligned} \mathcal{P}_s(w^*) &= \sum_{0 \leq w_i \leq w^*} p(w_i) \\ \mathcal{P}_w(w^*) &= \sum_{w^* < w_i \leq \max_{1 \leq j \leq m} w_j} p(w_i) \end{aligned} \quad (14)$$

and the corresponding entropies are formulated as:

$$\begin{aligned} H_s(w^*) &= -\ln \sum_{0 \leq w_i \leq w^*} \left[\frac{p(w_i)}{\mathcal{P}_s(w^*)} \right]^2 \\ H_w(w^*) &= -\ln \sum_{w^* < w_i \leq \max_{1 \leq j \leq m} w_j} \left[\frac{p(w_i)}{\mathcal{P}_w(w^*)} \right]^2 \end{aligned} \quad (15)$$

The total amount of entropy is expressed as $H(w^*) = H_s(w^*) + H_w(w^*)$. The threshold can then be determined by:

$$\hat{w} = \arg \max_{w^*} H(w^*) \quad (16)$$

We can solve it efficiently by exploring all combinations here, as the number of combinations is relatively small.

APPENDIX B

CLUSTERING OVER PERMUTATIONS

Given the permutations $\mathbf{R}(\Theta_s)$, we model them with the probabilistic mixture model of t parts, i.e.:

$$Pr(\boldsymbol{\pi} | \{\mathbf{s}_i, c_i\}_{i=1}^t) = \sum_{i=1}^t \alpha_i \cdot M(\boldsymbol{\pi} | \mathbf{s}_i, c_i) \quad (17)$$

where the α_i 's are the mixture weights with $0 \leq \alpha_i \leq 1$ and $\sum_{i=1}^t \alpha_i = 1$. Each part of the mixture model observes a Mallows distribution [6]:

$$M(\boldsymbol{\pi} | \mathbf{s}, c) = \exp(-c \cdot K(\boldsymbol{\pi}, \mathbf{s}) - \log Z) \quad (18)$$

where \mathbf{s} is a permutation called the location parameter; $c \in \mathbb{R}_+$ is the concentration parameter; Z is a normalization constant; and $K(\cdot, \cdot)$ defines the Kendall's tau distance [34] on $\mathbf{R}(\Theta_s)$, which measures the similarity between patches.

If two patches are from the same structure, the distance value is small; otherwise, it is large.

Learning the mixture model in Eq. (17) can be conducted using the EM method [4] that requires to know the number of parts in advance, which is, however, unknown in our context. To this end, we employ the medoid shift technique [27] for mode clustering over permutations.

Given the permutations $\mathbf{R}(\Theta_s) = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots, \boldsymbol{\pi}_l\}$, kernel density estimation can be used to evaluate its underlying distribution [26]:

$$f(\boldsymbol{\pi}) = c_0 \sum_{i=1}^l \Phi\left(\frac{K(\boldsymbol{\pi}, \boldsymbol{\pi}_i)}{h^2}\right) \quad (19)$$

where $\Phi(\cdot)$ is a kernel function using the profile notation; c_0 is a positive scalar dependent on l and the bandwidth h . To initialize the mode-seeking, the mode of each candidate $\boldsymbol{\pi}_i$ is set as $\boldsymbol{\pi}_i^{(0)} = \boldsymbol{\pi}_i$. Each iteration of medoid shift moves along the direction of highest gradient to one element of the permutations. The permutation $\boldsymbol{\pi}_i^{(k)}$ in the k -th iteration step is updated as:

$$\boldsymbol{\pi}_i^{(k+1)} = \arg \min_{\boldsymbol{\pi} \in \mathbf{R}(\Theta_s)} \sum_{j=1}^l -K(\boldsymbol{\pi}_j, \boldsymbol{\pi}) \cdot \Phi'\left(\frac{K(\boldsymbol{\pi}_j, \boldsymbol{\pi}_i^{(k)})}{h^2}\right) \quad (20)$$

The iteration is repeated until convergence, i.e., $|\boldsymbol{\pi}_i^{(k+1)} - \boldsymbol{\pi}_i^{(k)}| < 10^{-6}$, which is guaranteed according to [27].

From Equation (20), we can see the mode yielded from medoid shift corresponds to the element in the permutations which best minimizes the function. As a result, for each $\boldsymbol{\pi}_i$, the corresponding convergent mode can be found from $\mathbf{R}(\Theta_s)$. Meanwhile, two permutations $\boldsymbol{\pi}_i, \boldsymbol{\pi}_j$, converge to the same mode if they are similar. i.e. the Kendall's tau distance between them is small. Accordingly, the similar permutations are grouped into the same cluster.

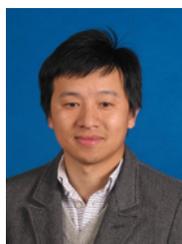
ACKNOWLEDGMENTS

We thank the reviewers for their comments and suggestions for improving the paper. Special thanks to Florent Lafarge, Aron Monszpart for their source code and datasets. This work was supported in part by the Fundamental Research Funds for the Central Universities (NE2014402, NE2016004), the Natural Science Foundation of Jiangsu Province (No. BK2014833), the Jiangsu Specially-Appointed Professorship, and the National Natural Science Foundation of China (No. 61402224, 61572507, 61532003, 61622212).

REFERENCES

- [1] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. 2001. Point set surfaces. In *Visualization 2001*. 21–28.
- [2] Murat Arikan, Michael Schwarzler, Simon Flory, Michael Wimmer, and Stefan Maierhofer. 2013. O-Snap: Optimization-based snapping for modeling architecture. *ACM Trans. on Graphics* 32, 1 (2013), 6:1–15.
- [3] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Joshua A. Levine, Andrei Sharf, and Claudio Silva. 2014. State of the Art in Surface Reconstruction from Point Clouds. *Eurographics STAR (Proc. of EG14)* (2014).
- [4] Ludwig M. Busse, Peter Orbanz, and Joachim M. Buhmann. 2007. Cluster analysis of heterogeneous rank data. In *Proc. Int. Conf. on Machine Learning*. 113–120.

- [5] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. 2004. Variational Shape Approximation. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 23, 3 (2004), 905–914.
- [6] Douglas E. Critchlow. 1985. *Metric Methods for Analyzing Partially Ranked Data*. Springer New York.
- [7] Fernando De la Torre and Michael J. Black. 2003. A Framework for Robust Subspace Learning. *Int. J. Computer Vision* 54, 1–3 (Aug. 2003), 117–142.
- [8] Xavier Decoret, Fredo Durand, Francois X. Sillion, and Julie Dorsey. 2003. Billboard Clouds for Extreme Model Simplification. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 22, 3 (2003), 689–696.
- [9] Ehsan Elhamifar and Rene Vidal. 2009. Sparse subspace clustering. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*. 2790–2797.
- [10] Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (June 1981), 381–395.
- [11] Shachar Fleishman, Daniel Cohen-Or, and Claudio T. Silva. 2005. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.* 24, 3 (July 2005), 544–552.
- [12] Ran Gal, Ariel Shamir, Tal Hassner, Mark Pauly, and Daniel Cohen-Or. 2007. Surface Reconstruction using Local Shape Priors. In *Proc. Eurographics Symp. on Geometry Processing*. 253–262.
- [13] Ali S. Hadi. 1992. Identifying Multiple Outliers in Multivariate Data. *Journal of the Royal Statistical Society* 54, 3 (1992), 761–771.
- [14] Hossam Isack, Yuri Boykov. 2012. Energy-based geometric multi-model fitting. *IJCV* 97, 2, 123–147.
- [15] Leif Kobbelt Jianhua Wu. 2005. Structure Recovery via Hybrid Variational Surface Approximation. *Computer Graphics Forum* 24, 3 (2005), 277–284.
- [16] G. Joseph. 2005. *Fundamentals of Remote Sensing*. Universities Press.
- [17] Florent Lafarge and Pierre Alliez. 2013. Surface reconstruction through point set structuring. *Computer Graphics Forum* 32, 2:225–234.
- [18] Yangyan Li, Xiaokun Wu, Yiorgos Chrysanthou, Andrei Sharf, Daniel Cohen-Or and Niloy J. Mitra. 2011. GlobFit: Consistently fitting primitives by discovering global relations. *ACM TOG* 30.
- [19] Hui Lin, Jizhou Gao, Yu Zhou, Guiliang Lu, Mao Ye, Chenxi Zhang, Ligang Liu, and Ruigang Yang. 2013. Semantic decomposition and reconstruction of residential scenes from LiDAR data. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 32, 4 (2013), 66:1–10.
- [20] Jinjie Lin, Daniel Cohen-Or, Hao Zhang, Liang Cheng, Andrei Sharf, Oliver Deussen, and Baoquan Chen. 2011. Structure-Preserving Retargeting of Irregular 3D Architecture. *ACM Trans. on Graphics* 30, 6 (2011), 183:1–10.
- [21] Przemyslaw Musialski, Peter Wonka, Daniel G. Aliaga, Michael Wimmer, Luc van Gool, and Werner Purgathofer. 2013. A Survey of Urban Reconstruction. *Computer Graphics Forum* 32, 6 (2013), 146–177.
- [22] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. 2003. Multi-level Partition of Unity Implicit. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* (2003), 463–470.
- [23] Cengiz Oztireli, Gael Guennebaud, and Markus Gross. 2009. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. *Computer Graphics Forum* 28(2):493–501.
- [24] Lance Parsons, Ehtesham Haque, and Huan Liu. 2004. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.* 6, 1 (June 2004), 90–105.
- [25] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. 2007. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum* 26, 2 (2007), 214–226.
- [26] Wand, M.P., M. Jones, *Kernel Smoothing*. Chapman & Hall, 1995.
- [27] Yaser Ajmal Sheikh, Erum Arif Khan, and Takeo Kanade. 2007. Mode-Seeking by Medoidshifts. In *Proc. Int. Conf. on Computer Vision*. 1–8.
- [28] Chao-Hui Shen, Shi-Sheng Huang, Hongbo Fu, and Shi-Min Hu. 2011. Adaptive partitioning of urban facades. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 30, 6 (2011), 184:1–10.
- [29] Hanzhi Wang, Tat-Jun Chin, and D. Suter. 2012. Simultaneously Fitting and Segmenting Multiple-Structure Data with Outliers. *IEEE Trans. Pattern Analysis & Machine Intelligence* 34, 6 (2012), 1177–1192.
- [30] Roberto Toldo and Andrea Fusiello. 2008. Robust multiple structures estimation with j-linkage. In *Proc. Euro. Conf. on Computer Vision*. 537–547.
- [31] Chen, H., P. Meer. 2003. Robust Regression with Projection Based M-estimators. *IEEE International Conference on Computer Vision*. 878–885.
- [32] L. Xu, E. Oja, and P. Kultanen. 1990. A new curve detection method: randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338.
- [33] Chin, T.-J., H. Wang, D. Suter. 2009. Robust Fitting of Multiple Structures: The Statistical Learning Approach. *IEEE International Conference on Computer Vision*. 413–420.
- [34] Jun Wang, Kai Xu, Ligang Liu, Junjie Cao, Shengjun Liu, Zeyun Yu, and Xianfeng David Gu. 2013. Consolidation of Low-quality Point Clouds from Outdoor Scenes. *Computer Graphics Forum* 32, 5 (2013), 207–216.
- [35] Aloysius Wehr and Uwe Lohr. 1999. Airborne laser scanning: An introduction and overview. *ISPRS Journal of Photogrammetry and Remote Sensing* 54, 23 (1999), 68–82.
- [36] Jui-Cheng Yen, Fu-Juay Chang, and Shyang Chang. 1995. A new criterion for automatic multilevel thresholding. *IEEE Trans. on Image Processing* 4, 3 (1995), 370–378.
- [37] Qian Zheng, Andrei Sharf, Guowei Wan, Yangyan Li, Niloy J. Mitra, Daniel Cohen-Or, and Baoquan Chen. 2010. Non-local scan consolidation for 3D urban scenes. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 29, 4 (2010), 94:1–9.
- [38] Aron Monszpart and Nicolas Mellado and Gabriel Brostow and Niloy Mitra. 2015. RApter: Rebuilding Man-made Scenes with Regular Arrangements of Planes. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 34, 4 (2015), Article No. 103.
- [39] Sven Oesau, Florent Lafarge, Pierre Alliez. 2015. Planar Shape Detection and Regularization in Tandem. *Computer Graphics Forum. DOI : 10.1111/cgf.12720*.
- [40] Qifa Ke, Takeo Kanade. 2004. Robust Subspace Clustering by Combined Use of kNN Metric and SVD Algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition 2004*, 592–599.
- [41] Ren Vidal, Yi Ma, Jacopo Piazzi. 2004. A New GPCA Algorithm for Clustering Subspaces by Fitting, Differentiating and Dividing Polynomials. In *IEEE Conference on Computer Vision and Pattern Recognition 2004*, 510–517.



Jun Wang is currently a professor at Nanjing University of Aeronautics and Astronautics (NUAA), China. He received his Bachelor and PhD degrees in Computer-Aided Design from NUA in 2002 and 2007 respectively. From 2008 to 2010, he conducted research as a postdoctoral scholar at the University of California and the University of Wisconsin. From 2010 to 2013, he worked as a senior research engineer at Leica Geosystems, USA. In 2013, he paid an academic visit to the Department of Mathematics at Harvard University. His research interests include geometry processing and geometric modeling.



Kai Xu received his PhD in Computer Science at National University of Defense Technology (NUDT). He is currently a postdoctoral researcher at Shenzhen Institutes of Advanced Technology and also holds a faculty position at NUDT. From Nov. 2008 to Oct. 2010, he visited the GrUVi Lab of Simon Fraser University, supported by the Chinese government. His research interests include geometry processing and geometric modeling, especially on data-driven approach to the problems in those directions. He is the recipient of LU Zengyong High-tech Award on CAD&CG (2nd place) in 2013. He serves on the editorial board of *Computers and Graphics journal*.