





Geometric Primitives in LiDAR Point Clouds: A Review

Shaobo Xia , Dong Chen , *Member, IEEE*, Ruisheng Wang , *Senior Member, IEEE*, Jonathan Li , *Senior Member, IEEE*, and Xinchang Zhang

Abstract—To the best of our knowledge, the most recent light detection and ranging (lidar)-based surveys have been focused only on specific applications such as reconstruction and segmentation, as well as data processing techniques based on a specific platform, e.g., mobile laser. However, in this article, lidar point clouds are understood from a new and universal perspective, i.e., geometric primitives embedded in versatile objects in the physical world. In lidar point clouds, the basic unit is the point coordinate. Geometric primitives that consist of a group of discrete points may be viewed as one kind of abstraction and representation of lidar data at the entity level. We categorize geometric primitives into two classes: shape primitives, e.g., lines, surfaces, and volumetric shapes, and structure primitives, represented by skeletons and edges. In recent years, many efforts from different communities, such as photogrammetry, computer vision, and computer graphics, have been made to finalize geometric primitive detection, regularization, and in-depth applications. Interpretations of geometric primitives from multiple disciplines try to convey the significance of geometric primitives, the latest processing techniques regarding geometric primitives, and their potential possibilities in the context of lidar point clouds. To this end, primitive-based applications are reviewed with an emphasis on object extraction and reconstruction to clearly show the significances of this article. Next, we survey and compare methods for geometric primitive extraction and then review primitive regularization methods that add real-world constraints to initial primitives. Finally, we summarize the challenges, expected applications, and describe possible future for primitive extraction methods that can achieve globally optimal results efficiently, even with disorganized, uneven, noisy, incomplete, and large-scale lidar point clouds.

Index Terms—Edges, geometric primitives, light detection and ranging (lidar), lines, planes, point clouds, regularization, skeletons, volumetric shapes.

I. INTRODUCTION

LIGHT detection and ranging (lidar) is a kind of remote sensing technique, which uses light in the form of a pulsed laser to measure the distance from the sensor to the objects. From the perspective of surveying principles, the existing lidar systems can be categorized into time-of-flight and phase-shift systems. The former measures distance by recording the transmission time of light between the laser sensor and objects, while the latter uses the phase difference between emitted and backscattered signals to determine the measure distance. Although the time-of-flight system is less accurate than the phase-shift system, it has the capability to survey longer-range based on the different operational platforms, i.e., space, airborne, and ground-based. Space-based lidar is installed on satellites such as ICESat [1]. Airborne lidar or airborne laser scanning (ALS) refers to the system running on aircraft, including unmanned aerial vehicles. Ground-based lidar includes terrestrial laser scanning (TLS) and mobile laser scanning (MLS) systems. Due to advances in manufacturing technologies, lidar has become more popular, more accessible, and more widely used than in the first decade of the 21st century. For example, ground-based lidar's ability to acquire precise high-resolution 3-D details has become an attractive technology for environmental mapping, including pavement identification [2], [3], road marker extraction [4], [5], curb modeling [6], [7], vegetation mapping [8], [9], pole-like object localization [10], [11], vehicle detection [12], [13], and building reconstruction [14], [15]. (A recent review of MLS data processing and applications can be found in [16].) However, gaps between lidar research and practice remain, with notable deficiencies in scalability and robustness of current processing methods.

Except for some satellite-based lidar systems that output only waveforms [17], most lidar systems output rich information, including intensity, full waveforms, pulse count, incidence angle, and point coordinates. The intensity is the amplitude of the received signal, which is influenced by various factors, such as range, emitted energy, incidence angle, and the backscattering characteristics of an object [18]. These systems require intensity calibration before further processing, but intensity calibration remains difficult as the intensity value is not as reliable as coordinates [19], [20]. ALS processing sometimes decomposes the full

Manuscript received February 20, 2019; revised October 1, 2019 and December 25, 2019; accepted January 18, 2020. Date of publication January 31, 2020; date of current version February 13, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 41971415, in part by the National Key R&D Program of China under Grant 2018YFB2100702, and in part by the Natural Sciences and Engineering Research Council. The work of S. Xia was supported in part by the China Scholarship Council and in part by the University of Calgary. (*Shaobo Xia and Dong Chen contributed equally to this work.*) (*Corresponding authors: Ruisheng Wang; Xinchang Zhang.*)

S. Xia is with the Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: shaobo.xia@ucalgary.ca).

D. Chen is with the College of Civil Engineering, Nanjing Forestry University, Nanjing 210037, China (e-mail: chendong@njfu.edu.cn).

R. Wang is with the School of Geographical Sciences, Guangzhou University, Guangzhou 510006, China, and also with the Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: ruiswang@ucalgary.ca).

J. Li is with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

X. Zhang is with the School of Geographical Sciences, Guangzhou University, Guangzhou 510006, China (e-mail: eeszxc2018sysu@outlook.com).

Digital Object Identifier 10.1109/JSTARS.2020.2969119

waveform to increase the density of point clouds [21] or estimate physical parameters of vegetation [22]. However, performing waveform decomposition is not required in most applications. To obtain color information, extra cameras, registration, and calibration are needed. Furthermore, most lidar point clouds are unorganized when compared with 2-D images obtained from cameras and point clouds captured by equipment, such as Microsoft Kinect and Google Tango Tablet. In such unorganized point clouds, the spatial relationship between neighboring points is not stored, and data cannot be indexed by row or column numbers. As a result, processing of lidar data in most contexts refers to the use of algorithms for processing unorganized and colorless 3-D point clouds using only 3-D coordinates.

The core of point cloud processing is to infer geometric information from 3-D discrete and unordered points. The basic geometric unit is the point, with groups of points forming geometric primitives. We group geometric primitives into two categories: shape primitives and structure primitives. Shape primitives include lines (e.g., straight and curved lines), surfaces (e.g., planes), and volumetric shapes (e.g., cubes and cylinders). Structure primitives include skeletons, 2-D outlines, and 3-D edges. The term “structure” is used because primitives in this category often constitute simplified object profiles. These structure primitives may also be viewed as advanced primitives derived from shape primitives. Similarly, contours are also regarded as geometric primitives in both 2-D images [23] and 3-D point clouds [24]. Although the lines in shape primitives and the 2-D/3-D edges in structure primitives look similar, they are quite different. Line primitives are linear objects such as power lines and thin poles, while edges delineate surfaces and volumetric shapes. They are used in different scenarios. For example, utility poles are represented by line primitives, which are being used in city asset management. Road curbs modeled by edges contribute to high-definition mapping for autonomous driving. Fig. 1 shows an overview of these two classes of primitives.

Objects in our environment, especially human-made ones, can be described by various geometric primitives and their combinations. For example, straight and curved lines often model curbs [25], [26]. Power lines are fitted by a series of piecewise straight lines [27] and/or a catenary curve model [28]. Volumetric shapes like cylinders approximate streetlight poles and tree trunks [29], [30]. Thus, geometric primitives are useful in object recognition, scene understanding, and reconstruction. For instance, the detection of volumetric shapes, such as cylinders and cubes, happens first, followed by a probabilistic graph matching to verify the shapes in the scene [31]. Furthermore, planar primitives [32], [33], 2-D outlines [34], and 3-D edges [35] are all basic elements in reconstructing various man-made objects.

Except for the aforementioned tasks, geometric primitives are also the basis for many other applications. For ground filtering, ground points can be roughly identified and removed by estimating the largest plane from the input point clouds [36]. For ground interpolation, edges detected from point clouds are the basis for breakline-preserving interpolation when filling holes in MLS point clouds [37]. For segmentation, edges in MLS point clouds enable the instance-level segmentation of buildings [38]. In lidar calibration, geometric primitives are also useful. Chan *et al.* [39]

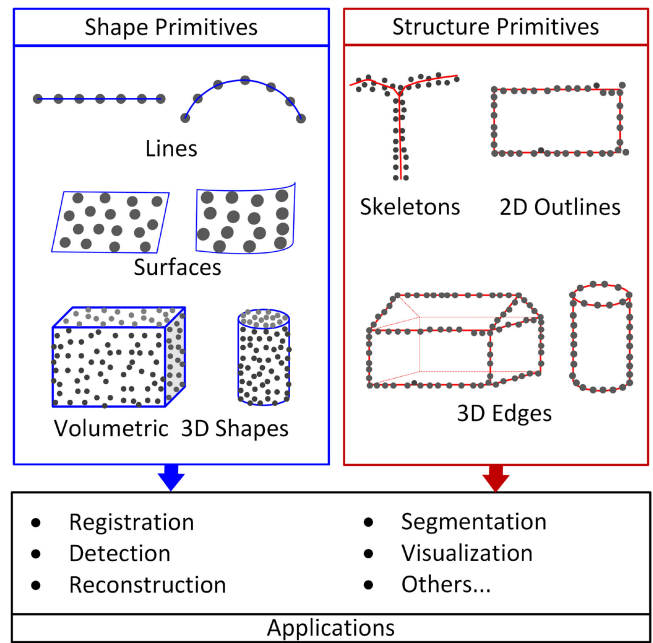


Fig. 1. Classification of geometric primitives in lidar point clouds.

proposed a rigorous self-calibration algorithm based on cylinder detection in a scene. The authors found that cylinder-based calibration outperformed plane-based calibration in several aspects; in particular, the former method relied less on observations than the latter method.

3-D edges in point clouds are also useful for registering point clouds. For example, the extracted crest curves of objects become the basis for registration [40]. Lin *et al.* [41] presented a registration method for outdoor point clouds using line segments detected from edges. Their method identifies corresponding matches and refines the matches using the iterative closest point algorithm. For the registration of images and lidar point clouds, Habib *et al.* [42] proposed a method using linear edge primitives estimated from plane intersections. Some registration techniques have used other shape primitives. Rabbani *et al.* [43] proposed a point cloud registration framework with no artificial targets and incorporated primitives, such as planes, spheres, and cylinders. This method first segments point clouds into regions, which are fitted using predefined models. Subsequently, registration is performed by minimizing the differences between the model parameters in different scans or the orthogonal distance between the points and fitted models. Von Hansen [44] presented a plane-based registration method for ground-based lidar point clouds. The method first divides point clouds into voxels, whose dominant planes are estimated using random sample consensus (RANSAC). The neighboring coplanar planes are then merged into larger shape primitives. Subsequently, the registration is conducted in two steps. The first step consists of finding matched primitives using an exhaustive search, and in the second step, the transformation parameters are calculated using least-squares adjustment. Recently, Xu *et al.* [45] have proposed a coarse registration algorithm based on planar patches in lidar point clouds.

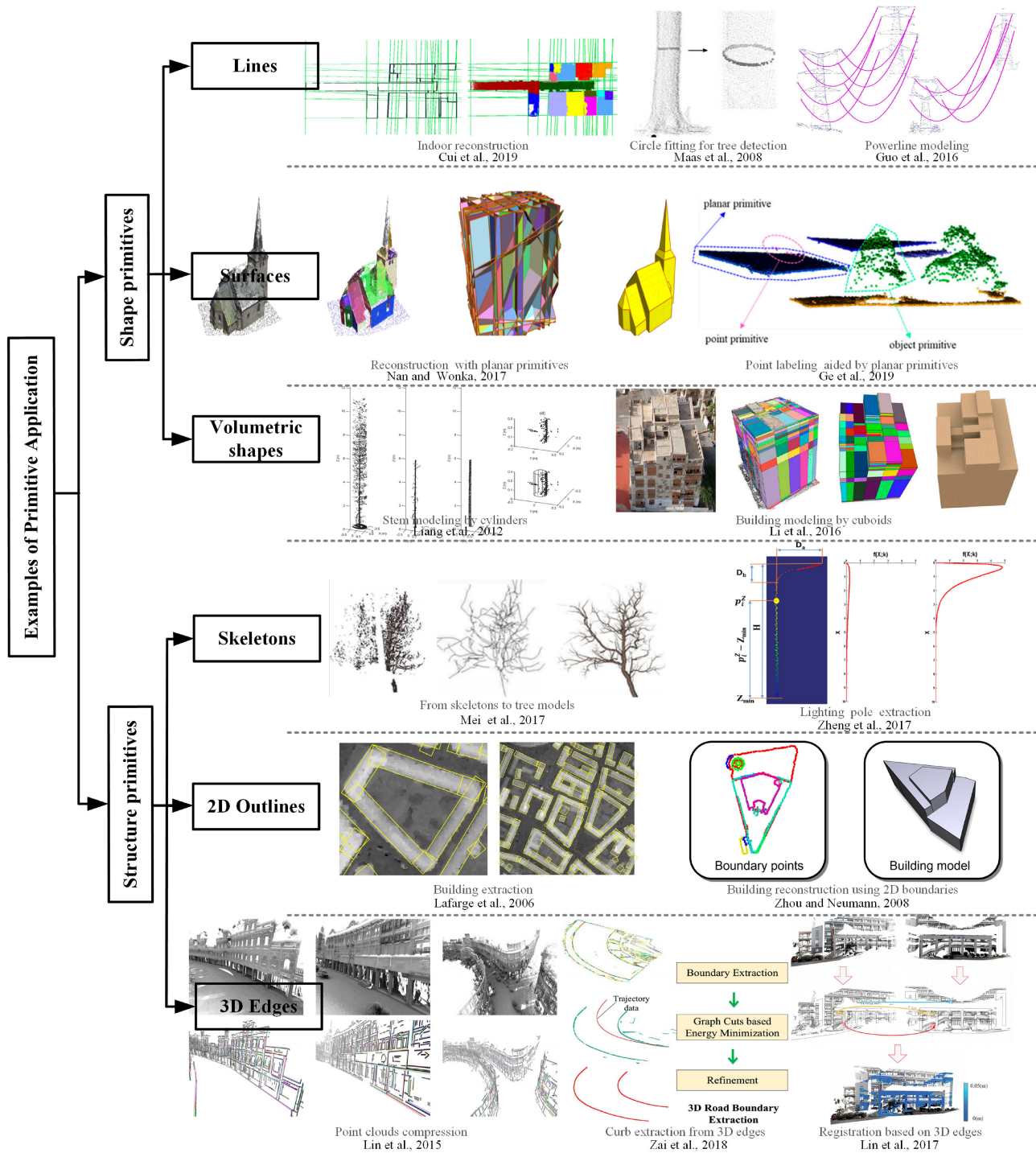


Fig. 2. Examples of geometric primitive-based applications.

To highlight the role of geometric primitives and convey the motivation why our review on this topic is needed, we provide an overview of typical applications using geometric primitives (see Fig. 2) to clarify the role of geometric primitives and explain our motivation for conducting a review on this topic. Object extraction and reconstruction are two application areas that have attracting the most attention in recent years.

- 1) *Object extraction*: The extraction of various objects from raw point clouds, including human-made objects and natural objects, such as vegetation, has been widely studied. Wang and Shan [46] introduced the use of roof contours to separate structures and vegetation for building detection. Buildings with rectangular outlines can be identified in ALS data applying a rectangle detection algorithm on

DEM derived from the point clouds [47]. In MLS data, buildings are often treated as objects consisting of multiple planes that are detectable with segmentation methods [48]. For instance, Rutzinger *et al.* [49] presented a rule-based method to detect buildings; non-wall segments were removed from the region-growing results of the input point clouds. Rules defining the segment size, number of points, and orientations facilitated filtering out the non-wall segments [49], [50]. Xia and Wang [51] proposed a building localization method for MLS point clouds. Their method extracts planar primitives and projects them as 2-D line segments. Subsequently, buildings are localized through the rectangles generated by these line segments. Planar primitives are also useful as geometric constraints that effectively discriminate between buildings and adjacent vegetation. In [52], the idea of using geometric primitive priors is also applied to improve the classification accuracy of ALS point clouds.

The extraction of pole-like objects, such as utility poles, light poles, and tree trunks, is also of research interest. Zheng *et al.* [53] presented a light pole extraction method that incorporated prior knowledge of the pole skeleton into the segmentation step. More specifically, it approximated the skeleton of light poles using a gamma function and used the distance between points and the potential skeleton as an important criterion for segmentation. Skeleton information was also used for detecting pole-like objects in [54]. In this work, the skeletons of pole-like object candidates were derived using Laplacian smoothing, and detection was achieved with a PCA-based object recognition method. Cylinder or circle detection methods can detect tree trunks in a forest automatically [30], [55], [56]. For example, Li *et al.* [57] performed pole detection using adaptive cylinder fitting, which is also used in the tree trunk detection. Maas *et al.* [58] extracted horizontal slices from raw point clouds and detected tree trunks using multiple circles extracted from the slices.

Algorithms for power line extraction project the candidate points of power line horizontally and then detect power line using a Hough transform (HT) [59]. Curbs, which are linear ground structures, are key features in many applications, including autonomous driving and road mapping. Road curbs in MLS point clouds can be viewed as contours. For example, Xu *et al.* [25] detected curb candidates by detecting edges among ground points. Others used a similar method for curb extraction by starting with 3-D edges [26].

The importance of geometric primitives in object extraction is twofold. First, geometric primitives can be treated as part of the objects of interest, which make object detection roughly equivalent to extracting primitives as in the case of detecting buildings using planes [51]. Second, applications can use geometric primitives as constraints or supplementary information during segmentation, such as the use of skeletons to extract poles [53]. Compared with pointwise features [60], geometric primitives are tolerant of noise and data gaps. Additionally, geometric primitives

provide shape priors of various objects, which provides useful high-level information for further processing.

- 2) *Reconstruction*: Object reconstruction from points has attracted increased attention recently because the models provide an elegant way to achieve abstraction and geometric representation of object entities embedded in discrete points. Next, we focus on reconstruction methods that make direct use of shape and structure primitives.

Line primitives are extensively used in model reconstruction. For example, Guo *et al.* [61] used the RANSAC method to fit curved line models to power-line point clouds in one span. Oesau *et al.* [62] projected wall points horizontally, detected line segments using multi-scale RANSAC, and obtained the indoor structures of a building by analyzing the intersections between line segments. Similar procedures for line primitive-based reconstruction were also used in [63]–[65]. Yang *et al.* [66] projected wall points in MLS point clouds on the ground and then detected and connected linear primitives to construct 2-D building footprints.

Other shape primitives such as planes can also be used to develop models, as in the building reconstruction study by Li *et al.* [32]. The authors detected planar segments using RANSAC and then extended the plane primitives to obtain 3-D intersections, thus generating multiple boxes. Building reconstruction was achieved by using a box selection algorithm. Nan and Wonka [33] generalized this idea by discarding the Manhattan world assumption. Plane primitives can also approximate polygonal surfaces. Monszpart *et al.* [67] focused on representing objects with regularized primitives. Similar projects also produced models by integrating multiple primitives [68], [69]. Furthermore, basic shape primitives can be used to determine complex components of buildings. Poullis and You [70] proposed a model-driven building reconstruction method, where simple modules consisted of planes defined to fit the input data with symmetry constraints. Lin *et al.* [15] segmented building point clouds into planar structures that were then combined into building components, such as chimneys, a box, and a couple of walls, using prior knowledge of the buildings at different levels. The final model was constructed by integrating various components with semantic labels.

Structure primitives are often treated as the basis in many reconstruction studies. Cao *et al.* [35] first extracted edges by setting thresholds for local geometric features and then connected them into a curve network that was completed with a clustering method. The closed and regularized curves were combined to generate the resulting model. Roof boundaries also help with retrieving of building models from ALS point clouds. Poullis [34] presented a framework for reconstructing buildings from ALS data. In this framework, the points were first segmented into planar segments using an unsupervised clustering method, and then, segment boundaries were extracted and regularized for roof modeling. Zhou and Neumann [71] developed models of buildings using detected 2-D outlines of plane

segments from ALS point clouds. Chen *et al.* [72] used roof boundaries as geometric primitives and generated multiple levels of detail using boundary assembly strategies. Yi *et al.* [73] first extracted the contour points of buildings and then divided the data into blocks. In each block, 2-D line primitives were extracted from the sliced point clouds. The final model was created by assembling the block models at different elevations. 3-D edges in point clouds also serve as geometric constraints during reconstruction [74]. Surface models can be expanded from detected skeletons [75]. In the case of tree modeling, models are often expanded from tree skeletons. For example, Livny *et al.* [76] first constructed and refined 3-D skeletons of trees using input point clouds. Subsequently, the authors used cylinders to inflate the skeleton into a volumetric tree model. Other researchers have proposed similar methods for tree reconstruction using skeletons [77], [78]; most of the methods emphasized robust skeleton generation and handling of missing trunks.

To sum up, geometric primitives are not only the basis for reconstructions, but also serve as the key in many other point clouds processing workflows (e.g., segmentation), whose performance often depends on the quality of the extracted geometric primitives. Thus, many lidar data processing methods may benefit from a comprehensive review of geometric extraction methods used in different research areas. Our focus in this article is on inferring shape and structure geometric primitives and their applications in lidar point clouds. We argue that many data processing problems can be partly solved or even well addressed if the potential of geometric primitives is fully explored.

Several reviews on lidar data processing have been published in recent years [16], [79]–[83]. These surveys only focused on certain applications, such as reconstruction [79], [80], [82], or specific data processing steps, such as segmentation, classification, and object recognition [81], [83], or point clouds acquired by a specific type of platform such as mobile lidar [16], [83]. In contrast, our review focuses on the basic elements in lidar point clouds and the intrinsic problems in data processing and is not limited to data collection platforms or specific applications. In other words, this review provides an overview and also a perspective on how to perceive and understand lidar point clouds based on geometric primitives.

The rest of this article is organized as follows. In Section II, we review the state-of-the-art methods for extracting different primitives. Section III summarizes the methods for primitive regularization. In Section IV, the relationship between pointwise features and geometric primitives, and the differences between lidar point clouds and other point clouds in terms of processing primitives are discussed. We conclude this article along with a few suggestions for future research topics in Section V.

II. PRIMITIVE EXTRACTION METHODS

A. Shape Primitives

In general, the majority of shape primitives refer to lines, surfaces, and volumetric elements. Existing methods can be classified into four classes: *region-growing methods*, *accumulation*

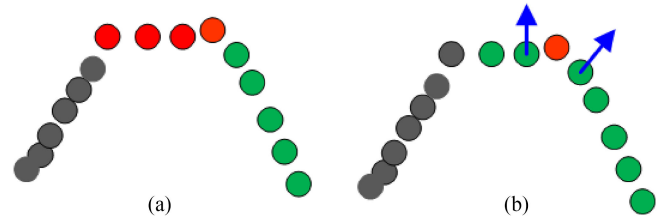


Fig. 3. Simple seed selection example. (a) Points in three segments have different colors. (b) If the red point at the corner is selected as a seed, only two segments would be identified.

methods, *hypothesis and selection methods*, and *clustering-based methods*.

1) *Region-Growing Methods*: Region growing in point clouds refers to algorithms that segment points into homogeneous regions based on local indicators. After selecting seed points, these algorithms determine whether nearby points should be grouped into the same segment according to the similarity between local features such as normals. They terminate if every point belongs to a particular segment.

The choice of neighborhood size, predefined merging rules, and seed selection are critically important for region growing. Most methods define the neighborhood size of each point using the k -nearest neighbors or fixed distance neighbor methods [84]. Point density influences the performance of both methods. Neighborhood size is usually selected empirically based on the data properties and held constant during the growing process. In general, the algorithm uses the normal difference to evaluate similarity within a point neighborhood. If the included angle between normals at points falls below a given threshold, the points will be grouped into the same cluster [84]. We note that a large threshold for the normal difference tends to undersegmentation results, while a small threshold may result in oversegmentation. In practice, the angle threshold is often set empirically based on data quality and accuracy requirements in different tasks.

In general, there are two ways to improve the performance of region growing. The first is to improve the accuracy of estimated local features [85]. For example, Nurunnabi *et al.* [86] proposed a normal estimation method using robust principal component analysis with neighboring points, yielding significant improvements in their segmentation tests on the cylinder, house, and traffic furniture scenes. The second is to reduce the threshold dependence. For example, Maalek *et al.* [87] identified outliers before clustering neighboring segments. Local statistical features are also extensively used as growth indicators. For example, Lin *et al.* [41] proposed a number of false alarm indicators to control the grouping process for line segments.

In the case of seed selection, erroneous results may occur when seeds are located along sharp edges or in noisy areas. For example, the points in Fig. 3(a) can be divided into three segments of different colors. Selecting the red point on the corner in Fig. 3(b) as a seed causes the normal directions (blue arrows) around the seed point to be close to each other, which may lead to undersegmentation (i.e., only two segments will be detected). A standard solution is the addition of smoothness constraints when selecting seeds. For example, the algorithm

selects seeds from flat regions based on local geometric features like curvature or flatness [84], [88], [89]. Region growing can also be accelerated by replacing pointwise growing with voxel-wise growing [90]–[92]. For example, Xu *et al.* [93] proposed a voxel-based primitive detection method that organizes points into voxels that are then aggregated into segments based on the similarity (e.g., distance and geometric saliency) between adjacent voxels. Finally, model fitting enables further recognition of the primitive types (e.g., plane and cylinder) of detected segments [43].

In general, region growing is a local-based method for shape primitive extraction from point clouds. It is easy to implement and often produces promising segmentation results [89]. Additionally, region growing can be implemented in parallel [94], thereby making processing large-scale point clouds possible. However, it suffers from noise, low accuracy of local features, inappropriate thresholds, and poor seed selection. It should be noted that region growing is also a typical locally optimal method, which means that the previous results can affect later ones, and repeated execution would not always output identical results.

2) *Accumulation Methods*: Accumulation methods are algorithms that work in parameter space and always contain a voting step. The HT [95] is a standard accumulation method with three broad steps: build the parameter space, accumulate votes from the input data, and output primitives identified in the parameter space [96]. The initial HT was proposed in [95] and then was generalized by Ballard [97] to detect various curves like lines and circles.

Taking the straight-line model as an example, there are two parameters: the distance ρ between the coordinate origin and the closest point on the line, and the angle θ between the line direction and a reference axis. The commonly used accumulation space is lattice-like structures, like rectangular cells in 2-D models, but inappropriate construction of the accumulation space may lead to detection failures. For example, when detecting a plane in 3-D, dividing the parameter space by latitude and longitude causes cells near the poles to be smaller than cells near the equator, biasing accumulation toward large cells and away from small cells [98]. To address this problem, Borrmann *et al.* [98] proposed a ball accumulator to divide the voting space equally.

As one survey of HT implementations has summarized, there are many variants but with the voting method being the significant difference between them [99]. In traditional voting approaches, every point casts a group of votes in the parameter space by fitting shape models with different parameters [100], which is time-consuming for massive lidar point clouds. The randomized HT [101] used a different approach for line detection: two pixels in the image space form one vote in the parameter space. Borrmann *et al.* [98] implemented a 3-D randomized HT for plane detection by randomly choosing three points to form a vote. The experimental results with point clouds show that the randomized method performed better than other variants. Camurri *et al.* [102] implemented both traditional and randomized HTs to detect spheres, but they used every point together with its associated normal to determine the plane in the 3-D space corresponding to one vote in the Hough space. This was in lieu of

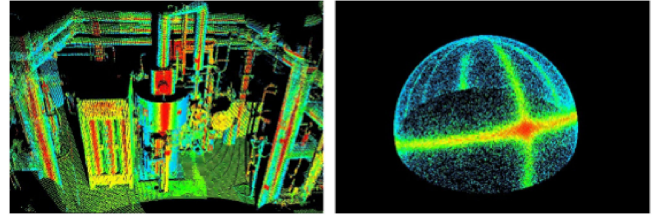


Fig. 4. Example of Gaussian sphere (on the right), where normal directions of the input point clouds (on the left) are accumulated [106].

randomly generating planes to cast votes [103], [104]. To detect peaks in the parameter space, a straightforward way is setting a threshold to find cells corresponding to shape primitives in the accumulation [100]. In [98], a cell with accumulated votes over 90% of the highest score will be retrieved as a plane. Peaks and their associated votes can also be identified by mean shift [105] and region growing [104]. Many accumulation algorithms are unable to separate coplanar segments without further processing.

Accumulation methods can also apply to the detection of volumetric shapes simply by using more parameters. For example, a cylinder description requires five parameters: two angular parameters for axis direction, one parameter for the radius, and two parameters for the center position. However, a Hough space with more than three dimensions is impractical due to the complexity of the parameter space. Thus, research into using accumulation methods for detecting volumetric primitives has focused primarily on improving computational efficiency.

One possible way to handle high-dimensional problems is to divide the accumulation space into smaller spaces to accumulate votes separately [106]. Hulik *et al.* [103] introduced a hierarchical structure to build such a voting space. Another solution for reducing the size of the problem is to handle the angular parameters (θ , ϕ) and distance ρ separately by using normals and Gaussian spheres [106]. For cylinder detection, point normals are used to identify the orientations of potential cylinders in a Gaussian sphere, as shown in Fig. 4, with points then projected onto a plane whose normal is in line with the cylinder axis. After applying a 2-D HT to detect circles on the projection plane, the position and radius of the cylinder can be calculated. For sphere detection, normal directions are used to vote for the sphere center followed by the determination of the sphere's radius.

Qiu *et al.* [107] used a Gaussian sphere to accumulate axis orientations. Compared with the work of Vosselman *et al.* [106], a different 2-D circle fitting method is used, which determines the circle center by line intersections on the projected plane. Rabani and Van Den Heuvel [108] implemented a similar process of decomposing a complex problem into smaller ones by using two Gaussian spheres: one for counting point normals and the other for accumulating cylinder orientations. After recovering the axis orientations, their method estimates the cylinder radius and position. In a shift away from decomposition, Drost and Ilic [109] proposed a local HT method that introduces point pair features to primitive detection and a vote casting method in submanifolds. A non-maximum suppression can refine the extracted primitives. Experiments with this approach show competitive results.

It is worth noting that the idea of voting and accumulation has also been applied in object detection. For example, Leibe *et al.* [110] proposed an implicit shape model (ISM) for detecting objects in images, which first trains a codebook of patchwise features. Then, if a new patch is matched to a cluster in the codebook, a vote for the potential object center will be cast based on the spatial probability distribution obtained in the training step. The ISM is also extended to detect objects in 3-D point clouds [111]. Although the ISM is designed for object detection, this voting framework may also help identify geometric primitives in point clouds.

In summary, accumulation methods are easy to implement and robust when the data have noise or omissions. The voting space structure and thresholds critically influence detection results as well. Oesau *et al.* [94] also pointed out that accumulation-based methods are sensitive to the selection of the coordinate origin, and the sequential extraction procedures of these methods are not suitable for global regularization. Finally, accumulation techniques have a high computational cost, especially for shape primitives with many parameters.

3) *Hypothesis and Selection*: Hypothesis and selection approaches have become popular in recent years. These methods first make primitive guesses followed by a model verification step. There are three broad types for this approach: *RANSAC*, *energy-based optimization*, and *preference analysis*.

- 1) *RANSAC*: The *RANSAC* method is a simple but powerful tool with wide application in outlier detection, shape detection, and registration. Fischler and Bolles [112] first developed *RANSAC* with others making additional improvements in the subsequent decades. For shape detection, *RANSAC* first selects a subset of the input data and then fits a predefined model and obtains the inliers of this hypothetical model from the input data. The algorithm repeats the above steps to generate enough model candidates, and the one with the most inliers can be regarded as an optimal model. An alternative way to evaluate an optimal model is to use the median of the distances to all points in the data, and the candidate with least median is selected as an optimal model [113]. Once the cost function has been defined, the maximum number of iterations can be implicitly defined based on the process of adaptive estimation, starting with the worst case and updating the estimation as the computation progress. If the proportion of inliers in data can be estimated in advance, the maximum number of iterations can also be inferred by a probabilistic inference. If there are multiple primitives, *RANSAC* executes multiple times to extract primitives one by one.

Elaborating further, *RANSAC* generates model guesses and then verifies the proposed models. Typically, the method implements a minimal subset sampling strategy, whereby a minimum number of data points is used to estimate the model parameters (e.g., two points for a line model and three points for a plane model). The advantage of this strategy is that using fewer points from the input data reduces the probability of using noise or other contaminated data. When doing minimal subset

sampling, points are often randomly sampled, which precludes using any prior knowledge of the input data. However, *RANSAC*'s efficiency can be improved when using sampling methods that take such prior knowledge into account. Ni *et al.* [114] assumed that the inliers tend to be close to each other and proposed a *GroupSAC* method that divides the input data into clusters and draws samples from clusters containing more inliers. Chum and Matas [115] enhanced the random sampling process by substituting sampling data with significant scores, which derived from prior or domain-specific knowledge, such as local features in 2-D images. Tennakoon *et al.* [116] studied the use of more than minimal subset sampling strategies with a goal of improving the model generation speed and the quality of guesses.

The verification step determines whether a model proposal is optimal for the given input data. Standard *RANSAC* selects the hypothetical model with the most supporters (i.e., the size of the consensus set of the sample that defines the inliers), a time-consuming step often requiring calculation of the distance between every input data point and the hypothetical model. To improve the efficiency, it is possible to filter out model proposals containing a large portion of outliers using fewer input data points. Chum and Matas [117] proposed a sequential probability ratio test to decrease the runtime of the verification step. It uses a statistical test on the sequential data and rejects the model if the likelihood ratio is larger than a given threshold. The runtime of this method is much less than standard *RANSAC*'s as the test often stops before examining all of the data. There are other statistical test methods for verification, such as the $T_{d,d}$ test [118] and the bail-out test [119]. All of these statistical methods improve the efficiency of standard *RANSAC* but increase the risk of rejecting correct models during the earlier steps and often require more hypothetical models. The aforementioned statistical methods all verify the hypothesized models one by one. Nistér [120] proposed a preemptive *RANSAC* for real-time applications that uses a breadth-first traversal to verify the proposals on a small set of input data. After generating a predefined number of model hypotheses and scoring them based on a subset of the input data, it selects models with high scores for further evaluation. A detailed discussion of standard *RANSAC* and its variants for image processing and photogrammetry applications can be found in [121] and [122]. In the following paragraphs, we focus only on *RANSAC* and its variants applicable to geometric shape detection in lidar point clouds.

Many researchers have studied the use of *RANSAC* for shape detection in lidar point clouds. Tarsha-Kurdi *et al.* [100] presented a detailed roof plane detection method using the standard deviation of the point-to-plane distance to verify model proposals rather than selecting the proposal with the most points as in standard *RANSAC*. This extension avoids selecting planes with a large number of supporters that are not real roof planes. Tarsha-Kurdi *et al.* [100] also noted that *RANSAC* is more efficient than

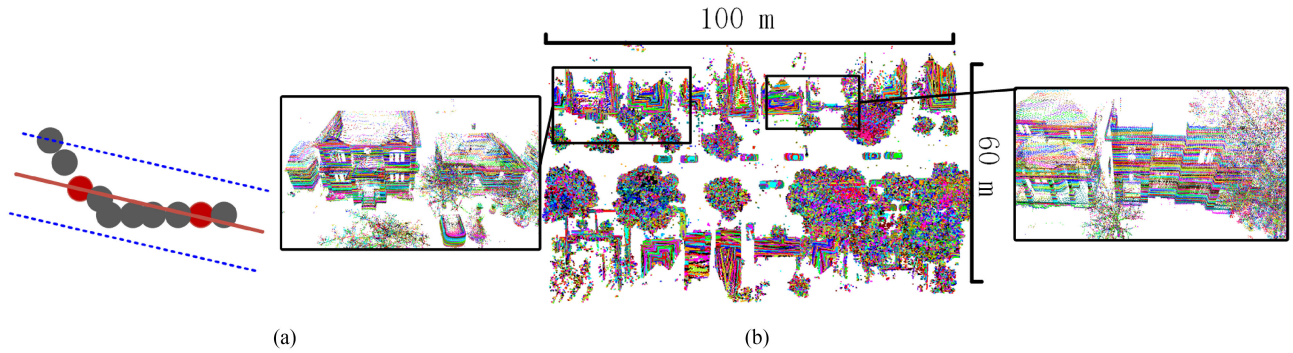


Fig. 5. Illustration of degeneracy. (a) Point clouds of small linear structures. (b) Degeneracy occurs when applying the standard RANSAC to a large scene of MLS point clouds. The input data are sliced vertically, and each slice, i.e., detected planes, is colored differently.

HTs in detecting roof planes. Chen *et al.* [123] applied a grid-based data structure to improve sampling efficiency and to decrease the number of wrong proposals. To handle the complex structure of planes, Yang and Förstner [124] proposed a plane detection method integrating RANSAC and the minimum description length (MDL) rule. After dividing the input data into groups and detecting planes in each group with RANSAC, the method estimates a number of detected planes in each group using the MDL rule. The detected planes are further merged by region growing. Schnabel *et al.* [125] proposed a framework to extract different shape primitives from point clouds that improves both the hypothesis and verification steps compared with standard RANSAC. Its model hypothesis step samples points with the help of an octree structure to maintain the spatial relationships between inliers and to increase the probability of generating good hypotheses. Its verification step scores and ranks model candidates by point features such as normals. Tran *et al.* [126] performed a primitive-type recognition step before extracting primitives. This method identifies the primitive type by finding the best-fit primitive model that has the smallest fitting error. Finally, planar surfaces are extracted using the RANSAC method proposed by Schnabel *et al.* [125]. The RANSAC variants mentioned above may detect shapes even when the input data contain more than 50% outliers. However, most of them only work well in point clouds of uniform point density for simple scenes. In point clouds from terrestrial lidar, data density varies, and scenes are complex. These factors greatly influence RANSAC's performance. The term “degeneracy” refers to the situation when data provide insufficient information to obtain a unique solution [122]. In Fig. 5(a), points form two line primitives. RANSAC will obtain a red line model if two red points are sampled. In this situation, all of the points are model inliers as indicated by two dashed blue lines. Thus, RANSAC will extract only one line having all of the points as inliers. Fig. 5(b) shows a real example of running RANSAC in large-scale MLS point clouds. Instead of extracting independent walls in each building, it selects models consisting of points in close elevations. Thus, the

algorithm detects stacked vertical planes while missing the actual planar structures. These problems with RANSAC processing of lidar clouds are difficult to solve. Similar to the idea of LO-RANSAC [116], which was proposed for epipolar geometry detection, one possible way is to improve the quality of model hypothesis. It tries to obtain better models via localized optimization methods, such as iterative model fitting. Another solution may lie in improving RANSAC to use more prior knowledge. For example, Oesau *et al.* [62] presented a two-scale RANSAC framework for line detection from 2-D projected point clouds. This approach makes the line hypotheses at multiple scales, combines two-line proposals into one proposal pairs, and selects pairs to preserve the detailed structure information of intersecting walls. Finally, RANSAC only detects one object each time, necessitating a sequential extraction strategy for handling point clouds with multiple objects. This approach detects the shape with the largest number of supporters, removes its inliers from the input data, and repeats the process until no detectable objects remain. However, this sequential strategy leads to incorrect detection at times because later results depend highly on previous ones [127]. Combining global structure information and detecting multiple primitives simultaneously may offer a solution.

- 2) *Energy-based optimization*: A recent approach to the multimodel extraction problem is based on energy minimization. It first generates many hypotheses for anticipated models and then selects a subset from the hypotheses by minimizing an objective function. A step-by-step example is shown in Fig. 6. Fig. 6(a) shows the 2-D points of a rectangle, and Fig. 6(b) shows the generated hypotheses from sampling two random sample points. This reduces to a problem of selecting the minimum number of hypotheses that cover most of the input data. For instance, Fig. 6(c) shows the four-line models selected after minimizing some predefined objective functions. Yu *et al.* [128] proposed a new multiple model fitting method using global optimization. After generating model hypotheses from sampling small subsets of the input data, the system minimizes the objective function. An objective function

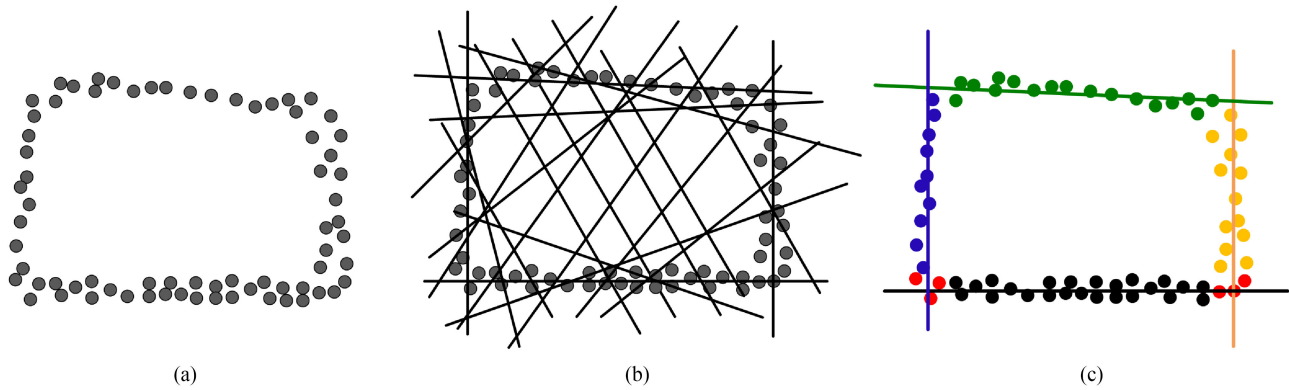


Fig. 6. Line extraction example based on the ideas of hypotheses and selection strategy. (a) 2-D point clouds. (b) Generated hypotheses. (c) Selected line models. Red points are intersections between models.

consists of three terms: model fidelity, inlier similarity, and a regularization parameter. Minimizing this function is a standard quadratic problem, which can be solved by various methods. A more general form of the objective function consists of data and smoothness terms [129]. The data term often quantifies the model fitting errors, and the smoothness term adds constraints (prior knowledge) to neighboring data points. Isack and Boykov [129] proposed a new framework called Propose Expand and Re-estimate Labels for the extraction of multiple geometric models from discrete points. It generates model proposals by a random sampling method and treats the model fitting problem as a labeling process. The data term is the sum of the distances between the data points and the generated models. The method builds a graph of neighboring data points by applying the Delaunay triangulation to the input data. After that, the smoothness term recognizes the prior knowledge that neighboring data points in the graph are likely from the same model. The sum of the data and smoothness terms can be minimized by an α -expansion algorithm [130] with a subsequent re-estimation of the proposed models using the optimized results. The method repeats the “optimize then re-estimate” steps until the energy stops decreasing. Finally, the system selects proposed models with their numbers of inliers above a given threshold as the extracted models. Delong *et al.* [131] optimized the number of labels (potential models) using an extended α -expansion algorithm. The smoothness terms of the Isack–Boykov method in [129] and [131] both incorporate low-level constraints (i.e., constraints on neighboring points) that are said to be insufficient [132]. Integrating high-level prior knowledge into the smoothness term [132], such as topological relationships like parallel or orthogonality between planes in the real world improves these methods. Pham *et al.* [133] proposed another improvement using a large rather than a minimal sampling strategy to generate model hypotheses along with a simulated annealing framework using graph cuts to minimize the objective function. Recently, these frameworks and their variants have

been used in ALS point cloud processing. For example, Yan *et al.* [134] solved the problem of detecting roof planes from ALS point clouds by minimizing a general object function, which considers label costs [131]. In [135], multiple line models were extracted from ALS data via solving an energy minimization formulation, which considers all relevant factors, including model consistency, piecewise continuity, and label compactness.

It is also possible to design objective functions from a data coverage point of view. Magri and Fusiello [136] generated hypotheses using a RANSAC sampling strategy, with the objective function defined as the sum of n binary values, each of which is 1 if the input point is an inlier of some selected models (covered) and 0 otherwise, where n is the number of input data points. Maximizing the objective function means maximizing the number of inliers for the selected models. Xia and Wang [137] proposed a line primitive extraction method for projected lidar point clouds that first uses a randomized HT to generate model guesses. It then considers data coverage, model intersections, and model numbers in the proposed objective function that can be optimized by linear programming. This idea is also useful for other 3-D data. For example, Jiang and Xiao [138] proposed a new framework for detecting cuboids in RGBD images. Their method generates cuboid hypotheses based on image segmentation results with prior knowledge, such as sizes of potential cubes. The objective function maximizes the coverage of data under constraints, such as overlap and cube fitting errors.

Compared with standard RANSAC, energy-based methods extract all of the primitives simultaneously and avoid the drawbacks of the greedy extraction strategy. Furthermore, the energy-based framework makes implementation of prior knowledge easy. These methods often result in globally optimal results. However, the setting of parameters may be difficult for some datasets, and missing data can lead to wrong results when using prior knowledge. Also, a large number of points dramatically reduces the selection efficiency. Thus, these methods are rarely applied to real lidar point clouds, especially for those with

variable density or large volumes from terrestrial lidar systems.

- 3) *Preference analysis methods*: In the RANSAC algorithms, the consensus step verifies each proposed model based on its supporters. This relationship can be inverted: the preference hypotheses for each input point can be examined in turn. Such methods are known as preference analysis, a term originating from [139]. This approach first generates hypotheses using a RANSAC-like sampling method and, then, for each point, calculates residuals to all the hypotheses. Observing that the residuals of hypotheses generated from outliers will be randomly distributed while the residuals of models generated from inliers will be aggregated, the peaks in the residual histogram of each point correspond to potential models. This method can automatically determine the number of potential primitives, but the steps for model identification are cumbersome and easily affected by noise [140]. This method was further developed by Toldo and Fusiello [140], who introduced an $N \times M$ preference matrix for N input data elements and M generated hypotheses. Each column of this matrix records the preferred set of hypotheses for an input data point. The method then groups points with a proposed agglomerative clustering method, called J-linkage, according to the Jaccard distance that measures the overlap degree between two preference sets. This method was further improved by using soft thresholds for the preference function and a new similarity measure called the Tanimoto distance [141]. Also, a bottom-up clustering method groups points into different structures. More recently, the idea of preference analysis for the extraction of multiple structures has been explored. Xiao *et al.* [142] introduced the hypergraph to integrate the information from input data. A vertex in the hypergraph corresponds to an input data point, and the edges represent the model hypotheses. Multiple primitives are detected by partitioning the original hypergraph into subhypergraphs using spectral clustering [143]. These methods have achieved promising results but are seldom used for real lidar point clouds due to efficiency limitations [142]. Furthermore, these results rely on a clustering step that is not always optimal and fail to consider the intersections between multiple structures.

- 4) *Clustering Method*: Clustering analysis is also a way for extracting geometric primitives from lidar point clouds. Feng *et al.* [144] divided point clouds into small regions and then selected the region with the minimum plane fitting errors as the clustering seed. Later, a hierarchical clustering algorithm has been proposed to merge neighboring regions if the fitting error is below a given threshold. However, this method is intended for organized point clouds. Sampath and Shan [145] proposed a fuzzy k -means clustering method with a predefined number of clusters applied to group lidar points. As the parallel and coplanar roof planes may be mistakenly combined during the clustering, the method uses model fitting and connective analysis to fix these errors. Poullis [34] proposed an unsupervised clustering algorithm using 6-D features that include the normal (n_x, n_y, n_z) , height value, local height variance, and local

normal variance. Primitives are propagated by testing whether the neighboring points meet a 6-D probability distribution defined by the 6-D Gaussian function. Once a point is assigned to a primitive, it is accepted as belonging to the primitive and the 6-D Gaussian distribution is recomputed to reflect the mean and covariance in that primitive. The propagation process is repeated until no more neighboring points satisfied the Gaussian model. The above propagation process is executed multiple times if the point set contains multiple primitives. To avoid setting a plethora of thresholds, the tensor-based clustering algorithm was proposed by Poullis [146] for segmenting the tensors into different primitives, such as surfaces, curves, or junctions, respectively. Similarly, Chen *et al.* [72] presented some refinements after the initial clustering to eliminate the errors due to an inappropriate selection of neighborhood size and thresholds. Xu *et al.* [147] proposed a hierarchical clustering method for MLS point clouds using bipartite graph theory, and their framework allows planar segments to be kept after postprocessing, such as rule-based merging.

Yi *et al.* [73] proposed a spectral residual clustering algorithm to extract multiple line primitives from point clouds using the idea of subspace clustering [148]. This method first builds a point-to-point similarity matrix and then utilizes the spectral clustering method [143] to divide input data into small groups (overclustering). These small clusters are then merged according to mutual information theory [149]. Data defects and varying densities may affect this method's results. The merging process is also significant in obtaining good results. Wang *et al.* [150] proposed a scoring-based clustering method for primitive extraction. It first generated model hypotheses using the RANSAC approach and sampling a minimal number of points to estimate the parameters of the potential model. Then, the method combined estimated thresholds and model residuals to score the hypotheses. Hypotheses with high scores were clustered into final models according to mutual information theory [149]. Wang and Xu [68] further improved the method and applied it to planes extracted from lidar point clouds. These improvements include consideration of the spatial distance when scoring hypotheses to reduce the impact of point densities and using a medoid shift clustering algorithm [151]. The methods in this paragraph rely on the robustness of model scoring methods, with the merging step noted to be of critical importance.

Clustering algorithms for primitive extraction are unsupervised learning methods. Different clustering methods have distinct advantages and disadvantages in various tasks. In general, current clustering algorithms applied in point clouds are often density sensitive, and extraction accuracy depends greatly on extra refinement steps, for example, the initial primitives extracted via probabilistic clustering in [34] contain a small number of trivial clusters, which were merged with their adjacent dominant primitives by a metric of Bhattacharya distance. Similarly, in [146], the trial primitives produced by tensor clustering were merged by analyzing primitive's adjacent relationships established on an adjacent graph.

In summary, the most frequently used methods for detecting primitives from lidar point clouds are region growing, RANSAC, and HT. Although these methods can achieve acceptable results

TABLE I
LIST OF REVIEWED PRIMITIVE EXTRACTION METHODS

Methods		Descriptions	Strengths	Limitations
Region Growing [41], [43], [84]–[94], [152]		Merge adjacent points into segments based on feature differences	* Easy to implement * Flexible in adding rules	* Parameter tuning * Locally optimal * Sensitive to noise
Accumulation [94]–[96], [98]–[104], [106]–[109]		Voting in parameter spaces where maxima are used to identify primitives	* Easy to implement * Robust to noise * Robust to missing data	* Voting space construction * Space/time complexity * Locally optimal
Hypothesis and Selection	RANSAC [62], [100], [112]–[120], [122]–[127]	Generate potential models among which the one with most inliers is selected	* Easy to access * Robust to outliers	* Degeneracy * Locally optimal
	Energy-based Optimization [128], [129], [131]–[138]	Generate guesses and select models by optimizing an objective function	* Add prior knowledge * Globally optimal solutions * Flexible in adding rules	* Time consuming * Parameter tuning * Little lidar experience
	Preference Analysis [139]–[142]	Select models based on the preference of each point	* Robust to outliers * A global view of guesses	* Time consuming * Numerous guesses * Little lidar experience
Clustering [34], [68], [72], [73], [143]–[151]		Using existing clustering algorithms	* Many available methods	* Density sensitive * Need further refinements * Often local optimal

in many applications, their robustness to noise, density variation, and missing data still needs improvement. More specifically, for the region-growing algorithm, the degree of noise determines which neighboring points can be selected to participate in clustering, while point density determines what extent of local point points can be aggregated together. For the RANSAC algorithm, different strategies for calculation of inliers have different sensitivities to noise, point density, and missing data. For example, adopting quadratic distance measure is robust to noise, whereas using least median measure for estimating inliers is more sensitive to density and missing data. For the HT algorithm, how to set up an accumulator defined in the parameter space will directly affect the accuracy of extracted primitives. In addition, complex structures in the real world also bring new challenges that make globally optimal solutions difficult to achieve. Emerging methods, such as energy-based frameworks, have already demonstrated their ability to detect multiple structures from synthetic data and 2-D points, but their performance with real lidar point clouds has not been fully examined.

Table I provides a quick reference to relevant shape primitive perceptive techniques along with their strengths and limitations. To sum up, users can select suitable methods in their applications based on five major factors: the degree of implementation difficulty, robustness, global optimum, time cost, and flexibility. Methods such as region growing, RANSAC, HT, and clustering-based ones are easy to be implemented, and lots of implementations in different programming languages have already been released to the public, compared with other methods in the table. The robustness of methods is critical when handling lidar point clouds that are noisy, incomplete, and containing a large percentage of outliers. In these cases, RANSAC and accumulation methods may be preferred. Besides, energy-based optimization methods [128], [131], and [132] that consider noise and outliers may also be applicable. If the global optimal results are needed in one application, users may have to turn to energy-based methods for help. But, it should be noticed that energy-based methods and preference analysis are both time-consuming and, thus, are not recommended when low time cost is required. The last factor is the flexibility of methods. A method with high flexibility means that prior knowledge and

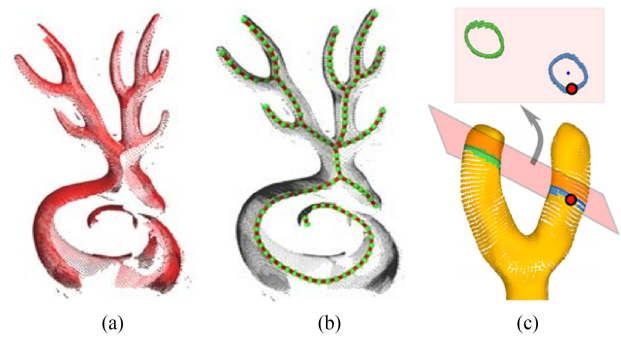


Fig. 7. Demonstration of skeletons and cutting planes. (a) Input points [155]. (b) Skeleton [155]. (c) Cutting plane [106].

various constraints can be easily considered during the primitive extraction process. Currently, the most flexible method group is energy-based optimization methods. However, high flexibility often means more parameters, which need to be tuned carefully for different datasets.

B. Structure Primitives

The structure primitives of interest include skeletons and 2-D/3-D edges. Specifically, 2-D edges occur in conjunction with 2-D outlines in projected lidar data or 2-D images derived from ALS point clouds. 3-D edges occur with all kinds of boundaries detected in 3-D point clouds.

1) *Skeletons*: In the context of lidar processing, a skeleton is “a thin and centered structure which jointly describes the topology and the geometry of the shape” [153]. For instance, the image in Fig. 7(a) has the skeleton shown in Fig. 7(b). Skeletons are useful in animations and shape analysis [153] with most existing methods for extracting 3-D skeletons designed for meshes. For example, Au *et al.* [154] proposed a mesh contraction method for skeleton detection. The method first processes the watertight meshes with Laplacian smoothing and then obtains a 1-D skeleton with connectivity surgery methods using smoothed meshes. A skeleton-mesh mapping method refines the final skeleton. However, these methods cannot be directly applied to raw lidar point clouds because constructing meshes from point clouds is complicated and may be incorrect. The

procedure is also time-consuming. Similar to other topics, we examine only skeleton extraction methods applicable to discrete point clouds. Tagliasacchi *et al.* had a review of skeleton methods for other data sources [153].

Sharf *et al.* [156] presented a curve skeleton extraction method from point sets. Using a deformable model to describe the volumetric shape of the input data, this method tracks and merges the fronts' centers to obtain curve skeletons. This method may fail if the data have significant omissions. Cao *et al.* [157] generalized the idea of contraction in [154] for processing point clouds. This generalized form projects points onto a local tangent plane, organizes them by the Delaunay triangulation, and performs iterative Laplacian smoothing. The topological thinning method produces the final curved skeleton. This approach tolerates missing data but requires clear dense input points [155]. Tagliasacchi *et al.* [75] presented a curve skeleton extraction method with a claimed tolerance to missing data. This method treats the skeleton as a set of generalized rotational symmetry axes (ROSA) of a shape and introduces cutting planes [see Fig. 7(c)] to obtain the slices of input points that generate the ROSA points. The method then joins and smooths the ROSA points to make skeletons. ROSA's stability makes the method robust to incomplete structures, but its input data should have a low noise level, and pointwise normal should be accurate. Huang *et al.* [155] proposed a skeleton construction method that is robust to noise by using the L_1 -median of local points to represent the optimal center. However, some researchers find that the method is sensitive to occlusions [78].

In some specific applications, such as tree reconstruction, prior knowledge is useful in approximation of the skeleton. For example, Bucksch and Lindenbergh [158] presented a skeleton extraction method using an octree-based graph and removal of nonskeleton edges according to predefined rules. Livny *et al.* [76] constructed a graph of points and extract a subgraph as the stem skeleton. Similarly, Wang *et al.* [77] started from a graph of input points and transformed the problem of skeleton extraction into one of finding a minimum spanning tree (MST) in the original graph. In these studies, the roots of the trees are important in graph analysis. For example, the shortest path in the graph should start from root points. Based on the prior knowledge of tree structures, Wang *et al.* [77] and Mei *et al.* [78] also presented point cloud optimization algorithms to repair missing structures. However, this prior knowledge is not applicable to other objects.

In summary, although there are multiple methods for 3-D skeleton extraction from point clouds, most of them have point clouds of one single object as input and are not applicable to raw lidar point clouds of real scenes. Also, the contraction methods for constructing skeletons require significant execution time and can achieve good results only in clean complete point clouds. By using prior knowledge, the skeletons of some specific objects such as trees may be detected easily. Efficient and robust skeleton extraction methods for raw lidar point clouds should be developed in the future.

2) *2-D Outlines*: 2-D outlines are boundaries extracted from projected point clouds or 2-D patches in ALS processing [159]. Conversion of 3-D point clouds to 2-D organized digital surface models (DSMs) or normalized DSM (nDSM) images simply

requires projecting points onto 2-D grids. Each grid corresponds to an image pixel with the pixel value equal to the maximum height of the points in that grid. The conversion to 2-D enables the use of traditional image processing algorithms.

Yu *et al.* [160] converted nDSMs derived from ALS lidar point clouds into binary images, labeling pixels with values higher than a given threshold as foreground and all others as background. The method then smooths the binary nDSM with morphological operations like closing and erosion and identifies pixels adjacent to background pixels as building boundaries. The algorithm then connects the boundaries using an extended boundary algorithm [161]. Grigillo and Kanjir [162] also applied morphological operations on nDSM images to remove spurious objects and extract 2-D building boundaries by detecting straight line segments using an HT. Lafarge *et al.* [47] used a marked point process technique to extract rectangular outlines from a DEM derived from ALS clouds. The technique associated each input data point with a shape proposal, e.g., a rectangle, and defined its objective function with three parts. The first part measures the coherence between the boundary points and rectangles. The second term qualifies the relationship between neighboring rectangles. The last term penalizes the intersection of parallel models. The proposed energy function can be optimized by a reversible jump Monte Carlo Markov chain sampler [163] and a simulated annealing framework [164]. This method adapts to 3-D situations by replacing the 2-D rectangle models with cuboids [165].

The α -shape [166] algorithm is a widely used tool to extract boundaries from 2-D points or pixels. The algorithm works by first selecting two points to generate circles with the given radius α . If there is no point that falls in the generated circle, the two points will be labeled as boundary points. The first two steps repeat until all points have been visited. The processing of ALS roof points makes frequent use of this outline extraction method [167], [168].

Other 2-D image processing methods can detect 2-D boundaries for planar patches. For example, Poullis [34] extracted boundaries of roof patches in 2-D images with a contour detection algorithm. Thanks to advances in 2-D image processing, most 2-D outline extraction problems with lidar data are solvable by converting input data to 2-D images. However, surrounding occlusions (e.g., vegetation) easily affect DSM generation and result in lower accuracy. In addition, 2-D outlines of objects can also be delineated from ground-based lidar point clouds. For example, Yang *et al.* [66] extracted 2-D outlines from MLS point clouds by projecting detected 3-D planes horizontally. Xia and Wang [137] first projected 3-D points horizontally, and line segments are detected from projected points, which are then connected into building outlines. In the recent years, it has been increasingly popular to extract 3-D edges or boundaries directly from lidar point clouds, especially from MLS and TLS point clouds with much higher point density and more details than ALS data.

3) *3-D Edges*: To the best of our knowledge, there is no universally accepted definition of 3-D edges or contours in the context of lidar point clouds. Hackel *et al.* [169] understood contours to be linear structures with a large number of

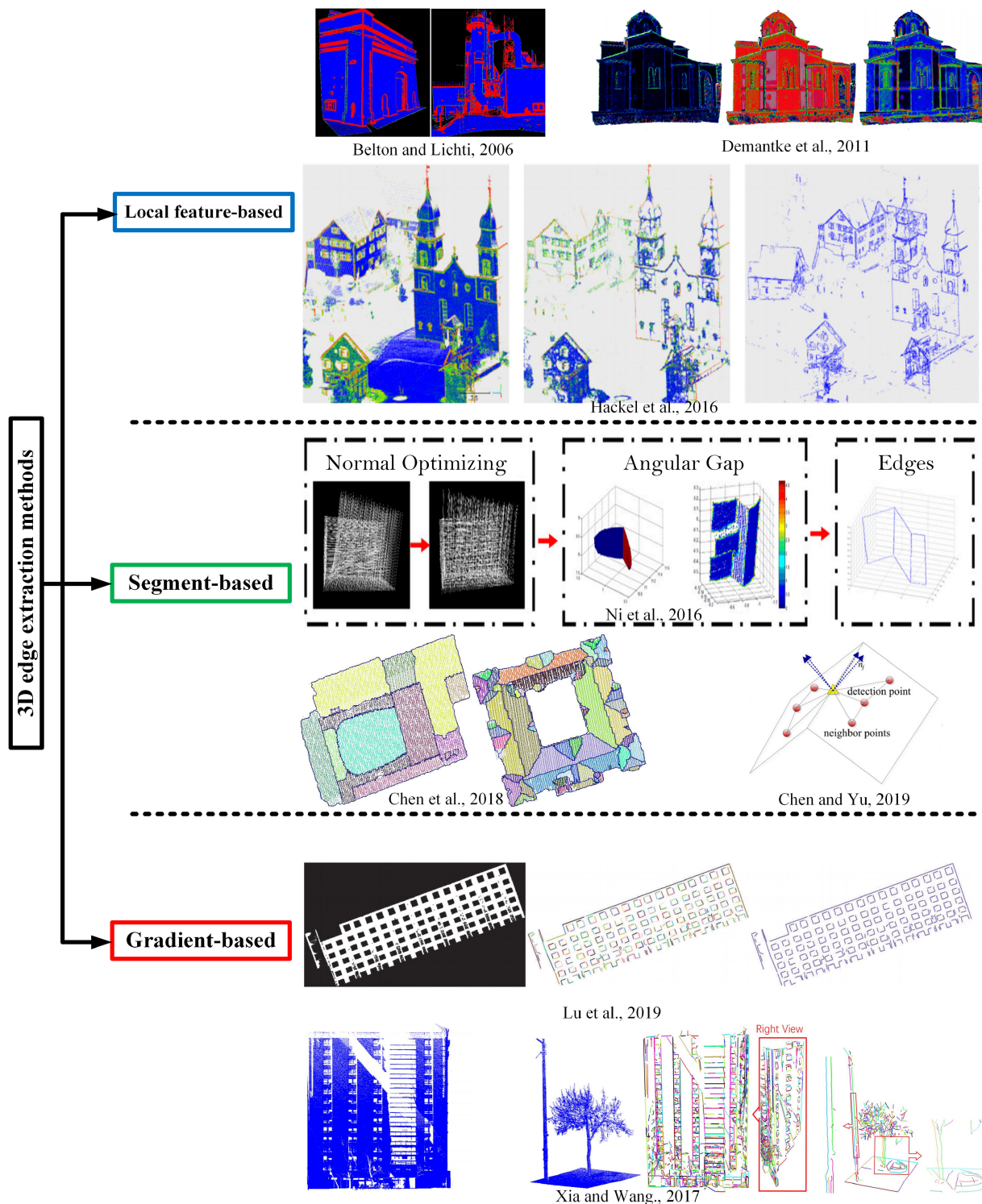


Fig. 8. Overview of 3-D edge extraction methods.

discontinuous among adjacent orientations or surface normals. Ni *et al.* [170] classified 3-D edges into boundary elements (e.g., surface outlines) and fold edges (e.g., surface intersection lines). We refer to all boundaries and contours in lidar point clouds as edges for clarity. Fig. 8 shows a summary of the three methods for edge extraction from lidar point clouds: local feature-based, segment-based, and gradient-based.

1) *Local feature-based methods:* Local geometric properties, such as flatness, are good indicators for identifying edge regions in point clouds. Based on the PCA of neighboring points, these methods estimate local curvature measures using eigenvalues and recognize data points with high curvature values as edges [171]. Demarsin *et al.* [172] used normal differences to identify edge points. In fact,

various eigenvalue-based features are useful for detecting rough regions in point clouds. Hackel *et al.* [173] proposed a supervised classification framework to extract contours from terrestrial lidar point clouds. Their method used features derived from PCA for classification, although the method may fail in areas lacking training data. Also, many factors such as data quality and feature scales all affect the values of these features [174]. Dittrich *et al.* [175] presented a comprehensive study on the accuracy and robustness of eigenvalue-based features in 3-D point clouds. Thus, these methods for edge detection are often applied to high-quality point clouds of small scenes.

- 2) *Segment-based methods*: Fold edges in 3-D point clouds are the intersections between segments or surfaces. In processing roof point clouds from ALS, Chen *et al.* [72] proposed a boundary extraction method using Voronoi diagrams. Their method clusters roof points into planar segments and then constructs a Voronoi graph and Delaunay triangles from points with labels to indicate segment membership. Finally, the method extracts the edges of triangles belonging to segment boundaries by analyzing the properties of adjacent labels. Borges *et al.* [176] also segmented points into planes and then calculated the intersection lines between adjacent planes.

Ni *et al.* [170] and Kang *et al.* [177] projected 3-D points onto a plane and then identified edge points based on the distribution of the projected points such as the pattern of angular gaps [177]. A region-growing method can connect edge points using point orientations estimated by RANSAC to constrain the growth. Chen and Yu [178] also used the angular properties of points on a projected plane to detect boundary points. In addition, the k -means clustering method is applied to identify fold edges. Lin *et al.* [41] obtained more detailed edges, instead of large planar segments, by generating oversegmented facets and then employed the α -shape algorithm to retrieve boundaries of each facet. After that, the algorithm eliminated boundaries surrounded by other coplanar facets and recognized the remaining boundaries as edges. Finally, the algorithm groups all the edge candidates using a constrained region-growing method. One drawback of this method is the difficulty in obtaining planar segments from nonplanar surfaces.

These methods all achieve promising results in small scenes or in specific applications (e.g., roof boundaries), but they rely too much on results from the still-evolving field of point cloud segmentation. Furthermore, retrieval of adjacent planes and estimation of the intersections between small primitives are difficult and inaccurate in complex scenes. Long execution time also remains a problem.

- 3) *Gradient-based methods*: Contour and edge detection within 2-D images are well-developed fields [179]. Classical methods, such as the Canny detector [180], have proven their ability to extract contours using the basic technique of gradient (difference between features) analysis, and others have extended existing 2-D solutions to point clouds. Lin *et al.* [181] created 2-D images by

projecting 3-D points onto planes from multiple views and applying a fast line detection method called line segment detector [182] to extract lines from the multiple perspectives. Projecting the 2-D line segments into a 3-D space enables extraction of the 3-D edges. Similarly, Lu *et al.* [183] proposed a fast line segment extraction method for lidar point clouds through detecting 2-D lines via contour finding method in projected planes. Xu *et al.* [25] extended the 2-D Sobel detector to 3-D and applied it to voxelized MLS point clouds. Their method selects points with significant response to the detector as edge candidates and filters non-edge points using an edge-linking algorithm. These methods often lose 3-D information and accuracy. Xia and Wang [184] proposed a gradient method for direct processing of 3-D point clouds and redefined gradient as the difference between features of neighboring points. Their method extends the Harris detector [185] to 3-D and filters edge candidates using the ratio between eigenvalues of a symmetric matrix instead of calculating all eigenvalues. It links edges with a graph snapping algorithm. However, the performance of this method depends on the accuracy of local features and requires careful threshold selection tailored to different datasets.

Among the three types of edge extraction methods, the local feature-based approach is usually the fastest, but its performance depends greatly on the accuracy of feature calculation, which is easily affected by factors such as noise. The segment intersection-based method is the slowest of the three, performing well only with good segmentation results. The gradient-based method appears to balance efficiency with performance but always requires voxelization or projection. The gradient definition in discrete point clouds is also unclear. Last but not least, edge linking methods that aim at grouping edge candidates into primitive-level clusters need further investigations, as they are not robust against noisy edge candidates and data missing. Overall, the existing edge extraction methods for 3-D point clouds still need further improvement.

III. PRIMITIVE REGULARIZATION

The results of primitive extraction are groups of points, each of which corresponds to a detected primitive. Some primitive extraction methods (e.g., RANSAC) will estimate model parameters for point groups during the extraction process, while many other methods (e.g., clustering-based methods) are not. To further obtain primitive parameters, least-squares fitting algorithms can be used to estimate geometric model parameters given groups of points [186]. PCA using model inliers can also fit planar primitives. For example, the normal of the potential plane equals the eigenvector corresponding to the smallest eigenvalue [187]. However, noise and outliers easily affect PCA-based results. Some researchers propose other plane fitting methods to solve this problem, such as a robust distance-based PCA algorithm that excludes outliers before plane fitting [188].

Primitive models fitted by inliers are data-driven models, which means that the models are only faithful to the input data. However, outliers, density variation, and occlusions can bias

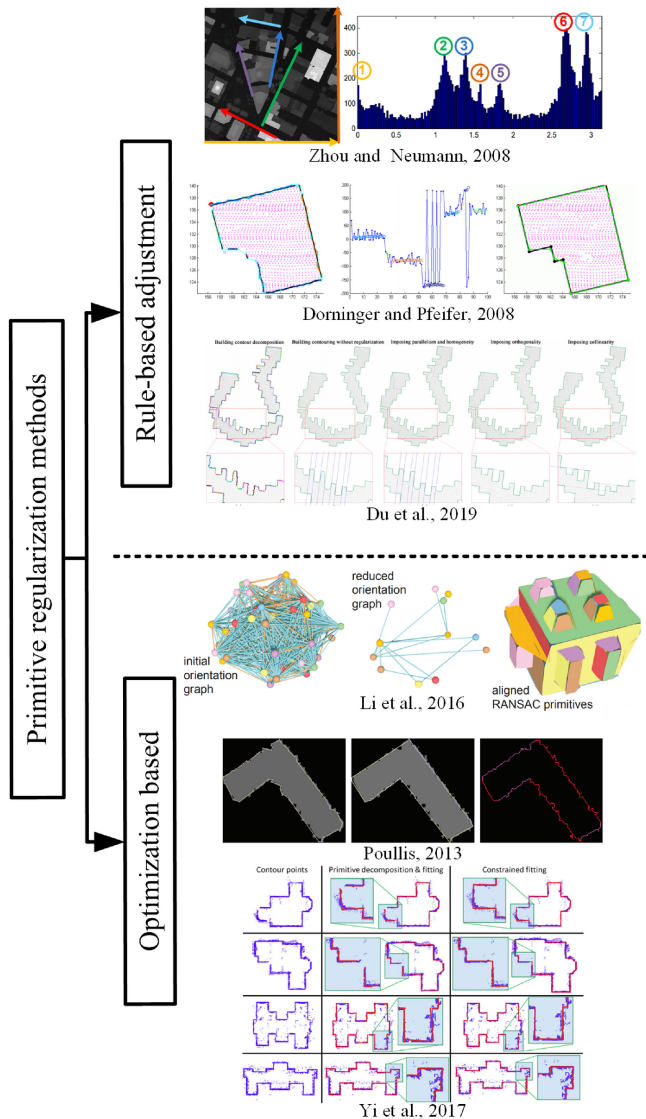


Fig. 9. Overview of primitive regularization methods.

results from such models. Primitive regularization means adjusting the parameters of the extracted primitives based on prior knowledge of targets. The use of regularization improves the accuracy of the resulting models and preserves the topological relationships (e.g., parallelism, orthogonality, and coplanarity) among primitives. Occlusions and density changes in the data from terrestrial lidar make regularization both difficult and necessary. For example, incomplete data often bias the initial fitted models. While there are numerous methods for primitive regularization, most of them exist for specific applications, such as reconstruction. Fig. 9 shows the two broad categories of regularization methods: *rule-based* and *optimization-based* adjustments.

1) *Rule-based adjustment*: Parameters of fitted primitive models are often adjusted to satisfy some specific configurations one by one in rule-based adjustment methods. Sampath and Shan [167] proposed a roof boundary regularization method for ALS point clouds that uses long segments as bases and adjusts the perpendicular and parallel relationships between boundary

lines hierarchically. Similarly, Dorninger and Pfeifer [168] adjusted topological relationships between boundaries by fixing the directions of long line segments and changing the parameters of the short ones. Zhang *et al.* [189] assumed that buildings contain only two dominant orientations and compulsively align detected building contours with the dominant orientations. Zhou and Neumann [71] proposed a dominant orientation detection method that first estimates the principal directions of local boundaries and then accumulates a histogram of directions. The peaks in the histogram identify the dominant directions, and the method fixes the boundary directions to the closest peak in the histogram. Oesau *et al.* [94] proposed a greedy regularization method using a graph structure. After detecting parallel configurations between primitives by running a mean-shift algorithm on the sphere of normals, the method detects orthogonality by comparing the directions between parallel clusters. It selects the cluster with most points as the basis and removes it. The process is repeated for all of the parallel clusters. It applies a clustering method with a distance metric to group adjacent parallel structures and determine coplanarity. A hierarchical regularization concept is extensively used to finalize global regularization for linear primitives [135] and planar primitives [190]. The hierarchical framework can avoid conflict of different types of regularized operations and costly pairwise comparisons of different relationships of geometric primitives.

2) *Optimization-based adjustment*: Optimization-based methods approach regularization by integrating prior knowledge into objective functions that are minimized to determine the optimal primitive configurations. The regularization can be performed during the primitive extraction step. For example, in [132], parallel and orthogonal constraints are added to the energy function to obtain regularized primitives. However, most research regularizes primitives after the extraction process. The objective function developed by Lafarge *et al.* [191] favors perpendicular and parallel configurations of primitives through the addition of an energy term that measures the differences in the direction between two primitives. Similarly, Holzmann *et al.* [69] incorporated the Manhattan world assumption for modeling background clutter into a regularization term that penalizes planes that violate this assumption. Li *et al.* [192] proposed GlobFit to adjust the configuration of shape primitives globally. This method uses an efficient RANSAC algorithm [125] to generate initial primitives from the input point cloud and then considers orientation (e.g., parallelism and orthogonality) and placement (e.g., coplanar and coaxial) relationships between primitives with the help of a relationship graph. This turns the regularization step into a constrained nonlinear optimization problem.

Monszpart *et al.* [67] detected initial primitives by region growing, which generates many candidates. The regularization step identifies a subset of regularized primitives among all candidates. This approach favors planes with close orientations and penalizes irregular primitive configurations. Quadratic programming was used to optimize the final objective function. Poullis [34] treated the regularization of 2-D edges as a multilabeling process. This approach first detects the principal directions of roof outlines using a Gaussian mixture model

and then initializes labels corresponding to these directions. The method uses an objective function that incorporates data, smoothness, and label terms. After optimization, line segments with the same direction share the same label, and the system also discards redundant labels. Recently, Yi *et al.* [73] have presented a contour regularization method that finds pairs of nearly parallel or nearly perpendicular line segments. The method includes a model for line segment refitting with topological constraints that is optimized by the Levenberg–Marquardt algorithm. Primitive regularization concepts also apply to the optimization of normals and orientations in lidar point clouds [193], [194]. To make full use of hard constraints, Zhou and Neumann [195] proposed a framework for rooftop reconstruction by imposing multiple global regularities (e.g., roof–roof, roof–boundary, and boundary–boundary regularities). The principle of this method is that the global regularity reveals the topological relations between the primitives and also provides information on the architectural design of buildings.

In summary, primitive regularization implements and incorporates prior knowledge of human-made objects. The rule-based methods are efficient and straightforward in most applications, but the greedy search procedure does not guarantee globally optimal solutions. The performance of these methods may also be affected by the predefined rules. Optimization-based methods are flexible, allowing for incorporating prior knowledge for model detection, and often provide globally optimal results. However, the methods are complicated and time-consuming, and it is challenging to design an appropriate objective function. Overall, globally optimal results are preferable, and optimization-based methods offer more potential for the future if their time cost is reduced.

IV. DISCUSSION

In the previous sections, we reviewed methods for primitive extraction and regularization. In this section, the relationship between pointwise features and geometric primitives in lidar point clouds is explored. Moreover, the differences in primitive processing and applications between lidar point clouds and point clouds acquired by other sensors are discussed.

A. Pointwise Features and Geometric Primitives

In point clouds processing, pointwise features are often used as the basis for many applications such as classification [196] and object recognition [60]. It is necessary to describe the relationship between pointwise features and geometric primitives in lidar point clouds processing. In this review, the relationship is illustrated from two aspects, i.e., the hand-crafted pointwise features and the deep-learning-based features.

Many widely used hand-crafted features in point clouds are based on eigenvalues. Jutzi and Gross [197] proposed an object structure analysis method based on eigenvalues of moments defined within a spherical neighborhood of each point. Local structures in the point clouds, such as planes and intersections of planes, can be identified by analyzing the magnitude of three eigenvalues. This eigenvalue-based method cannot extract geometric primitives directly, but it may be useful for finding regions

of potential geometric primitives. The eigenvalue-based method is sensitive to the size of the neighborhood and can be improved by using an adaptive neighborhood selection method [174]. A list of eigenvalue-based features and feature relevance analysis can be found in [196]. On the other hand, geometric primitives in point clouds can also be used as a reference for feature calculation. Guo *et al.* [198] proposed a robust local reference frame based on local surfaces for calculating pointwise features. Mallet *et al.* [199] used local-plane-based features to improve lidar point classification accuracy. These pointwise features, such as angle deviation and distance between points and local planes, are calculated based on planes formed by neighboring points. In [200], several pointwise features were calculated based on local surfaces. For example, the plane index, which is the standard deviation of the distance between points and a local plane, and the surface index (averaged plane index) were designed. Similar to the spin image descriptor [201], Guo *et al.* [200] also calculated the pointwise spin image by projecting local points onto a predefined plane. In [202], the projected plane of the spin image is fitted by neighboring points. In short, hand-crafted pointwise features can reveal local properties of geometric primitives, which can also be regarded as the basis for feature design and calculation.

Deep learning is a supervised learning method and has become popular in point clouds processing in the past three years. Geometric primitive detection based on deep learning frameworks has also been proposed for images and 3-D models. A deep neural network named PlaneNet, which can predict the depth and plane parameters from RGB images, was proposed in [203]. In this framework, a number of planar surfaces are extracted in a network branch that uses global averaged pooling features as input, and the plane parameters can be predicted when camera's intrinsic parameters are known. Zou *et al.* [204] presented a 3-D primitive recurrent neural network for extracting primitives from RGBD images. The key idea of this framework is applying a long short-term memory unit, which predicts primitives in sequences. The advantage is that this network unit preserves symmetry and long-range structural similarity of human-made objects. In [205], four geometric primitives (plane, sphere, cylinder, and cone) were detected in RGBD images using their network. In the experiments, the positions and normal vectors of pixels were fed to the network to obtain probability maps of primitives. Then, the potential primitives were verified using RANSAC model fitting. Tulsiani *et al.* [206] proposed a CNN framework for extracting volumetric primitives (cuboid) from 3-D object models. In this model, the loss function consisted of two terms, i.e., the coverage loss and consistency loss. The first term is large if the primitives do not cover the object, and the second term is a penalty when the detected primitives are outside the objects.

The aforementioned deep learning frameworks cannot process discrete point clouds. To address this problem, Qi *et al.* [207] proposed a classical neural network to extract high-dimensional pointwise features from unorganized and discrete point sets. In this framework, a multilayer perceptron was used to learn pointwise features from the training data, and the global features were obtained by max pooling. The PointNet++ [208] was developed by improving the PointNet using multiple-scale

sampling operations. Li *et al.* [209] proposed a supervised primitive fitting network for extracting geometric primitives from 3-D point clouds. This network is based on PointNet++, which outputs three pointwise features in their architecture: the point-to-primitive relationship, normal vectors, and the associated primitive types. These learned features are combined to detect various primitives in point clouds. Four primitive types (plane, sphere, cylinder, and cone) were used in the study. The authors established a new loss function consisting of five terms, i.e., segmentation loss, normal angle loss, primitive-type loss, primitive fitting loss, and primitive axis angle loss. The first term is small if the point sets are well segmented. The second term is a penalty if there are differences between the predicted normals and the ground truth. The third term penalizes an inaccurate relationship between the points and the associated primitive types. The objective of their method is to obtain a low model fitting error (the fourth term) and to minimize the axis differences (the last term) between the predicted model and the ground truth. This article offers a practical way to obtain geometric primitives based on pointwise features learned from neural networks. Geometric primitives may also improve the performance of deep learning-based applications in point clouds. For example, Landrieu and Simonovsky [210] first grouped point clouds into simple shape primitives to construct a so-called superpoint graph, which is useful for the retrieval of contextual information using graph convolution.

Only a few deep learning frameworks have been proposed for the extraction of 3-D shape primitives, and there are almost no published papers on structure primitive extraction from discrete point clouds based on deep learning. However, some deep-learning methods exist for the extraction of structure primitives from images. For example, Yu *et al.* [211] presented a semantic edge extraction framework for images, but the depth information was not considered in their framework. In the area of skeleton extraction, Atienza [212] applied the pyramid U-Net to extract 2-D skeletons, and in their method, point sets are transformed into 2-D images. The authors tested the method in the SkelNetOn dataset [213], which contains labeled skeletons of 2-D point sets.

To sum up, a major problem in training primitive detection networks is the need for sufficient large-scale ground truth data of various primitives. Ground truthing using manual labeling is time-consuming and labor-intensive. Ground truth can also be generated using automatized methods [203], simulations [205], and CAD models [209]. However, this type of ground truth data may contain numerous errors or be biased toward real scenes. Notably, there exist no datasets of 3-D edges and skeletons, which has become a significant obstacle in developing deep learning frameworks for primitive extraction from lidar point clouds.

B. Primitives in Other Point Clouds

In general, the primitive extraction and regularization methods reviewed in this article can be used not only for lidar point clouds, but also for point clouds generated from other sources such as mesh models, RGBD depth data, and multiview stereo images. There are three key differences between lidar point

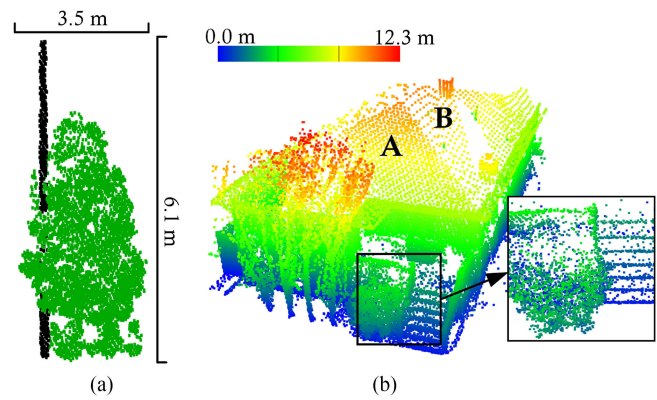


Fig. 10. Penetration and density variances in lidar point clouds. (a) Cylinder pole (black) is surrounded by vegetation (green). (b) Walls (planes) close to vegetation, and roof planes with a diverse point density.

clouds and other point clouds, namely, precision and accuracy, homogeneity, and application scenarios. Therefore, different challenges are associated with the different types of point clouds and affect the choice of the method.

1) *Precision and Accuracy*: The precision and accuracy of lidar point clouds are often higher than those point clouds generated by other sensors. These two factors need to be considered when choosing algorithms. Precision refers to the consistency of repeated measurements. For example, the thickness of a wall will be large if the wall points have low precision. In this case, primitive extraction methods (e.g., region growing) relying on local features (e.g., normals) should consider the feature reliability. Accuracy refers to the closeness of point clouds to the ground truth. For example, according to [214], point clouds generated by structure from motion (SfM) are often more distorted than ALS point clouds. For example, the edges in SfM-derived point clouds are not as sharp as those in lidar point clouds [214]. Thus, when using geometric primitives as constraints for object detection or segmentation, primitives extracted from lidar point clouds are often more reliable than primitives detected from other point clouds. It should be noted that the accuracy of SfM point clouds is not always lower than that of lidar point clouds. In some cases, the accuracy depends on the application and the environment [215].

2) *Homogeneity of Point Clouds*: Since lidar can penetrate vegetation and windows, it is more challenging to separate spatially connected objects in lidar point clouds than in other point clouds. For example, as shown in Fig. 10(a), a cylinder pole is surrounded by vegetation, and the wall planes are spatially connected to the neighboring trees. It is preferable for the extraction of primitives (cylinder and planes) to choose methods that are robust against noise. Also, prior knowledge, such as the perpendicular and parallel relationships between wall planes, can be used as constraints for primitive extraction and regularization [191]. Apart from the property of penetration, the point density of lidar point clouds, especially those collected by ground-based lidar, is much varied than other point clouds due to occlusion and the scanning mode. For example, as shown in Fig. 10(b), although the roof planes A and B are spatially

connected, the point density of the roof plane A is much higher than that of roof plane B. Therefore, methods for the extraction of geometric primitives should be insensitive to point density. That is, clustering-based methods that strongly depend on local features (e.g., normal, density) are not suitable in this case.

3) *Application Scenarios*: In many lidar-based applications such as urban reconstruction, the data volume of point clouds is always large (e.g., city-scale or street-block level). In contrast, point clouds obtained by RGBD cameras or generated from mesh models have small data volumes (e.g., room level). Given the large data volume, the preferred primitive extraction methods are expected to be efficient. This is why the energy-based optimization methods are not widely used for processing large-scale lidar point clouds. In addition, lidar point clouds are often used in surveying and mapping, where high precision and accuracy are required in the final products. For example, geometric primitives such as lines derived from edges or planes are vital elements that bridge the gap between discrete points and vectorized maps. Thus, primitive detection and regularization in lidar point clouds often require higher fidelity than in other applications such as visualization and model abstraction [216].

V. CONCLUSION AND FUTURE WORKS

In this review, we focused on geometric primitives as core elements in lidar point clouds and examined point cloud processing from a geometric perspective. Geometric primitives fall into two classes: shapes (e.g., lines, planes, and cylinders) and structures (e.g., skeletons and edges). We have discussed the extraction and regularization of geometric primitives in lidar processing in detail. We summarize the existing challenges and provide suggestions for future studies.

- 1) *Data inconsistency*: Raw point clouds are often noisy, have uneven point distribution, and are incomplete, especially if data are acquired from terrestrial lidar platforms, such as TLS and MLS. These problems often result in low performance of commonly used primitive extraction methods, such as region growing and RANSAC. Point cloud consolidation [194] offers a promising approach to solve the data quality problem, but the method is time-consuming and not suitable for large-scale scenes at present. It is necessary to develop methods that address data inconsistencies as well as algorithms that are tolerant of data inconsistency.
- 2) *Local versus global solutions*: In the context of primitive extraction, locally optimal solutions are always achieved. Thus, regularization is often used during postprocessing to obtain globally optimal results [192]. These postprocessing methods always require prior knowledge of object structures, which may be challenging to define for large and complex scenes. Current extraction and regularization methods also suffer from these weakness when dealing with large data volumes. Therefore, efficient and globally optimal methods for extracting multiple primitives from data with complex structures are still needed. For example, Dong *et al.* [217] improved the classical region-growing approach by introducing a global optimization method. Most existing methods are tailored to detect one specific

primitive type from the input data. The optimal extraction of multiple types of primitives requires further study.

- 3) *The potential of structure primitives*: Many applications use geometric primitives as discussed in the first section. Compared with shape primitives, the potential of structure primitives is relatively unexplored. For example, segmenting point clouds into regions using contour information has received little attention, while many contour-based segmentation methods [179] have been developed for 2-D images. In the future, we look forward to more attention being given to edge extraction and its application in 3-D point clouds, such as data abstraction and edge-based segmentation.
- 4) *Benchmarks of geometric primitives*: Currently, there are no benchmark datasets for geometric primitives in lidar point clouds. Furthermore, except for widely used methods such as region growing and RANSAC, many implementations of the reviewed methods are not publicly available. Therefore, the guideline of method selection provided in this review is primarily based on analysis and experience. In the future, existing methods for geometric primitive extraction and regularization should be benchmarked to enable quantitative comparisons. The main objective of developing benchmarks is helping researchers to select suitable primitive extraction and regularization methods for their applications. Besides, point cloud datasets of geometric primitives are also critical for deep-learning-based studies [209].
- 5) *Primitive inference in deep learning*: Deep learning based on the CNN has become popular in recent years and has achieved satisfactory results in different areas, such as image contour extraction [218]. However, key technologies, such as convolution operations, are not suitable for discrete, disorganized 3-D point clouds, although it is possible to take advantage of regular 3-D grids [219]. Some pioneering researchers have designed deep learning frameworks that can be used for discrete points directly [207]. Recently, geometric primitive inference from point clouds has attracted increasing attention [204]. The future looks promising for primitive extraction and applications using deep learning frameworks.

ACKNOWLEDGMENT

S. Xia would like to thank Prof. D. Lichti and Prof. M. Shahbazi at the University of Calgary for their helpful suggestions.

REFERENCES

- [1] X. Wang, X. Cheng, P. Gong, H. Huang, Z. Li, and X. Li, "Earth science applications of ICESat/GLAS: A review," *Int. J. Remote Sens.*, vol. 32, no. 23, pp. 8837–8864, 2011.
- [2] A. Serna and B. Marcotegui, "Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning," *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 243–255, 2014.
- [3] B. Yang, L. Fang, and J. Li, "Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 79, pp. 80–93, 2013.

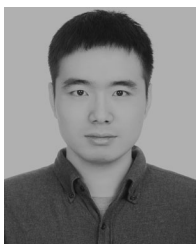
- [4] H. Guan, J. Li, Y. Yu, C. Wang, M. Chapman, and B. Yang, "Using mobile laser scanning data for automated extraction of road markings," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 93–107, 2014.
- [5] M. Soillán, B. Riveiro, J. Martínez-Sánchez, and P. Arias, "Segmentation and classification of road markings using MLS data," *ISPRS J. Photogramm. Remote Sens.*, vol. 123, pp. 94–103, 2017.
- [6] P. Kumar, C. P. McElhinney, P. Lewis, and T. McCarthy, "An automated algorithm for extracting road edges from terrestrial mobile LIDAR data," *ISPRS J. Photogramm. Remote Sens.*, vol. 85, pp. 44–55, 2013.
- [7] G. Vosselman and L. Zhou, "Detection of curbstones in airborne laser scanning data," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 38, no. Part 3/W8, pp. 111–116, 2009.
- [8] J. Heinzl and M. Huber, "Detecting tree stems from volumetric TLS data in forest environments with rich understory," *Remote Sens.*, vol. 9, no. 1, 2016, Art. no. 9.
- [9] L. Zhong, L. Cheng, H. Xu, Y. Wu, Y. Chen, and M. Li, "Segmentation of individual trees from TLS and MLS data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 774–787, Feb. 2017.
- [10] Y. Yu, J. Li, H. Guan, C. Wang, and J. Yu, "Semiautomated extraction of street light poles from mobile LIDAR point-clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1374–1386, Mar. 2015.
- [11] L. Yan, Z. Li, H. Liu, J. Tan, S. Zhao, and C. Chen, "Detection and classification of pole-like road objects from mobile LIDAR data in motorway environment," *Opt. Laser Technol.*, vol. 97, pp. 272–283, 2017.
- [12] W. Xiao, B. Vallet, K. Schindler, and N. Paparoditis, "Street-side vehicle detection, classification and change detection using mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 166–178, 2016.
- [13] J. Eum, M. Bae, J. Jeon, H. Lee, S. Oh, and M. Lee, "Vehicle detection from airborne LIDAR point clouds based on a decision tree algorithm with horizontal and vertical features," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 409–418, 2017.
- [14] L. Nan, A. Sharf, H. Zhang, D. Cohen-Or, and B. Chen, "Smartboxes for interactive urban reconstruction," *ACM Trans. Graph.*, vol. 29, no. 4, 2010, Art. no. 93.
- [15] H. Lin *et al.*, "Semantic decomposition and reconstruction of residential scenes from LIDAR data," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 66.
- [16] L. Ma, Y. Li, J. Li, C. Wang, R. Wang, and M. Chapman, "Mobile laser scanned point-clouds for road object detection and extraction: A review," *Remote Sens.*, vol. 10, no. 10, 2018, Art. no. 1531.
- [17] S. Nie, C. Wang, H. Zeng, X. Xi, and S. Xia, "A revised terrain correction method for forest canopy height estimation using ICESat/GLAS data," *ISPRS J. Photogramm. Remote Sens.*, vol. 108, pp. 183–190, 2015.
- [18] W. Wagner, A. Ullrich, V. Ducic, T. Melzer, and N. Studnicka, "Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner," *ISPRS J. Photogramm. Remote Sens.*, vol. 60, no. 2, pp. 100–112, 2006.
- [19] W. Wagner, "Radiometric calibration of small-footprint full-waveform airborne laser scanner measurements: Basic physical concepts," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 6, pp. 505–513, 2010.
- [20] W. Fang, X. Huang, F. Zhang, and D. Li, "Intensity correction of terrestrial laser scanning data by estimating laser transmission function," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 942–951, Feb. 2015.
- [21] C. Mallet, F. Lafarge, M. Roux, U. Soergel, F. Bretar, and C. Heipke, "A marked point process for modeling LIDAR waveforms," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3204–3221, Dec. 2010.
- [22] X. Yang *et al.*, "Three-dimensional forest reconstruction and structural parameter retrievals using a terrestrial full-waveform lidar instrument (echidna)," *Remote Sens. Environ.*, vol. 135, pp. 36–51, 2013.
- [23] D. F. Fouhey, A. Gupta, and M. Hebert, "Data-driven 3D primitives for single image understanding," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3392–3399.
- [24] A. Berner, J. Li, D. Holz, J. Stückler, S. Behnke, and R. Klein, "Combining contour and shape primitives for object detection and pose estimation of prefabricated parts," in *Proc. IEEE Int. Conf. Image Process.*, 2013, pp. 3326–3330.
- [25] S. Xu, R. Wang, and H. Zheng, "Road curb extraction from mobile LIDAR point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 996–1009, Feb. 2017.
- [26] D. Zai *et al.*, "3-D road boundary extraction from mobile laser scanning data via supervoxels and graph cuts," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 802–813, Mar. 2018.
- [27] L. Cheng, L. Tong, Y. Wang, and M. Li, "Extraction of urban power lines from vehicle-borne LIDAR data," *Remote Sens.*, vol. 6, no. 4, pp. 3302–3320, 2014.
- [28] Y. Jwa and G. Sohn, "A piecewise catenary curve model growing for 3D power line reconstruction," *Photogramm. Eng. Remote Sens.*, vol. 78, no. 12, pp. 1227–1240, 2012.
- [29] F. Wu *et al.*, "Rapid localization and extraction of street light poles in mobile LIDAR point clouds: A supervoxel-based approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 292–305, Feb. 2017.
- [30] X. Liang, J. Hyypä, A. Kukko, H. Kaartinen, A. Jaakkola, and X. Yu, "The use of a mobile laser scanning system for mapping large forest plots," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 9, pp. 1504–1508, Sep. 2014.
- [31] M. Nieuwenhuisen, J. Stückler, A. Berner, R. Klein, and S. Behnke, "Shape-primitive based object recognition and grasping," in *Proc. 7th German Conf. Robot.*, 2012, pp. 1–5.
- [32] M. Li, P. Wonka, and L. Nan, "Manhattan-world urban reconstruction from point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 54–69.
- [33] L. Nan and P. Wonka, "PolyFit: Polygonal surface reconstruction from point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2353–2361.
- [34] C. Poullis, "A framework for automatic modeling from point cloud data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2563–2575, Nov. 2013.
- [35] Y. Cao, L. Nan, and P. Wonka, "Curve networks for surface reconstruction," 2016. [Online]. Available: <http://arxiv.org/abs/1603.08753>
- [36] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 2154–2161.
- [37] S. Xia, D. Chen, and R. Wang, "A breakline-preserving ground interpolation method for MLS data," *Remote Sens. Lett.*, vol. 10, no. 12, pp. 1201–1210, 2019.
- [38] S. Xia and R. Wang, "Façade separation in ground-based LIDAR point clouds based on edges and windows," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 3, pp. 1041–1052, Mar. 2019.
- [39] T. O. Chan, D. D. Lichti, and D. Belton, "A rigorous cylinder-based self-calibration approach for terrestrial laser scanners," *ISPRS J. Photogramm. Remote Sens.*, vol. 99, pp. 84–99, 2015.
- [40] B. Yang and Y. Zang, "Automated registration of dense terrestrial laser-scanning point clouds using curves," *ISPRS J. Photogramm. Remote Sens.*, vol. 95, pp. 109–121, 2014.
- [41] Y. Lin, C. Wang, B. Chen, D. Zai, and J. Li, "Facet segmentation-based line segment extraction for large-scale point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 4839–4854, Sep. 2017.
- [42] A. Habib, M. Ghanma, M. Morgan, and R. Al-Ruzouq, "Photogrammetric and LIDAR data registration using linear features," *Photogramm. Eng. Remote Sens.*, vol. 71, no. 6, pp. 699–707, 2005.
- [43] T. Rabbani, S. Dijkman, F. van den Heuvel, and G. Vosselman, "An integrated approach for modelling and global registration of point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 61, no. 6, pp. 355–370, 2007.
- [44] W. Von Hansen, "Robust automatic marker-free registration of terrestrial scan data," *Photogramm. Comput. Vis.*, vol. 36, pp. 105–110, 2006.
- [45] Y. Xu, R. Boerner, W. Yao, L. Hoegner, and U. Stilla, "Pairwise coarse registration of point clouds in urban scenes using voxel-based 4-planes congruent sets," *ISPRS J. Photogramm. Remote Sens.*, vol. 151, pp. 106–123, 2019.
- [46] J. Wang and J. Shan, "Segmentation of LIDAR point clouds for building extraction," in *Proc. Amer. Soc. Photogramm. Remote Sens. Annu. Conf.*, Baltimore, MD, USA, 2009, pp. 9–13.
- [47] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, "Automatic 3D building reconstruction from DEMs: An application to pleiades simulations," in *Proc. Int. Soc. Photogramm. Remote Sens. Commission I Symp.*, Marne La Vallée, France, 2006, pp. 1–6. [Online]. Available: <https://www.isprs.org/proceedings/xxxvii/part1/papers/T07-32.pdf>
- [48] A. Jochem, B. Höfle, and M. Rutzinger, "Extraction of vertical walls from mobile laser scanning data for solar potential assessment," *Remote Sens.*, vol. 3, no. 4, pp. 650–667, 2011.
- [49] M. Rutzinger, B. Höfle, S. O. Elberink, and G. Vosselman, "Feasibility of facade footprint extraction from mobile laser scanning data," *Photogramm.-Fernerkundung-Geoinf.*, vol. 2011, no. 3, pp. 97–107, 2011.
- [50] S. Pu and G. Vosselman, "Knowledge based reconstruction of building models from terrestrial laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 64, no. 6, pp. 575–584, 2009.
- [51] S. Xia and R. Wang, "Extraction of residential building instances in suburban areas from mobile LIDAR data," *ISPRS J. Photogramm. Remote Sens.*, vol. 144, pp. 453–468, 2018.
- [52] X. Ge, B. Wu, Y. Li, and H. Hu, "A multi-primitive-based hierarchical optimal approach for semantic labeling of ALS point clouds," *Remote Sens.*, vol. 11, no. 10, 2019, Art. no. 1243.

- [53] H. Zheng, R. Wang, and S. Xu, "Recognizing street lighting poles from mobile LIDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 1, pp. 407–420, Jan. 2017.
- [54] J. Yang, Z. Kang, and P. H. Akwensi, "A skeleton-based hierarchical method for detecting 3-D pole-like objects from mobile LIDAR point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 5, pp. 801–805, May 2019.
- [55] B. Yang, W. Dai, Z. Dong, and Y. Liu, "Automatic forest mapping at individual tree levels from terrestrial laser scanning point clouds with a hierarchical minimum cut method," *Remote Sens.*, vol. 8, no. 5, 2016, Art. no. 372.
- [56] S. Xu, S. Xu, N. Ye, and F. Zhu, "Individual stem detection in residential environments with MLS data," *Remote Sens. Lett.*, vol. 9, no. 1, pp. 51–60, 2018.
- [57] L. Li, Y. Li, and D. Li, "A method based on an adaptive radius cylinder model for detecting pole-like objects in mobile laser scanning data," *Remote Sens. Lett.*, vol. 7, no. 3, pp. 249–258, 2016.
- [58] H.-G. Maas, A. Bienert, S. Scheller, and E. Keane, "Automatic forest inventory parameter determination from terrestrial laser scanner data," *Int. J. Remote Sens.*, vol. 29, no. 5, pp. 1579–1593, 2008.
- [59] H. Guan, Y. Yu, J. Li, Z. Ji, and Q. Zhang, "Extraction of power-transmission lines from vehicle-borne LIDAR data," *Int. J. Remote Sens.*, vol. 37, no. 1, pp. 229–247, 2016.
- [60] Y. Guo, M. Bennamoun, F. Soheli, M. Lu, and J. Wan, "3D object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014.
- [61] B. Guo, Q. Li, X. Huang, and C. Wang, "An improved method for power-line reconstruction from point cloud data," *Remote Sens.*, vol. 8, no. 1, 2016, Art. no. 36.
- [62] S. Oesau, F. Lafarge, and P. Alliez, "Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut," *ISPRS J. Photogramm. Remote Sens.*, vol. 90, pp. 68–82, 2014.
- [63] S. Ochmann, R. Vock, R. Wessel, and R. Klein, "Automatic reconstruction of parametric building models from indoor point clouds," *Comput. Graph.*, vol. 54, pp. 94–103, 2016.
- [64] R. Wang, L. Xie, and D. Chen, "Modeling indoor spaces using decomposition and reconstruction of structural elements," *Photogramm. Eng. Remote Sens.*, vol. 83, no. 12, pp. 827–841, 2017.
- [65] Y. Cui, Q. Li, B. Yang, W. Xiao, C. Chen, and Z. Dong, "Automatic 3-D reconstruction of indoor environment with mobile laser scanning point clouds," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 3117–3130, Aug. 2019.
- [66] B. Yang, Z. Wei, Q. Li, and J. Li, "Semiautomated building facade footprint extraction from mobile LIDAR point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 4, pp. 766–770, Jul. 2013.
- [67] A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra, "RAPTER: Rebuilding man-made scenes with regular arrangements of planes," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 103.
- [68] J. Wang and K. Xu, "Shape detection from raw LIDAR data with subspace modeling," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 9, pp. 2137–2150, Sep. 2017.
- [69] T. Holzmann, M. R. Oswald, M. Pollefeys, F. Fraundorfer, and H. Bischof, "Planebased surface regularization for urban 3D reconstruction," in *Proc. 28th Brit. Mach. Vis. Conf.*, 2017, vol. 28, pp. 1–14.
- [70] C. Poullis and S. You, "Photorealistic large-scale urban city model reconstruction," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 4, pp. 654–669, Jul./Aug. 2009.
- [71] Q.-Y. Zhou and U. Neumann, "Fast and extensible building modeling from airborne LIDAR data," in *Proc. 16th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2008, pp. 7–15.
- [72] D. Chen, R. Wang, and J. Peethambaran, "Topologically aware building rooftop reconstruction from airborne laser scanning point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7032–7052, Dec. 2017.
- [73] C. Yi, Y. Zhang, Q. Wu, Y. Xu, O. Remil, M. Wei, and J. Wang, "Urban building reconstruction from raw LIDAR point data," *Comput.-Aided Des.*, vol. 93, pp. 1–14, 2017.
- [74] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation," *Int. J. Comput. Vis.*, vol. 99, no. 1, pp. 69–85, 2012.
- [75] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 71.
- [76] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds," *ACM Trans. Graph.*, vol. 29, no. 6, 2010, Art. no. 151.
- [77] Z. Wang *et al.*, "A structure-aware global optimization method for reconstructing 3-D tree models from terrestrial laser scanning data," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5653–5669, Sep. 2014.
- [78] J. Mei, L. Zhang, S. Wu, Z. Wang, and L. Zhang, "3D tree modeling from incomplete point clouds via optimization and L1-MST," *Int. J. Geograph. Inf. Sci.*, vol. 31, no. 5, pp. 999–1021, 2017.
- [79] N. Haala and M. Kada, "An update on automatic 3D building reconstruction," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 6, pp. 570–580, 2010.
- [80] P. Mulsalski, P. Wonka, D. G. Aliaga, M. Wimmer, L. Van Gool, and W. Purgathofer, "A survey of urban reconstruction," *Comput. Graph. Forum*, vol. 32, no. 6, pp. 146–177, 2013.
- [81] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," in *Proc. 6th IEEE Conf. Robot., Autom. Mechatronics*, 2013, pp. 225–230.
- [82] R. Wang, J. Peethambaran, and D. Chen, "Lidar point clouds to 3-D urban models: A review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 2, pp. 606–627, Feb. 2018.
- [83] E. Che, J. Jung, and M. J. Olsen, "Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review," *Sensors*, vol. 19, no. 4, 2019, Art. no. 810.
- [84] T. Rabbani, F. Van den Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraint," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 36, pp. 248–253, 2006.
- [85] E. Che and M. J. Olsen, "Multi-scan segmentation of terrestrial laser scanning data based on normal variation analysis," *ISPRS J. Photogramm. Remote Sens.*, vol. 143, pp. 233–248, 2018.
- [86] A. Nurunnabi, D. Belton, and G. West, "Robust segmentation for large volumes of laser scanning three-dimensional point cloud data," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4790–4805, Aug. 2016.
- [87] R. Maalek, D. Lichti, and J. Ruwanpura, "Robust classification and segmentation of planar and linear features for construction site progress monitoring and structural dimension compliance control," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. II-3/W5, pp. 129–136, 2015.
- [88] A.-L. Chauve, P. Labatut, and J.-P. Pons, "Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data," in *Proc. IEEE Int. Conf. Robot. Autom., 2010*, pp. 1261–1268.
- [89] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1–4.
- [90] J.-E. Deschaud and F. Goulette, "A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing," in *Proc. 3D Process., Vis. Transmiss. Conf.*, Paris, France, May, 2010, pp. 1–8.
- [91] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 104, pp. 88–100, 2015.
- [92] Y. Xu, W. Yao, L. Hoegner, and U. Stilla, "Segmentation of building roofs from airborne LIDAR point clouds using robust voxel-based region growing," *Remote Sens. Lett.*, vol. 8, no. 11, pp. 1062–1071, 2017.
- [93] Y. Xu *et al.*, "Geometric primitive extraction from point clouds of construction sites using VGS," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 424–428, Mar. 2017.
- [94] S. Oesau, F. Lafarge, and P. Alliez, "Planar shape detection and regularization in tandem," *Comput. Graph. Forum*, vol. 35, no. 1, pp. 203–215, 2016.
- [95] P. V. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3 069 654, Dec. 18, 1962.
- [96] G. Vosselman and S. Dijkman, "3D building model reconstruction from point clouds and ground plans," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 34, no. 3/W4, pp. 37–44, 2001.
- [97] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.
- [98] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design," *3D Res.*, vol. 2, no. 2, pp. 1–13, 2011.
- [99] P. Mukhopadhyay and B. B. Chaudhuri, "A survey of Hough transform," *Pattern Recognit.*, vol. 48, no. 3, pp. 993–1010, 2015.
- [100] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer, "Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from LIDAR data," in *Proc. ISPRS Workshop Laser Scanning/SilviLaser*, 2007, vol. 36, pp. 407–412.

- [101] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," *Pattern Recognit. Lett.*, vol. 11, no. 5, pp. 331–338, 1990.
- [102] M. Camurri, R. Vezzani, and R. Cucchiara, "3D Hough transform for sphere recognition on point clouds," *Mach. Vis. Appl.*, vol. 25, no. 7, pp. 1877–1891, 2014.
- [103] R. Hulik, M. Spänel, P. Smrz, and Z. Materna, "Continuous plane detection in point-cloud data based on 3D Hough transform," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 86–97, 2014.
- [104] X. Leng, J. Xiao, and Y. Wang, "A multi-scale plane-detection method based on the Hough transform and region growing," *Photogramm. Rec.*, vol. 31, no. 154, pp. 166–192, 2016.
- [105] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.
- [106] G. Vosselman, B. G. Gorte, G. Sithole, and T. Rabbani, "Recognising structure in laser scanner point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 46, no. 8, pp. 33–38, 2004.
- [107] R. Qiu, Q.-Y. Zhou, and U. Neumann, "Pipe-run extraction and reconstruction from point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 17–30.
- [108] T. Rabbani and F. Van Den Heuvel, "Efficient Hough transform for automatic detection of cylinders in point clouds," in *Proc. ISPRS Workshop Laser Scanning*, 2005, vol. 3, pp. 60–65.
- [109] B. Drost and S. Ilic, "Local Hough transform for 3D primitive detection," in *Proc. IEEE Int. Conf. 3D Vis.*, 2015, pp. 398–406.
- [110] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 259–289, 2008.
- [111] A. Velizhev, R. Shapovalov, and K. Schindler, "Implicit shape models for object detection in 3D point clouds," in *Proc. Int. Soc. Photogramm. Remote Sens. Congr.*, 2012, vol. 2, pp. 179–184.
- [112] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [113] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
- [114] K. Ni, H. Jin, and F. Dellaert, "GroupSAC: Efficient consensus in the presence of groupings," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 2193–2200.
- [115] O. Chum and J. Matas, "Matching with PROSAC-progressive sample consensus," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 220–226.
- [116] R. B. Tennakoon, A. Bab-Hadiashar, Z. Cao, R. Hoseinnezhad, and D. Suter, "Robust model fitting using higher than minimal subset sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 350–362, Feb. 2016.
- [117] O. Chum and J. Matas, "Optimal randomized RANSAC," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 8, pp. 1472–1482, Aug. 2008.
- [118] O. Chum and J. Matas, "Randomized RANSAC with Td,d test," in *Proc. Brit. Mach. Vis. Conf.*, 2002, vol. 2, pp. 448–457.
- [119] D. P. Capel, "An effective bail-out test for RANSAC consensus scoring," in *Proc. Brit. Mach. Vis. Conf.*, 2005, pp. 629–638.
- [120] D. Nistér, "Preemptive RANSAC for live structure and motion estimation," *Mach. Vis. Appl.*, vol. 16, no. 5, pp. 321–329, 2005.
- [121] M. Shahbazi, G. Sohn, and J. Théau, "Evolutionary optimization for robust epipolar-geometry estimation and outlier detection," *Algorithms*, vol. 10, no. 3, 2017, Art. no. 87.
- [122] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "USAC: A universal framework for random sample consensus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 2022–2038, Aug. 2013.
- [123] D. Chen, L. Zhang, J. Li, and R. Liu, "Urban building roof segmentation from airborne LIDAR point clouds," *Int. J. Remote Sens.*, vol. 33, no. 20, pp. 6497–6515, 2012.
- [124] M. Y. Yang and W. Förstner, "Plane detection in point cloud data," in *Proc. 2nd Int. Conf. Mach. Control Guid.*, Bonn, Germany, 2010, vol. 1, pp. 95–104.
- [125] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, 2007.
- [126] T.-T. Tran, V.-T. Cao, and D. Laurendeau, "Extraction of reliable primitives from unorganized point clouds," *3D Res.*, vol. 6, no. 4, 2015, Art. no. 44.
- [127] M. Zuliani, C. S. Kenney, and B. Manjunath, "The multiRANSAC algorithm and its application to detect planar homographies," in *Proc. IEEE Int. Conf. Image Process.*, 2005, vol. 3, pp. 153–156.
- [128] J. Yu, T.-J. Chin, and D. Suter, "A global optimization approach to robust multi-model fitting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2041–2048.
- [129] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *Int. J. Comput. Vis.*, vol. 97, no. 2, pp. 123–147, 2012.
- [130] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
- [131] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *Int. J. Comput. Vis.*, vol. 96, no. 1, pp. 1–27, 2012.
- [132] T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter, "Interacting geometric priors for robust multimodel fitting," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4601–4610, Oct. 2014.
- [133] T. T. Pham, T.-J. Chin, J. Yu, and D. Suter, "The random cluster model for robust geometric fitting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1658–1671, Aug. 2014.
- [134] J. Yan, J. Shan, and W. Jiang, "A global optimization approach to roof segmentation from airborne LIDAR point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 94, pp. 183–193, 2014.
- [135] J. Du, D. Chen, R. Wang, J. Peethambaran, P. T. Mathiopoulos, L. Xie, and T. Yun, "A novel framework for 2.5-D building contouring from large-scale residential scenes," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4121–4145, Jun. 2019.
- [136] L. Magri and A. Fusiello, "Multiple model fitting as a set coverage problem," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3318–3326.
- [137] S. Xia and R. Wang, "Semiautomatic construction of 2-D façade footprints from mobile LIDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4005–4020, Jun. 2019.
- [138] H. Jiang and J. Xiao, "A linear approach to matching cuboids in RGBD images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2171–2178.
- [139] W. Zhang and J. Ksecká, "Nonparametric estimation of multiple structures with outliers," in *Dynamical Vision*. New York, NY, USA: Springer, 2006, pp. 60–74.
- [140] R. Toldo and A. Fusiello, "Robust multiple structures estimation with j-linkage," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 537–547.
- [141] L. Magri and A. Fusiello, "T-linkage: A continuous relaxation of j-linkage for multi-model fitting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3954–3961.
- [142] G. Xiao, H. Wang, T. Lai, and D. Suter, "Hypergraph modelling for geometric model fitting," *Pattern Recognit.*, vol. 60, pp. 748–760, 2016.
- [143] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Departmental Papers (CIS)*, p. 107, 2000.
- [144] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 6218–6225.
- [145] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial LIDAR point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1554–1567, Mar. 2010.
- [146] C. Poullis, "Large-scale urban reconstruction with tensor clustering and global boundary refinement," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
- [147] S. Xu, R. Wang, H. Wang, and H. Zheng, "An optimal hierarchical clustering approach to mobile LIDAR point clouds," *IEEE Trans. Intell. Transp. Syst.*, to be published.
- [148] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
- [149] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.
- [150] H. Wang, T.-J. Chin, and D. Suter, "Simultaneously fitting and segmenting multiple-structure data with outliers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1177–1192, Jun. 2012.
- [151] Y. A. Sheikh, E. A. Khan, and T. Kanade, "Mode-seeking by medoid-shifts," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
- [152] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, Jun. 1994.
- [153] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea, "3D skeletons: A state-of-the-art report," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 573–597, 2016.
- [154] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton extraction by mesh contraction," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. no. 44.

- [155] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen, "L1-medial skeleton of point cloud," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 65.
- [156] A. Sharf, T. Lewiner, A. Shamir, and L. Kobbelt, "On-the-fly curve-skeleton computation for 3D shapes," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 323–328, 2007.
- [157] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point cloud skeletons via Laplacian based contraction," in *Proc. Shape Model. Int. Conf.*, 2010, pp. 187–197.
- [158] A. Bucksch and R. Lindenbergh, "Campinoa skeletonization method for point cloud processing," *ISPRS J. Photogramm. Remote Sens.*, vol. 63, no. 1, pp. 115–127, 2008.
- [159] Q.-Y. Zhou and U. Neumann, "2.5 D dual contouring: A robust approach to creating building models from aerial LIDAR point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 115–128.
- [160] B. Yu, H. Liu, J. Wu, Y. Hu, and L. Zhang, "Automated derivation of urban building density information using airborne LIDAR data and object-based method," *Landscape Urban Planning*, vol. 98, nos. 3/4, pp. 210–219, 2010.
- [161] Y.-T. Liow, "A contour tracing algorithm that preserves common boundaries between regions," *CVGIP: Image Understanding*, vol. 53, no. 3, pp. 313–321, 1991.
- [162] D. Grigillo and U. Kanjir, "Urban object extraction from digital surface model and digital aerial images," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 215–220, 2012.
- [163] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.
- [164] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*. Berlin, Germany: Springer, 1987, pp. 7–15.
- [165] B. Yang, W. Xu, and Z. Dong, "Automated extraction of building outlines from airborne laser scanning point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 6, pp. 1399–1403, Nov. 2013.
- [166] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 4, pp. 551–559, Jul. 1983.
- [167] A. Sampath and J. Shan, "Building boundary tracing and regularization from airborne LIDAR point clouds," *Photogramm. Eng. Remote Sens.*, vol. 73, no. 7, pp. 805–812, 2007.
- [168] P. Dorninger and N. Pfeifer, "A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds," *Sensors*, vol. 8, no. 11, pp. 7323–7343, 2008.
- [169] T. Hackel, J. D. Wegner, and K. Schindler, "Contour detection in unstructured 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1610–1618.
- [170] H. Ni, X. Lin, X. Ning, and J. Zhang, "Edge detection and feature line tracing in 3D-point clouds by analyzing geometric properties of neighborhoods," *Remote Sens.*, vol. 8, no. 9, pp. 1–20, 2016.
- [171] D. Belton and D. D. Lichti, "International archives of the photogrammetry, remote sensing and spatial information sciences," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 36, no. 5, pp. 44–49, 2006.
- [172] K. Demarsin, D. Vanderstraeten, T. Volodine, and D. Roose, "Detection of closed sharp edges in point clouds using normal estimation and graph theory," *Comput.-Aided Des.*, vol. 39, no. 4, pp. 276–283, 2007.
- [173] T. Hackel, J. D. Wegner, and K. Schindler, "Joint classification and contour extraction of large 3D point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 231–245, 2017.
- [174] J. Demantké, C. Mallet, N. David, and B. Vallet, "Dimensionality based scale selection in 3D LIDAR point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 38, no. 5/W12, pp. 97–102, 2011.
- [175] A. Dittrich, M. Weinmann, and S. Hinz, "Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3d point cloud data," *ISPRS J. Photogramm. Remote Sens.*, vol. 126, pp. 195–208, 2017.
- [176] P. Borges, R. Zlot, M. Bosse, S. Nuske, and A. Tews, "Vision-based localization using an edge map extracted from 3D laser range data," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 4902–4909.
- [177] Z. Kang, R. Zhong, A. Wu, Z. Shi, and Z. Luo, "An efficient planar feature fitting method using point cloud simplification and threshold-independent BaySAC," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1842–1846, Dec. 2016.
- [178] X. Chen and K. Yu, "Feature line generation and regularization from point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9779–9790, Dec. 2019.
- [179] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
- [180] J. Canny, "A computational approach to edge detection," in *Readings in Computer Vision*. New York, NY, USA: Elsevier, 1987, pp. 184–203.
- [181] Y. Lin, C. Wang, J. Cheng, B. Chen, F. Jia, Z. Chen, and J. Li, "Line segment extraction for large scale unorganized point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 102, pp. 172–183, 2015.
- [182] R. G. Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.
- [183] X. Lu, Y. Liu, and K. Li, "Fast 3D line segment detection from unorganized point cloud," 2019, *arXiv:1901.02532*.
- [184] S. Xia and R. Wang, "A fast edge extraction method for mobile LIDAR point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1288–1292, Aug. 2017.
- [185] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–151.
- [186] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.
- [187] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *ACM SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, 1992.
- [188] A. Nurunnabi, D. Belton, and G. West, "Robust statistical approaches for local planar surface fitting in 3D laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 96, pp. 106–122, 2014.
- [189] K. Zhang, J. Yan, and S.-C. Chen, "Automatic construction of building footprints from airborne LIDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2523–2533, Sep. 2006.
- [190] Y. Verdier, F. Lafarge, and P. Alliez, "LOD generation for urban scenes," *ACM Trans. Graph.*, vol. 34, no. 3, 2015, Art. no. 30.
- [191] F. Lafarge, R. Keriven, M. Brédif, and H.-H. Vu, "A hybrid multiview stereo algorithm for modeling urban scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 5–17, Jan. 2013.
- [192] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra, "GlobFit: Consistently fitting primitives by discovering global relations," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. no. 52.
- [193] N. Schertler, B. Savchynskyy, and S. Gumhold, "Towards globally optimal normal orientations for large point clouds," *Comput. Graph. Forum*, vol. 36, no. 1, pp. 197–208, 2017.
- [194] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, 2009, Art. no. 176.
- [195] Q.-Y. Zhou and U. Neumann, "2.5 D building modeling by discovering global regularities," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 326–333.
- [196] M. Weinmann, B. Jutzi, and C. Mallet, "Feature relevance assessment for the semantic interpretation of 3D point cloud data," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 5, no. W2, pp. 313–318, 2013.
- [197] B. Jutzi and H. Gross, "Nearest neighbour classification on laser point clouds to gain object structures from buildings," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 38, no. 1, pp. 4–7, 2009.
- [198] Y. Guo, F. Sohler, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," *Int. J. Comput. Vis.*, vol. 105, no. 1, pp. 63–86, 2013.
- [199] C. Mallet, F. Bretar, M. Roux, U. Soergel, and C. Heipke, "Relevance assessment of full-waveform LIDAR data for urban area classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 6, pp. S71–S84, 2011.
- [200] B. Guo, X. Huang, F. Zhang, and G. Sohn, "Classification of airborne laser scanning data using jointboost," *ISPRS J. Photogramm. Remote Sens.*, vol. 100, pp. 71–83, 2015.
- [201] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.
- [202] Z. Wang *et al.*, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2409–2425, May 2015.
- [203] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa, "Planenet: Piecewise planar reconstruction from a single RGB image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2579–2588.
- [204] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem, "3D-PRNN: Generating shape primitives with recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 900–909.

- [205] D. Li and C. Feng, "Primitive fitting using deep boundary aware geometric segmentation," 2018, *arXiv:1810.01604*.
- [206] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik, "Learning shape abstractions by assembling volumetric primitives," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2635–2643.
- [207] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
- [208] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
- [209] L. Li, M. Sung, A. Dubrovina, L. Yi, and L. J. Guibas, "Supervised fitting of geometric primitives to 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2652–2660.
- [210] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4558–4567.
- [211] Z. Yu *et al.*, "Simultaneous edge alignment and learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 388–404.
- [212] R. Atienza, "Pyramid U-network for skeleton extraction from shape points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1–5.
- [213] I. Demir *et al.*, "Skelneton 2019: Dataset and challenge on deep learning for geometric shape understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1–4.
- [214] E. Widyaningrum and B. Gorte, "Comprehensive comparison of two image-based point clouds from aerial photos with airborne LIDAR for large-scale mapping," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 42, pp. 557–565, 2017.
- [215] A. Salach, K. Bakula, M. Pilarska, W. Ostrowski, K. Górski, and Z. Kurczyński, "Accuracy assessment of point clouds from LIDAR and dense image matching acquired using the UAV platform for DTM creation," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 9, 2018, Art. no. 342.
- [216] H. Fang, F. Lafarge, and M. Desbrun, "Planar shape detection at structural scales," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2965–2973.
- [217] Z. Dong, B. Yang, P. Hu, and S. Scherer, "An efficient global energy optimization approach for robust 3D plane segmentation of point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 137, pp. 112–133, 2018.
- [218] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam, "Casenet: Deep category-aware semantic edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5964–5973.
- [219] F. Liu *et al.*, "3DCNN-DQN-RNN: A deep reinforcement learning framework for semantic parsing of large-scale 3D point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5678–5687.



Shaobo Xia received the bachelor's degree in geodesy and geomatics, in 2013 from the School of Geodesy and Geomatics, Wuhan University, Wuhan, China, and the master's degree in cartography and geographic information systems, in 2016 from the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China. He is currently working toward the Ph.D. degree with the Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada.

His research interests include mobile mapping and remote sensing.



Dong Chen (Member, IEEE) received the bachelor's degree in computer science from the Qingdao University of Science and Technology, Qingdao, China, in 2005, the master's degree in cartography and geographical information engineering from the Xi'an University of Science and Technology, Xi'an, China, in 2009, and the Ph.D. degree in geographical information sciences from Beijing Normal University, Beijing, China, in 2013.

He is currently an Assistant Professor with Nanjing Forestry University, Nanjing, China. He is also a

Postdoctoral Fellow with the Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada. His research interests include image- and LiDAR-based segmentation and reconstruction, full-waveform LiDAR data processing, and related remote sensing applications in the field of forest ecosystems.



Ruisheng Wang (Senior Member, IEEE) received the B.Eng. degree in photogrammetry from Wuhan University, Wuhan, China, in 1993, the M.Sc.E. degree in geomatics engineering from the University of New Brunswick, Fredericton, NB, Canada, in 2004, and the Ph.D. degree in electrical and computer engineering from McGill University, Montreal, QC, Canada, in 2011.

He is currently an Associate Professor with the Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada. His research interests include geomatics and computer vision, especially point cloud processing.



Jonathan Li (Senior Member, IEEE) received the Ph.D. degree in remote sensing and geographic information systems, in 2016 from the University of Cape Town, Cape Town, South Africa.

He is currently a Professor of Geomatics and Head of the Geospatial Technology and Remote Sensing Group, Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON, Canada. He is also an Adjunct Professor with York University, Toronto, ON, Canada, and a Guest Professor with several top-ranked universities in China.

He has published extensively in leading remote sensing journals. His current research interests include the use of mobile laser scanning point clouds for 3-D surface modeling.

Dr. Li is the recipient of several prestigious awards. He was the Chair of ISPRS ICWG I/Va on Mobile Scanning and Imaging Systems from 2012 to 2016, the Vice Chair of the ICA Commission on Mapping from Remote Sensor Imagery from 2011 to 2015, and the Vice Chair of the FIG Commission IV on Hydrography as well as the Chair of FIG WG4.4 on Maritime and Marine Spatial Information Management from 2014 to 2018. He has been the Remote Sensing Editor for *Geomatica* since 2007.



Xinchang Zhang received the Ph.D. degree in mapping and geographic information engineering, in 2004 from Wuhan University, Wuhan, China.

He is currently a Second Grade Professor with the School of Geographical Science, Guangzhou University, Guangzhou, China. He is also a Professor and Supervisor of Ph.D. students with Sun Yat-Sen University, Guangzhou, China. He was elected as IEAS Academician in 2017. His research interests include the integration of multisource geospatial information data, intelligent management, and automatic updating of databases.