A Review of Point Cloud Semantic Segmentation

Yuxing Xie, Jiaojiao Tian, *Member, IEEE* and Xiao Xiang Zhu, *Senior Member, IEEE*

Abstract

This is the preprint version. To read the final version please go to IEEE Geoscience and Remote Sensing Magazine on IEEE XPlore.

3D Point Cloud Semantic Segmentation (PCSS) is attracting increasing interest, due to its applicability in remote sensing, computer vision and robotics, and due to the new possibilities offered by deep learning techniques. In order to provide a needed up-to-date review of recent developments in PCSS, this article summarizes existing studies on this topic. Firstly, we outline the acquisition and evolution of the 3D point cloud from the perspective of remote sensing and computer vision, as well as the published benchmarks for PCSS studies. Then, traditional and advanced techniques used for Point Cloud Segmentation (PCS) and PCSS are reviewed and compared. Finally, important issues and open questions in PCSS studies are discussed.

Index Terms

review, point cloud, segmentation, semantic segmentation, deep learning.

I. MOTIVATION

Semantic segmentation, in which pixels are associated with semantic labels, is a fundamental research challenge in image processing. Point Cloud Semantic Segmentation (PCSS) is the 3D form of semantic segmentation, in which regular or irregular distributed points in 3D space are used instead of regular distributed pixels in a 2D image. The point cloud can be acquired directly from sensors with distance measurability, or generated from stereo- or multi-view imagery. Due to recently developed stereovision algorithms and the deployment of all kinds of 3D sensors, point clouds, basic 3D data, have become

easily accessible. High-quality point clouds provide a way to connect the virtual world to the real one. Specifically, they generate 2.5D/3D geometric structures, with which modeling is possible.

A. Segmentation, classification, and semantic segmentation

Research on PCSS has a long tradition involving different fields and defining distinct concepts for similar tasks. A brief clarification of some concepts is therefore necessary to avoid misunderstandings. The term PCSS is widely used in computer vision, especially in recent deep learning applications [1]–[3]. However, in photogrammetry and remote sensing, PCSS is usually called "point cloud classification" [4]–[6]. Or in some cases, this task is also called "point labeling" [7]–[9]. In this article, to avoid confusion and to make this literature review keep up with latest deep learning techniques, we refer to point cloud semantic segmentation/classification/labeling, i.e., the task of associating each point of a point cloud with a semantic label, as PCSS.

Before effective supervised learning methods were widely applied in semantic segmentation, unsupervised Point Cloud Segmentation (PCS) was a significant task for 2.5D/3D data. PCS aims at grouping points with similar geometric/spectral characteristics without considering semantic information. In the PCSS workflow, PCS can be utilized as a presegmentation step, influencing the final results. Hence, PCS approaches are also included in this paper.

Single objects or the same classes of structures cannot be acquired from a raw point cloud directly. However, instance-level or class-level objects are required for object recognition. For example, urban planning and Building Information Modeling (BIM) need buildings and other man-made ground objects for reference [10], [11]. Forest remote sensing monitoring needs individual tree information based on their geometric structures [12], [13]. Robotics applications, like Simultaneous Localization And Mapping (SLAM), need detailed indoor objects for mapping [7], [14]. In some applications related to computer vision, such as autonomous driving, object detection, segmentation, and classification are necessary with the construction of a High Definition (HD) Map [15]. For the mentioned cases, PCSS and PCS are basic and critical tasks for 3D applications.

B. *New challenges and possibilities*

Papers [16] and [17] provide two of the best available reviews for PCS and PCSS, but lack detailed information, especially for PCSS. Futhermore, in the past two years, deep learning has largely driven studies in PCSS. To meet the demand of deep learning, 3D datasets have improved, both in quality and diversity. Therefore, an updated study on current PCSS techniques is necessary. This paper starts with the introduction of existing techniques to acquire point clouds and the existing benchmarks for point cloud study (section II). In section III and IV, the major categories of algorithms are reviewed, for both PCS and PCSS. In section V, some issues related to data and techniques are discussed. Section VI concludes this paper with a technical outlook.

II. AN INTRODUCTION TO POINT CLOUD

A. *Point cloud data acquisition*

In computer vision and remote sensing, point clouds can be acquired with four main techniques: 1) Image-derived methods; 2) Light Detection And Ranging (LiDAR) systems; 3) Red Green Blue -Depth (RGB-D) cameras; and 4) Synthetic Aperture Radar (SAR) systems. Due to the differences in survey principles and platforms, their data features and application ranges are very diverse. A brief introduction to these techniques is provided below.

1) Image-derived point cloud: Image-derived methods generate a point cloud indirectly from spectral imagery. First, they acquire stereotype images through electro-optical systems, e.g., cameras. Then they calculate 3D isolated point information according to principles in photogrammetry or computer vision theory, either automatically or semi-automatically [18], [19]. Based on distinct platforms, stereo- and multi-view image-derived systems can be divided into airborne, spaceborne, UAV-based, and close-range categories.

Early aerial traditional photogrammetry produced 3D points with semi-automatic human-computer interaction in digital photogrammetric systems, characterized by strict geometric constraints and high

survey accuracy [20]. To produce this type of point data was time expensive due to many manual works. Therefore it was not feasible to generate dense points for large areas in this way. In the surveying and remote sensing industry, those early-form "point clouds" were used in mapping and producing Digital Surface Models (DSMs) and Digital Elevation Models (DEMs). Due to the limitation of image resolutioan and the ability of processing multi-view images, traditional photogrammetry could only acquire close to nadir views with few building façades from aerial/satellite platforms, which only generated a 2.5D point cloud rather than full 3D. At this stage, photogrammetry principles could also be applied as close-range photogrammetry in order to obtain points from certain objects or small-area scenes, but manual editing would also be necessary in the point cloud generating procedure.

Dense matching [21]–[23], Multiple View Stereovision (MVS) [24], [25], and Structure from Motion (SfM) [19], [26], [27], changed the image-derived point cloud, and opened the era of multiple view stereovision. SfM can estimate camera positions and orientations automatically, making it capable of processing multiview images simultaneously, while dense matching and MVS algorithms provide the ability to generate large volume of point clouds. In recent years, city-scale full 3D dense point clouds can be acquired easily through an oblique photography technique based on SfM and MVS. However, the quality of point clouds from SfM and MVS is not as good as those generated by traditional photogrammetry or LiDAR techniques, and it is especially unreliable for large regions [28].

Compared to airborne photogrammetry, satellite stereo system is disadvantaged in terms of spatial resolution and availability of multi-view imagery. However, satellite cameras are able to map large regions in a short period of time with relatively lower cost. Also due to new dense matching techniques and their improved spatial resolution, satellite imagery is becoming an important data source for image-derived point clouds.

2) LiDAR point cloud: Light Detection And Ranging (LiDAR) is a surveying and remote sensing technique. As its name suggests, LiDAR utilizes laser energy to measure the distance between the sensor and the object to be surveyed [29]. Most LiDAR systems are pulse-based. The basic principle of pulse-

based measuring is to emit a pulse of laser energy and then measure the time it takes for that energy to travel to a target. Depending on sensors and platforms, the point density or resolution varies greatly, from less than 10 points per m^2 (pts/m^2) to thousands of points per m^2 [30]. Based on platforms, LiDAR systems are divided into airborne LiDAR scanning (ALS), terrestrial LiDAR scanning (TLS), mobile LiDAR scanning (MLS) and unmanned LiDAR scanning (ULS) systems.

ALS operates from airborne platforms. Early ALS LiDAR data are 2.5D point clouds, which are similar to traditional photogrammetric point clouds. The density of ALS points is normally low, as the distance from an airborne platform to the ground is large. In comparison to traditional photogrammetry, ALS point clouds are more expensive to acquire and normally contain no spectral information. Vaihingen point cloud semantic labeling dataset [31] is a typical ALS benchmark dataset. Multispectral airborne LiDAR is a special form of an ALS system that obtains data using different wavelengths. Multispectral LiDAR performs well for the extraction of water, vegetation and shadows, but the data are not easily available [32], [33].

TLS, also called static LiDAR scanning, scans with a tripod-mounted stationary sensor. Since it is used in a middle- or close-range environment, the point cloud density is very high. Its advantage is its ability to provide real, high quality 3D models. Until now TLS has been commonly used for modeling small urban or forest sites, and heritage or artwork documentation. Semantic3D.net [34] is a typical TLS benchmark dataset.

MLS operates from a moving vehicle on the ground, with the most common platforms being cars. Currently, research and development on autonomous driving is a hot topic, for which HD maps are essential. The generation of HD maps is therefore the most significant application for MLS. Several mainstream point cloud benchmark datasets belong to MLS [35], [36].

ULS systems are usually deployed on drones or other unmanned vehicles. Since they are relatively cheap and very flexible, this recent addition to the LiDAR family is currently becoming more and more popular. Compared to ALS, where the platform is working above the objects, ULS can provide a shorter-

distance LiDAR survey application, collecting denser point clouds with higher accuracy. Thanks to the small size and light weight of its platform, ULS offers high operational flexibility. Therefore, in addition to traditional LiDAR tasks (e.g., acquiring DSMs), ULS has advantages in agriculture and forestry surveying, disaster monitoring and mining surveying [37]–[39].

For LiDAR scanning, since the system is always moving with the platform, it is necessary to combine points' positions with Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) data to ensure a high-quality matching point cloud. Until now, LiDAR has been the most important data source for point cloud research and has been used to provide ground truth to evaluate the quality of other point clouds.

3) RGB-D point cloud: An RGB-D camera is a type of sensor that can acquire both RGB and depth information. There are three kinds of RGB-D sensors, based on different principles: (a) structured light [40], (b) stereo [41], and (c) time of flight [42]. Similar to LiDAR, the RGB-D camera can measure the distance between the camera to the objects, but pixel-wise. However, a RGB-D sensor is much cheaper than a LiDAR system. Microsoft's Kinect is the most well-known and most used RGB-D sensor [40], [42]. In a RGB-D camera, relative orientation elements between or among different sensors are calibrated and known, so co-registered synchronized RGB images and depth maps can be easily acquired. Obviously, the point cloud is not the direct product of RGB-D scanning. But since the position of the camera's center point is known, the 3D space position of each pixel in a depth map can be easily obtained, and then directly used to generate the point cloud. RGB-D cameras have three main applications: object tracking, human pose or signature recognition, and SLAM-based environment reconstruction. Since mainstream RGB-D sensors are close-range, even much closer than TLS, they are usually employed in indoor environments. Several mainstream indoor point cloud segmentation benchmarks are RGB-D data [43], [44].

4) SAR point cloud: Interferometric Synthetic Aperture Radar (InSAR), a radar technique crucial to remote sensing, generates maps of surface deformation or digital elevation based on the comparison of multiple SAR image pairs. A rising star, InSAR-based point cloud has showed its value over the past

few years and is creating new possibilities for point cloud applications [45]–[49]. Synthetic Aperture Radar tomography (TomoSAR) and Persistent Scatterer Interferometry (PSI) are two major techniques that generate point clouds with InSAR, extending the principle of SAR into 3D [50], [51]. Compared with PSI, TomoSAR's advantage is its detailed reconstruction and monitoring of urban areas, especially man-made infrastructure [51]. The TomoSAR point cloud has a point density that is comparable to LiDAR [52], [53]. These point clouds can be employed for applications in building reconstruction in urban areas, as they have the following features [46]:

(a) TomoSAR point clouds reconstructed from spaceborne data have a moderate 3D positioning accuracy on the order of 1 m [54], even able to reach a decimeter level by geocoding error correction techniques [55], while ALS LiDAR provides accuracy typically on the order of 0.1 m [56].

(b) Due to their coherent imaging nature and side-looking geometry, TomoSAR point clouds emphasize different objects with respect to LiDAR systems: a) The side-looking SAR geometry enables TomoSAR point clouds to possess rich façade information: results using pixel-wise TomoSAR for the high-resolution reconstruction of a building complex with a very high level of detail from spaceborne SAR data are presented in [57]; b) temporarily incoherent objects, e.g., trees, cannot be reconstructed from multipass spaceborne SAR image stacks; and c) to obtain the full structure of individual buildings from space, façade reconstruction using TomoSAR point clouds from multiple viewing angles is required [45], [58].

(c) Complementary to LiDAR and optical sensors, SAR is so far the only sensor capable of providing fourth dimension information from space, i.e., temporal deformation of the building complex [59], and microwave scattering properties of the façade reflect geometrical and material features.

InSAR point clouds have two main shortcomings that affect their accuracy: (1) Due to limited orbit spread and the small number of images, the location error of TomoSAR points is highly anisotropic, with an elevation error typically one or two orders of magnitude higher than in range and azimuth; (2) Due to multiple scattering, ghost scatterers may be generated, appearing as outliers far away from a realistic 3D position [60].

Compared with the aforementioned image-derived, LiDAR-based, and RGB-D-based point cloud, the data from SAR have not yet been widely used for studies and applications. However, mature SAR satellites, such as TerraSAR-X, have collected rich global SAR data, which are available for InSAR-based reconstruction at global scale [61]. Hence, the SAR point cloud can be expected to play a conspicuous role in the future.

B. Point cloud characters

From the perspective of sensor development and various applications, we have cataloged point clouds into: (a) sparse (less than 20 pts/m^2), (b) dense (hundreds of pts/m^2), and (c) multi-source.

(a) In their early stage, which was limited by matching techniques and computation ability, photogrammetric point clouds were sparse and small in volume. At that time, laser scanning systems had limited types and were not widely used. ALS point clouds, mainstream laser data, were also sparse. Limited by the point density, point clouds at this stage were not able to represent land covers in object level. Therefore there was no specific demand for precise PCS or PCSS. Researchers mainly focused on 3D mapping (DEM generation), and simple object extraction (e.g., rooftops).

(b) Computer vision algorithms, such as dense matching, and high-efficiency point cloud generators, such as various LiDAR systems and RGB-D sensors, opened the big data era of the dense point cloud. Dense and large-volume point clouds created more possibilities in 3D applications but also had a stronger desire for practicable algorithms. PCS and PCSS were newly proposed and became increasingly necessary, since only a class-level or instance-level point cloud further connect virtual word to the real one. Both computer vision and remote sensing need PCS and PCSS solutions to develop class-level interactive applications.

(c) From the perspective of general computer vision, research on the point cloud and its related algorithms remain at stage (b). However, as a benefit to the development of spaceborne platforms and multi-sensors, remote sensing researchers developed a new understanding of the point cloud. New-generation point clouds, such as satellite photogrammetric point clouds and TomoSAR point clouds, stimulated

demand for relevant algorithms. Multi-source data fusion has become a trend in remote sensing [62]–[64], but current algorithms in computer vision are insufficient for such remote sensing datasets. To fully exploit multi-source point cloud data, more research is needed.

As we have reviewed, different point clouds have different features and application environments. Table I provides an overview of basic information about various point clouds, including point density, advantages, disadvantages, and applications.

C. Point cloud application

In the studies on PCS and PCSS, data and algorithm selections are driven by the requirements of specific applications. In this section, we outline most of the studies focusing on PCS and PCSS reviewed in this article (see Table II). These works are classified according to their point cloud data types and working environments. The latter include urban, forest, industry, and indoor settings. In Table II, texts in brackets, after each reference, contain the corresponding publishing year and main methods. Algorithm types are represented as abbreviations.

Several issues can be summarized from Table II: (a) LiDAR point clouds are the most commomly used data in PCS. They have been widely used for buildings (urban environments) and trees (forests). Buildings are also the most popular research objects in traditional PCS. As buildings are usually constructed with regular planes, plane segmentation is a fundamental topic in building segmentation.

(b) Image-derived point clouds have been frequently used in real-world scenarios. However, mainly due to the limitation of available annotated benchmarks, there are not many PCS and PCSS studies on image-based data. Currently, there is only one public influential dataset based on image-derived points, whose range is only a very small area around one single building [132]. More efforts are therefore needed in this area.

(c) RGB-D sensors are limited by their close range, so they are usually applied in an indoor environment. In PCS studies, plane segmentation is the main task for RGB-D data. In PCSS studies, since there are several benchmark datasets from RGB-D sensors, many deep learning-based methods are tested on them.

TABLE I
AN OVERVIEW OF VARIOUS POINT CLOUDS

		Point density	Advantages	Disadvantages	Applications
Image-derived		From sparse ($<10 pts/m^2$) to very high ($>400 pts/m^2$), depending on the spatial resolution of the stereo- or multi-view images	With color (RGB, multi-spectral) information; suitable for large area (airborne, spaceborne)	Influenced by light; accuracy depends on available precise camera models, image matching algorithms, stereo angles, image resolution and image quality; not suitable for areas or objects without texture, such as water or snow-covered regions; influenced by shadows in images	Urban monitoring; vegetation monitoring; 3D object reconstruction; etc.
LiDAR	ALS	Sparse ($<20 pts/m^2$)	High accuracy ($<15 cm$); suitable for large area; not affected by weather	Expensive; affected by mirror reflection; long scanning time	Urban monitoring; vegetation monitoring; power line detection; etc.
	MLS	Dense ($>100 pts/m^2$), the survey distance is smaller, the density is higher	High accuracy (cm-level)		HD map; urban monitoring
	TLS	Dense ($>100 pts/m^2$), the survey distance is smaller, the density is higher	High accuracy (mm-level)		Small-area 3D reconstruction
	ULS	Dense ($>100 pts/m^2$), the survey distance is smaller, the density is higher	High accuracy (cm-level)		Forestry survey; mining survey; disaster monitoring; etc.
RGB-D		Middle-density	Cheap; flexible	Close-range; limited accuracy	Indoor reconstruction; object tracking; human pose recognition; etc.
InSAR		Sparse ($<20 pts/m^2$)	Global data is available; compared to ALS, complete building façade information is available; 4D information; middle-accuracy; not affected by weather	Expensive data; ghost scatterers; preprocessing techniques are needed	Urban monitoring; forest monitoring; etc.

TABLE II
AN OVERVIEW OF PCS AND PCSS APPLICATIONS SORTED ACCORDING TO DATA ACQUISITIONS

RG is short for Region Growing. HT is short for Hough Transform. R is short for RANSAC. C is short for Clustering-based. O is short for Oversegmentation. ML is short for Machine Learning. DL is short for Deep Learning.

	Urban	Forest	Industry	Indoor
Image-derived	**Building façades**: [65] (2018/RG), [66] (2005/RG); **PCSS**: [67] (2018/DL), [68] (2018/DL), [69] (2017/DL), [70] (2019/DL)			**Plane PCS**: [71] (2015/HT)
ALS	**Building plane PCS**: [72] (2015/R), [73] (2014/R), [74] (2007/R, HT), [75] (2002/HT), [76] (2006/C), [77] (2010/C), [78] (2012/C), [79] (2014/C); **Urban scene**: [80] (2007/C), [81] (2009/C); **PCSS**: [82] (2007/ML), [83] (2009/ML), [84] (2009/ML), [85] (2010/ML), [86] (2012/ML), [87] (2014/ML), [88] (2017/HT, R, ML), [89] (2011/ML), [90] (2014/ML), [4] (2013/HT, ML)	**Tree structure PCS**: [91](2004/C); **Forest structure**: [92] (2010/C)		
MLS	**Buildings**: [93] (2015/RG); **Urban objects**: [94] (2012/RG); **PCSS**: [89] (2011/ML), [95] (2015/ML), [5] (2015/ML), [8] (2012/ML), [90] (2014/ML), [96] (2009/ML), [97] (2017/ML), [98] (2017/DL), [99] (2018/DL), [100] (2019/O, DL)			**Plane PCS**: [101] (2013/R), [102] (2017/R)
TLS	**Building/building structure PCS**: [103] (2007/R), [93] (2015/RG), [104] (2018/RG, C), [105] (2008/C); **Buildings and trees**: [106] (2009/RG); **Urban scene**: [107] (2016/O, C), [108] (2017/O, C), [109] (2018/O, C); **PCSS**: [6] (2015/ML), [110] (2009/O, ML), [111] (2016/ML), [67] (2018/DL), [98] (2017/DL), [2] (2018/O, DL), [112] (2019/DL) [70] (2019/DL)	**Tree PCSS**: [113] (2005/ML)		**Plane PCS**: [114] (2011/HT)
RGB-D				**Plane PCS**: [115] (2014/HT), [104] (2018/RG, C); **PCSS**: [116] (2012/ML), [117] (2013/ML), [118] (2018/DL), [119] (2018/DL), [98] (2017/DL), [1] (2017/DL), [120] (2017/DL), [3] (2018/DL), [2] (2018/DL), [99] (2018/DL), [121] (2018/DL), [70] (2019/DL), [112] (2019/DL), [122] (2019/DL), [123] (2019/DL), [124] (2019/DL), [125] (2019/DL), [126] (2019/DL), [100] (2019/O, DL); **Instance segmentation**: [127] (2018/DL), [128] (2019/DL), [123] (2019/DL), [124] (2019/DL)
InSAR	**Building/building structure**: [47] (2015/C), [45] (2012/C), [46] (2014/C)	**Tree PCS**: [48] (2015/C)		
Not mentioned data			[129](2005/HT), [130] (2015/R), [131] (2018/R)	

(d) As for InSAR point clouds, although there are not many PCS or PCSS studies, these have shown potential in urban monitoring, especially building structure segmentation.

D. Benchmark datasets

Public standard benchmark datasets achieve significant effectiveness for algorithm development, evaluation and comparison. It should be noted that most of them are labeled for PCSS rather than PCS. Since 2009, several benchmark datasets have been available for PCSS. However, early datasets have plenty of shortcomings. For example, the Oakland outdoor MLS dataset [96], the Sydney Urban Objects MLS dataset [133], the Paris-rue-Madame MLS dataset [134], the IQmulus & TerraMobilita Contest MLS dataset [35] and ETHZ CVL RueMonge 2014 multiview stereo dataset [132] can not sufficiently provide both different object representations and labeled points. KITTI [135] and NYUv2 [136] have more objects and points than the aforementioned datasets, but they do not provide a labeled point cloud directly. These must be generated from 3D bounding boxes in KITTI or depth images in NYUv2.

To overcome the drawbacks of early datasets, new benchmark data have been made available in recent years. Currently, mainstream PCSS benchmark datasets are from either LiDAR or RGB-D. A nonexhaustive list of these datasets follows.

1) Semantic3D.net: The semantic3D.net [34] is a representative large-scale outdoor TLS PCSS dataset. It is a collection of urban scenes with over four billion labeled 3D points in total for PCSS purposes only. Those scenes contain a range of diverse urban objects, divided into eight classes, including man-made terrain, natural terrain, high vegetation, low vegetation, buildings, hardscape, scanning artefacts, and cars. In consideration of the efficiency of different algorithms, two types of sub-datasets were designed, semantic-8 and reduced-8. Semantic-8 is the full dataset, while reduced-8 uses training data in the same way as semantic-8, but only includes four small-sized subsets as test data. This dataset can be downloaded at http://www.semantic3d.net/. To learn the performance of different algorithms on this dataset, readers are recommended to refer to [2], [67], [112].

2) Stanford Large-scale 3D Indoor Spaces Dataset (S3DIS): Unlike semantic3D.net, S3DIS [44] is a large-scale indoor RGB-D dataset, which is also a part of the 2D-3D-S dataset [137]. It is a collection of over 215 million points, covering an area of over 6,000 m^2 in six indoor regions originating from three buildings. The main covered areas are for educational and office use. Annotations in S3DIS have been prepared at an instance level. Objects are divided into structural and movable elements, which are further classified into 13 classes (structural elements: ceiling, floor, wall, beam, column, window, door; movable elements: table, chair, sofa, bookcase, board, clutter for all other elements). The dataset can be requested from http://buildingparser.stanford.edu/dataset.html. To learn the performance of different algorithms on this dataset, readers are recommended to refer to [2], [70], [100], [119].

3) Vaihingen point cloud semantic labeling dataset: This dataset [31] has been the only published benckmark dataset in the remote sensing field in recent years. It is a collection of ALS point cloud, consisting of 10 strips captured by a Leica ALS50 system with a $45°$ field of view and 500 m mean flying height over Vaihingen, Germany. The average overlap between two neighboring strips is around 30% and the median point density is 6.7 $points/m^2$ [31]. This dataset had no label at a point level at first. Niemeyer et al. [87] first used it for a PCSS test and labeled points in three areas. Now the labeled point cloud is divided into nine classes as an algorithm evaluation standard. Although this dataset has significantly fewer points compared with semantic3D.net and S3DIS, it is an influential ALS dataset for remote sensing. The dataset can be requested from http://www2.isprs.org/commissions/comm3/wg4/3d-semantic-labeling.html.

4) Paris-Lille-3D: The Paris-Lille-3D [36] is a brand new benchmark for PCSS, as it was published in 2018. It is an MLS point cloud dataset with more than 140 million labelled points, including 50 different urban object classes along 2 km of streets in two French cities, Paris and Lille. As an MLS dataset, it also could be used for autonomous vehicles. As this is a recent dataset, only a few validated results are shown on the related website. This dataset is available at http://npm3d.fr/paris-lille-3d.

5) ScanNet: ScanNet [43] is an instance-level indoor RGB-D dataset that includes both 2D and 3D data. In contrast to the benchmarks mentioned above, ScanNet is a collection of labeled voxels rather than

points or objects. Up to now, ScanNet v2, the newest version of ScanNet, has collected 1513 annotated scans with an approximate 90% surface coverage. In the semantic segmentation task, this dataset is marked in 20 classes of annotated 3D voxelized objects. Each class corresponds to one category of furniture. This dataset can be requested from http://www.scan-net.org/index#code-and-data. To learn the performance of different algorithms on this dataset, readers are recommended to refer to [70], [120], [123], [124].

III. POINT CLOUD SEGMENTATION TECHNIQUES

PCS algorithms are mainly based on strict hand-crafted features from geometric constraints and statistical rules. The main process of PCS aims at grouping raw 3D points into non overlapping regions. Those regions correspond to specific structures or objects in one scene. Since no supervised prior knowledge is required in such a segmentation procedure, the delivered results have no strong semantic information. Those approaches could be categorized into four major groups: edge-based, region growing, model fitting, and clustering-based.

A. Edge-based

Edge-based PCS approaches were directly transferred from 2D images to 3D point clouds, which were mainly used in the very early stage of PCS. As the shapes of objects are described by edges, PCS can be solved by finding the points that are close to the edge regions. The principle of edge-based methods is to locate the points that have a rapid change in intensity [16], which is similar to some 2D image segmentation approaches.

According to the definition from [138], an edge-based segmentation algorithm is formed by two main stages: (1) edge detection, where the boundaries of different regions are extracted, and (2) grouping points, where the final segments are generated by grouping points inside the boundaries from (1). For example, in [139], the authors designed a gradient-based algorithm for edge detection, fitting 3D lines to a set of points and detecting changes in the direction of unit normal vectors on the surface. In [140], the authors proposed a fast segmentation approach based on high-level segmentation primitives (curve segments), in

which the amount of data could be significantly reduced. Compared to the method presented in [139], this algorithm is both accurate and efficient, but it is only suitable for range images, and may not work for uneven-density point clouds. Moreover, paper [141] extracted close contours from a binary edge map for fast segmentation. Paper [142] introduced a parallel edge-based segmentation algorithm extracting three types of edges. An algorithm optimization mechanism, named reconfigurable multiRing network, was applied in this algorithm to reduce its runtime.

The edge-based algorithms enable a fast PCS due to its simplicity, but their good performance can only be maintained when simple scenes with ideal points are provided (e.g., low noise, even density). Some of them are only suitable for range images rather than 3D points. Thus this approach is rarely applied for dense and/or large-area point cloud datasets nowadays. Besides, in 3D space, such methods often deliver disconnected edges, which cannot be used to identify closed segments directly, without a filling or interpretation procedure [17], [143].

B. Region growing

Region growing is a classical PCS method, which is still widely used today. It uses criteria, combining features between two points or two region units in order to measure the similarities among pixels (2D), points (3D), or voxels (3D), and merge them together if they are spatially close and have similar surface properties. Besl and Jain [144] introduced a two-step initial algorithm: (1) coarse segmentation, in which seed pixels are selected based on the mean and Gaussian curvature of each point and its sign; and (2) region growing, in which interactive region growing is used to refine the result of step (1) based on a variable order bivariate surface fitting. Initially, this method was primarily used in 2D segmentation. As in the early stage of PCS research most point clouds were actually 2.5D airborne LiDAR data, in which only one layer has a view in the z direction, the general preprocessing step was to transform points from 3D space into a 2D raster domain [145]. With the more easily available real 3D point clouds, region growing was soon adopted directly in 3D space. This 3D region growing technique has been widely applied in the segmentation of building plane structures [75], [93], [94], [101], [104].

Similar to the 2D case, 3D region growing comprises two steps: (1) select seed points or seed units; and (2) region growing, driven by certain principles. To design a region growing algorithm, three crucial factors should be taken into consideration: criteria (similarity measures), growth unit, and seed point selection. For the **criteria** factor, geometric features, e.g., Euclidean distance or normal vectors, are commonly used. For example, Ning et al. [106] employed the normal vector as criterion, so that the coplanar may share the same normal orientation. Tovari et al. [146] applied normal vectors, the distance of the neighboring points to the adjusting plane, and the distance between the current point and candidate points as the criteria for merging a point to a seed region that was randomly picked from the dataset after manually filtering areas near edges. Dong et al. [104] chose normal vectors and the distance between two units.

For **growth unit** factor, there are usually three strategies: (1) single points, (2) region units, e.g., voxel grids and octree structures, and (3) hybrid units. Selecting single points as region units was the main approach in the early stages [106], [138]. However, for massive point clouds, point-wise calculation is time-consuming. To reduce the data volume of the raw point cloud and improve calculation efficiency, e.g., neighborhood search with a k-d tree in raw data [147], the region unit is an alternative idea of direct points in 3D region growing. In a point cloud scene, the number of voxelized units is smaller than the number of points. In this way, the region growing process can be accelerated significantly. Guided by this strategy, Deschaud et al. [147] presented a voxel-based region growing algorithm to improve efficiency by replacing points with voxels during the region growing procedure. Vo et al. [93] proposed an adaptive octree-based region growing algorithm for fast surface patch segmentation by incrementally grouping adjacent voxels with a similar saliency feature. As a balance of accuracy and efficiency, hybrid units were also proposed and tested by several studies. For example, Xiao et al. [101] combined single points with subwindows as growth units to detect planes. Dong et al. [104] utilized a hybrid region growing algorithm, based on units of both single points and supervoxels, to realize coarse segmentation before global energy optimization.

For **Seed point selection**, since many region growing algorithms aim at plane segmentation, a usual

practice is designing a fitting plane for a certain point and its neighbor points first, and then choosing the point with minimum residual to the fitting plane as a seed point [106], [138]. The residual is usually estimated by the distance between one point and its fitting plane [106], [138] or the curvature of the point [94], [104].

Nonuniversality is a nontrivial problem for region growing [93]. The accuracy of these algorithms relies on the growth criteria and locations of the seeds, which should be predefined and adjusted for different datasets. In addition, these algorithms are computationally intensive and may require a reduction in data volume for a trade-off between accuracy and efficiency.

C. Model fitting

The core idea of model fitting is matching the point clouds to different primitive geometric shapes, thus it has been normally regarded as a shape detection or extraction method. However, when dealing with scenes with parameter geometric shapes/models, e.g., planes, spheres, and cylinders, model fitting can also be regarded as a segmentation approach. Most widely used model-fitting methods are built on two classical algorithms, Hough Transform (HT) and RANdom SAmple Consensus (RANSAC).

1) HT: HT is a classical feature detection technique in digital image processing. It was initially presented in [148] for line detection in 2D images. There are three main steps in HT [149]: (1) mapping every sample (e.g., pixels in 2D images and points in point clouds) of the original space into a discretized parameter space; (2) laying an accumulator with a cell array on the parameter space and then, for each input sample, casting a vote for the basic geometric element of which they are inliers in the parameter space; and (3) selecting the cell with the local maximal score, of which parameter coordinates are used to represent a geometric segment in original space. The most basic version of HT is Generalized Hough Transform (GHT), also called the Standard Hough Transform (SHT), which is introduced in [150]. GHT uses an angle-radius parameterization instead of the original slope-intercept form, in order to avoid the

infinite slope problem and simplify the computation. The GHT is based on:

$$\rho = x\cos(\theta) + y\sin(\theta) \tag{1}$$

where x and y are the image coordinates of a corresponding sample pixel, ρ is the distance between the origin and the line through the corresponding pixel, and θ is the angle between the normal of the above-mentioned line and the x-axis. Angle-radius parameterization can also be extended into 3D space, and thus can be used in 3D feature detection and regular geometric structure segmentation. Compared with the 2D form, in 3D space, there is one more angle parameter, ϕ:

$$\rho = x\cos(\theta)\sin(\phi) + y\sin(\theta)\sin(\phi) + z\cos(\phi) \tag{2}$$

where x, y, and z are corresponding coordinates of a 3D sample (e.g., one specific point from the whole point cloud), and θ and ϕ are polar coordinates of the normal vector of the plane, which includes the 3D sample.

One of the major disadvantages of GHT is the lack of boundaries in the parameter space, which leads to high memory consumption and long calculation time [151]. Therefore, some studies have been conducted to improve the performance of HT by reducing the cost of the voting process [71]. Such algorithms include Probabilistic Hough transform (PHT) [152], Adaptive probabilistic Hough transform (APHT) [153], Progressive Probabilistic Hough Transform (PPHT) [154], Randomized Hough Transform (RHT) [149], and Kernel-based Hough Transform (KHT) [155]. In addition to computational costs, choosing a proper accumulator representation is also a way to optimize HT performance [114].

Several review articles involving 3D HT are available [71], [114], [151]. As with region growing in the 3D field, planes are the most frequent research objects in HT-based segmentation [71], [74], [115], [156]. In addition to planes, other basic geometric primitives can also be segmented by HT. For example, Rabbani et al. [129] used a Hough-based method to detect cylinders in point clouds, similar to plane detection. In addition, a comprehensive introduction to sphere recognition based on HT methods is presented in [157].

To evaluate different HT algorithms on point clouds, Borrmann et al. [114] compared improved HT algorithms and concluded that RHT was the best one for PCS at that time, due to its high efficiency. Limberger et al. [71] extended KHT [155] to 3D space, and proved that 3D KHT performed better than previous HT techniques, including RHT, for plane detection. The 3D KHT approach is also robust to noise and even to irregularly distributed samples [71].

2) RANSAC: The RANSAC technique is the other popular model fitting method [158]. Several reviews about general RANSAC-based methods have been published. Learning more about the RANSAC family and their performance is highly recommended, particularly in [159]–[161]. The RANSAC-based algorithm has two main phases: (1) generate a hypothesis from random samples (hypothesis generation), and (2) verify it to the data (hypothesis evaluation/model verification) [159], [160]. Before step (1), as in the case of HT-based methods, models have to be manually defined or selected. Depending on the structure of 3D scenes, in PCS, these are usually planes, spheres, or other geometric primitives that can be represented by algebraic formulas.

In hypothesis generation, RANSAC randomly chooses N sample points and estimates a set of model parameters using those sample points. For example, in PCS, if the given model is a plane, then $N = 3$ since 3 non-collinear points determine a plane. The plane model can be represented by:

$$aX + bY + cZ + d = 0 \tag{3}$$

where $[a, b, c, d]^T$ is the parameter set to be estimated.

In hypothesis evaluation, RANSAC chooses the most probable hypothesis from all estimated parameter sets. RANSAC uses Eq. 4 to solve the selection problem, which is regarded as an optimization problem [159]:

$$\hat{M} = \arg\min_{M}\{\sum_{d \in \mathscr{D}} Loss(Err(d; M))\} \tag{4}$$
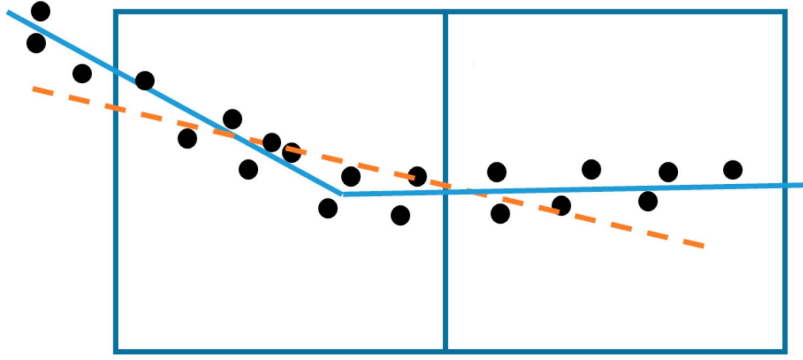
Fig. 1. An example of a spurious plane [102]. Two well-estimated hypothesis planes are shown in blue. A spurious plane (in orange) is generated using the same threshold.

where $\mathscr{D}$ is data, $Loss$ represents a loss function, and Err is an error function such as geometric distance.

As an advantage of random sampling, RANSAC-based algorithms do not require complex optimization or high memory resources. Compared to HT methods, efficiency and the percentage of successful detected objects are two main advantages for RANSAC in 3D PCS [74]. Moreover, RANSAC algorithms have the ability to process data with a high amount of noise, even outliers [162]. For PCS, as with HT and region growing, RANSAC is widely used in plane segmentation, such as building façades [65], [66], [103], building roofs [73], and indoor scenes [102]. In some fields there is demand for the segmentation of more complex structures than planes. Schnabel et al. [162] proposed an automatic RANSAC-based algorithm framework to detect basic geometric shapes in unorganized point clouds. Those shapes include not only planes, but also spheres, cylinders, cones, and tori. RANSAC-based PCS segmentation algorithms were utilized for cylinder objects in [130] and [131].

RANSAC is a nondeterministic algorithm, and thus its main shortcoming is its spurious surface: the probability exists that models detected by RANSAC-based algorithm do not exist in reality (Fig. 1). To overcome the adverse effect of RANSAC in PCS, a soft-threshold voting function was presented to improve the segmentation quality in [72], in which both the point-plane distance and the consistency between the normal vectors were taken into consideration. Li et al. [102] proposed an improved RANSAC method based on NDT cells [163], also in order to avoid spurious surface problem in 3D PCS.
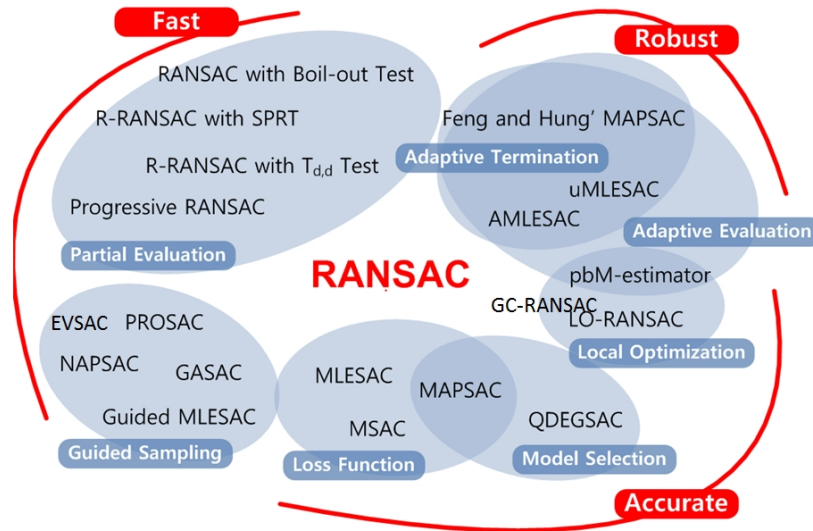
Fig. 2. RANSAC family with algorithms categorized according to their performance and basic strategies [159], [164], [165].

As with HT, many improved algorithms based on RANSAC have emerged over the past decades to further improve its efficiency, accuracy and robustness. These approaches have been categorized by their research objectives and are shown in Fig. 2. The figure has been originally described in [159], in which seven subclasses according to seven strategies are used. Venn diagrams are utilized here to describe connections between methods and strategies, since a method may use two strategies. For detail description and explanation on those strategies, please refer to [159]. Considering that [159] is obsolete, we add two recently published methods, EVSAC [164] and GC-RANSAC [165] on original figure to make it keep up with the times.

D. Unsupervised clustering-based

Clustering-based methods are widely used for unsupervised PCS task. Strictly speaking, clustering-based methods are not based on a specific mathematical theory. This methodology family is a mixture of different methods that share a similar aim, which is grouping points with similar geometric spectral features or spatial distribution into the same homogeneous pattern. Unlike region growing and model fitting, these patterns usually are not defined in advance [166], and thus clustering-based algorithms can be employed for irregular object segmentation, e.g., vegetation. Moreover, seed points are not required by

clustering-based approaches, in contrast to region growing methods [109]. In the early stage, K-means [45], [46], [76], [77], [91], mean shift [47], [48], [80], [92], and fuzzy clustering [77], [105] were the main algorithms in the clustering-based point cloud segmentation family. For each clustering approach, several similarity measures with different features can be selected, including Euclidean distance, density, and normal vector [109]. From the perspective of mathematics and statistics, the clustering problem can be regarded as a graph-based optimization problem, so several graph-based methods have been experimented in PCS [78], [79], [167].

1) K-means: K-means is a basic and widely used unsupervised cluster analysis algorithm. It separates the point cloud dataset into K unlabeled classes. The clustering centers of K-means are different than the seed points of region growing. In K-means, every point should be compared to every cluster center in each iteration step, and the cluster centers will change when absorbing a new point. The process of K-means is "clustering" rather than "growing". It has been adopted for single tree crown segmentation on ALS data [91] and planar structure extraction from roofs [76]. Shahzad et al. [45] and Zhu et al. [46] utilized K-means for building façade segmentation on TomoSAR point clouds.

One advantage of K-means is that it can be easily adapted to all kinds of feature attributes, and can even be used in a multidimensional feature space. The main drawback of K-means is that it is sometimes difficult to predefine the value of K properly.

2) Fuzzy clustering: Fuzzy clustering algorithms are improved versions of K-means. K-means is a hard clustering method, which means the weight of a sample point to a cluster center is either 1 or 0. In contrast, fuzzy methods use soft clustering, meaning a sample point can belong to several clusters with certain nonzero weights.

In PCS, a no-initialization framework was proposed in [105], by combining two fuzzy algorithms, Fuzzy C-Means (FCM) algorithm and Possibilistic C-Means (PCM). This framework was tested on three point clouds, including a one-scan TLS outdoor dataset with building structures. Those experiments showed that fuzzy clustering segmentation worked robustly on planer surfaces. Sampath et al. [77] employed fuzzy

K-means for segmentation and reconstruction of building roofs from an ALS point cloud.

3) Mean-shift: In contrast to K-means, mean-shift is a classic nonparametric clustering algorithm and hence avoids the predefined K problem in K-means [168]–[170]. It has been applied effectively on ALS data in urban and forest terrain [80], [92]. Mean-shift have also been adopted on TomoSAR point clouds, enabling building façades and single trees to be extracted [47], [48].

As both the cluster number and the shape of each cluster are unknown, mean-shift delivers with high probability oversegmented result [81]. Hence, it is usually used as a presegmentation step before partitioning or refinement.

4) Graph-based: In 2D computer vision, introducing graphs to represent data units such as pixels or superpixels has proven to be an effective strategy for the segmentation task. In this case, the segmentation problem can be transformed into a graph construction and partitioning problem. Inspired by graph-based methods from 2D, some studies have applied similar strategies in PCS and achieved results in different datasets.

For instance, Golovinskiy and Funkhouser [167] proposed a PCS algorithm based on min-cut [171], by constructing a graph using k-nearest neighbors. The min-cut was then successfully applied for outdoor urban object detection [167]. Ural et al. [78] also used min-cut to solve the energy minimization problem for ALS PCS. Each point is considered to be a node in the graph, and each node is connected to its 3D voronoi neighbors with an edge. For the roof segmentation task, Yan et al. [79] used an extended α-expansion algorithm [172] to minimize the energy function from the PCS problem. Moreover, Yao et al. [81] applied a modified normalized cut (N-cut) in their hybrid PCS method.

Markov Random Field (MRF) and Conditional Random Field (CRF) are machine learning approaches to solve graph-based segmentation problems. They are usually used as supervised methods or postprocessing stages for PCSS. Major studies using CRF and supervised MRFs belong to PCSS rather than PCS. For more information about supervised approaches, please refer to section IV-A.

E. Oversegmentation, supervoxels, and presegmentation

To reduce the calculation cost and negative effects from noise, a frequently used strategy is to over-segment a raw point cloud into small regions before applying computationally expensive algorithms. Voxels can be regarded as the simplest oversegmentation structures. Similar to superpixels in 2D images, supervoxels are small regions of perceptually similar voxels. Since supervoxels can largely reduce the data volume of a raw point cloud with low information loss and minimal overlapping, they are usually utilized in presegmentation before executing other computationally expensive algorithms. Once oversegments like supervoxels are generated, these are fed to postprocessing PCS algorithms rather than initial points.

The most classical point cloud oversegmentation algorithm is Voxel Cloud Connectivity Segmentation (VCCS) [173]. In this method, a point cloud is first voxelized by the octree. Then a K-means clustering algorithm is employed to realize supervoxel segmentation. However, since VCCS adopts fixed resolution and relies on initialization of seed points, the quality of segmentation boundaries in a non-uniform density cannot be guaranteed. To overcome this problem, Song et al. [174] proposed a two-stage supervoxel oversegmentation approach, named Boundary-Enhanced Supervoxel Segmentation (BESS). BESS preserves the shape of the object, but it also has an obvious limitation for the assumption that points are sequentially ordered in one direction. Recently, Lin et al. [175] summarized the limitations of previous studies, and formalized oversegmentation as a subset selection problem. This method adopts an adaptive resolution to preserve boundaries, a new practice in supervoxel generation. Landrieu and Boussaha [100] presented the first supervised framework for 3D point cloud oversegmentation, achieving significant improvements compared to [173], [175]. For PCS tasks, several studies have been based on supervoxel-based presegmentation [107]–[109], [176], [177].

As mentioned in section III-D, in addition to supervoxels, other methods can also be employed as presegmentation. For example, Yao et al. [81] utilized mean-shift to oversegment ALS data in urban areas.

IV. POINT CLOUD SEMANTIC SEGMENTATION TECHNIQUES

The procedure of PCSS is similar to clustering-based PCS. But in contrast to non-semantic PCS methods, PCSS techniques generate semantic information for every point, and are not limited to clustering. Therefore, PCSS is usually realized by supervised learning methods, including "regular" supervised machine learning and state-of-the-art deep learning.

A. Regular supervised machine learning

In this section, regular supervised machine learning refers to non-deep supervised learning algorithms. Comprehensive and comparative analysis on different PCSS methods based on regular supervised machine learning has been provided by previous researchers [87], [88], [95], [97].

Paper [5] pointed out that supervised machine learning applied to PCSS could be divided into two groups. One group, individual PCSS, classifies each point or each point cluster based only on its individual features, such as Maximum Likelihood classifiers based on Gaussian Mixture Models [113], Support Vector Machines [4], [111], AdaBoost [6], [82], a cascade of binary classifiers [83], Random Forests [84], and Bayesian Discriminant Classifiers [116]. The other group is statistical contextual models, such as Associative and Non-Associative Markov Networks [85], [90], [96], Conditional Random Fields [86]–[88], [110], [178], Simplified Markov Random Fields [8], multistage inference procedures focusing on point cloud statistics and relational information over different scales [89], and spatial inference machines modeling mid- and long-range dependencies inherent in the data [117].

The general procedure of the individual classification for PCSS has been well described in [95]. As Fig. 3 shows, the procedure entails four stages: neighborhood selection, feature extraction, feature selection, and semantic segmentation. For each stage, paper [95] summarized several crucial methods and tested different methods on two datasets to compare their performance. According to the authors' experiment, in individual PCSS, the Random Forest classifier had a good trade-off between accuracy and efficiency on two datasets. It should be noted that [95] used a so-called "deep learning" classifier in their experiments,
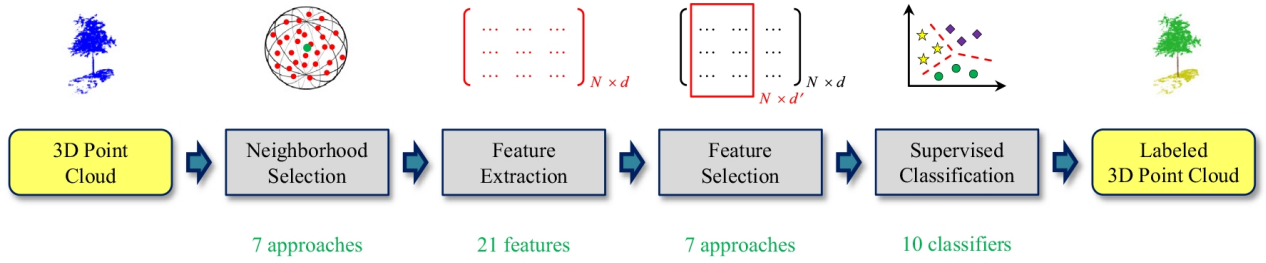
Fig. 3. The PCSS framework by [95]. The term "semantic segmentation" in our review is defined as "supervised classification" in [95].

but that is an old neural network appearing in the time of regular machine learning, not the recent deep learning methods described in section IV-B.

Since individual PCSS does not take contextual features of points into consideration, individual classifiers work efficiently but generate unavoidable noise that cause unsmooth PCSS results. Statistical context models can mitigate this problem. Conditional Random Fields (CRF) is the most widely used context model in PCSS. Niemeyer et al. [87] provided a very clear introduction about how CRF has been used on PCSS, and tested several CRF-based approaches on the Vaihingen dataset. Based on the individual PCSS framework [95], Landrieu et al. [97] proposed a new PCSS framework that combines individual classification and context classification. As shown in Fig. 4, in this framework a graph-based contextual strategy was introduced to overcome the noise problem of initial labeling, from which the process was named structured regularization or "smoothing".

For the regularization process, Li et al. [111] utilized a multilabel graph-cut algorithm to optimize the initial segmentation result from Support Vector Machine (SVM). Landrieu et al. [97] compared various postprocess methods in their studies, which proved that regularization indeed improved the accuracy of PCSS.

B. Deep learning

Deep learning is the most influential and fastest-growing current technique in pattern recognition, computer vision, and data analysis [179]. As its name indicates, deep learning uses more than two hidden layers to obtain high-dimension features from training data, while traditional handcrafted features are
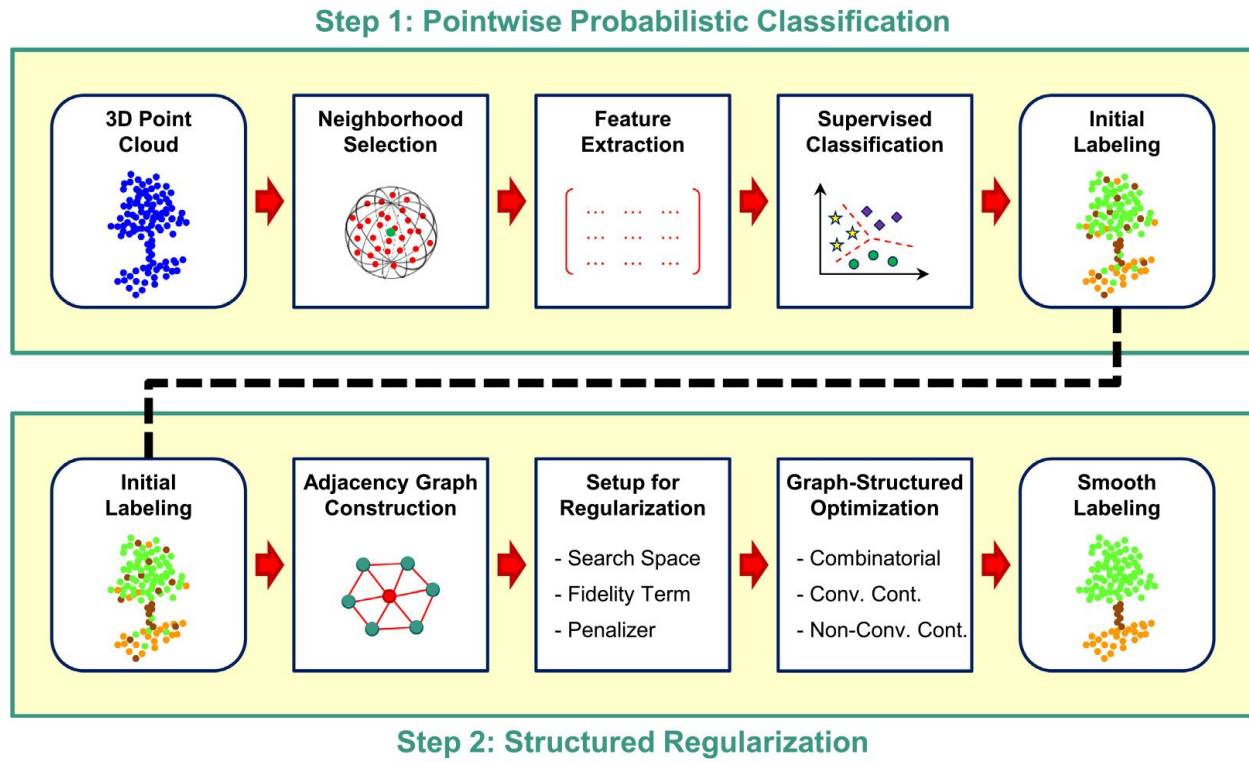
Step 1: Pointwise Probabilistic Classification

Step 2: Structured Regularization

Fig. 4. The PCSS framework by [97]. The term "semantic segmentation" in our review is defined as "supervised classification" in [97].

designed with domain-specific knowledge. Before being applied in 3D data, deep learning appeared as an effective power in a variety of tasks in 2D computer vision and image processing, such as image recognition [180], [181], object detection [182], [183], and semantic segmentation [184], [185]. It has been attracting more interest in 3D analysis since 2015, driven by the multiview-based idea proposed by [186], and voxel-based 3D Convolutional Neural Network (CNN) by [187].

Standard convolutions originally designed for raster images cannot easily be directly applied to PCSS, as the point cloud is unordered and unstructured/irregular/non-raster. Thus, in order to solve this problem, a transformation of the raw point cloud becomes essential. Depending on the format of the data ingested into neural networks, deep learning-based PCSS approaches can be divided into three categories: multiview-based, voxel-based, and point-based, respectively.

1) Multiview-based: One of the early solutions to applying deep learning in 3D is dimensionality reduction. In short, the 3D data is represented by multi-view 2D images, which can be processed based on 2D CNNs. Subsequently, the classification results can be restored into 3D. The most influential multi-
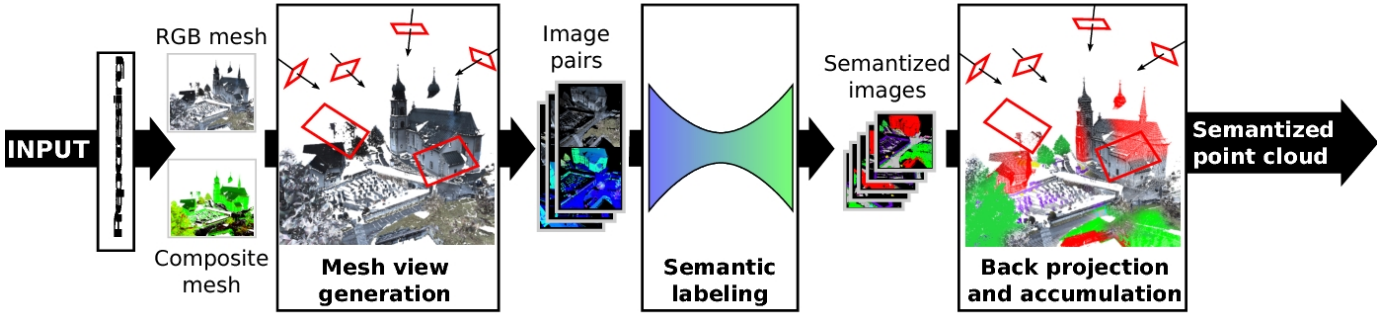
Fig. 5. The Workflow of SnapNet [67].

view deep learning in 3D analysis is MVCNN [186]. Although the original MVCNN algorithm did not experiment on PCSS, it is a good example for learning about the multiview concept.

The multiview-based methods have solved the structuring problems of point cloud data well, but there are two serious shortcomings in these methods. Firstly, they cause numerous limitations and a loss in geometric structures, as 2D multiview images are just an approximation of 3D scenes. As a result, complex tasks such as PCSS could yield limited and unsatisfactory performances. Secondly, multiview projected images must cover all spaces containing points. For large, complex scenes, it is difficult to choose enough proper viewpoints for multiview projection. Thus, few studies used multiview-based deep learning architecture for PCSS. One exception is SnapNet [9], [67], which uses full dataset semantic-8 of semantic3D.net as the test dataset. Fig. 5 shows the workflow of SnapNet. In SnapNet, the preprocessing step aims at decimating the point cloud, computing point features and generatinga mesh. Snap generation is to generate RGB images and depth composite images of the mesh, based on various virtual cameras. Semantic labeling is to realize image semantic segmentation from the two input images, by image deep learning. The last step is to project 2D semantic segmentation results back to 3D space, thereby 3D semantics can be acquired.

2) Voxel-based: Combining voxels with 3D CNNs is the other early approach in deep learning-based PCSS. Voxelization solves both unordered and unstructured problems of the raw point cloud. Voxelized data can be further processed by 3D convolutions, as in the case of pixels in 2D neural networks.

Voxel-based architectures still have serious shortcomings. In comparison to the point cloud, the voxel
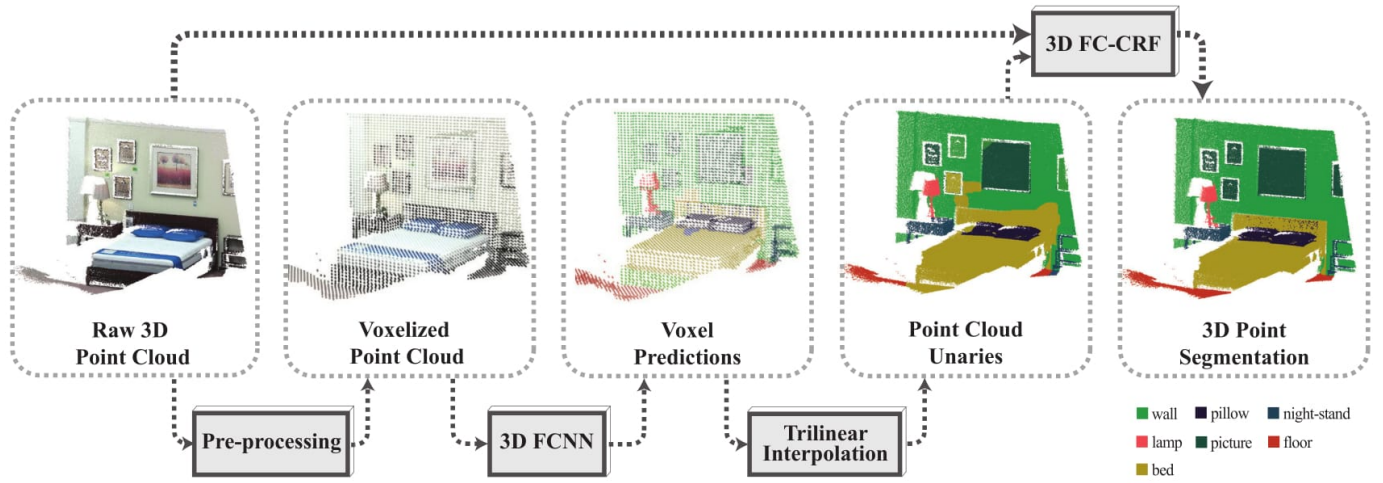
Fig. 6. The Workflow of SegCloud [98].

structure is a low-resolution form. Obviously, there is a loss in data representation. In addition, voxel structures not only store occupied spaces, but also store free or unknown spaces, which can result in high computational and memory requirements.

The most well-known voxel-based 3D CNN is VoxNet [187], but this was only tested for object detection. On the PCSS task, some papers, like [69], [98], [188] and [189], proposed representative frameworks. SegCloud [98] is an end-to-end PCSS framework that combines 3D-FCNN, trilinear interpolation (TI), and fully connected Conditional Random Fields (FC-CRF) to accomplish the PCSS task. Fig. 6 shows the framework of SegCloud, which also provides a basic pipeline of voxel-based semantic segmentation. In SegCloud, the preprocessing step is to voxelize raw point clouds. Then a 3D fully convolutional neural netwotk is applied to generate downsampled voxel labels. After that, a trilinear interpolation layer is employed to transfer voxel labels back to 3D point labels. Finally, a 3D fully connected CRF method is utilized to regularize previous 3D PCSS results, and acquire final results. SegCloud used to be the state-of-art approach in both S3DIS and semantic3D.net, but it did not take any steps to optimize high computational and memory problem from fixed-sized voxels. With more advanced methods springing up, SegCloud has fallen from favor in recent years.

To reduce unnecessary computation and memory consumption, the flexible octree structure is an effective

replacement for fixed-size voxels in 3D CNNs. OctNet [69] and O-CNN [188] are two representative approaches. Recently, VV-NET [189] extended the use of voxels. VV-Net utilized a radial basis function-based Variational Auto-Encoder (VAE) network, which provided a more information-rich representation for point cloud compared with fixed-size voxels.

3) Directly process point cloud data: As there are serious limitations in both multiview- and voxel-based methods (e.g., loss in structure resolution), exploring PCSS methods directly on point is a natural choice. Up to now, many approaches have emerged and are still emerging [1]–[3], [119], [120]. Unlike employing separated pretransformation operation in multiview-based and voxel-based cases, in these approaches the canonicalization is binding with the neural network architecture.

PointNet [1] is a pioneering deep learning framework which has been performed directly on point. Different with recently published point cloud networks, there is no convolution operator in PointNet. The basic principle of PointNet is:

$$f(\{x_1, ..., x_n\}) \approx g(h(x_1), ..., h(x_n)) \tag{5}$$

where $f : 2^{\mathbb{R}^N} \to \mathbb{R}$ and $h : \mathbb{R}^N \to \mathbb{R}^K$. $g : \underbrace{\mathbb{R}^K \times ... \times \mathbb{R}^K}_{n} \to \mathbb{R}$ is a symmetric function, used to solve the ordering problem of point clouds. As Fig. 7 shows, PointNet uses MultiLayer Perceptrons (MLPs) to approximate h, which represents the per-point local features corresponding to each point. The global features of point sets g are aggregated by all per-point local features in a set, through a symmetric function, max pooling. For the classification task, output scores for k classes can be produced by a MLP operation on global features. For the PCSS task, in addition to global features, per-point local features are demanded. PointNet concatenates aggregated global features and per-point local features into combined point features. Subsequently, new per-point features are extracted from the combined point features by MLPs. On their basis, semantic labels are predicted.

Although more and more newly published networks outperform PointNet on various benchmark datasets, PointNet is still a baseline for PCSS research. The original PointNet uses no local structure information
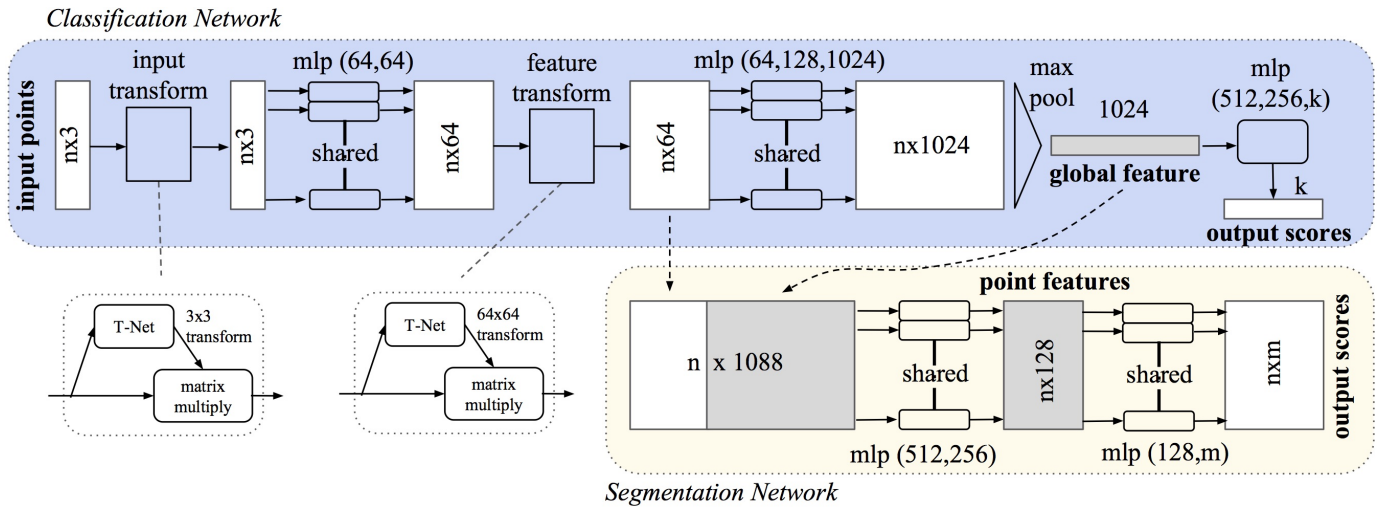
Fig. 7. The Workflow of PointNet [1]. In this figure, "Classification Network" is used for object classification. "Segmentation Network" is applied for the PCSS mission.

within neighboring points. In a further study, Qi et al. [120] used a hierarchical neural network to capture local geometric features to improve the basic PointNet model and proposed PointNet++. Drawing inspiration from PointNet/PointNet++, studies on 3D deep learning focus on feature augmentation, especially to local features/relationships among points, utilizing knowledge from other fields to improve the performance of the basic PointNet/PointNet++ algorithms. For example, Engelmann et al. [190] employed two extensions on the PointNet to incorporate larger-scale spatial context. Wang et al. [3] considered that missing local features was still a problem in PointNet++, since it neglected the geometric relationships between a single point and its neighbors. To overcome this problem, Wang et al. [3] proposed Dynamic Graph CNN (DGCNN). In this network, the authors designed a procedure called EdgeConv to extract edge features while maintaining permutation invariance. Inspired by the idea of the attention mechanism, Wang et al. [112] designed a Graph Attention Convolution (GAC), of which kernels could be dynamically adapted to the structure of an object. GAC can capture the structural features of point clouds while avoiding feature contamination between objects. To exploit richer edge features, Landrieu and Simonovsky [2] introduced the SuperPoint Graph (SPG), offering both compact and rich representation of contextual relationships among object parts rather than points. The partition of the superpoint can be regarded as a nonsemantic presegmentation and downsampling step. After SPG construction, each superpoint is

embedded in a basic PointNet network and then refined in Gated Recurrent Units (GRUs) for PCSS. Benefiting from information-rich downsampling, SPG is highly efficient for large-volume datasets.

Also in order to overcome the drawback of no local features represented by neighboring points in PointNet, 3P-RNN [99] adopted a Pointwise Pyramid Pooling (3P) module to capture the local feature of each point. In addition, it employed a two-direction Recurrent Neural Network (RNN) model to integrate long-range context in PCSS tasks. The 3P-RNN technique has increased overall accuracy at a negligible extra overhead. Komarichev et al. [125] introduced an annular convolution, which could capture the local neighborhood by specifying the ring-shaped structures and directions in the computation, and adapt to the geometric variabil1ity and scalability at the signal processing level. Due to the fact that the K-nearest neighbor search in PointNet++ may lead to the K neighbors falling in one orientation, Jiang et al. [121] designed PointSIFT to capture local features from eight orientations. In the whole architecture, the PointSIFT module achieves multiscale representation by stacking several Orientation-Encoding (OE) units. The PointSIFT module can be integrated into all kinds of PointNet-based 3D deep learning architectures to improve the representational ability for 3D shapes. Built upon PointNet++, PointWeb [126] utilized the Adaptive Feature Adjustment (AFA) module to find the interaction between points. The aim of AFA is also to capture and aggregate local features of points.

Besides, based on PointNet/PointNet++, instance segmentation can also be realized, even accompanied by PCSS. For instance, Wang et al. [127] presented the Similarity Group Proposal Network (SGPN). SGPN is the first published point cloud instance segmentation framework. Yi et al. [128] presented a Region-based PointNet (R-PointNet). The core module of R-PointNet is named as Generative Shape Proposal Network (GSPN), of which the base is PointNet. Pham et al. [124] applied a Multi-task Pointwise Network (MT-PNet) and a Multi-Value Conditional Random Field (MV-CRF) to address PCSS and instance segmentation simultaneously. MV-CRF jointly realized the optimization of semantics and instances. Wang et al. [123] proposed an Associatively Segmenting Instances and Semantics (ASIS) module, making PCSS and instance segmentation take advantage of each other, leading to a win-win situation. In [123], the backbone that

networks employed are also PointNet and PointNet++.

An increasing number of researchers have chosen an alternative to PointNet, employing the convolution as a fundamental and significant component, with their deeper understanding on point-based learning. Some of them, like [3], [112], [125], have been introduced above. In addition, PointCNN used a $\mathcal{X}$-transformation instead of symmetric functions to canonicalize the order [119], which is a generalization of CNNs to feature learning from unorderd and unstructured point clouds. Su et al. [68] provided a PCSS framework that could fuse 2D images with 3D point clouds, named SParse LATtice Networks (SPLATNet), preserving spatial information even in sparse regions. Recurrent Slice Networks (RSN) [118] exploited a sequence of multiple 1×1 convolution layers for feature learning, and a slice pooling layer to solve the unordered problem of raw point clouds. A RNN model was then applied on ordered sequences for the local dependency modeling. Te et al. [191] proposed Regularized Graph CNN (RGCNN) and tested it on a part segmentation dataset, ShapeNet [192]. Experiments show that RGCNN can reduce computational complexity and is robust to low density and noise. Regarding convolution kernels as nonlinear functions of the local coordinates of 3D points comprised of weight and density functions, Wu et al. [122] presented PointConv. PointConv is an extension to the Monte Carlo approximation of the 3D continuous convolution operator. PCSS is realized by a deconvolution version of PointConv. What is more, Choy et al. [70] proposed 4-dimensional convolutional neural networks (MinkowskiNets) to process 3D-videos, which are a series of CNNs for high-dimensional spaces including the 4D spatio-temporal data. MinkowskiNets can also be applied on 3D PCSS tasks. They have achieved good performance on a series of PCSS benchmark datasets, especially a significant accuracy improvement on ScanNet [43].

As SPG [2], DGCNN [3], RGCNN [191] and GAC [112] employed graph structures in neural networks, they can also be regarded as Graph Neural Networks (GNNs) in 3D [193], [194].

The research on PCSS based on deep learning is still ongoing. New ideas and approaches on the topic of 3D deep learning-based frameworks are keeping popping up. Current achievements have proved that it is a great boost for the accuracy of 3D PCSS.

C. Hybrid methods

In PCSS, hybrid segment-wise methods have been attracting researchers' attention in recent years. A hybrid approach is usually made up of at least two stages: (1) utilize an oversegmentation or PCS algorithm (introduced in section III as the presegmentation), and (2) apply PCSS on segments from (1) rather than points. In general, as with presegmentation in PCS, presegmentation in PCSS also has two main functions: to reduce the data volume and to conduct local features. Oversegmentation for supervoxels is a kind of presegmentation algorithm in PCSS [110], since it is an effective way to reduce the data volume with light accuracy loss. In addition, because nonsemantic PCS methods can provide rich natural local features, some PCSS studies also use them as presegmentation. For example, Zhang et al. [4] employed region growing before SVM. Vosselman et al. [88] applied HT to generate planar patches in their PCSS algorithm framework as the presegmentation. In deep learning, Landrieu and Simonovsky [2] exploited a superpoint graph structure as the presegmentation step, and provided a contextual PCSS network combining superpoint graphs with PointNet and contextual segmentation. Landrieu and Boussaha [100] used a supervised algorithm to realize the presegmentation, which is the first supervised framework for 3D point cloud oversegmentation.

V. DISCUSSION

A. Open issues in segmentation techniques

1) Features: One of the core questions in pattern recognition is how to obtain effective features. Essentially, the biggest differences among the various methods in PCSS or PCS are the differences of feature design, selection, and application. Feature selection is a trade-off between algorithm accuracy and efficiency. Focusing on PCSS, Weinmann et al. [95] analyzed features from three aspects: neighborhood selection (fixed or individual); feature extraction (single-scale or multi-scale); and classifier selection (individual classifier or contextual classifier). Deep learning-based algorithms face similar problems. The local feature is the most significant aspect to be improved after the birth of PointNet [1].

Even in a PCS task, different methods also show different understandings of features. Model fitting is actually searching for a group of points connected with certain geometric primitives, which also can be defined as features. For this reason, deep learning has been introduced into model fitting recently [195]. The criteria or the similarity measure in region growing or clustering is the feature of a point essentially. The improvement of an algorithm reflects its ability to more strongly capture features.

2) Hybrid: As mentioned in section IV-C, hybrid is a strategy for PCSS. Presegmentation can provide local features in a natural way. Once the development of neural network architectures stabilizes, nonsemantic presegmentation might become a progressive course for PCCS.

3) Contextual information: In PCSS tasks, contextual models are crucial tools for regular supervised machine learning, widely exploited as a smoothing postprocessing step. In deep learning, several methods, like [98], [2], [124] and [70], have employed contextual segmentation, but there is still room for further improvements.

4) PCSS with GNNs: GNN is becoming increasingly popular in 2D image processing [193], [194]. For PCSS tasks, its excellent performance has been shown in [2], [3], [191] and [112]. Similar to contextual models, the GNN might also have some surprises for PCSS. But more research is required in order to evaluate its performance.

5) Regular machine learning vs. deep learning: Before deep learning emerged, regular machine learning was the choice of supervised PCSS. Deep learning has changed the way a point cloud is handled. Compared with regular machine learning, deep learning has notable advantages: (1) it is more efficient at handling large-volume datasets; (2) there is no need to handcraft feature design and selection, a difficult task in regular machine learning; and (3) it yields high ranks (high-accuracy results) on public benchmark datasets. Nevertheless, deep learning is not a universal solution. Firstly, its principal shortcoming is poor interpretability. Currently, it is well known how each type of layers (e.g., convolution, pooling) works in a neural network. In pioneering PCSS works, such knowledge has been used to develop a series of functional networks [1], [119], [122]. However, a detailed internal decision-making process for deep learning is not

yet understood, and therefore cannot be fully described. As a result, certain fields demanding high-level safety or stability cannot trust deep learning completely. A typical example that is relevant to PCSS is autonomous driving. Secondly, data limit the application of deep learning-based PCSS. Compared with annotating 2D images, acquiring and annotating a point cloud is much more complicated. Finally, although current public datasets provide several indoor and outdoor scenes, they cannot meet the demand in real applications sufficiently.

B. Remote sensing meets computer vision

Remote sensing and general computer vision might be two of the most active groups interested in point clouds, having published many pioneering studies. The main difference between these two groups is that computer vision focuses on new algorithms to further improve the accuracy of the results. Remote sensing researchers, on the other hand, are trying to apply these techniques on different types of datasets. However, in many cases the algorithms proposed by computer vision studies cannot be adopted in remote sensing directly.

1) Evaluation system: In generic computer vision, in order to evaluate the accuracy, the overall accuracy is a significant index. However, some remote sensing applications care more about the accuracy of certain objects. For instance, for urban monitoring the accuracy of buildings is crucial, while the segmentation or the semantic segmentation of other objects is less important. Thus, compared to computer vision, remote sensing needs a different evaluation system for selecting proper algorithms.

2) Multi-source Data: As discussed in section II, point clouds in remote sensing and computer vision appear differently. For example, airborne/spaceborne 2.5D and/or sparse point clouds are also crucial components of remote sensing data, while computer vision focuses on denser full 3D.

3) Remote sensing algorithms: Published computer vision algorithms are usually tested on a small-area dataset with limited categories of objects. However, for remote sensing applications, large-area data with more complex and specific ground object categories are demanded. For example, in agricultural remote

sensing, vegetation is expected to be separated into certain specific species, which is difficult for current computer vision algorithms to solve.

4) Noise and outliers: Current computer vision algorithms do not pay much attention to noise, while in remote sensing, sensor noise is unavoidable. Currently, noise adaptive algorithms are unavailable.

C. Limitation of public benchmark datasets

In section II-D, several popular benchmark datasets are listed. Obviously, in comparison to the situation several years ago, the number of large-scale datasets with dense point clouds and rich information available to researchers has increased considerably. Some datasets, such as semantic3D.net and S3DIS, have hundreds of millions of points. However, those benchmark datasets are still insufficient for PCSS tasks.

1) Limited data types: Despite the fact that several large datasets for PCSS are available, there is still demand for more varied data. In the real world, there are much more object categories than the ones considered in current benchmark datasets. For example, semantic3D.net provides a large-scale urban point cloud benchmark. However, it only covers one kind of cities. If researchers chose a different city for a PCSS task, in which building styles, vegetation species, and even ground object types would differ, algorithm results might in turn be different.

2) Limited data sources: Most mainstream point cloud benchmark datasets are acquired from either LiDAR or RGB-D. But in practical applications, image-derived point clouds cannot be ignored. As previously mentioned, in remote sensing the airborne 2.5D point cloud is an important category, but for PCSS tasks only the Vaihingen dataset [31], [87] is published as a benchmark dataset. New data types, such as satellite photogrammetric point clouds, InSAR point clouds, and even multi-source fusion data, are also necessary to establish corresponding baselines and standards.

VI. CONCLUSION

This paper provided a review of current PCSS and PCS techniques. This review not only summarizes the main categories of relevant algorithms, but also briefly introduces the acquisition methodology and evolution of point clouds. In addition, the advanced deep learning methods that have been proposed in recent years are compared and discussed. Due to the complexity of the point cloud, PCSS is more challenging than 2D semantic segmentation. Although many approaches are available, they have each been tested on very limited and dissimilar datasets, so it is difficult to select the optimal approach for practical applications. Deep learning-based methods have ranked high for most of the benchmark-based evaluations, yet there is no standard neural network publicly available. Improved neural networks for the solution of PCSS problems can be expected to be designed in coming years.

Most current methods have only considered point features, but in practical applications such as remote sensing the noise and outliers are still problems that cannot be avoided. Improving the robustness of current approaches, and combining initial point-based algorithms with different sensor theories to denoise the data are two potential future fields of research for semantic segmentation.

REFERENCES

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017.

[2] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567, 2018.

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *arXiv preprint arXiv:1801.07829*, 2018.

[4] J. Zhang, X. Lin, and X. Ning, "Svm-based classification of segmented airborne lidar point clouds in urban areas," *Remote Sensing*, vol. 5, no. 8, pp. 3749–3775, 2013.

[5] M. Weinmann, A. Schmidt, C. Mallet, S. Hinz, F. Rottensteiner, and B. Jutzi, "Contextual classification of point cloud data by exploiting individual 3d neigbourhoods," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3 (2015), Nr. W4*, vol. 2, no. W4, pp. 271–278, 2015.

[6] Z. Wang, L. Zhang, T. Fang, P. T. Mathiopoulos, X. Tong, H. Qu, Z. Xiao, F. Li, and D. Chen, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2409–2425, 2015.

[7] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *Advances in neural information processing systems*, pp. 244–252, 2011.

[8] Y. Lu and C. Rasmussen, "Simplified markov random fields for efficient semantic labeling of 3d point clouds," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2690–2697, IEEE, 2012.

[9] A. Boulch, B. Le Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks.," in *3DOR*, 2017.

[10] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques," *Automation in construction*, vol. 19, no. 7, pp. 829–843, 2010.

[11] R. Volk, J. Stengel, and F. Schultmann, "Building information modeling (bim) for existing buildingsliterature review and future needs," *Automation in construction*, vol. 38, pp. 109–127, 2014.

[12] K. Lim, P. Treitz, M. Wulder, B. St-Onge, and M. Flood, "Lidar remote sensing of forest structure," *Progress in physical geography*, vol. 27, no. 1, pp. 88–106, 2003.

[13] L. Wallace, A. Lucieer, C. Watson, and D. Turner, "Development of a uav-lidar system with application to forest inventory," *Remote Sensing*, vol. 4, no. 6, pp. 1519–1543, 2012.

[14] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[15] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

[16] A. Nguyen and B. Le, "3d point cloud segmentation: A survey," in *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*, pp. 225–230, IEEE, 2013.

[17] E. Grilli, F. Menna, and F. Remondino, "A review of point clouds segmentation and classification algorithms," in *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 339, 2017.

[18] E. P. Baltsavias, "A comparison between photogrammetry and laser scanning," *ISPRS Journal of photogrammetry and Remote Sensing*, vol. 54, no. 2-3, pp. 83–94, 1999.

[19] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. Reynolds, "structure-from-motionphotogrammetry: A low-cost, effective tool for geoscience applications," *Geomorphology*, vol. 179, pp. 300–314, 2012.

[20] E. M. Mikhail, J. S. Bethel, and J. C. McGlone, "Introduction to modern photogrammetry," *New York*, 2001.

[21] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 807–814, 2005.

[22] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[23] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[24] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[25] F. Nex and F. Remondino, "Uav for 3d mapping applications: a review," *Applied geomatics*, vol. 6, no. 1, pp. 1–15, 2014.

[26] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM transactions on graphics (TOG)*, vol. 25, pp. 835–846, ACM, 2006.

[27] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *International journal of computer vision*, vol. 80, no. 2, pp. 189–210, 2008.

[28] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1625–1632, 2013.

[29] J. Shan and C. K. Toth, *Topographic laser ranging and scanning: principles and processing*. CRC press, 2018.

[30] R. Qin, J. Tian, and P. Reinartz, "3d change detection–approaches and applications," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 122, pp. 41–56, 2016.

[31] F. Rottensteiner, G. Sohn, M. Gerke, and J. D. Wegner, "Isprs test project on urban classification and 3d building reconstruction," *Commission III-Photogrammetric Computer Vision and Image Analysis, Working Group III/4-3D Scene Analysis*, pp. 1–17, 2013.

[32] F. Morsdorf, C. Nichol, T. Malthus, and I. H. Woodhouse, "Assessing forest structural and physiological information content of multi-spectral lidar waveforms by radiative transfer modelling," *Remote Sensing of Environment*, vol. 113, no. 10, pp. 2152–2163, 2009.

[33] A. Wallace, C. Nichol, and I. Woodhouse, "Recovery of forest canopy parameters by inversion of multispectral lidar data," *Remote*

Sensing, vol. 4, no. 2, pp. 509–531, 2012.

[34] T. Hackel, N. Savinov, L. Ladicky, J. Wegner, K. Schindler, and M. Pollefeys, "Semantic3d. net: a new large-scale point cloud classification benchmark," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 91–98, 2017.

[35] M. Brédif, B. Vallet, A. Serna, B. Marcotegui, and N. Paparoditis, "Terramobilita/iqmulus urban point cloud classification benchmark," in *Workshop on Processing Large Geospatial Data*, 2014.

[36] X. Roynard, J.-E. Deschaud, and F. Goulette, "Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification," *The International Journal of Robotics Research*, vol. 37, no. 6, pp. 545–557, 2018.

[37] T. Sankey, J. Donager, J. McVay, and J. B. Sankey, "Uav lidar and hyperspectral fusion for forest monitoring in the southwestern usa," *Remote Sensing of Environment*, vol. 195, pp. 30–43, 2017.

[38] X. Zhang, R. Gao, Q. Sun, and J. Cheng, "An automated rectification method for unmanned aerial vehicle lidar point cloud data based on laser intensity," *Remote Sensing*, vol. 11, no. 7, p. 811, 2019.

[39] J. Li, B. Yang, Y. Cong, L. Cao, X. Fu, and Z. Dong, "3d forest mapping using a low-cost uav laser scanning system: Investigation and comparison," *Remote Sensing*, vol. 11, no. 6, p. 717, 2019.

[40] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[41] S. Mattoccia and M. Poggi, "A passive rgbd sensor for accurate and real-time depth sensing self-contained into an fpga," in *Proceedings of the 9th International Conference on Distributed Smart Cameras*, pp. 146–151, ACM, 2015.

[42] E. Lachat, H. Macher, M. Mittet, T. Landes, and P. Grussenmeyer, "First experiences with kinect v2 sensor for close range 3d modelling," in *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, p. 93, 2015.

[43] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017.

[44] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1534–1543, 2016.

[45] M. Shahzad, X. X. Zhu, and R. Bamler, "Facade structure reconstruction using spaceborne tomosar point clouds," in *2012 IEEE International Geoscience and Remote Sensing Symposium*, pp. 467–470, IEEE, 2012.

[46] X. X. Zhu and M. Shahzad, "Facade reconstruction using multiview spaceborne tomosar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 6, pp. 3541–3552, 2014.

[47] M. Shahzad and X. X. Zhu, "Robust reconstruction of building facades for large areas using spaceborne tomosar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 2, pp. 752–769, 2015.

[48] M. Shahzad, M. Schmitt, and X. X. Zhu, "Segmentation and crown parameter extraction of individual trees in an airborne tomosar point cloud," in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, pp. 205–209, 2015.

[49] M. Schmitt, M. Shahzad, and X. X. Zhu, "Reconstruction of individual trees from multi-aspect tomosar data," *Remote Sensing of*

Environment, vol. 165, pp. 175–185, 2015.

[50] R. Bamler, M. Eineder, N. Adam, X. X. Zhu, and S. Gernhardt, "Interferometric potential of high resolution spaceborne sar," *Photogrammetrie-Fernerkundung-Geoinformation*, vol. 2009, no. 5, pp. 407–419, 2009.

[51] X. X. Zhu and R. Bamler, "Very high resolution spaceborne sar tomography in urban environment," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 12, pp. 4296–4308, 2010.

[52] S. Gernhardt, N. Adam, M. Eineder, and R. Bamler, "Potential of very high resolution sar for persistent scatterer interferometry in urban areas," *Annals of GIS*, vol. 16, no. 2, pp. 103–111, 2010.

[53] S. Gernhardt, X. Cong, M. Eineder, S. Hinz, and R. Bamler, "Geometrical fusion of multitrack ps point clouds," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 1, pp. 38–42, 2012.

[54] X. X. Zhu and R. Bamler, "Super-resolution power and robustness of compressive sensing for spectral estimation with application to spaceborne tomographic sar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 1, pp. 247–258, 2012.

[55] S. Montazeri, F. Rodríguez González, and X. X. Zhu, "Geocoding error correction for insar point clouds," *Remote Sensing*, vol. 10, no. 10, p. 1523, 2018.

[56] F. Rottensteiner and C. Briese, "A new method for building extraction in urban areas from high-resolution lidar data," in *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, pp. 295–301, 2002.

[57] X. X. Zhu and R. Bamler, "Demonstration of super-resolution for tomographic sar imaging in urban environment," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 8, pp. 3150–3157, 2012.

[58] X. X. Zhu, M. Shahzad, and R. Bamler, "From tomosar point clouds to objects: Facade reconstruction," in *2012 Tyrrhenian Workshop on Advances in Radar and Remote Sensing (TyWRRS)*, pp. 106–113, IEEE, 2012.

[59] X. X. Zhu and R. Bamler, "Let's do the time warp: Multicomponent nonlinear motion estimation in differential sar tomography," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 735–739, 2011.

[60] S. Auer, S. Gernhardt, and R. Bamler, "Ghost persistent scatterers related to multiple signal reflections," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 5, pp. 919–923, 2011.

[61] Y. Shi, X. X. Zhu, and R. Bamler, "Nonlocal compressive sensing-based sar tomography," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 5, pp. 3015–3024, 2019.

[62] Y. Wang and X. X. Zhu, "Automatic feature-based geometric fusion of multiview tomosar point clouds in urban area," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 3, pp. 953–965, 2014.

[63] M. Schmitt and X. X. Zhu, "Data fusion and remote sensing: An ever-growing relationship," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 4, pp. 6–23, 2016.

[64] Y. Wang, X. X. Zhu, B. Zeisl, and M. Pollefeys, "Fusing meter-resolution 4-d insar point clouds and optical images for semantic urban infrastructure monitoring," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 1, pp. 14–26, 2017.

[65] A. Adam, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "H-ransac: A hybrid point cloud segmentation combining 2d and 3d data.," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, no. 2, 2018.

[66] J. Bauer, K. Karner, K. Schindler, A. Klaus, and C. Zach, "Segmentation of building from dense 3d point-clouds," in *Proceedings of the ISPRS. Workshop Laser scanning Enschede*, pp. 12–14, 2005.

[67] A. Boulch, J. Guerry, B. Le Saux, and N. Audebert, "Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks," *Computers & Graphics*, vol. 71, pp. 189–198, 2018.

[68] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2530–2539, 2018.

[69] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3577–3586, 2017.

[70] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084, 2019.

[71] F. A. Limberger and M. M. Oliveira, "Real-time detection of planar regions in unorganized point clouds," *Pattern Recognition*, vol. 48, no. 6, pp. 2043–2053, 2015.

[72] B. Xu, W. Jiang, J. Shan, J. Zhang, and L. Li, "Investigation on the weighted ransac approaches for building roof plane segmentation from lidar point clouds," *Remote Sensing*, vol. 8, no. 1, p. 5, 2015.

[73] D. Chen, L. Zhang, P. T. Mathiopoulos, and X. Huang, "A methodology for automated segmentation and reconstruction of urban 3-d buildings from als point clouds," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 10, pp. 4199–4217, 2014.

[74] 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 *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, vol. 36, pp. 407–412, 2007.

[75] B. Gorte, "Segmentation of tin-structured surface models," in *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, pp. 465–469, 2002.

[76] A. Sampath and J. Shan, "Clustering based planar roof extraction from lidar data," in *American Society for Photogrammetry and Remote Sensing Annual Conference, Reno, Nevada, May*, pp. 1–6, 2006.

[77] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds," *IEEE Transactions on geoscience and remote sensing*, vol. 48, no. 3, pp. 1554–1567, 2010.

[78] S. Ural and J. Shan, "Min-cut based segmentation of airborne lidar point clouds," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 167–172, 2012.

[79] J. Yan, J. Shan, and W. Jiang, "A global optimization approach to roof segmentation from airborne lidar point clouds," *ISPRS journal of photogrammetry and remote sensing*, vol. 94, pp. 183–193, 2014.

[80] T. Melzer, "Non-parametric segmentation of als point clouds using mean shift," *Journal of Applied Geodesy Jag*, vol. 1, no. 3, pp. 159–170, 2007.

[81] W. Yao, S. Hinz, and U. Stilla, "Object extraction based on 3d-segmentation of lidar data by combining mean shift with normalized cuts: Two examples from urban areas," in *2009 Joint Urban Remote Sensing Event*, pp. 1–6, IEEE, 2009.

[82] S. K. Lodha, D. M. Fitzpatrick, and D. P. Helmbold, "Aerial lidar data classification using adaboost," in *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, pp. 435–442, IEEE, 2007.

[83] M. Carlberg, P. Gao, G. Chen, and A. Zakhor, "Classifying urban landscape in aerial lidar using 3d shape analysis," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 1701–1704, IEEE, 2009.

[84] N. Chehata, L. Guo, and C. Mallet, "Airborne lidar feature selection for urban classification using random forests," in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, pp. 207–212, 2009.

[85] R. Shapovalov, E. Velizhev, and O. Barinova, "Nonassociative markov networks for 3d point cloud classification," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, pp. 103–108, 2010.

[86] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Conditional random fields for lidar point cloud classification in complex urban areas," in *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, vol. 3, pp. 263–268, 2012.

[87] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of lidar data and building object detection in urban areas," *ISPRS journal of photogrammetry and remote sensing*, vol. 87, pp. 152–165, 2014.

[88] G. Vosselman, M. Coenen, and F. Rottensteiner, "Contextual segment-based classification of airborne laser scanner data," *ISPRS journal of photogrammetry and remote sensing*, vol. 128, pp. 354–371, 2017.

[89] X. Xiong, D. Munoz, J. A. Bagnell, and M. Hebert, "3-d scene analysis via sequenced predictions over points and regions," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2609–2616, IEEE, 2011.

[90] M. Najafi, S. T. Namin, M. Salzmann, and L. Petersson, "Non-associative higher-order markov networks for point cloud classification," in *European Conference on Computer Vision*, pp. 500–515, Springer, 2014.

[91] F. Morsdorf, E. Meier, B. Kötz, K. I. Itten, M. Dobbertin, and B. Allgöwer, "Lidar-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management," *Remote Sensing of Environment*, vol. 92, no. 3, pp. 353–362, 2004.

[92] A. Ferraz, F. Bretar, S. Jacquemoud, G. Gonçalves, and L. Pereira, "3d segmentation of forest structure using a mean-shift based algorithm," in *2010 IEEE International Conference on Image Processing*, pp. 1413–1416, IEEE, 2010.

[93] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 104, pp. 88–100, 2015.

[94] A. Nurunnabi, D. Belton, and G. West, "Robust segmentation in laser scanning 3d point cloud data," in *2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2012.

[95] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 286–304, 2015.

[96] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 975–982, IEEE, 2009.

[97] L. Landrieu, H. Raguet, B. Vallet, C. Mallet, and M. Weinmann, "A structured regularization framework for spatially smoothing semantic labelings of 3d point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 132, pp. 102–118, 2017.

[98] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," in *2017 International Conference on 3D Vision (3DV)*, pp. 537–547, IEEE, 2017.

[99] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, "3d recurrent neural networks with context fusion for point cloud semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 403–417, 2018.

[100] L. Landrieu and M. Boussaha, "Point cloud oversegmentation with graph-structured deep metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7440–7449, 2019.

[101] J. Xiao, J. Zhang, B. Adler, H. Zhang, and J. Zhang, "Three-dimensional point cloud plane segmentation in both structured and unstructured environments," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1641–1652, 2013.

[102] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, "An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells," *Remote Sensing*, vol. 9, no. 5, p. 433, 2017.

[103] H. Boulaassal, T. Landes, P. Grussenmeyer, and F. Tarsha-Kurdi, "Automatic segmentation of building facades using terrestrial laser data," in *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, pp. 65–70, 2007.

[104] Z. Dong, B. Yang, P. Hu, and S. Scherer, "An efficient global energy optimization approach for robust 3d plane segmentation of point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 137, pp. 112–133, 2018.

[105] J. M. Biosca and J. L. Lerma, "Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 63, no. 1, pp. 84–98, 2008.

[106] X. Ning, X. Zhang, Y. Wang, and M. Jaeger, "Segmentation of architecture shape information from 3d point cloud," in *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, pp. 127–132, ACM, 2009.

[107] Y. Xu, S. Tuttas, and U. Stilla, "Segmentation of 3d outdoor scenes using hierarchical clustering structure and perceptual grouping laws," in *2016 9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS)*, pp. 1–6, IEEE, 2016.

[108] Y. Xu, L. Hoegner, S. Tuttas, and U. Stilla, "Voxel-and graph-based point cloud segmentation of 3d scenes using perceptual grouping laws," in *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2017.

[109] Y. Xu, W. Yao, S. Tuttas, L. Hoegner, and U. Stilla, "Unsupervised segmentation of point clouds from buildings using hierarchical clustering based on gestalt principles," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, no. 99, pp. 1–17, 2018.

[110] E. H. Lim and D. Suter, "3d terrestrial lidar classifications with super-voxels and multi-scale conditional random fields," *Computer-Aided Design*, vol. 41, no. 10, pp. 701–710, 2009.

[111] Z. Li, L. Zhang, X. Tong, B. Du, Y. Wang, L. Zhang, Z. Zhang, H. Liu, J. Mei, X. Xing, *et al.*, "A three-step approach for tls point cloud classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 9, pp. 5412–5424, 2016.

[112] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10296–10305, 2019.

[113] J.-F. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert, "Scale selection for classification of point-sampled 3d surfaces," in *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pp. 285–292, IEEE, 2005.

[114] 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 Research*, vol. 2, no. 2, p. 3, 2011.

[115] R. Hulik, M. Spanel, P. Smrz, and Z. Materna, "Continuous plane detection in point-cloud data based on 3d hough transform," *Journal of visual communication and image representation*, vol. 25, no. 1, pp. 86–97, 2014.

[116] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[117] R. Shapovalov, D. Vetrov, and P. Kohli, "Spatial inference machines," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2985–2992, 2013.

[118] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3d segmentation of point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2626–2635, 2018.

[119] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in Neural Information Processing Systems*, pp. 828–838, 2018.

[120] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, pp. 5099–5108, 2017.

[121] M. Jiang, Y. Wu, and C. Lu, "Pointsift: A sift-like network module for 3d point cloud semantic segmentation," *arXiv preprint arXiv:1807.00652*, 2018.

[122] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2019.

[123] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4096–4105, 2019.

[124] Q.-H. Pham, T. Nguyen, B.-S. Hua, G. Roig, and S.-K. Yeung, "Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8827–8836, 2019.

[125] A. Komarichev, Z. Zhong, and J. Hua, "A-cnn: Annularly convolutional neural networks on point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7421–7430, 2019.

[126] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "Pointweb: Enhancing local neighborhood features for point cloud processing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5565–5573, 2019.

[127] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2569–2578, 2018.

[128] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "Gspn: Generative shape proposal network for 3d instance segmentation in point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3947–3956, 2019.

[129] T. Rabbani and F. Van Den Heuvel, "Efficient hough transform for automatic detection of cylinders in point clouds," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, pp. 60–65, 2005.

[130] T.-T. Tran, V.-T. Cao, and D. Laurendeau, "Extraction of cylinders and estimation of their parameters from point clouds," *Computers & Graphics*, vol. 46, pp. 345–357, 2015.

[131] V.-H. Le, H. Vu, T. T. Nguyen, T.-L. Le, and T.-H. Tran, "Acquiring qualified samples for ransac using geometrical constraints," *Pattern Recognition Letters*, vol. 102, pp. 58–66, 2018.

[132] H. Riemenschneider, A. Bódis-Szomorú, J. Weissenberg, and L. Van Gool, "Learning where to classify in multi-view semantic segmentation," in *European Conference on Computer Vision*, pp. 516–532, Springer, 2014.

[133] M. De Deuge, A. Quadros, C. Hung, and B. Douillard, "Unsupervised feature learning for classification of outdoor 3d scans," in *Australasian Conference on Robitics and Automation*, vol. 2, 2013.

[134] A. Serna, B. Marcotegui, F. Goulette, and J.-E. Deschaud, "Paris-rue-madame database: a 3d mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods," in *4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014*, 2014.

[135] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[136] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*, pp. 746–760, Springer, 2012.

[137] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," *arXiv preprint arXiv:1702.01105*, 2017.

[138] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," in *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 36, pp. 248–253, 2006.

[139] B. Bhanu, S. Lee, C.-C. Ho, and T. Henderson, "Range data processing: Representation of surfaces by edges," in *Proceedings of the Eighth International Conference on Pattern Recognition*, pp. 236–238, IEEE Computer Society Press, 1986.

[140] X. Y. Jiang, U. Meier, and H. Bunke, "Fast range image segmentation using high-level segmentation primitives," in *Proceedings Third IEEE Workshop on Applications of Computer Vision*, pp. 83–88, IEEE, 1996.

[141] A. D. Sappa and M. Devy, "Fast range image segmentation by an edge detection strategy," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 292–299, IEEE, 2001.

[142] M. A. Wani and H. R. Arabnia, "Parallel edge-region-based segmentation algorithm targeted at reconfigurable multiring network," *The Journal of Supercomputing*, vol. 25, no. 1, pp. 43–62, 2003.

[143] E. Castillo, J. Liang, and H. Zhao, "Point cloud segmentation and denoising via constrained nonlinear least squares normal estimates," in *Innovations for Shape Analysis*, pp. 283–299, Springer, 2013.

[144] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 167–192, 1988.

[145] R. Geibel and U. Stilla, "Segmentation of laser altimeter data for building reconstruction: different procedures and comparison," in *International Archives of Photogrammetry and Remote Sensing*, vol. 33, pp. 326–334, 2000.

[146] D. Tóvári and N. Pfeifer, "Segmentation based robust interpolation-a new approach to laser data filtering," in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, pp. 79–84, 2005.

[147] 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 *3DPVT*, 2010.

[148] P. V. Hough, "Method and means for recognizing complex patterns," 1962. US Patent 3,069,654.

[149] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: randomized hough transform (rht)," *Pattern recognition letters*, vol. 11, no. 5, pp. 331–338, 1990.

[150] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[151] A. Kaiser, J. A. Ybanez Zepeda, and T. Boubekeur, "A survey of simple geometric primitives detection methods for captured 3d data," in *Computer Graphics Forum*, vol. 38, pp. 167–196, Wiley Online Library, 2019.

[152] N. Kiryati, Y. Eldar, and A. M. Bruckstein, "A probabilistic hough transform," *Pattern recognition*, vol. 24, no. 4, pp. 303–316, 1991.

[153] A. Yla-Jaaski and N. Kiryati, "Adaptive termination of voting in the probabilistic circular hough transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 911–915, 1994.

[154] C. Galamhos, J. Matas, and J. Kittler, "Progressive probabilistic hough transform for line detection," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition*, vol. 1, pp. 554–560, IEEE, 1999.

[155] L. A. Fernandes and M. M. Oliveira, "Real-time line detection through an improved hough transform voting scheme," *Pattern recognition*, vol. 41, no. 1, pp. 299–314, 2008.

[156] G. Vosselman, B. G. Gorte, G. Sithole, and T. Rabbani, "Recognising structure in laser scanner point clouds," in *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 46, pp. 33–38, 2004.

[157] M. Camurri, R. Vezzani, and R. Cucchiara, "3d hough transform for sphere recognition on point clouds," *Machine vision and applications*, vol. 25, no. 7, pp. 1877–1891, 2014.

[158] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[159] S. Choi, T. Kim, and W. Yu, "Performance evaluation of ransac family," in *Proceedings of the British Machine Vision Conference*, 2009.

[160] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus," in *European Conference on Computer Vision*, pp. 500–513, Springer, 2008.

[161] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "Usac: a universal framework for random sample consensus," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 2022–2038, 2013.

[162] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," in *Computer graphics forum*, vol. 26, pp. 214–226, Wiley Online Library, 2007.

[163] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453), vol. 3, pp. 2743–2748, IEEE, 2003.

[164] V. Fragoso, P. Sen, S. Rodriguez, and M. Turk, "Evsac: accelerating hypotheses generation by modeling matching scores with extreme value theory," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2472–2479, 2013.

[165] D. Barath and J. Matas, "Graph-cut ransac," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6733–6741, 2018.

[166] S. Filin, "Surface clustering from airborne laser scanning data," in *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, pp. 119–124, 2002.

[167] A. Golovinskiy and T. Funkhouser, "Min-cut based segmentation of point clouds," in *IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 39–46, IEEE, 2009.

[168] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1197–1203, IEEE, 1999.

[169] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 603–619, 2002.

[170] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[171] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International journal of computer vision*, vol. 70, no. 2, pp. 109–131, 2006.

[172] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *International journal of computer vision*, vol. 96, no. 1, pp. 1–27, 2012.

[173] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation-supervoxels for point clouds," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2027–2034, 2013.

[174] S. Song, H. Lee, and S. Jo, "Boundary-enhanced supervoxel segmentation for sparse outdoor lidar data," *Electronics Letters*, vol. 50, no. 25, pp. 1917–1919, 2014.

[175] Y. Lin, C. Wang, D. Zhai, W. Li, and J. Li, "Toward better boundary preserved supervoxel segmentation for 3d point clouds," *ISPRS journal of photogrammetry and remote sensing*, vol. 143, pp. 39–47, 2018.

[176] S. Christoph Stein, M. Schoeler, J. Papon, and F. Worgotter, "Object partitioning using local convexity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 304–311, 2014.

[177] B. Yang, Z. Dong, G. Zhao, and W. Dai, "Hierarchical extraction of urban objects from mobile laser scanning data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 99, pp. 45–57, 2015.

[178] A. Schmidt, F. Rottensteiner, and U. Sörgel, "Classification of airborne laser scanning data in wadden sea areas using conditional random fields," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 39, pp. 161–166, 2012.

[179] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive

review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[180] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[181] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[182] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[183] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[184] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[185] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[186] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, pp. 945–953, 2015.

[187] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928, 2015.

[188] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 72, 2017.

[189] H.-Y. Meng, L. Gao, Y. Lai, and D. Manocha, "Vv-net: Voxel vae net with group convolutions for point cloud segmentation," *arXiv preprint arXiv:1811.04337*, 2018.

[190] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3d semantic segmentation of point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 716–724, 2017.

[191] G. Te, W. Hu, A. Zheng, and Z. Guo, "Rgcnn: Regularized graph cnn for point cloud segmentation," in *ACM Multimedia Conference on Multimedia Conference*, pp. 746–754, ACM, 2018.

[192] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[193] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[194] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[195] L. Li, M. Sung, A. Dubrovina, L. Yi, and L. J. Guibas, "Supervised fitting of geometric primitives to 3d point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2652–2660, 2019.

Yuxing Xie (yuxing.xie@dlr.de) received the B.Eng. degree in remote sensing science and technology from Wuhan University, Wuhan, China, in 2015, and the M.Eng. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2018. He is currently pursuing the Ph.D. degree with the Remote Sensing Technology Institute, German Aerospace Center (DLR), Weßling, Germany, and the Technical University of Munich (TUM), Munich, Germany. His research interests include point cloud processing and the application of 3D geographic data.

Jiaojiao Tian (jiaojiao.tian@dlr.de) received her B.S in Geo-Information Systems at the China University of Geoscience (Beijing) in 2006, M. Eng in Cartography and Geo-information at the Chinese Academy of Surveying and Mapping (CASM) in 2009, and Ph.D. degree in mathematics and computer science from Osnabrück University, Germany in 2013. Since 2009, she has been with the Photogrammetry and Image Analysis Department, Remote Sensing Technology Institute, German Aerospace Center, Weßling, Germany, where she is currently the Head of the 3D Modeling Team. In 2011, she was a Guest Scientist with the Institute of Photogrammetry and Remote Sensing, ETH Zrich, Zurich, Switzerland. Her research interests include 3-D change detection, digital surface model (DSM) generation, 3D point cloud semantic segmentation, object extraction, and DSM-assisted building reconstruction and classification.

Xiao Xiang Zhu (xiaoxiang.zhu@dlr.de) received the Master (M.Sc.) degree, her doctor of engineering (Dr.-Ing.) degree and her Habilitation in the field of signal processing from Technical University of Munich (TUM), Munich, Germany, in 2008, 2011 and 2013, respectively.

She is currently the Professor for Signal Processing in Earth Observation (www.sipeo.bgu.tum.de) at Technical University of Munich (TUM) and German Aerospace Center (DLR); the head of the department "EO Data Science" at DLR's Earth Observation Center; and the head of the Helmholtz Young Investigator Group "SiPEO" at DLR and TUM. Since 2019, she is co-coordinating the Munich Data Science Research School (www.mu-ds.de). She is also leading the Helmholtz Artificial Intelligence Cooperation Unit (HAICU) – Research Field "Aeronautics, Space and Transport". Prof. Zhu was a guest scientist or visiting professor at the Italian National Research Council (CNR-IREA), Naples, Italy, Fudan University, Shanghai, China, the University of Tokyo, Tokyo, Japan and University of California, Los Angeles, United States in 2009, 2014, 2015 and 2016, respectively. Her main research interests are remote sensing and Earth observation, signal processing, machine learning and data science, with a special application focus on global urban mapping.

Dr. Zhu is a member of young academy (Junge Akademie/Junges Kolleg) at the Berlin-Brandenburg Academy of Sciences and Humanities and the German National Academy of Sciences Leopoldina and the Bavarian Academy of Sciences and Humanities. She is an associate Editor of IEEE Transactions on Geoscience and Remote Sensing.