

# Linking Points With Labels in 3D

## *A review of point cloud semantic segmentation*

YUXING XIE, JIAOJIAO TIAN, AND XIAO XIANG ZHU

**R**ipe with possibilities offered by deep-learning techniques and useful in applications related to remote sensing, computer vision, and robotics, 3D point cloud semantic segmentation (PCSS) and point cloud segmentation (PCS) are attracting increasing interest. This article summarizes available data sets and relevant studies on recent developments in PCSS and PCS.

### MOTIVATION

Semantic segmentation, a technique that associates pixels with semantic labels, is a fundamental research challenge in image processing. PCSS is the 3D form of semantic segmentation. With PCSS, 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 that measure distance or generated from stereo- or multiview imagery. Due to recently developed stereovision algorithms and the deployment of all kinds of 3D sensors, point clouds, which are 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 3D/2.5D geometric structures, which makes modeling possible.

### SEGMENTATION, CLASSIFICATION, AND SEMANTIC SEGMENTATION

Research on PCSS has a long tradition involving various fields. As a result, multiple terms for similar ideas have emerged. A brief clarification of some concepts is therefore necessary to avoid confusion. 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]. In some cases, this task is also called *point labeling* [7]–[9]. In this article, to ensure clarity and to be consistent with the latest deep-learning techniques, we refer to 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 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 (sometimes used as a presegmentation step) can influence the final results. Hence, PCS approaches are also included in this article.

Single objects or the same classes of structures cannot be acquired from a raw point cloud directly. However, instance- or class-level objects are required for object recognition. For example, urban planning applications and building information modeling need buildings and other man-made ground objects for reference [10], [11]. The use of sensors to remotely monitor forests requires individual tree information based on each tree's geometric structure [12], [13]. Robotics applications, such as 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]. In these cases, PCSS and PCS are basic and critical tasks for 3D applications.

### NEW CHALLENGES AND POSSIBILITIES

Two of the best available reviews for PCS and PCSS are found in [16] and [17]. However, they lack detailed information, especially for PCSS. Furthermore, in the past two years, deep learning has largely driven studies in PCSS. To meet the demand of deep learning, higher quality and more diverse 3D data sets have become available. Therefore, an updated study on current PCSS techniques is necessary.

## AN INTRODUCTION TO POINT CLOUDS

### POINT CLOUD DATA ACQUISITION

In computer vision and remote sensing, point clouds can be acquired using 1) image-derived methods; 2) lidar; 3) red, green, blue-depth (RGB-D) cameras; or 4) synthetic aperture radar (SAR) systems. Due to the differences in survey principles and platforms, their data features and application ranges are very diverse.

### 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, either automatically or semiautomatically, 3D isolated point information according to principles based on photogrammetry or computer vision theory [18], [19]. Stereo- and multiview image-derived systems can be divided into four categories according to platform: airborne, spaceborne, unmanned aerial vehicle based, and close range.

Traditional aerial photogrammetry early on produced 3D points with semiautomatic human-computer interaction in digital photogrammetric systems characterized by strict geometric constraints and high survey accuracy [20]. Producing this type of point data took a lot of time because it required so much labor. This made the generation of dense points for large areas impractical. 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 limitations related to image resolution and the ability to process multiview images, traditional photogrammetry could acquire only near-nadir views with few building façades from aerial/satellite platforms, which generated a 2.5D point cloud rather than full 3D. At this stage, photogrammetry principles could also be applied as close-range photogrammetry 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 MVS. SfM can estimate camera positions and orientations automatically, making it able to process multiview images simultaneously, while dense matching and MVS algorithms provide the ability to generate a 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 that generated by traditional photogrammetry or lidar techniques, and it is especially unreliable for large regions [28].

Compared to airborne photogrammetry, a satellite stereo system is inferior in terms of spatial resolution and availability of multiview imagery. However, satellite cameras are able to map large regions quickly for less 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.

### LIDAR POINT CLOUD

Lidar, a surveying and remote sensing technique, uses laser energy to measure the distance between the sensor and the object to be surveyed [29]. Most lidar systems are pulse based. With pulse-based measuring, a pulse of laser energy is emitted, and then the time it takes for that energy to travel to a target is measured. Depending on sensors and platforms, the point density or resolution varies greatly from fewer than 10 points per square meter (pts/m<sup>2</sup>) to thousands of pts/m<sup>2</sup> [30]. Various lidar platforms are available in the form of airborne lidar scanning (ALS), terrestrial LS (TLS), mobile LS (MLS), and unmanned LS (ULS) systems.

ALS systems operate from airborne platforms. Data from early ALS systems are 2.5D point clouds, which are similar to traditional photogrammetric point clouds. The density of ALS points is normally low because of the long distance from an airborne platform to the ground. In comparison to traditional photogrammetry, ALS point clouds cost more to acquire and normally contain no spectral information. The Vaihingen point cloud semantic labeling data set [31] is a typical ALS benchmark data set. Multispectral airborne lidar, a special kind of ALS system, obtains data using different wavelengths. Multispectral lidar performs well in

extracting water, vegetation, and shadows, but the data are not easily obtained [32], [33].

TLS, also called *static LS*, 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. TLS has been commonly used to model small urban or forest sites and document heritage sites or works of art. Semantic3D.net [34] is a typical TLS benchmark data set.

MLS systems operate from moving vehicles, usually cars. A current hot topic for research and development is autonomous driving, for which HD maps are essential. The generation of HD maps is therefore the most significant application for MLS systems. Several mainstream point cloud benchmark data sets are captured with MLS systems [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 gaining popularity. Compared to ALS systems, where the platform is also above the objects, ULS systems can conduct lidar surveys from a shorter distance. They can thus collect denser point clouds with higher accuracy. Because the platform is compact and lightweight, ULS systems offer high operational flexibility. Therefore, in addition to traditional lidar tasks (e.g., acquiring DSMs), ULS systems offer advantages for conducting agriculture, forestry, and mining surveys and for monitoring disasters [37]–[39].

Since the system is always moving with the platform, it is necessary for LS to combine the positions of points with GNSS and inertial measurement unit data to ensure a high-quality matching point cloud. Lidar has been the most important data source for point cloud research and has been used to compare and evaluate the quality of point clouds from other sources.

#### RGB-D POINT CLOUD

An RGB-D camera can acquire both RGB and depth information. There are three kinds of RGB-D sensors, each based on a different principle: 1) structured light [40], 2) stereo [41], and 3) time of flight [42]. Similar to lidar, the RGB-D camera measures the distance between the camera and the objects. But the camera generates pixel-wise depth data, rather than unstructured points. An RGB-D sensor is much cheaper than a lidar system. Microsoft's Kinect is a well-known and widely used RGB-D sensor [40], [42]. In an RGB-D camera, relative orientation elements between or among different sensors are calibrated and known, so coregistered, 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 for close-range applications, much closer even than those for TLS systems, they are usually employed in indoor environments. Several mainstream indoor PCSS benchmarks use RGB-D data [43], [44].

#### SAR POINT CLOUD

Interferometric SAR (InSAR), a radar technique crucial for remote sensing, generates maps that show surface deformations or digital elevations based on comparisons of multiple SAR image pairs. InSAR-based point clouds have demonstrated their value over the past few years and are opening up new possibilities for point cloud applications [45]–[49]. SAR tomography (TomoSAR) and persistent scatterer interferometry (PSI) are two major techniques that generate point clouds with InSAR, extending the principle of SAR into the 3D realm [50], [51]. TomoSAR's advantage over PSI is its ability to enable detailed reconstruction and monitoring of urban areas, especially of human-made infrastructures [51]. The TomoSAR point cloud has a point density comparable to that of ALS lidar [52], [53]. Especially useful for applications in building reconstruction in urban areas, these point clouds have the following features [46]:

- 1) TomoSAR point clouds reconstructed from spaceborne data have a moderate 3D positioning accuracy on the order of 1 m [54], enabling decimeter-level accuracy by geocoding error-correction techniques [55]. By comparison, ALS lidar provides accuracy typically on the order of 0.1 m [56].
- 2) Due to their coherent imaging nature and side-looking geometry, TomoSAR point clouds emphasize different objects with respect to lidar systems. The side-looking SAR geometry enables TomoSAR point clouds to possess rich façade information ([57] presents results using pixelwise TomoSAR for the high-resolution reconstruction of a building complex with a very high level of detail from spaceborne SAR data). Temporarily incoherent objects, e.g., trees, cannot be reconstructed from multipass spaceborne SAR image stacks. To obtain the full structure of individual buildings from space, façade reconstruction using TomoSAR point clouds from multiple viewing angles is required [45], [58].
- 3) 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; and 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, data from SAR systems have not yet been widely used for studies and applications. However, mature SAR satellites, such as *TerraSAR-X*, have collected abundant global SAR data, which are available for InSAR-based reconstruction at a global scale [61]. Hence, SAR point clouds can be expected to play a conspicuous role in the future.

### POINT CLOUD CHARACTERS

As sensors were developed and various applications emerged, point clouds evolved in three stages: 1) sparse (fewer than 20 pts/m<sup>2</sup>), 2) dense (hundreds of pts/m<sup>2</sup>), and 3) multisource.

- 1) In the early stage, photogrammetric point clouds were sparse, limited by matching techniques and computation ability. At that time, only a few types of laser scanning systems were available, and they were not widely used. ALS point clouds, which were the mainstream laser data, were also sparse. Limited by point density, point clouds at this stage were not able to represent land surface at an object level. 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).
- 2) 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 while also stimulating demands for practicable algorithms. PCS and PCSS were proposed and became increasingly necessary, since only a class-level or instance-level point cloud can further connect the virtual world to the real one. Both computer vision and remote sensing need PCS and PCSS solutions to develop class-level interactive applications.
- 3) From the perspective of general computer vision, research on the point cloud and its related algorithms remains at stage 2). However, driven by rapidly growing data from spaceborne platforms and multisensors, remote sensing researchers have a different understanding of point clouds. New-generation point clouds, such as satellite photogrammetric point clouds and TomoSAR point clouds, have stimulated demand for relevant algorithms. Multisource data fusion has become a trend in remote sensing [62]–[64], but current algorithms in computer vision are insufficient for such remote sensing data sets. To fully exploit multisource point cloud data, more research is needed.

Table 1 provides an overview of basic information about various point clouds, including point density, advantages, disadvantages, and applications.

### POINT CLOUD APPLICATION

In studies about PCS and PCSS, requirements of specific applications drive the selection of data and algorithms. In this section, we outline most of the studies focusing on PCS and PCSS reviewed in this article (Table 2). These studies are classified according to their point cloud data types and working environments, such as urban, forest, industry, and indoor settings.

Several issues can be summarized from Table 2:

- 1) Lidar point clouds are the most commonly used data in PCS applications. They have been widely used for buildings (urban environments) and trees (forests). Buildings are also the most popular research objects in traditional PCS applications. As buildings are usually constructed with regular planes, plane segmentation is a fundamental topic in building segmentation.
- 2) Image-derived point clouds have been frequently used in real-world scenarios. However, mainly due to the limitations of available annotated benchmarks, there are not many PCS and PCSS studies on image-based data. Currently, only one public, influential data set is based on image-derived points, and its range is just a very small area around a single building [132]. More efforts are therefore needed.
- 3) RGB-D sensors are limited by their close range, so they are usually applied in indoors. In PCS studies, plane segmentation is the main task for RGB-D data. In PCSS studies, since several benchmark data sets are derived from RGB-D sensors, many deep-learning-based methods are tested on them.
- 4) Few relevant PCS or PCSS studies have been done involving InSAR point clouds, but these have shown potential in urban monitoring, especially in regard to building structure segmentation.

### BENCHMARK DATA SETS

Public standard benchmark data sets have demonstrated significant effectiveness for algorithm development, evaluation, and comparison. Most of them are labeled for PCSS rather than PCS. Since 2009, several benchmark data sets have been available for PCSS. However, early data sets have many shortcomings. For example, Neither the Oakland 3D Point Cloud MLS data set [96], the Sydney Urban Objects MLS data set [133], the Paris-Rue-Madame MLS data set [134], the IQmulus and TerraMobilita Contest MLS data set [35], nor the ETHZ CVL RueMonge 2014 multiview stereo data set [132] can sufficiently provide both different object representations and labeled points. The Karlsruhe Institute of Technology and Toyota Technological Institute of Chicago (KITTI) data set [135] and New York University Depth Dataset V2 (NYUv2) data set [136] have more objects and points than the aforementioned data sets, 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.

**TABLE 1. AN OVERVIEW OF VARIOUS POINT CLOUDS.**

	<b>POINT DENSITY</b>	<b>ADVANTAGES</b>	<b>DISADVANTAGES</b>	<b>APPLICATIONS</b>
<b>IMAGE DERIVED</b>	From sparse (<10 pts/m <sup>2</sup> ) to very high (>400 pts/m <sup>2</sup> ), depending on the spatial resolution of the stereo- or multiview images	With color (RGB, multispectral) information; suitable for large areas (airborne, spaceborne)	Influenced by light; accuracy depends on available precise camera models, image-matching algorithms, stereo angles, image resolution, and 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; and more
<b>LIDAR</b>				
<b>ALS</b>	Sparse (<20 pts/m <sup>2</sup> ); when the survey distance is shorter, the density is higher	High accuracy (<15 cm); suitable for large areas; not affected by weather		Urban monitoring; vegetation monitoring; power line detection; and more
<b>MLS</b>	Dense (>100 pts/m <sup>2</sup> ); when the survey distance is shorter, the density is higher	High accuracy (centimeter level)	Expensive; affected by mirror reflection; long scanning time	HD map; urban monitoring
<b>TLS</b>	Dense (>100 pts/m <sup>2</sup> ); when the survey distance is shorter, the density is higher	High accuracy (millimeter level)		Small-area 3D reconstruction
<b>ULS</b>	Dense (>100 pts/m <sup>2</sup> ); when the survey distance is shorter, the density is higher	High accuracy (centimeter level)		Forestry surveying; mining surveying; disaster monitoring; and others
<b>RGB-D</b>	Middle density	Cheap; flexible	Close range; limited accuracy	Indoor reconstruction; object tracking; human pose recognition; and others
<b>InSAR</b>	Sparse (<20 pts/m <sup>2</sup> )	Global data are available; compared to ALS, complete building façade information is available; 4D information; middle accuracy; not affected by weather	Expensive data; ghost scatterers; preprocessing techniques needed	Urban monitoring; forest monitoring; and others

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

#### SEMANTIC3D.NET

Semantic3D.net [34] is a representative, large-scale outdoor TLS PCSS data set. It constitutes a collection of urban scenes with more than four billion labeled 3D points for PCSS purposes only. Those scenes contain a range of urban objects, divided into eight classes, including manmade terrain, natural terrain, high vegetation, low vegetation, buildings, hardscape, scanning artifacts, and cars. In consideration of the efficiency of different algorithms, two types of subdata sets were designed, Semantic-8 and Reduced-8. Semantic-8 is the full data set, while Reduced-8 uses training data in the same way as Semantic-8 but includes only four small subsets as test data. This data set can be downloaded at <http://www.semantic3d.net/>. To learn about the performance of different algorithms on this data set, refer to [2], [67], and [112].

#### STANFORD LARGE-SCALE 3D INDOOR SPACES DATA SET

Unlike Semantic3D.net, the Stanford Large-Scale 3D Indoor Spaces Data Set (S3DIS) [44] is a large-scale indoor

RGB-D data set, which is also a part of the 2D-3D-S data set [137]. It is a collection of more than 215 million points, covering an area of more than 6,000 m<sup>2</sup> in six indoor spaces 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 sorted as structural or movable elements, which are further divided 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 data set can be requested from <http://buildingparser.stanford.edu/dataset.html>. To learn the performance of different algorithms on this data set, see [2], [70], [100], and [119].

#### VAIHINGEN POINT CLOUD SEMANTIC LABELING DATA SET

The Vaihingen Point Cloud Semantic Labeling Data Set [31] is the most well-known published benchmark data set in the photogrammetry and remote sensing field in recent years. A collection of ALS point clouds, it consists 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

**TABLE 2. AN OVERVIEW OF PCS AND PCSS APPLICATIONS SORTED ACCORDING TO DATA ACQUISITIONS.**

	<b>URBAN</b>	<b>FOREST</b>	<b>INDUSTRY</b>	<b>INDOOR</b>
<b>IMAGE DERIVED</b>	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)
<b>ALS</b>	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)		
<b>MLS</b>	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)
<b>TLS</b>	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)
<b>RGB-D</b>				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)
<b>InSAR</b>	Building/building structure: [47] (2015/C), [45] (2012/C), [46] (2014/C)	Tree PCS: [48] (2015/C)		
<b>NOT MENTIONED DATA</b>			[129](2005/HT), [130] (2015/R), [131] (2018/R)	

RG: region growing; R: RANSAC; C: clustering based; O: oversegmentation; ML: machine learning; DL: deep learning.

the median point density is 6.7 pts/m<sup>2</sup> [31]. This data set 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 data set has significantly fewer points compared with the Semantic3D.net and S3DIS data sets, it is an influential ALS data set for photogrammetry and remote sensing. The data set can be requested from <http://www2.isprs.org/commissions/comm3/wg4/3d-semantic-labeling.html>.

#### PARIS-LILLE-3D

The Paris-Lille-3D data set [36], published in 2018, is a new benchmark for PCSS. It is an MLS point cloud data set with more than 140 million labeled points, including 50 different urban object classes along 2 km of streets in two French

cities, Paris and Lille. As an MLS data set, it also could be used for autonomous vehicles. As this is a recent data set, only a few validated results are shown on the related website. This data set is available at <http://npm3d.fr/paris-lille-3d>.

#### SCANNET

ScanNet [43] is an instance-level indoor RGB-D data set that includes both 2D and 3D data. In contrast to the benchmarks already mentioned, ScanNet is a collection of labeled voxels rather than points or objects. ScanNet v2, the newest version of ScanNet, has collected 1,513 annotated scans with approximately 90% surface coverage. In the semantic segmentation task, this data set is marked in 20 classes of annotated 3D voxelized objects. Each class corresponds to one category of furniture. This

data set can be requested from <http://www.scan-net.org/>. To learn about the performance of different algorithms on this data set, refer to [70], [120], [123], and [124].

### POINT CLOUD SEGMENTATION TECHNIQUES

PCS algorithms are based mainly on strict handcrafted features from geometric constraints and statistical rules. The main process of PCS aims at grouping raw 3D points into nonoverlapping regions. Those regions correspond to specific structures or objects in each 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.

#### 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 in 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 extracted by edge detection. 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, [141] extracted close contours from a binary edge map for fast segmentation. Reference [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, because they are so simple, enable a fast PCS. However, this performance can be maintained only when simple scenes with ideal points are provided (e.g., low noise, even density). Some are suitable for range images only rather than 3D points. Thus, this approach is now rarely applied for dense or large-area point cloud data sets. 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].

### REGION GROWING

Region growing, a classical PCS method, is still widely used. It combines features from two points or two region units to measure the similarities among pixels (2D), points (3D), or voxels (3D) and merges them 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 coarse segmentation 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 and 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) selecting seed points or seed units; and 2) region growing, driven by certain principles. To design a region-growing algorithm, three crucial factors should be considered: 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 the 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 randomly picked from the data set after manually filtering areas near edges. Dong et al. [104] chose normal vectors and the distance between two units.

For the growth unit factor, one of three strategies is commonly applied. The first involves single points; the second uses region units, e.g., voxel grids and octree structures; and the third is based on hybrid units. Selecting single points as region units was the main approach in the early stages [106], [138]. However, for massive point clouds, pointwise 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. In efforts to balance accuracy and efficiency, researchers proposed and tested hybrid units. For example, Xiao et al. [101] combined single points with subwindows as growth units to detect planes. Dong et al. [104] used a hybrid region-growing algorithm based on units of both single points and supervoxels to realize coarse segmentation before global energy optimization.

Since many region-growing algorithms aim at plane segmentation, the usual practice in seed point selection is to design a fitting plane for a certain point and its neighbor points first and then choose 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 significant problem for region growing [93]. The accuracy of these algorithms depends on the growth criteria and locations of the seeds, which should be predefined and adjusted for different data sets. In addition, these algorithms are computationally intensive and may require a reduction in data volume for a tradeoff between accuracy and efficiency.

### MODEL FITTING

The core idea of model fitting is matching the point clouds to different primitive geometric shapes. Thus, model fitting has been normally regarded as a shape-detection or -extraction method. However, when dealing with scenes having 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).

#### HT

HT is a classical feature-detection technique in digital image processing. It was initially presented in [148] for line detection in 2D images. The HT technique involves three main steps [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 representing the inliers in the parameter space
- 3) selecting the cell with the local maximal score and using the parameter coordinates of that cell to represent a geometric segment in original space. The most basic version of HT is the Generalized HT (GHT), also called *the Standard HT (SHT)*, which is introduced in [150]. To avoid the infinite slope problem and simplify the computation, the GHT uses an angle-radius parameterization instead of the original slope-intercept form. The GHT is based on

$$\rho = x \cos(\theta) + y \sin(\theta), \quad (1)$$

where  $x$  and  $y$  are the image coordinates of a corresponding sample pixel,  $\rho$  is the distance between the origin and the line through the corresponding pixel, and  $\theta$  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. In 3D space, compared with the 2D form, there is one more angle parameter,  $\phi$ :

$$\rho = x \cos(\theta) \sin(\phi) + y \sin(\theta) \sin(\phi) + z \cos(\phi), \quad (2)$$

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

One of the major disadvantages of the 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 the HT by reducing the cost of the voting process [71]. Such algorithms include the Probabilistic HT (PHT) [152], Adaptive PHT [153], Progressive PHT [154], Randomized HT (RHT) [149], and Kernel-Based HT (KHT) [155]. Like streamlining computational effort, choosing a proper accumulator representation is an effective 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 in a way similar to plane detection. Reference [157] presents a comprehensive introduction to sphere recognition based on HT methods.

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].

#### RANSAC

Several reviews about the RANSAC technique, the other popular model-fitting method [158], have been published. More about members of the RANSAC family and their performance can be found in [159]–[161]. The RANSAC-based algorithm has two main phases. In the first, it generates a hypothesis from random samples (hypothesis generation). In the second, it uses the data to verify the hypothesis (hypothesis evaluation/model verification) [159], [160]. As in

the case of HT-based methods, models have to be manually defined or selected before the first phase. In PCS, depending on the structure of 3D scenes, 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 noncollinear points determine a plane. The plane model can be represented by

$$aX + bY + cZ + d = 0, \quad (3)$$

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

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

$$\hat{M} = \arg \min_M \left\{ \sum_{d \in \mathcal{D}} \text{Loss}(\text{Err}(d; M)) \right\}, \quad (4)$$

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

Because of random sampling, RANSAC-based algorithms do not require complex optimization or ample memory resources. The RANSAC method has two main advantages over HT methods in 3D PCS: it is more efficient, and it detects a higher percentage of objects [74]. Moreover, RANSAC algorithms are able to process data with a high amount of noise and even outliers [162]. For PCS, as with HT and region growing, the RANSAC method is widely used in plane segmentation, such as building façades [65], [66], [103], building roofs [73], and indoor scenes [102]. Some fields demand 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 used for cylindrical objects in [130] and [131].

RANSAC is a nondeterministic algorithm, and thus its main shortcoming is its spurious surface: models detected by the RANSAC-based algorithm may not exist (Figure 1). To overcome the adverse effect of RANSAC in PCS and improve the segmentation quality, a soft-threshold voting function was created in [72]. This function considers both the point-plane distance and the consistency between the normal vectors. Li et al. [102] proposed an improved RANSAC method based on normal distributions transform cells [163] to avoid the spurious surface problem in 3D PCS.

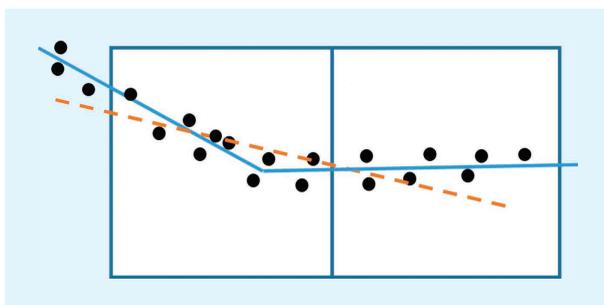
As with HT, many algorithms based on RANSAC have emerged over the past decades to further improve its efficiency, accuracy, and robustness. These approaches are categorized by their research objectives (Figure 2). The figure, originally described in [159], uses seven subclasses based on strategies. Venn diagrams are used here to describe connections between methods and strategies, since a method may use two strategies. For detailed descriptions and explanations of those strategies, see [159]. Considering that [159] is obsolete, we added two recently published methods, Extreme Value Sample Consensus [164] and Graph-Cut RANSAC [165], to the original figure to bring it up to date.

### UNSUPERVISED CLUSTERING BASED

Clustering-based methods are widely used for unsupervised PCS tasks. Strictly speaking, clustering-based methods are not grounded in a specific mathematical theory. This methodology family is made up of a mixture of different methods that share a similar aim: grouping points with similar geometric features, 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, in contrast to region-growing methods [109], seed points are not required by clustering-based approaches. Early on,  $K$ -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 tried in experiments involving PCS [78], [79], [167].

### $K$ -MEANS

$K$ -means is a basic and widely used unsupervised cluster analysis algorithm. It separates the point cloud data set into  $K$  unlabeled classes. The clustering centers of  $K$ -means are different from 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 change



**FIGURE 1.** The black dots represent points in a point cloud. Two well-estimated hypothesis planes are indicated by the blue lines running through the blue squares. A spurious plane (dotted orange line) is generated using the same threshold [102]. (Reprinted from [102].)

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] used  $K$ -means for building façade segmentation on TomoSAR point clouds.

$K$ -means 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.

### 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, the fuzzy  $C$ -means algorithm and the possibilistic  $C$ -means algorithm. This framework was tested on three point clouds including a one-scan TLS outdoor data set 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.

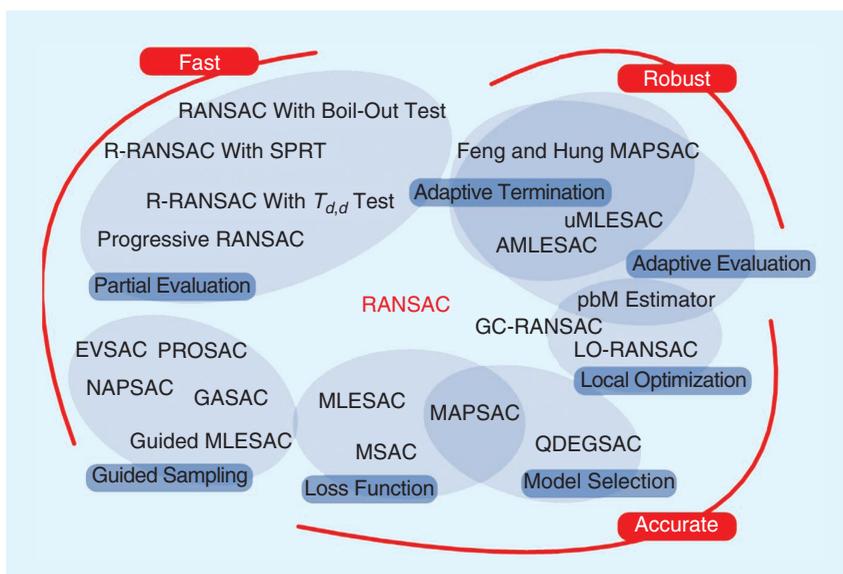
### MEAN SHIFT

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 terrains [80], [92]. Mean shift has also been adopted for use with TomoSAR point clouds, enabling the extraction of building façades and single trees [47], [48].

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

### GRAPH BASED

In 2D computer vision, the introduction of 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



**FIGURE 2.** The RANSAC family, with algorithms categorized according to their performance and basic strategies [159], [164], [165]. SPRT: sequential probability ratio test; R-RANSAC: randomized-RANSAC; MAPSAC: maximum a posterior estimation sample consensus; pbM: projection-based  $M$ -estimator; EVSAC: extreme value sample consensus; GC-RANSAC: graph-cut RANSAC; PROSAC: progressive sample consensus; NAPSAC:  $N$  adjacent points sample consensus; GASAC: genetic algorithm sample consensus; MLESAC: maximum likelihood estimation sample consensus; AMLESAC: a new MLESAC; uMLESAC: user-independent MLESAC; MSAC:  $M$ -estimator SAC; QDEGSAC: RANSAC for quasi-degenerate data; LO-RANSAC: locally optimized RANSAC. (Used with permission from [159].)

construction and partitioning problem. Inspired by graph-based methods from 2D, some studies have applied similar strategies in PCS and achieved results in different data sets.

For instance, Golovinskiy and Funkhouser [167] proposed a PCS algorithm based on minimum cut (min-cut) [171] by constructing a graph using  $k$ -nearest neighbors. The min-cut was then successfully applied for detecting outdoor urban objects [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  $\alpha$ -expansion algorithm [172] to minimize the energy function from the PCS problem. Moreover, Yao et al. [81] applied a modified normalized cut in their hybrid PCS method.

Markov random field (MRF) and conditional random field (CRF) are machine-learning approaches for solving 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, see the section “Regular Supervised Machine Learning.”

### OVERSEGMENTATION, SUPERVOXELS, AND PRESEGMENTATION

To reduce calculation costs and the harmful effects of noise, a common strategy is to oversegment 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 used in presegmentation before executing other computationally expensive algorithms. Once oversegments like supervoxels are generated, these, rather than initial points, are fed to postprocessing PCS algorithms.

The classic 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 nonuniform density cannot be guaranteed. To overcome this problem, Song et al. [174] proposed a two-stage supervoxel oversegmentation approach, Boundary-Enhanced Supervoxel Segmentation (BESS). BESS preserves the shape of the object, but it also has an obvious limitation: it assumes 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] and [175]. For PCS tasks, several studies have explored supervoxel-based presegmentation [107]–[109], [176], [177].

As mentioned in the section “Unsupervised Clustering Based,” other methods besides those involving supervoxels can also be employed in presegmentation. For example, Yao et al. [81] used mean shift to oversegment ALS data in urban areas.

## POINT CLOUD SEMANTIC SEGMENTATION TECHNIQUES

The PCSS procedure is similar to clustering-based PCS. But, in contrast to nonsemantic 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.

### REGULAR SUPERVISED MACHINE LEARNING

In this section, the term *regular supervised machine learning* refers to nondeep supervised learning algorithms. Various researchers offer comprehensive and comparative analyses of different PCSS methods based on regular supervised machine learning [87], [88], [95], [97].

Reference [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], a support vector machine (SVM) [4], [111], AdaBoost [6], [82], a cascade of binary classifiers [83], random forests [84], and Bayesian discriminant classifiers [116]. The other group is made up of statistical contextual models, such as associative and nonassociative Markov networks [85], [90], [96], CRF [86]–[88], [110], [178], simplified MRFs [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 is well described in [95]. As Figure 3 shows, the procedure entails four stages: neighborhood selection, feature extraction, feature selection, and semantic segmentation. For each stage, [95] summarized several crucial methods and tested different methods on two data sets to compare their performance. According to the authors’ experiment, in individual PCSS, the random forest classifier had a good tradeoff between accuracy and efficiency on two data sets. It should be noted that the authors of [95] used a so-called deep-learning classifier in their experiments, but that is an old neural network appearing in the time of regular machine learning, not the recent deep-learning methods described in the section “Deep Learning.”

Since individual PCSS does not consider the contextual features of points, individual classifiers work efficiently but generate unavoidable noise that causes unsmooth PCSS results. Statistical context models can mitigate this problem. 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 data set. 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 Figure 4, a graph-based contextual strategy in this framework was introduced to overcome the noise problem of initial labeling, establishing a process called *structured regularization* or *smoothing*.

For the regularization process, Li et al. [111] used a multilabel graph-cut algorithm to optimize the initial segmentation result from the SVM. Landrieu et al. [97], comparing various postprocessing methods, proved that regularization indeed improved the accuracy of PCSS.

### 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 designed with domain-specific knowledge.

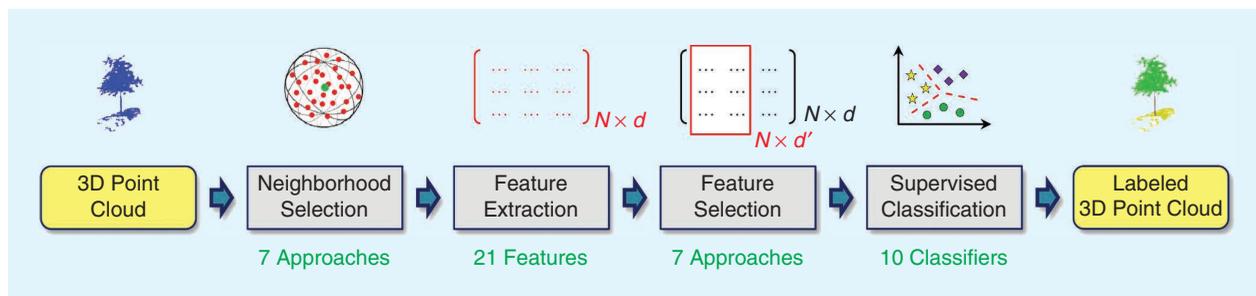
Before being applied in 3D data, deep learning appeared as an effective power in a variety of tasks in 2D computer vision and image processing, such as image recognition [180], [181], object detection [182], [183], and semantic segmentation [184], [185]. It has been attracting more interest in 3D analysis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

Standard convolutions originally designed for raster images cannot easily be directly applied to PCSS, as the point cloud is disordered and unstructured. (Unstructured is sometimes expressed as *irregular* or *nonraster*.) Thus, to solve this problem, the raw point cloud must be transformed. Depending on the format of the data ingested into neural networks, deep-learning-based PCSS approaches can be sorted into three categories: multiview based, voxel based, and point based.

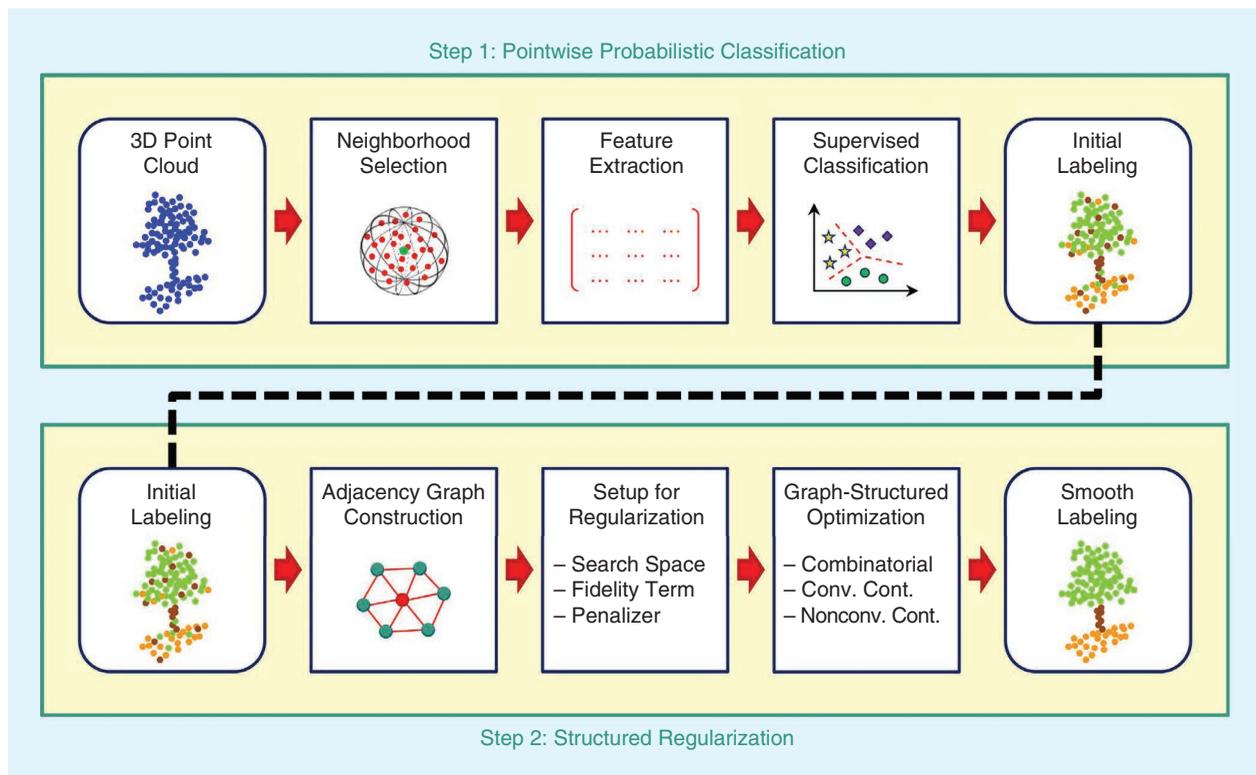
### MULTIVIEW BASED

One of the early ways to apply deep learning in 3D was with dimensionality reduction. With this method, the 3D data are represented by multiview 2D images, which can be processed based on 2D CNNs. Subsequently, the classification results can be restored to 3D. The most influential multiview deep learning in 3D analysis is multiview CNN (MVCNN) [186]. Although no experiments were performed on PCSS using the original MVCNN algorithm, 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. First, they bring about many limitations and a loss in geometric structures, as 2D multiview images are just an approximation of 3D scenes. As a result, the performance of such complex tasks PCSS could be unsatisfactory. Second, multiview projected



**FIGURE 3.** The PCSS framework described by [95]. The term *semantic segmentation* in our review is defined as “supervised classification” in [95]. (Source: [95]; reprinted with permission from Elsevier.)



**FIGURE 4.** The PCSS framework by [97]. The term *semantic segmentation* in our review is defined as “supervised classification” in [97]. Conv. Cont.: convex continuous. (Source: [97]; reprinted with permission from Elsevier.)

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 have used multiview-based deep-learning architecture for PCSS. One exception is SnapNet [9], [67], which uses full data set Semantic-8 of Semantic3D.net as the test data set. Figure 5 depicts the workflow of SnapNet. In SnapNet, the preprocessing step aims at decimating the point cloud, computing point features, and generating a mesh. Snap generation uses various virtual cameras to generate RGB images and depth composite images of the mesh. Semantic labeling realizes image semantic segmentation from the two input images by image deep learning. In the last step, 2D semantic segmentation results are projected back to 3D space, thereby enabling the acquisition of 3D semantics.

### VOXEL BASED

Combining voxels with 3D CNNs is the other approach used early on 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. The voxel structure has a low resolution compared to the point cloud. 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 has been tested only 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 full CNN, trilinear interpolation, and fully connected conditional random fields to complete the PCSS task. Figure 6 shows the framework of SegCloud and illustrates the basic pipeline for voxel-based semantic segmentation. In SegCloud, raw point clouds are voxelized in the preprocessing step. Then a 3D fully CNN (FCNN) 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 (FC-CRF) method is used to regularize previous 3D PCSS results and acquire final results. SegCloud

used to be the state-of-the-art approach in both S3DIS and Semantic3D.net, but it provided no steps for addressing high computational and memory problems related to fixed-size 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, the voxel variational encoder net (VV-NET) [189] extended the use of voxels. VV-Net uses a radial basis function-based variational autoencoder network, which provides a more information-rich representation for a point cloud compared with that provided by binary voxels. What is more, Choy et al. [70] proposed 4D CNNs (Minkowski-Nets) to process 3D videos; these are a series of CNNs for high-dimensional spaces including 4D spatiotemporal data. MinkowskiNets can also be applied for performing 3D PCSS tasks. They have achieved good performance on a series of PCSS benchmark data sets. Accuracy on ScanNet showed especially significant improvement [43].

### 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 appearing [1]–[3], [119], [120]. In these approaches, unlike those employing separated pretransformation operation in multiview- and voxel-based cases, the canonicalization is binding with the neural network architecture.

PointNet [1] is a pioneering deep-learning framework that has been performed directly on point. Unlike point cloud networks described in recently published reports, PointNet does not use a convolution operator. The basic principle of PointNet is

$$f(\{x_1, \dots, x_n\}) \approx g(h(x_1), \dots, h(x_n)), \quad (5)$$

where  $f: 2^{\mathbb{R}^N} \rightarrow \mathbb{R}$  and  $h: \mathbb{R}^N \rightarrow \mathbb{R}^K$ .  $g: \underbrace{\mathbb{R}^K \times \dots \times \mathbb{R}^K}_n \rightarrow \mathbb{R}$  is a symmetric function used to solve the ordering problem of

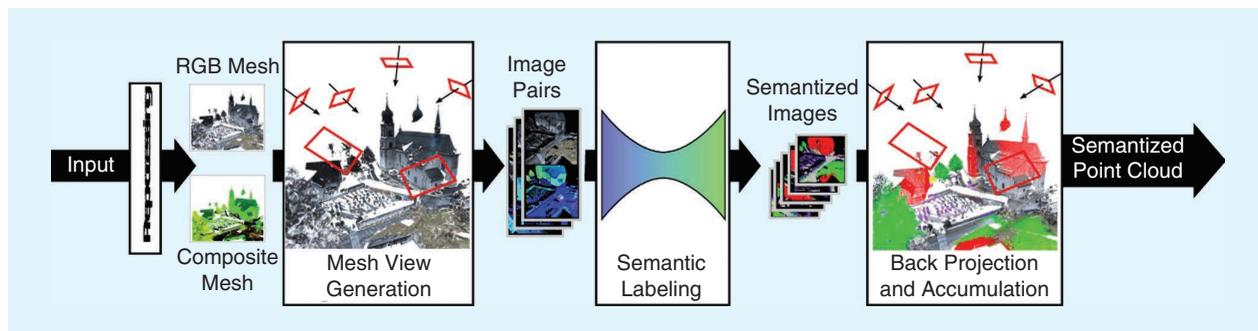


FIGURE 5. The workflow of SnapNet. (Source: [67]; reprinted with permission from Elsevier.)

point clouds. As shown in Figure 7, PointNet uses multi-layer 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 an MLP operation on global features. For the PCSS task, per-point local features are demanded in addition to global features. 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 the basis of these features, semantic labels are predicted.

Although an increasing number of newly crafted networks outperform PointNet on various benchmark data sets, PointNet is still a baseline for PCSS research. The original PointNet uses no local structure information within neighboring points. In a further study, Qi et al. [120] used a hierarchical neural network to capture local geometric

features and improve the basic PointNet model. This resulted in PointNet++. Drawing inspiration from PointNet/PointNet++, studies on 3D deep learning focus on augmenting features, especially local features/relationships among points, using knowledge from other fields to improve the performance of the basic PointNet/PointNet++ algorithms. For example, Engelmann et al. [190] employed two extensions on PointNet to incorporate larger-scale spatial context. Wang et al. [3] saw missing local features as a lingering 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), which enabled kernels to 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

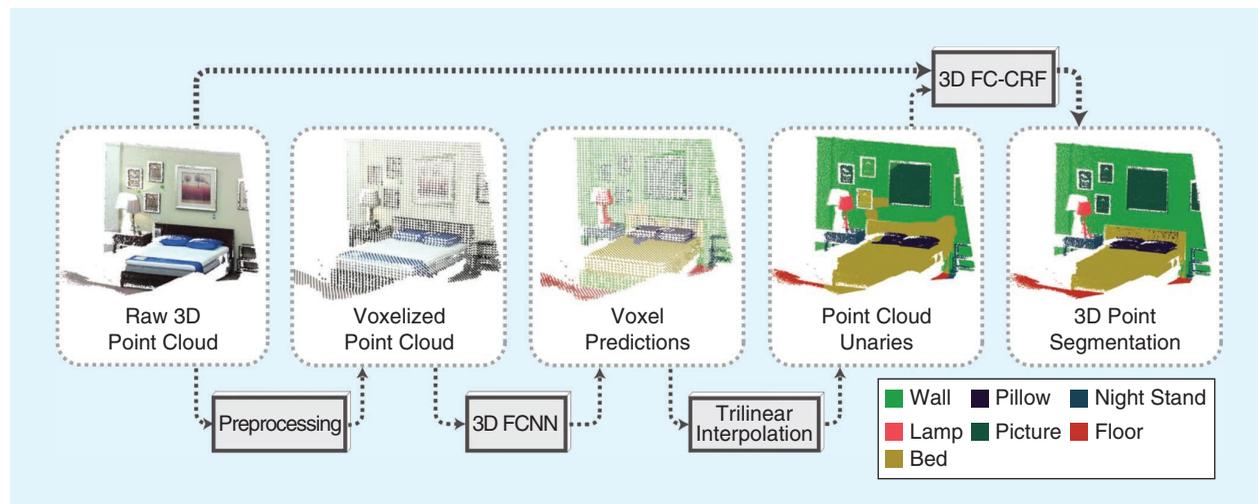


FIGURE 6. The workflow of SegCloud [98].

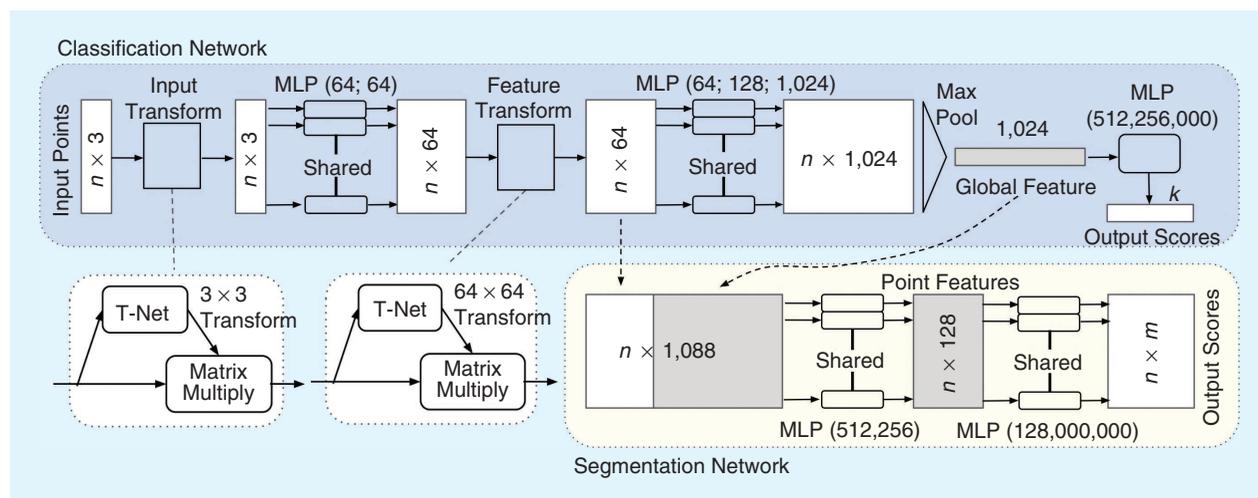


FIGURE 7. The workflow of PointNet [1]. In this figure, the classification network is used for object classification. The segmentation network is applied for the PCSS mission.

exploit richer edge features, Landrieu and Simonovsky [2] introduced the superpoint graph (SPG), which offers both compact and rich representations 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 for PCSS. Benefiting from information-rich downsampling, SPG is highly efficient for large-volume data sets.

To overcome the lack of local features represented by neighboring points in PointNet, a pointwise pyramid pooling (3 P) module was developed to capture the local feature of each point [99]. This module employed a two-direction recurrent neural network (RNN) model to integrate long-range context in PCSS tasks. This technique, 3 P-RNN, has increased overall accuracy at a negligible extra overhead cost. Komarichev et al. [125] introduced an annular convolution that could capture the local neighborhood by specifying the ring-shaped structures and directions in the computation and adapt to the geometric variability and scalability at the signal processing level. Because 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 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] uses 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.

Meanwhile, instance segmentation based on PointNet/PointNet++ can also be realized, even accompanied by PCSS. For instance, Wang et al. [127] presented the similarity group proposal network, 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, called the *generative shape proposal network*, is based on PointNet. Pham et al. [124] applied a multitask pointwise network and a multivalued 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 module, which enables PCSS and instance segmentation to take advantage of each other, leading to a win-win situation. In [123], the backbone that networks employed is also PointNet and PointNet++.

An increasing number of researchers who have chosen an alternative to PointNet nevertheless employ the convolution as a fundamental and significant component. Some of them, like [3], [112], and [125], have been introduced in this article. In addition, PointCNN used an  $\mathcal{X}$  transformation instead of symmetric functions to canonicalize the order [119], which is a generalization of CNNs

to feature learning from unordered and unstructured point clouds. Su et al. [68] provided a PCSS framework that could fuse 2D images with 3D point clouds. This framework, named *sparse lattice networks* or *SPLATNet*, preserves spatial information even in sparse regions. Recurrent slice networks (RSNs) [118] exploited a sequence of multiple  $1 \times 1$  convolution layers for feature learning and a slice pooling layer to solve the unordered problem of raw point clouds. An RNN model was then applied on ordered sequences for local dependency modeling. Te et al. [191] proposed regularized graph CNN (RGCNN) and tested it on a part segmentation data set, 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. Because 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 keep popping up. Current achievements have proved that deep learning is a great boost for the accuracy of 3D PCSS.

## HYBRID METHODS

Hybrid segmentwise methods in PCSS have attracted researchers' attention in recent years. A hybrid approach is usually made up of at least two stages. In the first stage, an oversegmentation or PCS algorithm (introduced in the section "Point Cloud Segmentation Techniques") is used as the presegmentation. In the second stage, PCSS is applied on segments from the first stage rather than points. In general, as with presegmentation in PCS, presegmentation in PCSS has two main functions: to reduce data volume and 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 little 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 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.

## DISCUSSION

### OPEN ISSUES IN SEGMENTATION TECHNIQUES

#### 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 and PCS are the differences in feature design, selection, and application. Feature selection is a tradeoff between algorithm accuracy and efficiency. Focusing on PCSS, Weinmann et al. [95] analyzed features from three perspectives: neighborhood selection (fixed or individual), feature extraction (single scale or multiscale), and classifier selection (individual or contextual classifier). Deep-learning-based algorithms face similar problems. The local feature is a significant aspect to be improved after the introduction of PointNet [1].

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

#### HYBRID

As mentioned in the section "Hybrid Methods," 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 PCSS.

#### CONTEXTUAL INFORMATION

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

#### 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 to evaluate its performance.

#### REGULAR MACHINE LEARNING VERSUS 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 for handling large data sets; 2) it requires no handcrafted feature design and selection process, a difficult task in regular machine learning; and 3) it delivers highly accurate results on public benchmark data sets. Nevertheless, deep learning is not a universal solution. Its principal shortcoming is poor interpretability. Currently, it is well-known how each type of layer (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 relevant to PCSS is autonomous driving. Another shortcoming of deep learning is the limitation on applications of deep-learning-based PCSS because deep learning relies so much on large amounts of data. Acquiring and annotating a point cloud is much more complicated than annotating 2D images. Finally, deep learning remains impractical for many uses because of a lack of appropriate data sets. Although current public data sets provide several indoor and outdoor scenes, they cannot sufficiently meet the demand in real applications.

### REMOTE SENSING MEETS COMPUTER VISION

On the basis of many published pioneering studies, researchers involved in remote sensing and general computer vision might be among the most active specialists interested in point clouds. Computer vision focuses on new algorithms to further improve accuracy. Remote sensing researchers, meanwhile, are trying to apply these techniques on different types of data sets. However, in many cases, the algorithms proposed by computer vision studies cannot be directly adopted for remote sensing.

#### EVALUATION SYSTEM

In generic computer vision, overall accuracy is a significant index. However, in some remote sensing applications, accuracy about certain objects may be of greater concern than accuracy about other objects. For instance, for urban monitoring, accuracy in representing 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.

#### MULTISOURCE DATA

As discussed in the section "An Introduction to Point Clouds," point clouds in remote sensing appear different from point clouds in computer vision. For example, airborne/spaceborne 2.5D and/or sparse point clouds are crucial components of remote sensing data, while computer vision focuses on denser full 3D.

## REMOTE SENSING ALGORITHMS

Published computer-vision algorithms are usually tested on a small-area data set with limited categories of objects. However, remote sensing applications demand large-area data with more complex and specific ground object categories. For example, in agricultural remote sensing, users expect vegetation to be identified according to specific species, a task difficult for current computer-vision algorithms to accomplish.

## NOISE AND OUTLIERS

Current computer-vision algorithms do not pay much attention to noise; in remote sensing, sensor noise is unavoidable. Currently, noise-adaptive algorithms are unavailable.

## LIMITATION OF PUBLIC BENCHMARK DATA SETS

The section "Benchmark Data Sets" lists several popular benchmark data sets. Obviously, the number of large-scale data sets with dense point clouds and rich information available to researchers has increased considerably in recent years. Some data sets, such as Semantic3D.net and S3DIS, have hundreds of millions of points. However, those benchmark data sets are still insufficient for PCSS tasks.

## LIMITED DATA TYPES

Though several large data sets for PCSS are available, there is still demand for more varied data. The real world has many more object categories than those considered in current benchmark data sets. For example, Semantic3D.net provides a large-scale urban point cloud benchmark. However, it covers cities only of a certain kind. If, for a PCSS task, researchers chose a different city with different building styles, vegetation, and even ground objects, algorithm results might in turn be different.

## LIMITED DATA SOURCES

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

## CONCLUSIONS

This article reviewed current PCSS and PCS techniques. This review not only summarized the main categories of relevant algorithms, but also introduced the acquisition methodology and evolution of point clouds. In addition, the advanced deep-learning methods proposed in recent years were compared and discussed. Due to the complexity of point clouds, PCSS is more challenging than 2D semantic

segmentation. Although many approaches are available, they have each been tested on very limited and dissimilar data sets, so it is difficult to select the optimal approach for practical applications. Deep-learning-based methods have ranked high for most benchmark-based evaluations. Yet no standard neural network is publicly available. In coming years, we can anticipate the appearance of improved neural networks for solving PCSS problems.

Most current methods have considered only point features, but in practical applications, such as remote sensing, 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.

## ACKNOWLEDGMENTS

We thank Dr. D. Cerra and P. Schwind for reviewing this article, and we acknowledge the anonymous reviewers and the associate editor for their insightful comments.

The work of Yuxing Xie was supported by a DLR-DAAD fellowship (57424731) funded by the German Academic Exchange Service and the German Aerospace Center.

The work of Xiao Xiang Zhu was jointly supported by the European Research Council (ERC), under the European Union's Horizon 2020 Research and Innovation Program (grant agreement ERC-2016-StG-714087); the Helmholtz Association, under the framework of the Young Investigators Group SiPEO (VH-NG-1018, [www.sipeo.bgu.tum.de](http://www.sipeo.bgu.tum.de)), and the Bavarian Academy of Sciences and Humanities in the framework of Junges Kolleg.

## AUTHOR INFORMATION

**Yuxing Xie** (Yuxing.Xie@dlr.de) received his B.Eng. degree in remote sensing science and technology and his M.Eng. degree in photogrammetry and remote sensing from Wuhan University, China, in 2015 and 2018, respectively. He is currently pursuing his Ph.D. degree with the Remote Sensing Technology Institute, German Aerospace Center, Wessling, Germany, and the Technical University of 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. degree in geoinformation systems from the China University of Geoscience, Beijing, in 2006, her M. Eng. degree in cartography and geoinformation at the Chinese Academy of Surveying and Mapping, Beijing, in 2009, and her 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 head of the 3D Modeling Team. In 2011, she was a guest scientist with the Institute of Photogrammetry and Remote Sensing, ETH Zürich, Switzerland. Her research interests include 3D change detection, digital surface model (DSM)

generation, 3D point cloud semantic segmentation, object extraction, and DSM-assisted building reconstruction and classification. She is a Member of the IEEE.

**Xiao Xiang Zhu** (xiaoxiang.zhu@dlr.de) received her M.Sc. degree, Dr.-Ing. degree, and habilitation in the field of signal processing from the Technical University of Munich (TUM), Germany, in 2008, 2011, and 2013, respectively. She is a professor of signal processing in Earth observation (EO) at TUM and head of the Department of EO Data Science with the Remote Sensing Technology Institute, German Aerospace Center. Since 2019, she has co-coordinated the Munich Data Science Research School. She also leads the Helmholtz Artificial Intelligence Cooperation Unit in the research field of aeronautics, space, and transport. Her main research interests are remote sensing and EO, signal processing, machine learning, and data science, with a special application focus on global urban mapping. She is an associate editor of *IEEE Transactions on Geoscience and Remote Sensing*. She is a Senior Member of the IEEE.

## 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 652–660.
- [2] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 4558–4567.
- [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." 2018. [Online]. Available: <https://arxiv.org/abs/1801.07829>
- [4] J. Zhang, X. Lin, and X. Ning, "SVM-based classification of segmented airborne lidar point clouds in urban areas," *Remote Sens.*, 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 neighbourhoods," in *Proc. Joint Int. Society for Photogrammetry and Remote Sensing Conf.*, 2015, vol. 2–3/W4, pp. 271–278.
- [6] Z. Wang et al., "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2409–2425, 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*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Cambridge, MA: MIT Press, 2011, pp. 244–252.
- [8] Y. Lu and C. Rasmussen, "Simplified Markov random fields for efficient semantic labeling of 3D point clouds," in *Proc. 2012 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 2690–2697.
- [9] A. Boulch, B. Le Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2017, pp. 17–24.
- [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," *Automat. Constr.*, vol. 19, no. 7, pp. 829–843, 2010.
- [11] R. Volk, J. Stengel, and F. Schultmann, "Building information modeling (BIM) for existing buildings—literature review and future needs," *Automat. Constr.*, 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 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," *Robot. Auton. Syst.*, 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.
- [16] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," in *Proc. 2013 6th IEEE Conf. Robotics, Automation and Mechatronics (RAM)*, pp. 225–230.
- [17] E. Grilli, F. Menna, and F. Remondino, "A review of point clouds segmentation and classification algorithms," *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. 42, pp. 339–344, 2017.
- [18] E. P. Baltsavias, "A comparison between photogrammetry and laser scanning," *ISPRS J. Photogrammetry 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-motion' photogrammetry: 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: Wiley, 2001.
- [21] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, pp. 807–814.
- [22] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2008.
- [23] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. 2007 IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8.
- [24] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, 2010.
- [25] F. Nex and F. Remondino, "UAV for 3D mapping applications: A review," *Appl. 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," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.
- [27] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from Internet photo collections," *Int. J. Comput. Vis.*, 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 *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 1625–1632.
- [29] J. Shan and C. K. Toth, *Topographic laser ranging and scanning: Principles and processing*. Boca Raton, FL: CRC, 2018.
- [30] R. Qin, J. Tian, and P. Reinartz, "3D change detection—approaches and applications," *ISPRS J. Photogrammetry 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 Comput. Vision Image Anal., Work. Group III/4-3D Scene Anal.*, 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 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 Sens.*, 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 Ann. Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. IV-1/W1, pp. 91–98, 2017.
- [35] B. Vallet, M. Brédif, A. Serna, B. Marcotegui, and N. Paparoditis, "TerraMobilita/iQmulus urban point cloud analysis benchmark," *Computers Graphics*, vol. 49, pp. 126–133, June 2015. doi: 10.1016/j.cag.2015.03.004.
- [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," *Int. J. Robotics Res.*, 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 Sens. Environ.*, 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 Sens.*, 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 Sens.*, 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 Trans. Cybern.*, 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 *Proc. 9th Int. Conf. Distributed Smart Cameras*, 2015, pp. 146–151.
- [42] E. Lachat, H. Macher, M. Mittet, T. Landes, and P. Grussenmeyer, "First experiences with Kinect v2 sensor for close range 3D modelling," *Int. Archives Photogrammetry, Remote Sens. Spat. Inform. Sci.*, 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.
- [44] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.
- [45] M. Shahzad, X. X. Zhu, and R. Bamler, "Facade structure reconstruction using spaceborne TomoSAR point clouds," in *2012 IEEE Int. Geoscience and Remote Sensing Symp.*, pp. 467–470.
- [46] X. X. Zhu and M. Shahzad, "Facade reconstruction using multi-view spaceborne TomoSAR point clouds," *IEEE Trans. Geosci. Remote Sens.*, 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 Trans. Geosci. Remote Sens.*, 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," *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, 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 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 Trans. Geosci. Remote Sens.*, 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," *Ann. 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 Geosci. Remote Sens. Lett.*, 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 Trans. Geosci. Remote Sens.*, 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 Sens.*, 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," *Int. Archives Photogrammetry Remote Sensing Spatial Inform. Sci.*, 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 Trans. Geosci. Remote Sens.*, 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 Advances in Radar and Remote Sensing (TyrrWRRS)*, pp. 106–113.
- [59] X. X. Zhu and R. Bamler, "Let's do the time warp: Multicomponent nonlinear motion estimation in differential SAR tomography," *IEEE Geosci. Remote Sens. Lett.*, 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 Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 919–923, 2011.

- [61] Y. Shi, X. X. Zhu, and R. Bamler, "Nonlocal compressive sensing-based SAR tomography," *IEEE Trans. Geosci. Remote Sens.*, 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 J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 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 Geosci. Remote Sens. Mag. (replaces Newsletter)*, 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 Trans. Geosci. Remote Sens.*, vol. 55, no. 1, pp. 14–26, 2017.
- [65] A. Adam, E. Chatzilaris, S. Nikolopoulos, and I. Kompatsiaris, "H-RANSAC: A hybrid point cloud segmentation combining 2D and 3D data," *ISPRS Ann. Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. 4, no. 2, pp. 1–8, 2018.
- [66] J. Bauer, K. Karner, K. Schindler, A. Klaus, and C. Zach, "Segmentation of building from dense 3D point-clouds," in *Proc. ISPRS Workshop Laser Scanning*, 2005, pp. 12–14.
- [67] A. Boulch, J. Guerry, B. Le Saux, and N. Audebert, "SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks," *Comput. Graph.*, vol. 71, pp. 189–198, 2018.
- [68] H. Su et al., "Splatnet: Sparse lattice networks for point cloud processing," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 2530–2539.
- [69] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 3577–3586.
- [70] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convnets: Minkowski convolutional neural networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.
- [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 Sens.*, 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 J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 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 *Proc. ISPRS Workshop Laser Scanning 2007*, vol. 36, pp. 407–412.
- [75] B. Gorte, "Segmentation of tin-structured surface models," *Int. Archives Photogrammetry Remote Sensing Spatial Inform. Sci.*, vol. 34, pp. 465–469, 2002.
- [76] A. Sampath and J. Shan, "Clustering based planar roof extraction from lidar data," in *Proc. American Society for Photogrammetry and Remote Sensing Annu. Conf.*, Reno, NV, May 2006, pp. 1–6.
- [77] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1554–1567, 2010.
- [78] S. Ural and J. Shan, "Min-cut based segmentation of airborne lidar point clouds," *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. XXXIX-B3, 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 J. Photogrammetry Remote Sensing*, vol. 94, pp. 183–193, 2014.
- [80] T. Melzer, "Non-parametric segmentation of ALS point clouds using mean shift," *J. Appl. Geodesy*, 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 *Proc. 2009 Joint Urban Remote Sensing Event*, pp. 1–6.
- [82] S. K. Lodha, D. M. Fitzpatrick, and D. P. Helmbold, "Aerial lidar data classification using AdaBoost," in *Proc. Sixth Int. Conf. 3-D Digital Imaging and Modeling (3DIM 2007)*, 2007, pp. 435–442.
- [83] M. Carlberg, P. Gao, G. Chen, and A. Zakhor, "Classifying urban landscape in aerial lidar using 3D shape analysis," in *Proc. 2009 16th IEEE Int. Conf. on Image Processing (ICIP)*, pp. 1701–1704.
- [84] N. Chehata, L. Guo, and C. Mallet, "Airborne lidar feature selection for urban classification using random forests," *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. 38, pp. 207–212, 2009.
- [85] R. Shapovalov, E. Velizhev, and O. Barinova, "Nonassociative Markov networks for 3D point cloud classification," *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, 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 Ann. Photogrammetry, Remote Sensing Spatial Inform. Sci.*, 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 J. Photogrammetry 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 J. Photogrammetry 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 *Proc. 2011 IEEE Int. Conf. Robotics and Automation*, pp. 2609–2616.
- [90] M. Najafi, S. T. Namin, M. Salzmann, and L. Petersson, "Non-associative higher-order Markov networks for point cloud classification," in *Proc. European Conf. Computer Vision*, pp. 500–515, 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 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 *Proc. 2010 IEEE Int. Conf. Image Processing*, pp. 1413–1416.
- [93] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Oc-tree-based region growing for point cloud segmentation," *ISPRS J. Photogrammetry 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 *Proc. 2012 Int. Conf. Digital Image Computing Techniques and Applications (DICTA)*, pp. 1–8.
- [95] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS J. Photogrammetry 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 *Proc. 2009 IEEE Conf. Computer Vision and Pattern Recognition*, pp. 975–982.
- [97] L. Landrieu, H. Raguét, B. Vallet, C. Mallet, and M. Weinmann, "A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds," *ISPRS J. Photogrammetry Remote Sensing*, vol. 132, pp. 102–118, 2017.
- [98] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "Seg-cloud: Semantic segmentation of 3D point clouds," in *Proc. 2017 Int. Conf. 3D Vision (3DV)*, pp. 537–547.
- [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 *Proc. European Conf. Computer Vision (ECCV)*, 2018, pp. 403–417.
- [100] L. Landrieu and M. Boussaha, "Point cloud oversegmentation with graph-structured deep metric learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 7440–7449.
- [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 Autonomous Syst.*, 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 Sens.*, vol. 9, no. 5, pp. 433, 2017.
- [103] H. Boulaassal, T. Landes, P. Grussenmeyer, and F. Tarsha-Kurdi, "Automatic segmentation of building facades using terrestrial laser data," in *Proc. ISPRS Workshop Laser Scanning 2007 and SilviLaser 2007*, pp. 65–70.
- [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 J. Photogrammetry 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 J. Photogrammetry 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 *Proc. 8th Int. Conf. Virtual Reality Continuum and Its Applications in Industry*, 2009, pp. 127–132.
- [107] Y. Xu, S. Tuttas, and U. Stilla, "Segmentation of 3D outdoor scenes using hierarchical clustering structure and perceptual grouping laws," in *Proc. 2016 9th IAPR Workshop Pattern Recognition in Remote Sensing (PRRS)*, pp. 1–6.
- [108] Y. Xu, L. Hoegner, S. Tuttas, and U. Stilla, "Voxel- and graph-based point cloud segmentation of 3D scenes using perceptual grouping laws," *ISPRS Ann. Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. 4, pp. 43–50, 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 J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 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," *Comput.-Aided Design*, vol. 41, no. 10, pp. 701–710, 2009.
- [111] Z. Li et al. "A three-step approach for TLS point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, 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 *Proc. IEEE Conf. 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 *Proc. Fifth Int. Conf. 3-D Digital Imaging and Modeling (3DIM'05)*, 2005, pp. 285–292.
- [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 Res.*, 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," *J. Vis. Commun. Image Represent.*, 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 2985–2992.
- [118] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation of point clouds," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 2626–2635.
- [119] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on x-transformed points," *Advances Neural Inform. Processing Syst.*, 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," *Advances Neural Inform. Processing Syst.*, pp. 5099–5108, 2017.
- [121] M. Jiang, Y. Wu, and C. Lu, PointSIFT: A SIFT-like network module for 3D point cloud semantic segmentation. 2018. [Online]. Available: <https://arxiv.org/abs/1807.00652>
- [122] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.
- [123] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 4096–4105.

- [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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 8827–8836.
- [125] A. Komarichev, Z. Zhong, and J. Hua, "A-CNN: Annularly convolutional neural networks on point clouds," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 7421–7430.
- [126] H. Zhao, L. Jiang, F. C.-W., and J. Jia, "Pointweb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 5565–5573.
- [127] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 2569–2578.
- [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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 3947–3956.
- [129] T. Rabbani and F. Van Den Heuvel, "Efficient Hough transform for automatic detection of cylinders in point clouds," *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, 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," *Comput. Graph.*, 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 Recog. Lett.*, 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 *Proc. European Conf. Computer Vision*, 2014, pp. 516–532.
- [133] M. De Deuge, A. Quadros, C. Hung, and B. Douillard, "Unsupervised feature learning for classification of outdoor 3D scans," in *Proc. Australasian Conf. Robotics and Automation*, 2013, pp. 1–9.
- [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 *Proc. 4th Int. Conf. Pattern Recognition, Applications and Methods*, 2014, pp. 819–824.
- [135] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robotics Res.*, 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 *Proc. European Conf. Computer Vision*, 2012, pp. 746–760.
- [137] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2D-3D-semantic data for indoor scene understanding. 2017. [Online]. Available: <https://arxiv.org/abs/1702.01105>
- [138] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. 36, pp. 248–253, 2006.
- [139] B. Bhanu, S. Lee, C. Ho, and T. Henderson, "Range data processing: Representation of surfaces by edges," in *Proc. Eighth Int. Conf. Pattern Recognition*, 1986, pp. 236–238.
- [140] X. Y. Jiang, U. Meier, and H. Bunke, "Fast range image segmentation using high-level segmentation primitives," in *Proc. Third IEEE Workshop Applications of Computer Vision*, 1996, pp. 83–88.
- [141] A. D. Sappa and M. Devy, "Fast range image segmentation by an edge detection strategy," in *Proc. Third Int. Conf. 3-D Digital Imaging and Modeling*, 2001, pp. 292–299.
- [142] M. A. Wani and H. R. Arabnia, "Parallel edge-region-based segmentation algorithm targeted at reconfigurable multiring network," *J. 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," *Innovations for Shape Analysis*, pp. 283–299, 2013.
- [144] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Anal. Mach. Intell.*, 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," *Int. Archives Photogrammetry 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," *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, 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 *Proc. 3DPVT Int. Conf.*, 2010, pp. 1–8.
- [148] P. V. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3069654, 1962.
- [149] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough Transform (RHT)," *Pattern Recog. Lett.*, 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," *Commun. 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," *Comput. Graph. Forum*, vol. 38, no. 1, pp. 167–196, 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 Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 9, pp. 911–915, 1994.
- [154] C. Galamhos, J. Matas, and J. Kittler, "Progressive probabilistic Hough transform for line detection," in *Proc. 1999 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 554–560.
- [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," *Int. Archives*

- Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. 46, pp. 33–38, 2004.
- [157] M. Camurri, R. Vezzani, and R. Cucchiara, “3D Hough transform for sphere recognition on point clouds,” *Mach. Vis. Appl.*, vol. 25, no. 7, pp. 1877–1891, 2014.
- [158] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [159] S. Choi, T. Kim, and W. Yu, “Performance evaluation of RANSAC family,” in *Proc. British Machine Vision Conf.*, 2009, pp. 1–12.
- [160] R. Raguram, J.-M. Frahm, and M. Pollefeys, “A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus,” in *Proc. European Conf. Computer Vision*, 2008, pp. 500–513.
- [161] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, “USAC: A universal framework for random sample consensus,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 2022–2038, 2013.
- [162] R. Schnabel, R. Wahl, and R. Klein, “Efficient RANSAC for point-cloud shape detection,” *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, 2007.
- [163] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” in *Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2003)*, vol. 3, pp. 2743–2748.
- [164] V. Fragoso, P. Sen, S. Rodriguez, and M. Turk, “EVSAC: Accelerating hypotheses generation by modeling matching scores with extreme value theory,” in *Proc. IEEE Int. Conf. Computer Vision*, pp. 2472–2479, 2013.
- [165] D. Barath and J. Matas, “Graph-Cut RANSAC,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 6733–6741.
- [166] S. Filin, “Surface clustering from airborne laser scanning data,” *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. 34, pp. 119–124, 2002.
- [167] A. Golovinskiy and T. Funkhouser, “Min-cut based segmentation of point clouds,” in *Proc. IEEE 12th Int. Conf. Computer Vision Workshops*, 2009, pp. 39–46.
- [168] D. Comaniciu and P. Meer, “Mean shift analysis and applications,” in *Proc. Seventh IEEE Int. Conf. Computer Vision*, 1999, vol. 2, pp. 1197–1203.
- [169] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 5, pp. 603–619, 2002.
- [170] Y. Cheng, “Mean shift, mode seeking, and clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, 1995.
- [171] Y. Boykov and G. Funka-Lea, “Graph cuts and efficient ND image segmentation,” *Int. J. Comput. Vis.*, 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,” *Int. J. Comput. Vis.*, 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 2027–2034.
- [174] S. Song, H. Lee, and S. Jo, “Boundary-enhanced supervoxel segmentation for sparse outdoor lidar data,” *Electron. Lett.*, 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 J. Photogrammetry Remote Sensing*, vol. 143, pp. 39–47, 2018.
- [176] S. C. Stein, M. Schoeler, J. Papon, and F. Worgotter, “Object partitioning using local convexity,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 304–311.
- [177] B. Yang, Z. Dong, G. Zhao, and W. Dai, “Hierarchical extraction of urban objects from mobile laser scanning data,” *ISPRS J. Photogrammetry 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,” *Int. Archives Photogrammetry, Remote Sensing Spatial Inform. Sci.*, vol. 39, pp. 161–166, 2012.
- [179] X. X. Zhu et al., “Deep learning in remote sensing: A comprehensive review and list of resources,” *IEEE Geosci. Remote Sens. Mag. (replaces Newsletter)*, vol. 5, no. 4, pp. 8–36, 2017.
- [180] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition.” 2014. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [181] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [182] R. Girshick, “Fast r-CNN,” in *Proc. IEEE Int. Conf. Computer Vision*, 2015, pp. 1440–1448.
- [183] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-CNN: Towards real-time object detection with region proposal networks,” *Advances Neural Inform. Processing Syst.*, pp. 91–99, 2015.
- [184] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
- [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 Trans. Pattern Anal. Mach. Intell.*, 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 *Proc. IEEE Int. Conf. Computer Vision*, 2015, pp. 945–953.
- [187] D. Maturana and S. Scherer, “Voxnet: A 3D convolutional neural network for real-time object recognition,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.
- [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 Trans. Graph.*, vol. 36, no. 4, pp. 1–11, 2017.
- [189] H.-Y. Meng, L. Gao, Y. Lai, and D. Manocha, “Voxel VAE net with group convolutions for point cloud segmentation.” 2018. [Online]. Available: <https://arxiv.org/abs/1811.04337>
- [190] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, “Exploring spatial context for 3D semantic segmentation of point clouds,” in *Proc. IEEE Int. Conf. Computer Vision*, 2017, pp. 716–724.

- [191] G. Te, W. Hu, A. Zheng, and Z. Guo, "RGCNN: Regularized graph CNN for point cloud segmentation," in *Proc. ACM Multimedia Conf.*, 2018, pp. 746–754.
- [192] A. X. Chang et al., Shapenet: An information-rich 3D model repository. 2015. [Online]. Available: <https://arxiv.org/abs/1512.03012>
- [193] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, Graph neural networks: A review of methods and applications. 2018. [Online]. Available: <https://arxiv.org/abs/1812.08434>
- [194] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, A comprehensive survey on graph neural networks. 2019. [Online]. Available: <https://arxiv.org/abs/1901.00596>
- [195] L. Li, M. Sung, A. Dubrovina, L. Yi, and L. J. Guibas, "Supervised fitting of geometric primitives to 3D point clouds," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 2652–2660.

**GRS**