Deep Learning for 3D Point Cloud Understanding: A Survey

Haoming Lu[1], Humphrey Shi[2,1]

[1]University of Illinois at Urbana-Champaign, [2]University of Oregon

Abstract

The development of practical applications, such as autonomous driving and robotics, has brought increasing attention to 3D point cloud understanding. While deep learning has achieved remarkable success on image-based tasks, there are many unique challenges faced by deep neural networks in processing massive, unstructured and noisy 3D points. To demonstrate the latest progress of deep learning for 3D point cloud understanding, this paper summarizes recent remarkable research contributions in this area from several different directions (classification, segmentation, detection, tracking, flow estimation, registration, augmentation and completion), together with commonly used datasets, metrics and state-of-the-art performances. More information regarding this survey can be found at: https://github.com/SHI-Labs/3D-Point-Cloud-Learning.

1. Introduction

Deep learning has shown outstanding performance in a wide range of computer vision tasks in the past years, especially image tasks. Meanwhile, in many practical applications, such as autonomous vehicles (Figure 1 shows a point cloud collected by an autonomous vehicle), we need more information than only images to obtain a better sense of the environment. 3D data from lidar or RGB-D cameras are considered to be a good supplement here. These devices generate 3D geometric data in the form of point clouds. With the growing demand from industry, utilization of point clouds with deep learning models is becoming a research hotspot recently.

In constrast to image data, point clouds do not directly contain spatial structure, and deep models on point clouds must therefore solve three main problems: (1) how to find a representation of high information density from a sparse point cloud, (2) how to build a network satisfying necessary restrictions like size-variance and permutation-invariance, (3) how to process large volumes of data with lower time and computing resource consumption. PointNet [79] is one of the representative early attempts to design a novel deep network for comsumption of unordered 3D point sets by taking advantage of MLP and T-Net. PointNet, together with its improved version PointNet++ [80], inspired a lot of follow-up works.

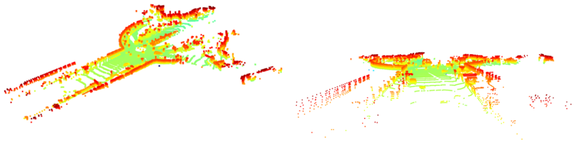

Figure 1. Point cloud data collected from outdoor scene, shown from two distinct angles.

Fundamental tasks in images, such as classification, segmentation and object detection also exist in point clouds. Most solutions to these problems benefit from research findings on the image side, while adequate adaptions are inevitable to suit the characteristics of 3D data. In this paper, recent works on point clouds are divided into the following categories: classification, segmentation, detection, matching and registration, augmentation, completion and reconstruction. Detailed descriptions of each category will be provided in the following sections.

A growing number of datasets are available for different tasks on point clouds. ShapeNet [5] and ModelNet [113] are two early datasets consisting of clean 3D models. These early datasets suffer from the lack of generalization. However, it is necessary to consider disturbance including noise and missing points to develop robust models. With that in mind, datasets such as ScanNet [10] and KITTI [22] are then created from scans of the actual environment. Datasets designed for autonomous vehicle tasks, like nuScenes [4] and Lyft [44], are further generalized by involving various environments at different times. Currently, ever more datasets are being proposed in order to meet the increasing demands of distinct niches.

The structure of this paper is as follows. Section 2 introduces existing 3D datasets and corresponding metrics for different tasks. Section 3 includes a survey of 3D shape classification methods. Section 4 reviews methods for 3D semantic segmentation and instance segmentation. Section

5 presents a survey of methods for 3D object detection and its derivative task. Section 6 introduces recent progress in 3D point cloud matching and registration. Section 7 provides a review of methods to improve data quality. Finally, section 8 concludes the paper.

2. Datasets and metrics

Datasets are of great importance in deep learning methods for 3D point cloud data. First, well-designed datasets provide convictive evaluation and comparison among different algorithms. Second, datasets with richer content and metadata help define more complicated tasks and raise new research topics. In this section, we will briefly introduce some most commonly used datasets and evaluation metrics.

2.1. Datasets

Table 1 shows the most commonly used 3D point cloud datasets for three matured tasks (classification, segmentation and detection), which will be mentioned often in the following sections. We will also introduce each of them with more details.

ShapeNet ShapeNet [5] is a rich-annotated dataset with 51300 3D models in 55 categories. It consists of several subsets. ShapeNetSem, which is one of the subsets, contains 12000 models spread over a broader set of 270 categories. This dataset, together with ModelNet40 [113], are relatively clean and small, so they are usually used to evaluate the capacity of backbones before applied to more complicated tasks.

ModelNet40 The ModelNet [113] project provides three benchmarks: ModelNet10, ModelNet40 and Aligned40. The ModelNet40 benchmark, where "40" indicates the number of classes, is the most widely used. To find the most common object categories, the statistics obtained from the SUN database [115] are utilized. After establishing the vocabulary, 3D CAD models are collected with online search engines and verified by human workers.

S3DIS The Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset is composed of 5 large-scale indoor scenes from three buildings to hold diverse in architectural style and appearance. The point clouds are automatically generated without manual intervention. 12 semantic elements including structural elements (floor, wall, etc.) and common furniture are detected.

Semantic3D Semantic3D [30] is the largest 3D point cloud dataset for outdoor scene segmentation so far. It contains over 4 billion points collected from around $110000m^2$ area with a static lidar. The natural of outdoor scene, such as the unevenly distribution of points and massive occlusions, makes the dataset challenging.

ScanNet ScanNet [10] is a video dataset consists of 2.5 million frames from more than 1000 scans, annotated with camera poses, surface reconstructions and instance-level semantic segmentation. The dataset provides benchmarks for mutiple 3D scene understanding tasks, such as classification, semantic voxel labeling and CAD model retrieval.

KITTI The KITTI [22, 23, 21, 69] vision benchmark suite is among the most famous benchmarks with 3D data. It covers benchmarks for 3D object detection, tracking and scene flow estimation. The multi-view data are captured with an autonomous driving platform with two high-resolution color and gray cameras, a Velodyne laser scanner and a GPS localization system. Only three kinds of objects which are important to autonomous driving are labelled: cars, pedestrians and cyclists.

Other datasets There are some other datasets of high quality but not widely used, such as Oakland [71], iQmulus [97] and Paris-Lille-3D [85]. 3DMatch [131] pushed the research in 3D matching and registration, which is a less popular direction in the past period. Recently, the rising demand from industry of autonomous driving has spawned several large-scale road-based datasets, represented by nuScenes [4], Lyft Level 5 [44] and Waymo Open Dataset [95]. They proposed complicated challenges requiring to leverage multi-view data and related metadata. The development of datasets is helping reduce the gap between research and practical applications.

2.2. Metrics

The comparison between different algorithms requires certain metrics. It is important to design and select appropriate metrics. Well-designed metrics can provide valid evaluation of different models, while unreasonable metrics might lead to incorrect conclusions.

Table 2 lists widely used metrics in different tasks. For classification methods, overall accuracy and mean accuracy are most frequently used. Segmentation models can be analyzed by accuracy or (m)IoU. In detection tasks, the result are usually evaluated region-wise, so (m)IoU, accuracy, precision and recall could apply. MOTA and MOTP are specially designed for object tracking modelts, while EPE is for scene for estimation. ROC curves, which is the derivative of precision and recall, help evaluate the performance of 3D match and registration models. Besides, visualization is always an effective supplement of numbers.

3. Classification

3.1. Overview

Classification on point clouds is commonly known as 3D shape classification. Similar to image classification models, models on 3D shape classification usually first generate a global embedding with an aggregation encoder, then pass the embedding through several fully connected layers to obtain the final result. Most 3D shape classification methods

Table 1. Commonly used 3D point cloud datasets in recent works

Dataset	Task	Classes	Scale	Feature	Year
ShapeNet [5]	Classification	55	51300 models	The categories are selected according to WordNet [70] synset.	2015
ModelNet40 [113]	Classification	40	12311 models	The models are collected with online search engines by querying for each established object category.	2015
S3DIS [1]	Segmentation	12	215 million points	Points are collected in 5 large-scale indoor scenes from 3 different buildings.	2016
Semantic3D [30]	Segmentation	8	4 billion points	Hand-labelled from a range of diverse urban scenes.	2017
ScanNet [10]	Segmentation	20	2.5 million frames	Collected with a scalable RGB-D capture system with automated surface reconstruction and crowdsourced semantic annotation.	2017
KITTI [22, 23, 21, 69]	Detection Tracking	3	80256 objects	Captured by a standard station wagon equipped with two cameras, a Velodyne laser scanner and a GPS localization system driving in different outdoor scenes.	2012
nuScenes [4]	Detection Tracking	23	1.4M objects	Captured with full sensor suite (1x LIDAR, 5x RADAR, 6x camera, IMU, GPS); 1000 scenes of 20s each.	2019
Waymo Open Dataset [95]	Detection Tracking	4	12.6M objects with tracking ID	Captured with 1 mid-range lidar, 4 short-range lidars and 5 cameras (front and sides); 1,950 segments of 20s each, collected at 10Hz.	2019

are tested with clean 3D models (as in Figure 2). Based on the point cloud aggregation method, classification models can be generally divided into two categories: projection-based methods and point-based methods.

3.2. Projection-based Methods

Projection-based methods project the unstructured 3D point clouds into specific presupposed modality (e.g. voxels, pillars), and extract features from the target format, which allows them to benefit from the previous research findings in the corresponding direction.

3.2.1 Multi-view representation

MvCNN [94] is a method based on a multi-view representation of point clouds. A 3D point cloud is represented by a group of 2D images by rendering snapshots from different angles. Each image in the group will be passed through a CNN to extract view-based features, pooled across views and passed through another CNN to build a compact descriptor. While MVCNN does not distinguish different views, it is helpful to consider the relationship among views. GVCNN [20] is a method that takes advantage of this relationship. By quantifying the discrimination of views, we are able to divided the set of views into groups based on their discrimination scores. The view descriptors will be passed through intra-group pooling and cross-group fusion for prediction. Aside from the models mentioned above, [128] and [124] also improve the recognition accuracy with multi-view representation.

3.2.2 Volumetric representation

VoxNet [67] is an early method using the volumetric representation. In this method, each point (x, y, z) is projected into a corresponding discrete voxel point (i, j, k). Each point cloud will be mapped into an occupancy grid of $32 \times 32 \times 32$ voxels, and the grid will then be passed through two 3D convolutional layers to obtain the final representation.

VoxNet simply uses adaption of CNN layers for the prediction head, which leads to potential loss of detailed spatial information. 3D ShapeNet [5] proposed a belief-based deep convolutional network to learn the distribution of point clouds in different 3D shapes. In this method, 3D shapes are represented by the probability distributions of binary variables on grids.

While volumetric methods already achieve satisfactory performance, most suffer from the cubic growth of computation complexity and memory footprint, hence the resolu-

Table 2. Commonly used metrics for different tasks. In this table, N denotes the number of samples, C denotes the number of categories, IDS denotes the number of identity switches, $I_{i,j}$ denotes the number of points that are from ground truth class/instance i and labelled as j, $TP/TN/FP/FN$ stands for the number of true positives, true negatives, false positives and false negatives respectively. Higher metrics indicate better results if not specified otherwise.

Metric	Formula	Explanation
Accuracy	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$	Accuracy indicates how many predictions are correct over all predictions. "Overall accuracy (OA)" indicates the accuracy on the entire dataset.
mACC	$mACC = \frac{1}{C}\sum_{c=1}^{C} Accuracy_c$	The mean of accuracy on different categories, useful when the categories are imbalanced.
Precision	$Precision = \frac{TP}{TP+FP}$	The ratio of correct predictions over all predictions.
Recall	$Recall = \frac{TP}{TP+FN}$	The ratio of correct predictions over positive samples in the ground truth.
F1-Score	$F_1 = 2 \times \frac{Precision \cdot Recall}{Precision+Recall}$	The harmonic mean of precision and recall.
IoU	$IoU_i = \frac{I_{i,i}}{\sum_{c=1}^{C}(I_{i,c}+I_{c,i})-I_{i,i}}$	Intersection over Union (of class/instance i). The intersection and union are calculated between the prediction and the ground truth.
mIoU	$mIoU = \frac{1}{C}\sum_{c=1}^{C} IoU_i$	The mean of IoU on all classes/instances.
MOTA	$MOTA = 1 - \frac{FN+FP+IDS}{TP+FN}$	Multi-object tracking accuracy (MOTA) synthesizes 3 error sources: false positives, missed targets and identity switches, and the number of ground truth (as $TP + FN$) is used for normalization.
MOTP	$MOTP = \frac{\sum_{i,t} e_{i,t}}{\sum_t d_t}$	Multi-object tracking precision (MOTP) indicates the precision of localization. d_t denotes the number of matches at time t, and $e_{i,t}$ denotes the error of the i-th pair at time t.
EPE	$EPE = \|\|\hat{sf} - sf\|\|_2$	End point error (EPE) is used in scene flow estimation, also referred as EPE2D/EPE3D for 2D/3D data respectively. $\hat{sf}$ denotes the predicted scene flow vector while sf denotes the ground truth.

Figure 2. 3D models from ShapeNet [5]. ShapeNet contains large-scale 3D models with manually verified annotation.

tion of the grid is strictly limited. OctNet [84] improved the efficiency by introducing a hybrid grid-octree structure to hierarchically partition point clouds. A point cloud is represented by several octrees along a regular grid, each octree is encoded as a bit string, and features are generated through naive arithmetic. Inspired by OctNet, OCNN [99] then proposed a method that introduces 3D-CNNs to extract features from octrees.

Methods based on volumetric representations as mentioned above are naturally coarse as only a small fraction of voxels are non-empty and the detailed context inside each voxel is hardly collected. The balance between resolution and computation is difficult to achieve in practice.

3.2.3 Basis point set

BPS [75] proposed a new approach that breaks the convention that point clouds, even with various sizes, are usually projected onto a grid of same size. In BPS, input points are first normalized into a unit ball, then a group of points is randomly sampled to make up a basis point set (BPS). The sampled BPS is constant for all point clouds in a dataset. For a given point cloud X, each point x_i is represented by the Euclidean distance between itself and its nearest neighbor in BPS. By passing such representation through the last two fully connected layers of PointNet, the model achieves performance similar to that of the original PointNet design.

3.3. Point-based Methods

Compared with projection-based methods that aggregate points from a spatial neighborhood, point-based methods attempt to learn features from individual points. Most of recent work focuses on this direction.

3.3.1 MLP networks

PointNet [79] is a famous architecture that takes advantage of multi-layer perceptrons (MLPs). The input (an $n \times 3$ 2D tensor) is first multiplied by an affine transformation matrix predicted by a mini-network (T-Net) to hold invariance under geometric transformations. The point set is then passed through a group of MLPs followed by another joint alignment network, and a max-pooling layer to obtain the final global feature. This backbone can be used for both classification and segmentation prediction. For classification, the global feature is passed through an MLP for output scores. For segmentation, the concatenations of the global feature and different levels of intermediate features from each point are passed through an MLP for the classification result of each point. Conventional CNNs take features at different scales by a stack of convolutional layers; inspired by that, PointNet++ [80] is proposed. In this work, the local region of a point x is defined as the points within a sphere centered at x. One set abstraction level here contains a sampling layer, a grouping layer to identify local regions and a PointNet layer. Stacking such set abstraction levels allows us to extract features hierarchically as CNNs for image tasks do.

The simple implementation and promising performance of PointNet [79] and PointNet++ [80] inspired a lot of follow-up work. PointWeb [136] is adapted from PointNet++ and improves quality of features by introducing Adaptive Feature Adjustment (AFA) to make use of context information of local neighborhoods. In addition, SRN [16] proposed Structural Relation Network (SRN) to equip PointNet++, and obtained better performance.

3.3.2 Convolutional networks

Convolution kernels on 2D data can be extended to work on 3D point cloud data. As mentioned before, VoxNet [67] is an early work that directly takes advantage of 3D convolution.

A-CNN [46] proposed another way to apply convolution on point clouds. In order to prevent redundant information from overlapped local regions (the same group of neighboring points might be repeatedly included in regions at different scales), A-CNN proposed a ring-based scheme instead of spheres. To convolve points within a ring, points are projected on a tangent plane at a query point q_i, then ordered in clockwise or counter-clockwise direction by making use of cross product and dot product, and eventually a 1-D convolution kernel will be applied to the ordered sequence. The output feature can be used for both classification and segmentation as in PointNet.

RS-CNN [62] is another convolutional network based on relation-shape convolution. An RS-Conv kernel takes a neighborhood around a certain point as its input, and learns the mapping from naive relations (e.g. Euclidean distance, relative position) to high-level relations among points, and encodes the spatial structure within the neighborhood with the learned mapping.

In PointConv [112], the convolution operation is defined as finding a Monte Carlo estimation of the hidden continuous 3D convolution w.r.t. an importance sampling. The process is composed with a weighting function and a density function, implemented by MLP layers and a kernelized density estimation. Furthermore, the 3D convolution is reduced into matrix multiplication and 2D convolution for memory and computational efficiency and easy deployment. A similar idea is used in MCCNN [34], where convolution is replaced by a Monte Carlo estimation based on the density function of the sample.

Geo-CNN [49] proposed another way to model the geometric relationship among neighborhood points. By taking six orthogonal bases, the space will be separated into eight quadrants, and all vectors in a specific quadrant can be composed by three of the bases. Features are extracted independently along each direction with corresponding direction-associated weight matrices, and are aggregated based on the angle between the geometric vector and the bases. The feature of some specific point at the current layer is the sum of features of the given point and its neighboring edge features from the previous layer.

In SFCNN [83], the input point cloud is projected onto regular icosahedral lattices with discrete sphere coordinates, hence convolution can be implemented by maxpooling and convolution on the concatenated features from vertices of spherical lattices and their neighbors. SFCNN holds rotation invariance and is robust to perturbations.

3.3.3 Graph networks

Graph networks consider a point cloud as a graph and the vertices of the graph as the points, and edges are generated based on the neighbors of each point. Features will be learned in spatial or spectral domains.

ECC [93] first proposed the idea of considering each point as a vertex of the graph and connected edges between pairs of points that are "neighbors". Then, edge conditioned convolution (ECC) is applied with a filter generating network such as MLP. Neighborhood information is aggregated by maxpooling and coarsened graph will be generated with VoxelGrid [86] algorithm. After that, DGCNN [105] uses a MLP to implement EdgeConv, followed by channel-wise symmetric aggregation on edge features from the neighborhood of each point, which allows the graph to be dynamically updated after each layer of the network.

Inspired by DGCNN, Hassani and Haley [31] proposed an unsupervised multi-task approach to learn shape features. The approach consists of an encoder and an decoder, where the encoder is constructed from multi-scale graphs, and the decoder is constructed for three unsupervised tasks (clustering, self-supervised classification and reconstruction) trained by a joint loss.

ClusterNet [6] uses rigorously rotation-invariant (RRI) module to generate rotation-invariant features from each point, and an unsupervised agglomerative hierarchical clustering method to construct hierarchical structures of a point cloud. Features of sub-clusters at each level are first learned with an EdgeConv block, then aggregated by maxpooling.

3.3.4 Other networks

Aside from OctNet [84], which uses octrees on voxel grids to hierarchically extract features from point clouds, Kd-Net [45] makes use of K-d trees to build a bottom-up encoder. Leaf node representations are normalized 3D coordinates (by setting the center of mass as origin and rescaled to $[-1, 1]^3$), and non-leaf node representations are calculated from its children nodes with MLP. The parameters of MLPs are shared within each level of the tree. Moreover, 3DContextNet [131] proposed another method based on K-d trees. While non-leaf representations are still computed with MLP from its children, the aggregation at each level is more complicated for considering both local cues and global cues. The local cues concern points in the corresponding local region, and the global cues concern the relationship between current position and all positions in the input feature map. The representation at the root will be used for prediction.

RCNet [111] introduced RNN to point cloud embedding. The ambient space is first partitioned into parallel beams, each beam is then fed into a shared RNN, and the output subregional features are considered as a 2D feature map and processed by a 2D CNN.

SO-Net [56] is a method based on the self-organized map (SOM). A SOM is a low-dimensional (two-dimensional in the paper) representation of the input point cloud, initialized by a proper guess (dispersing nodes uniformly in a unit ball), and trained with unsupervised competitive learning. A k-nearest-neighbor set is searched over the SOM for each point, and the normalized KNN set is then passed through a series of fully connected layers to generate individual point features. The point features are used to generate node features by maxpooling according to the association in KNN search, and the node features are passed through another series of fully connected layers and aggregated into a global representation of the input point cloud.

3.4. Experiments

Different methods choose to test their models on various datasets. In order to obtain a better comparison among methods, we select datasets that most methods are tested on, and list the experiment results for them in Table 3.

4. Segmentation

4.1. Overview

3D segmentation intends to label each individual point, which requires the model to collect both global context and detailed local information at each point. Figure 3 shows some examples from S3DIS [1] dataset. There are two main tasks in 3D segmentation: semantic segmentation and instance segmentation.

Since a large number of classification models are able to achieve very high performance on popular benchmarks, they tend to test their backbone on segmentation datasets to prove the novel contribution and generalization ability. We will not reintroduce these models if they have been mentioned above. There are also some models that benefit from the jointly training on multiple tasks, and we will discuss these methods later in section 3.4.

4.2. Semantic Segmentation

Similar to 3D shape classification models, based on how the raw point cloud is organized, semantic segmentation methods can be generally divided into projection-based methods and point-based methods.

Table 3. Experiment results on ModelNet40 classification benchmark. “OA” stands for overall accuracy and “mACC” stands for mean accuracy.

Methods	ModelNet40(OA)	ModelNet40(mAcc)
PointNet [79]	89.2%	86.2%
PointNet++ [80]	90.7%	90.7%
PointWeb [136]	92.3%	89.4%
SRN [16]	91.5%	-
Pointwise-CNN [37]	86.1%	81.4%
PointConv [112]	92.5%	-
RS-CNN [62]	92.6%	-
GeoCNN [49]	93.4%	91.1%
A-CNN [46]	92.6%	90.3%
Hassani and Haley [31]	89.1%	-
ECC [93]	87.4%	83.2%
SFCNN [83]	91.4%	-
DGCNN [105]	92.2%	90.2%
ClusterNet [6]	87.1%	-
BPS [75]	91.6%	-
KD-Net [45]	91.8%	88.5%
3DContextNet[131]	91.1%	-
RCNet [111]	91.6%	-
SO-Net [56]	90.9%	87.3%

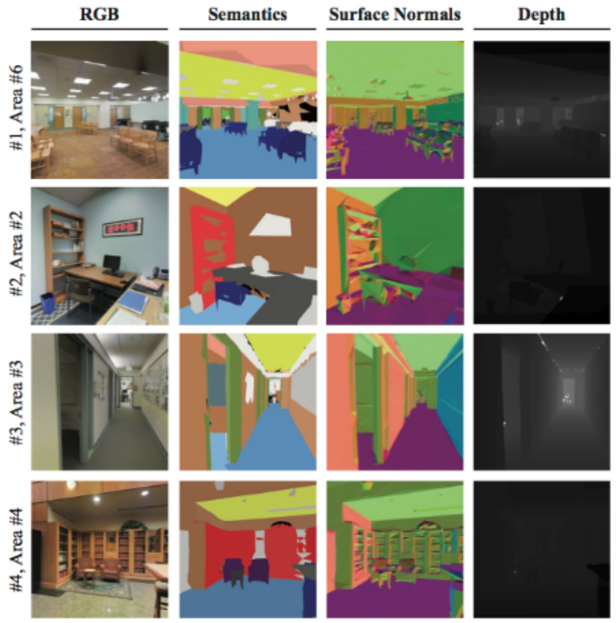

Figure 3. Stanford Large-Scale 3D Indoor Spaces Dataset [1] (S3DIS).

4.2.1 Projection-based methods

Huang and You [38] project the input point cloud into occupancy voxels, which are then fed into a 3D convolutional network to generate voxel-level labels. All points within a voxel are assigned with the same semantic label as the voxel. ScanComplete [12] utilizes fully convolutional networks to adapt to different input data sizes, and deploys a coarse-to-fine strategy to improve the resolution of predictions hierarchically. VV-Net [68] also transfers unordered points into regular voxel grids as the first step. After that, the local geometry information of each voxel will be encoded with a kernel-based interpolated variational auto-encoder (VAE). In each voxel, a radial basis function (RBF) is computed to generate a local continuous representation to deal with sparse distributions of points.

F. Jaremo-Lawin et al. [55] proposed a multi-view method that first projects a 3D cloud to 2D planes from multiple camera views, then pixel-wise scores on synthetic images are predicted with a multi-stream FCN, and the final labels are obtained by fusing scores over different views. PolarNet [135], however, proposed a polar BEV representation. By implicitly aligning attention with the long-tailed distribution, this representation reduces the imbalance of points across grid cells along the radial axis.

Some other methods leverage scans in multiple modalities. 3DMV [11] proposed a joint 3D-multi-view network that combines features from RGB images and point cloud. Features are extracted with a 3D CNN stream and a group of 2D streams respectively. MVPNet [42] proposed another aggregation to fuse features (from images and point cloud) in 3D canonical space with a point-based network.

4.2.2 Point-based methods

First of all, PointNet [79] and PointNet++ [80] can predict semantic labels with corresponding prediction branches attached. Engelmann et al. [18] proposed a method to define neighborhoods in both world space and feature space with k-means clustering and KNN. A pairwise distance loss and centroid loss are introduced to feature learning based on the assumption that points with the same semantic label are supposed to be closer. PointWeb [136], as mentioned in classification, can also be adapted to predict segmentation labels. PVCNN [63] proposed a comprehensive method that leverages both point and voxel representation to obtain memory and computation efficiency simultaneously.

Some extensions of the convolution operator are introduced for feature extraction on point cloud. PCCN [100] introduces parametric continuous convolutional layers. These layers are parameterized by MLPs and span full continuous vector spaces. The generalization allows models to learn over any data structure where the support relationship is computable. Pointwise-CNN [37] introduced a point-wise convolution where the neighbor points are projected into kernel cells and convolved with corresponding kernel weights. Engelmann et al. [17] proposed Dilated Point Convolution (DPC) to aggregate dilated neighbor features, instead of the conventional k-nearest neighbors.

Graph networks are also used in some segmentation models to obtain the underlying geometric structures of the input point clouds. SPG [52] introduced a structure called superpoint graph (SPG) to capture the organization of point clouds. The idea is further extended in [51], which introduces a oversegmentation (into pure superpoints) of the input point cloud. Aside from that, Graph Attention Convolution [98] (GAC) is proposed to learn relevant features from local neighborhoods selectively. By dynamically assigning attention weights to different neighbor points and different feature channels based on their spatial positions and feature differences, the model is able to learn discriminative features from the most relevant part of the neighbor point sets.

Compared with projection-based methods, point-based methods usually require more computation and therefore have more trouble dealing with large-scale data. Tatarchenko et al. [96] introduced tangent convolutions to solve this. A fully-convolutional network is designed based on the tangent convolution and successfully improved the performance on large-scale point clouds. RandLA-Net [36] attempted to reduce computation by replace conventional complex point sampling approaches with random sampling. And to avoid random sampling from discarding crucial information, a novel feature aggregation module is introduced to enlarge receptive fields of each point.

Based on the fact that the production of point-level labels is labor-intensive and time-consuming, some methods explored weakly supervised segmentation. Xu and Lee [118] proposed a weakly supervised approach which only requires a small fraction of points to be labelled at training stage. By learning gradient approximation and smoothness constraints in geometry and color, competitive results can be obtained with as few as 10% points labelled. On the other hand, Wei et al. [108] introduced a multi-path region mining module, which can provide pseudo point-level labels by a classification network over weak labels. The segmentation network is then trained with these pseudo labels in a fully supervised manner.

4.3. Instance Segmentation

Instance segmentation, compared with semantic segmentation, requires distinguishing points with same semantic meaning, which makes the task more challenging. In this section, instance segmentation methods are further divided into two categories: proposal-based methods and proposal-free methods.

4.3.1 Proposal-based methods

Proposal-based instance segmentation methods can be considered as the combination of object detection and mask prediction. 3D-SIS [35] is a fully convolutional network for 3D semantic instance segmentation where geometry and color signals are fused. For each image, 2D features for each pixel are extracted by a series of 2D convolutional layers, and then backprojected to the associated 3D voxel grids. The geometry and color features are passed through a series of 3D convolutional layers respectively and concatenated into a global semantic feature map. Then a 3D-RPN and a 3D-RoI layer are applied to generate bounding boxes, instance masks and object labels. Generative Shape Proposal Network (GSPN) [126] generates proposals by reconstructing shapes from the scene instead of directly regresses bounding boxes. The generated proposals are refined with a region-based PointNet (R-PointNet), and the labels are determined with a point-wise binary mask prediction over all class labels. 3D-BoNet [124] is a single-stage method that adapts PointNet++ [80] as backbone network to global features and local features at each point. Two prediction branches follow to generate instance-level bounding box and point-level mask respectively. Zhang el al. [134] proposed a method for large-scale outdoor point clouds. The point cloud is first encoded into a high-resolution BEV representation augmented by KNN, and features are then extracted by voxel feature encoding (VFE) layers and self-attention blocks. For each grid, a horizontal object center and its height limit are predicted, objects that are closed enough will be merged, and eventually these constraints will be leveraged to generate instance prediction.

4.3.2 Proposal-free methods

Proposal-free methods tend to generate instance-level label based on semantic segmentation by algorithms like clustering. Similarity Group Proposal Network (SGPN) [101] is a representative work that learns a feature and semantic map for each point, and a similarity matrix to estimate the similarity between pairs of features. A heuristic non-maximal suppression method follows to merge points into instances. Lahoud et al. [48] adopted multi-task metric learning to (1) learn a feature embedding such that voxels with the same instance label are close and those with different labels are separated in the feature space and (2) predict the shape of instance at each voxel. Instance boundaries are estimated with mean-shift clustering and NMS.

Zhang et al. [133] introduced a probabilistic embedding to encode point clouds. The embedding is implemented with multivariate Gaussian distribution, and the Bhattacharyya kernel is adopted to esimate the similarity between points. Proposal-free methods do not suffer from the computational complexity of region-proposal layers; however, it is usually difficult for them to produce discriminative object boundaries from clustering.

There are also several instance segmentation methods based on projection. SqueezeSeg [109] is one of the pioneer works in this direction. In this method, points are first projected onto a sphere for a grid-based representation. The transformed representation is of size $H \times W \times C$, where in practice H=64 is the number of vertical channels of lidar, W is manually picked to be 512, and C equals to 5 (3 dimensional coordinates + intensity measurement + range). The representation is then fed through a conventional 2D CNN and a conditional random field (CRF) for refined segmentation results. This method is afterwards improved by SqueezeSegv2 [110] with a context aggregation module and a domain adaptation pipeline.

The idea of projection-based methods is further explored by Lyu et al. [66]. Inspired by graph drawing algorithms, they proposed a hierarchical approximate algorithm to project point clouds into image representations with abundant local geometric information preserved. The segmentation will then be generated by a multi-scale U-Net from the image representation. With this innovative projection algorithm, the method obtained significant improvement.

PointGroup [43] proposed a bottom-up framework with two prediction branches. For each point, its semantic label and relative offset to its respective instance centroid are predicted. The offset branch helps better grouping of points into objects as well as separation of objects with the same semantic label. During the clustering stage, both original positions and shifted positions are considered, the association of these two results turns out to have a better performance. Along with NMS based on the newly designed ScoreNet, this method outperforms other works of the day by a great margin.

4.4. Joint Training

As mentioned above, some recent works jointly address more than one problems to better realized the power of models. The unsupervised multi-task approach proposed by Hassani and Haley [31] is an example in which clustering, self-supervised classification and reconstruction are jointly trained. The two tasks under segmentation, semantic segmentation and instance segmentation, are also proven to likely benefit from simultaneous training.

There are two naive ways to solve semantic segmentation and instance segmentation at the same time: (1) solve semantic segmentation first, run instance segmentation on points of certain labels based on the result of semantic segmentation, (2) solve instance segmentation first, and directly assign semantic labels with instance labels. These two step-wise paradigms highly depend on the output quality of the first step, and are not able to make full use of the shared information between two tasks.

JSIS3D [74] develops a pointwise network that predicts the semantic label of each point and high-dimensional embeddings at the same time. After these steps, instances of the same class will have similar embeddings, then a multi-value conditional random field model is applied to synthesize semantic and instance labels, formulating the problem as jointly optimizing labels in the field model. ASIS [103] is another method that makes the two tasks benefit from each other. Specifically, instance segmentation benefits from semantic segmentation by learning semantic-aware instance embedding at point level, while semantic features of the point set from the same instance will be fused together to generate accurate semantic predictions for every point.

4.5. Experiments

We select the benchmarks on which most methods are tested, S3DIS[1], to compare the performance of different methods. The performances are summarized in Table 4.

5. Detection, Tracking and Flow Estimation

5.1. Overview

Object detection is a recent research hotspot as the basis of many practical applications. It aims to locate all the objects in the given scene. 3D object detection methods can be generally divided into three categories: multi-view methods, projection-based methods and point-based methods. Figure 4.1 shows an example of 3D object detection on multiple (lidar and camera) views. Aside from image object detection models, the exclusive characteristics of point cloud data provide more potential of optimization. Also, since 3D object tracking and scene flow estimation are two

Table 4. Experiment results on on semantic segmentation in S3DIS benchmark. Only results that are reported in the original papers are listed, those reported as a reference by other papers are excluded because they are sometimes conflicting.

Methods	Area5(mACC)	Area5(mIoU)	6-fold(mACC)	6-fold(mIoU)
PointCNN [58]	63.9	57.3	75.6	65.4
PointWeb [136]	66.6	60.3	76.2	66.7
A-CNN [112]	-	-	-	62.9
DGCNN [105]	-	-	-	56.1
VV-Net [68]	-	-	82.2	78.2
PCCN [100]	-	58.3	-	-
GAC [98]	-	62.9	-	-
DPC [17]	68.4	61.3	-	-
SSP+SPG [51]	-	-	78.3	68.4
JSIS3D [74]	-	-	78.6	-
ASIS [103]	60.9	53.4	70.1	59.3
Xu and Lee [118]	-	48.0	-	-
RandLA-Net [36]	-	-	82.0	70.0
Tatarchenko et al. [96]	62.2	52.8	-	-

derivative tasks that highly depend on object detection, they will be discussed together in this section.

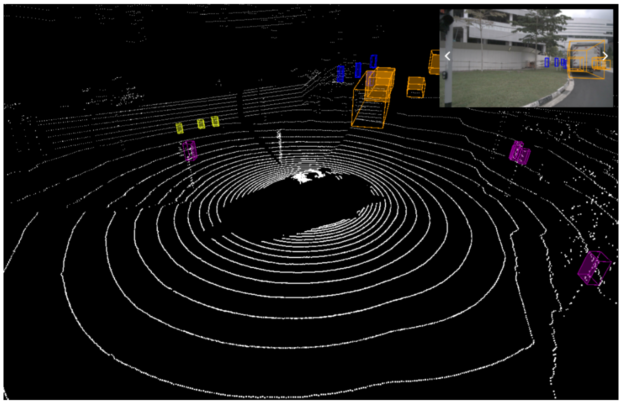

Figure 4. An outdoor scene from nuScenes [4], annotations in multi-view (lidar/camera) are provided.

5.2. Object Detection

5.2.1 Projection-based methods

The success of convolutional neural networks in image object detection inspired attempts to apply 3D CNN on projected point cloud data. VoxelNet [138] proposed an approach that applies random sampling to the point set within each voxel, and passes them through a novel voxel feature encoding (VFE) layer based on PointNet [79] and PointNet++ [80] to extract point-wise features. A region proposal network is used to produce detection results. Similar to classification models with volumetric representation, VoxelNet runs at a relatively low speed due to the sparsity of voxels and 3D convolutions. SECOND [119] then proposed an improvement in inference efficiency by taking advantage of sparse convolution network.

PointPillars [53] utilizes point cloud data in another way. Points are organized in vertical columns (called Pillars), and the features of pillars are extracted with PointNet to generate a pseudo image. The pseudo image is then considered as the input of a 2D object detection pipeline to predict 3D bounding boxes. PointPillars is more accurate than previous fusion approaches, and it is capable of real-time applications with a running speed of 62 FPS. Wang et al. [104] further proposed another anchor-free bounding box prediction based on a cylindrical projection into multi-view features.

Projection-based methods suffer from spatial information loss inevitably. Aside from using point-based networks instead, He et al. [32] proposed a structure-aware method to mitigate the problem. The convolutional layers are explicitly supervised to contain structural information by an auxiliary network. The auxiliary network converts the convolutional features from the backbone network to point-level representations and is jointly optimized. After the training process is finished, the auxiliary network can be detached to speed up the inference.

5.2.2 Point-based methods

Most point-based methods attempt to minimize information loss during feature extraction, and they are the group with the best performance so far. STD [123] introduced the idea of using sphere anchors for proposal generation, which achieves a high recall with significantly less computation than previous methods. Each proposal is passed through a PointsPool layer that converts proposal features from sparse expression to compact representation, and is robust under transformation. In addition to the regular regression branch, STD has another IoU branch to replace the role of classification score in NMS.

Some methods use foreground-background classification to improve the quality of proposals. PointRCNN [90] is such a framework, in which points are directly segmented to screen out foreground points, while semantic features and spatial features are then fused to produce high-quality 3D boxes. Compared with multi-view methods above, segmentation-based methods perform better for complicated scenes and occluded objects.

Furthermore, Qi et al. proposed VoteNet [77]. A group of points are sampled as seeds, and each seed independently generates a vote for potential center points of objects in the point cloud with the help of PointNet++ [80]. By taking advantage of voting, VoteNet outperforms previous approaches on two large indoor benchmarks. However, as the center point prediction of virtual center points is not as stable, the method performs less satisfactorily in wild scenes. As a follow-up work, ImVoteNet [76] inherited the idea of VoteNet and achieved prominent improvement by fusing 3D votes with 2D votes from images.

There are also attempts that consider domain knowledge as an auxiliary to enhance features. Associate-3Ddet [15] introduced the idea of perceptual-to-conceptual association. To enrich perception features that might be incomplete due to occlusion or sparsity, a perceptual-to-conceptual module is proposed to generate class-wise conceptual models from the dataset. The perception and conceptual features will be associated for feature enhancement.

Yang et al. [122] proposed a point-based anchor-free method 3DSSD. This method attempts to reduce computation by abandoning the upsampling layers (e.g. feature propagation layers in [123]) and refinement stages that are widely used in previous point-based methods. Previous set abstraction layers for downsampling only leverage furthest-point-sampling based on Euclidean distance (D-FPS), instances with a small number of interior points are easily lost under this strategy. In this case, removing upsampling layers could lead to huge performance drop. 3DSSD proposed F-FPS, a new sampling strategy based on feature distances, to preserve more foreground points for instances. The fusion of F-FPS and D-FPS, together with the candidate generation layer and 3D center-ness assignment in the prediction head, help this method outperform previous single-stage methods with a considerable margin.

Graph neural networks have also been introduced to 3D object detection for its ability to accommodate intrinsic characteristics of point clouds like sparsity. PointRGCN [130] is an early work that introduce graph-based representation for 3D vehicle detection refinement. After that, HGNet [7] introduces a hierarchical graph network based on shape-attentive graph convolution (SA-GConv). By capturing object shapes with relative geometric information and reasoning on proposals, the method obtained a significant improvement on previous results. Besides, Point-GNN [91] proposed a single-shot method based on graph neural networks. It first builds a fixed radius near-neighbors graph over the input point cloud. Then, the category and the bounding box of affiliation are predicted with the point graph. Finally, a box merging and scoring operation is used to obtain accurate combination of detection results from multiple vertices.

5.2.3 Multi-view methods

MV3D [8] is a pioneering method in multi-view object detection methods on point clouds. In this approach, candidate boxes are generated from BEV map and projected into feature maps of multiple views (RGB images, lidar data, etc.), then the region-wise features extracted from different views are combined to produce the final oriented 3D bounding boxes. While this approach achieves satisfactory performance, much like many other early multi-view methods, its running speed is too slow for practical use.

Attempts to improve multi-view methods generally take one of two directions. First, we could find a more efficient way to fuse information from different views. Liang et al. [59] use continuous convolutions to effectively fuse feature maps from images and lidar at different resolutions. Image features for each point in BEV space are utilized to generate a dense BEV feature map by bi-linear interpolation with projections of image features within the BEV plane. Experiments show that dense BEV feature maps perform better than discrete image feature maps and sparse point cloud feature maps. Second, many methods propose innovative feature extraction approaches to obtain representations of input data with higher robustness. SCANet [64] introduced a Spatial Channel Attention (SCA) module to make use of multi-scale contextual information. The SCA module captures useful features from the global and multi-scale context of given scene, while an Extension Spatial Unsample (ESU) module helps combine multi-scale low-level features to generate high-level features with rich spatial information, which then leads to accurate 3D object proposals. In RT3D [132], the majority of convolution operations prior to the RoI pooling module are removed. With such optimization, RoI convolutions only need to be performed once for all proposals, accelerating the method to run at 11.1 FPS, which is five times faster than MV3D [8].

Another approach to detect 3D objects is to generate candidate regions on 2D plane with 2D object detectors, then extract a 3D frustum proposal for each 2D candidate region. In F-PointNets [78], each 2D region generates a frustum proposal, and the features of each 3D frustum are learned with PointNet [79] or PointNet++ [80] and used for 3D bounding box estimation. PointFusion [117] uses both 2D image region and corresponding frustum points for more accurate 3D box regression. A fusion network is proposed to directly predict corner locations of boxes by fusing image features and global features from point clouds.

5.3. Object Tracking

Object tracking targets estimating the location of a certain object in subsequent frames given its state in the first frame. The success of Siamese networks [2] in 2D image object tracking inspired 3D object tracking, and Giancola et al. [24] extend Siamese networks to 3D. In this method, candidates are first generated by a Kalman filter, then passed through an encoding model to generate compact representations with shape regularization, and match the detected objects by cosine similarity. Zarzar et al. [130] proposed another method that captures target objects more efficiently by leveraging a 2D Siamese network to detect coarse object candidates on BEV representation. The coarse candidates are then refined by cosine similarity in the 3D Siamese network.

Chiu et al. [9] introduced the Kalman filter to encode the hidden states of objects. The state of an object is represented by a tuple of 11 variables, including position, orientation, size and speed. A Kalman filter is adopted to predict the object in next frame based on previous information, and a greedy algorithm is used for data association with Mahalanobis distance.

Besides, Qi et al. [81] proposed P2B, a point-to-box method for 3D object tracking. It divides the task into two parts. The first part is target-specific feature augmentation, seeds from the template and the search area are generated with a PointNet++ backbone, and the search area seeds will be enriched with target clues from the template. The second is target proposal and verification, candidate target centers are regressed and seed-wise targetness is evaluated for joint target proposal and verification.

5.4. Scene Flow Estimation

Similar to optical flow estimation on images, 3D scene flow estimation works on a sequence of point clouds. FlowNet3D [60] is a representative work that directly estimates scene flows from pairs of consecutive point clouds. The flow embedding layer is used to learn point-level features and motion features. The experiment results of FlowNet3D shows that it performs less than satisfactorily in non-static scenes, and the angles of predicted motion vectors sometimes significantly differ from the ground truth. FlowNet3D++ [107] is proposed to fix these issues by introducing a cosine distance loss in angles, and a point-to-plane distance loss to improve accuracy in dynamic scenes. HPLFlowNet [28], on the other hand, proposed a series of bilateral convolutional layers to fuse information from two consecutive frames and restore structural information from unconstructed point clouds.

In addition, MeteorNet [61] introduced direct grouping and chained-flow grouping to group temporal neighbors, and adopted information aggregation over neighbor points to generate representation for dynamic scenes. Derived from recurrent models in images, Fan and Yang [19] proposed PointRNN, PointGRU and PointLSTM to encode dynamic point clouds by capturing both spatial and temporary information.

5.5. Experiments

KITTI [22, 23, 21, 69] is one of the most popular benchmarks for many computer vision tasks, including those in images, point clouds, and multi-views. By taking advantage of autonomous driving platforms, KITTI provides raw data of real-world scenes, and allows evaluation on multiple tasks. Table 5 shows experimental results of different methods on KITTI. Some methods, such as VoteNet [14], which does not provide detailed test results on KITTI, are not listed.

6. Registration

6.1. Overview

In some scenarios like autopilot, it is of great value to find the relationship between point cloud data of the same scene collected in different ways. These data might be collected from different angles, or at different times. 3D point cloud registration (sometimes also called matching) attempts to align two or more different point clouds by estimating the transformation between them. It is a challenging problem affected by a lot of factors including noise, outliers and nonrigid spatial transformation.

6.2. Traditional Methods

The Iterative Closest Point (ICP) algorithm [3] is a pioneering work that solves 3D point set registration. The basic pipeline of ICP and its variants is as follows: (1) Sample a point set P from the source point cloud. (2) Compute the closest point set Q from the target point cloud. (3) Calculate the registration (transformation) with P and Q. (4) Apply the registration, and if the error is above some threshold, go back to step (2), otherwise terminate. A global refinement step is usually required for better performance. The performance of ICP highly depends on the quality of initialization and whether the input point clouds are clean. Generalized-ICP [88] and Go-ICP [121] are two representative follow-up works that mitigate the problems of ICP in different ways.

Coherent Point Drift (CPD) algorithm [72] considers the alignment as a problem of probability density estimation. Concretely, the algorithm consider the first point set as the Gaussian mixture model centroids, and the transformation is estimated by maximizing the likelihood in fitting them to

Table 5. Experiment results on KITTI 3D detection benchmark, E/M/H stands for easy/medium/hard samples.

Method	Category	Speed	Car			Pedestrians			Cyclists		
			E	M	H	E	M	H	E	M	H
MV3D [8]	multi-view	2.8	74.8	63.6	54.0	-	-	-	-	-	-
AVOD [47]	multi-view	12.5	89.8	85.0	78.3	42.6	33.6	30.1	64.1	48.1	42.4
SCANet [64]	multi-view	12.5	76.4	66.5	60.2	-	-	-	-	-	-
PIXOR [120]	projection	28.6	84.0	80.0	74.3	-	-	-	-	-	-
VoxelNet [138]	projection	2.0	77.5	65.1	57.7	39.5	33.7	31.5	61.2	48.4	44.4
SECOND [119]	projection	26.3	83.3	72.6	65.8	49.0	38.8	34.9	71.3	52.1	45.8
PointPillars [53]	projection	62.0	82.6	74.3	69.0	54.5	41.2	38.9	77.1	85.7	52.0
PointRCNN [90]	point	10.0	87.0	75.6	70.7	48.0	39.4	36.0	75.0	58.8	52.5
PointRGCN [130]	point	3.8	86.0	95.6	70.7	-	-	-	-	-	-
STD [123]	point	12.5	88.0	79.7	75.1	53.3	42.5	38.3	78.7	61.6	55.3
Point-GNN [91]	point	-	88.3	79.5	72.3	52.0	43.8	40.1	78.6	63.5	57.0
PV-RCNN [89]	point	-	90.2	81.4	76.8	52.1	43.3	40.3	78.6	63.7	57.7
3DSSD [122]	point	26.3	88.4	79.6	74.6	54.6	44.3	40.2	82.5	64.1	56.9

the second point set. The movement of these centroids are forced to be coherent to preserve the topological structure.

Robust Point Matching (RPM) [27] is another influential point matching algorithm. The algorithm starts with soft assignments of the point correspondences, and these soft assignments will get hardened through deterministic annealing. RPM is generally more robust than ICP, but still sensitive to initialization and noise.

Iglesias et al. [41] focused on the registration of several point clouds to a global coordinate system. In other words, with the original set of n points, we want to find the correspondences between (subsets of) the original set and m local coordinate systems respectively. Iglesias et al. consider the problem as a Semidefinite Program (SDP), and attempt to analyze it with the application of Lagrangian duality.

6.3. Learning-based Methods

DeepVCP [65] is the first end-to-end learning-based framework in point cloud registration. Given the source and target point cloud, PointNet++ [80] is applied to extract local features. A point weighting layer then helps select a set of N keypoints, after which $N \times C$ candidates from the target point cloud are selected and passed through a deep feature embedding operation together with keypoints from the source. Finally, a corresponding point generation layer takes the embeddings and generates the final result. Two losses are incurred: (1) the Euclidean distance between the estimated corresponding points and ground truth under the ground truth transformation, and (2) the distance between the target under the estimated transformation and ground truth. These losses are combined to consider both global geometric information and local similarity.

3DSmoothNet [26] is proposed to perform 3D point cloud matching with a compact learned local feature descriptor. Given two raw point clouds as input, the model first computes the local reference frame (LRF) of the neighborhood around the randomly sampled interest points. Then the neighborhoods are transformed into canonical representations and voxelized by Gaussian smoothing, and the local feature of each point is then generated by 3DSmoothNet. The features will then be utilized by a RANSAC approach to produce registration results. The proposed smooth density value (SDV) voxelization outperforms traditional binary-occupancy grids by reducing the impact of boundary effects and noise, and provides greater compactness. Following 3DSmoothNet, Gojcic et al. [25] proposed another method that formulates conventional two-stage approaches in an end-to-end structure. Earlier methods solve the problem in two steps, the pairwise alignment and the globally consistent refinement, by jointly learning both parts. Gojcic et al.'s method outperforms previous ones with higher accuracy and less computational complexity.

RPM-Net [125] inherits the idea of RPM [27] algorithm, and takes advantage of deep learning to enhance robustness against noise, outliers and bad initialization. In this method, the initialization assignments are generated based on hybrid features from a network instead of spatial distances between points. The parameters of annealing is predicted by a secondary network, and a modified Chamfer distance is introduced to evaluate the quality of registration. This method outperforms previous methods no matter the input is clean, noisy, or even partially visible.

7. Augmentation and Completion

7.1. Overview

Point clouds collected by lidar, especially those from outdoor scenes, suffer from different kinds of quality issues like noise, outliers, and missing points. Many attempts have been made to improve the quality of raw point clouds by completing missing points, removing outliers and so on. The motivation and implementation vary a lot among different approaches; in this paper, we divide them into two categories: discriminative models and generative models.

7.2. Discriminative Methods

Noise in point clouds collected from outdoor scenes is naturally inevitable. To prevent noise from influencing the encoding of point clouds, some denoising methods shall be applied in pre-processing. Conventional methods include local surface fitting, neighborhood averaging and guessing the underlying noise model. PointCleanNet [82] proposed a data-driven method to remove outliers and reduce noise. With a deep neural network adapted from PCPNet[29], the model first classifies outliers and discards them, then estimates a correction projection that projects noise to original surfaces.

Hermosilla et al. [33] proposed Total Denoising that achieved unsupervised denoising of 3D point clouds without additional data. The unsupervised image denoisers are usually built based on the assumption that the value of a noisy pixel follows a distribution around a clean pixel value. Under this assumption, the original clean value can be recovered by learning the parameters of the random distribution. However, such an idea cannot be directly extended to point clouds because there are multiple formats of noise in point clouds, such as a global position deviation where no reliable reference point exists. Total Denoising introduces a spatial prior term that finds the closest of all possible modes on a manifold. The model achieves competitive performance against supervised models.

While a lot of models benefit from rich information in dense point clouds, some others are suffering from the low efficiency with large amounts of points. Conventional downsampling approaches usually have to risk dropping critical points. Nezhadarya et al. [73] proposed the critical points layer (CPL) that learns to reduce the number of points while preserving the important ones. The layer is deterministic, order-agnostic and also efficient by avoiding neighbor search. Aside from that, SampleNet [54] proposed a differentiable relaxation of point sampling by approximating points after sampling as a mixture of original points. The method has been tested as a front to networks on various tasks, and obtains decent performance with only a small fraction of the raw input point cloud.

7.3. Generative Methods

Generative adversarial networks are widely studied for 2D images and CNNs, as they help locate the potential defects of networks by generating false samples. While typical applications of point cloud models, such as autonomous driving, consider safety as a critical concern, it is helpful to study how current deep neural networks on point clouds are affected by false samples.

Xiang et al. [114] proposed several algorithms to generate adversarial point clouds against PointNet. The adversarial algorithms work in two ways: point perturbation and point generation. Perturbation is implemented by shifting existing points negligibly, and generation is implemented by either adding some independent and scattered points or a small number of point clusters with predefined shapes. Shu et al. [92] proposed tree-GAN, a tree-structured graph convolution network. By performing graph convolution within a tree, the model takes advantage of ancestor information to enrich the capacity of features. Along with the development of adversarial networks, DUP-Net [137] is proposed to defend 3D adversarial models. The model contains a statistical outlier removal (SOR) module as denoiser and a data-driven upsampling network as upsampler.

Aside from adversarial generation, generative models are also used for point cloud upsampling. There are generally two motivations to upsample a point cloud. The first is to reduce the sparseness and irregularity of data, and the second is to restore missing points due to occlusion.

For the first aim, PU-Net [127] proposed upsampling in the feature space. For each point, multi-level features are extracted and expanded via a multi-branch convolution unit; after that, the expanded feature is split into multiple features and reconstructed to upsample the input set. Inspired by image super-resolution models, Wang et al. [106] proposed a cascade of patch-based upsampling networks, learning different levels of details at different steps, where at each step the network focuses only on a local patch from the output of the previous step. The architecture is able to upsample a sparse input point set to a dense set with rich details. Hui et al. [40] also proposed a learning-based deconvolution network that generates multi-resolution point clouds based on low-resolution input with bilateral interpolation performed in both the spatial and feature spaces.

Meanwhile, early methods in completion, such as [13], tend to voxelize the input point cloud at the very beginning. PCN [129] was the first framework to work on raw point clouds and in a coarse-to-fine fashion. Wang et al. [102] improved the results with a two-step reconstruction design. Besides, Huang et al. [39] proposed PF-Net that preserves the spatial structure of the original incomplete point cloud, and predicts the missing points hierarchically a multi-scale generating network. GRNet[116], on the other hand, proposed a gridding-based which retrieve structural context by performing cubic feature sampling per grid, and complete the output with ”Gridding Reverse” layers and MLPs.

Lan et al. [50] proposed a probabilistic approach to optimize outliers by applying EM algorithm with Cauchy-Uniform mixture model to suppress potential outliers. More generally, PU-GAN [57] proposed a data-driven generative adversarial network to learn point distributions from the data and upsample points over patches on the surfaces of objects. Furthermore, RL-GAN-Net [87] uses a reinforcement learning (RL) agent to provide fast and reliable control of a generative adversarial network. By first training the GAN on the dimension-reduced latent space representation, and then finding the correct input to generate the representation that fits the current input form the uncompleted point cloud with a RL agent, the framework is able to convert noisy, partial point cloud into a completed shape in real time.

8. Conclusion

In this paper, we reviewed milestones and recent progress on various problems in 3D point clouds. With the expectation of practical applications like autonomous driving, point cloud understanding has received increasing attention lately. In 3D shape classification, point-based models have achieved satisfactory performance on recognized benchmarks. Methods developed from image tasks, such as two-stage detector and the Siamese architecture, are widely introduced in 3D segmentation, object detection and other derivative tasks. Specific deep learning frameworks are proposed to match point clouds of the same scene from multiple scans, and generative networks are adapted to improve the quality of point cloud data with noise and missing points. Deep learning methods with proper adaption have been proven to efficiently help overcome the unique challenges in point cloud data.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, pages 850–865. Springer, 2016.

[3] Paul J Besl and Neil D McKay. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. Technical Report arXiv:1512.03012 [cs.GR], 2015.

[6] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. ClusterNet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4994–5002, 2019.

[7] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z. Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[8] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[9] Hsu-kuang Chiu, Antonio Prioletti, Jie Li, and Jeannette Bohg. Probabilistic 3D multi-object tracking for autonomous driving. *arXiv preprint arXiv:2001.05673*, 2020.

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[11] Angela Dai and Matthias Nießner. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018.

[12] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018.

[13] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5868–5877, 2017.

[14] Zhipeng Ding, Xu Han, and Marc Niethammer. VoteNet: A deep learning label fusion method for multi-atlas segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 202–210. Springer, 2019.

[15] Liang Du, Xiaoqing Ye, Xiao Tan, Jianfeng Feng, Zhenbo Xu, Errui Ding, and Shilei Wen. Associate-3ddet: Perceptual-to-conceptual association for 3d point cloud object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] Yueqi Duan, Yu Zheng, Jiwen Lu, Jie Zhou, and Qi Tian. Structural relational reasoning of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 949–958, 2019.

[17] Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dilated point convolutions: On the receptive field of point convolutions. *arXiv preprint arXiv:1907.12046*, 2019.

[18] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know what your neighbors do: 3D semantic segmentation of point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[19] Hehe Fan and Yi Yang. PointRNN: Point recurrent neural network for moving point cloud processing. *arXiv preprint arXiv:1910.08287*, 2019.

[20] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. GVCNN: Group-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2018.

[21] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[22] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[23] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[24] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3D siamese tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1359–1368, 2019.

[25] Zan Gojcic, Caifa Zhou, Jan D. Wegner, Leonidas J. Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3D point cloud matching with smoothed densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019.

[27] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern recognition*, 31(8):1019–1031, 1998.

[28] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. HPLFlowNet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3254–3263, 2019.

[29] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. PCPNet: Learning local shape properties from raw point clouds. In *Computer Graphics Forum*, volume 37, pages 75–85. Wiley Online Library, 2018.

[30] Timo Hackel, N. Savinov, L. Ladicky, Jan D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1, pages 91–98, 2017.

[31] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8160–8171, 2019.

[32] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[33] Pedro Hermosilla, Tobias Ritschel, and Timo Ropinski. Total denoising: Unsupervised learning of 3D point cloud cleaning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 52–60, 2019.

[34] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte Carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018.

[35] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019.

[36] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[37] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018.

[38] Jing Huang and Suya You. Point cloud labeling using 3D convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2670–2675. IEEE, 2016.

[39] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] Le Hui, Rui Xu, Jin Xie, Jianjun Qian, and Jian Yang. Progressive point cloud deconvolution generation network. In *The European Conference on Computer Vision (ECCV)*, 2020.

[41] Jose Pedro Iglesias, Carl Olsson, and Fredrik Kahl. Global optimality for point set registration using semidefinite programming. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[42] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3D scene understanding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[43] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[44] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova,

C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft Level 5 AV Dataset 2019. https://level5.lyft.com/dataset/, 2019.

[45] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3D point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872, 2017.

[46] Artem Komarichev, Zichun Zhong, and Jing Hua. A-CNN: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7421–7430, 2019.

[47] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3D proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[48] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3D instance segmentation via multi-task metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9256–9266, 2019.

[49] Shiyi Lan, Ruichi Yu, Gang Yu, and Larry S Davis. Modeling local geometric structure of 3D point clouds using Geo-CNN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 998–1008, 2019.

[50] Ziquan Lan, Zi Jian Yew, and Gim Hee Lee. Robust point cloud based reconstruction of large-scale outdoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9690–9698, 2019.

[51] Loic Landrieu and Mohamed Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7440–7449, 2019.

[52] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4558–4567, 2018.

[53] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[54] Itai Lang, Asaf Manor, and Shai Avidan. Samplenet: Differentiable point cloud sampling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[55] Felix Järemo Lawin, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Deep projective 3D semantic segmentation. In *International Conference on Computer Analysis of Images and Patterns*, pages 95–107. Springer, 2017.

[56] Jiaxin Li, Ben M Chen, and Gim Hee Lee. SO-Net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9397–9406, 2018.

[57] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. PU-GAN: A point cloud upsampling adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7203–7212, 2019.

[58] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, pages 820–830, 2018.

[59] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3D object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018.

[60] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. FlowNet3D: Learning scene flow in 3D point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019.

[61] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. MeteorNet: Deep learning on dynamic 3D point cloud sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9246–9255, 2019.

[62] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019.

[63] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, pages 965–975, 2019.

[64] Haihua Lu, Xuesong Chen, Guiying Zhang, Qiuhao Zhou, Yanbo Ma, and Yong Zhao. SCANet: Spatial-channel attention network for 3D object detection. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1992–1996. IEEE, 2019.

[65] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. DeepVCP: An end-to-end deep neural network for point cloud registration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 12–21, 2019.

[66] Yecheng Lyu, Xinming Huang, and Ziming Zhang. Learning to segment 3d point clouds in 2d image space. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[67] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015.

[68] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. VV-Net: Voxel vae net with group convolutions for point cloud segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8500–8508, 2019.

[69] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[70] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[71] Daniel Munoz, J Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual classification with functional

max-margin markov networks. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 975–982. IEEE, 2009.

[72] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.

[73] Ehsan Nezhadarya, Ehsan Taghavi, Ryan Razani, Bingbing Liu, and Jun Luo. Adaptive hierarchical down-sampling for point cloud classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[74] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. JSIS3D: Joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2019.

[75] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[76] Charles R. Qi, Xinlei Chen, Or Litany, and Leonidas J. Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[77] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[78] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum PointNets for 3D object detection from RGB-D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[79] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[80] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[81] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2b: Point-to-box network for 3d object tracking in point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[82] Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J Mitra, and Maks Ovsjanikov. PointCleanNet: Learning to denoise and remove outliers from dense point clouds. In *Computer Graphics Forum*, volume 39, pages 185–203, 2020.

[83] Yongming Rao, Jiwen Lu, and Jie Zhou. Spherical fractal convolutional neural networks for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 452–460, 2019.

[84] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017.

[85] Xavier Roynard, Jean-Emmanuel Deschaud, and Franois Goulette. Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *The International Journal of Robotics Research*, 37(6):545–557, 2018.

[86] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point cloud library (PCL). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4. IEEE, 2011.

[87] Muhammad Sarmad, Hyunjoo Jenny Lee, and Young Min Kim. RL-GAN-Net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5898–5907, 2019.

[88] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[89] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[90] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[91] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[92] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3D point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3859–3868, 2019.

[93] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017.

[94] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[95] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[96] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3D. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018.

[97] Bruno Vallet, Mathieu Brédif, Andrés Serna, Beatriz Marcotegui, and Nicolas Paparoditis. Terramobilita/iqmulus urban point cloud analysis benchmark. *Computers & Graphics*, 49:126–133, 2015.

[98] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10296–10305, 2019.

[99] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017.

[100] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597, 2018.

[101] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018.

[102] Xiaogang Wang, Marcelo H. Ang Jr. , and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[103] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4096–4105, 2019.

[104] Yue Wang, Alireza Fathi, Abhijit Kundu, David A. Ross, Caroline Pantofaru, Thomas A. Funkhouser, and Justin M. Solomon. Pillar-based object detection for autonomous driving. In *The European Conference on Computer Vision (ECCV)*, 2020.

[105] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12, 2019.

[106] Yifan Wang, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3D point set upsampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5958–5967, 2019.

[107] Zirui Wang, Shuda Li, Henry Howard-Jenkins, Victor Prisacariu, and Min Chen. FlowNet3D++: Geometric losses for deep scene flow estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 91–98, 2020.

[108] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and Lihua Xie. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[109] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018.

[110] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[111] Pengxiang Wu, Chao Chen, Jingru Yi, and Dimitris Metaxas. Point cloud processing via recurrent set encoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5441–5449, 2019.

[112] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

[113] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[114] Chong Xiang, Charles R Qi, and Bo Li. Generating 3D adversarial point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9136–9144, 2019.

[115] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010.

[116] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *ECCV*, 2020.

[117] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. PointFusion: Deep sensor fusion for 3D bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018.

[118] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[119] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[120] Bin Yang, Wenjie Luo, and Raquel Urtasun. PIXOR: Real-time 3D object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[121] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015.

[122] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[123] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. STD: Sparse-to-dense 3D object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1951–1960, 2019.

[124] Ze Yang and Liwei Wang. Learning relationships for multiview 3D object recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7505–7514, 2019.

[125] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11824–11833, 2020.

[126] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019.

[127] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. PU-Net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2018.

[128] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3D object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 186–194, 2018.

[129] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018.

[130] Jesus Zarzar, Silvio Giancola, and Bernard Ghanem. PointRGCN: Graph convolution networks for 3D vehicles detection refinement. *arXiv preprint arXiv:1911.12236*, 2019.

[131] Wei Zeng and Theo Gevers. 3DContextNet: Kd tree guided hierarchical learning of point clouds using local and global contextual cues. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[132] Yiming Zeng, Yu Hu, Shice Liu, Jing Ye, Yinhe Han, Xiaowei Li, and Ninghui Sun. RT3D: Real-time 3-D vehicle detection in lidar point cloud for autonomous driving. *IEEE Robotics and Automation Letters*, 3(4):3434–3440, 2018.

[133] Biao Zhang and Peter Wonka. Point cloud instance segmentation using probabilistic embeddings. *arXiv preprint arXiv:1912.00145*, 2019.

[134] Feihu Zhang, Chenye Guan, Jin Fang, Song Bai, Ruigang Yang, Philip Torr, and Victor Prisacariu. Instance segmentation of lidar point clouds. *ICRA, Cited by*, 4(1), 2020.

[135] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[136] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5565–5573, 2019.

[137] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. DUP-Net: Denoiser and upsampler network for 3D adversarial point clouds defense. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1961–1970. IEEE, 2019.

[138] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.