robotics

MDPI

Review

Recent Advances and Perspectives in Deep Learning Techniques for 3D Point Cloud Data Processing

Zifeng Ding [1], Yuxuan Sun [1], Sijin Xu [2], Yan Pan [3], Yanhong Peng [4,*] and Zebing Mao [5]

[1] School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Haidian District, Beijing 100044, China
[2] School of Electronics and Computer Science, The University of Southampton, University Rd., Southampton SO17 1BJ, UK
[3] Shenzhen Institute of Advanced Electronic Materials, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518100, China
[4] School of Mechanical Electrical Engineering, Beijing Information Science and Technology University, NO.12 Xiaoying East Road, Qinghe, Haidian District, Beijing 100192, China
[5] Department of Mechanical Engineering, Tokyo Institute of Technology, 2-12-1 Ookayama Meguro-Ku, Tokyo 152-8550, Japan
* Correspondence: yhpeng@nagoya-u.jp

Abstract: In recent years, deep learning techniques for processing 3D point cloud data have seen significant advancements, given their unique ability to extract relevant features and handle unstructured data. These techniques find wide-ranging applications in fields like robotics, autonomous vehicles, and various other computer-vision applications. This paper reviews the recent literature on key tasks, including 3D object classification, tracking, pose estimation, segmentation, and point cloud completion. The review discusses the historical development of these methods, explores different model architectures, learning algorithms, and training datasets, and provides a comprehensive summary of the state-of-the-art in this domain. The paper presents a critical evaluation of the current limitations and challenges in the field, and identifies potential areas for future research. Furthermore, the emergence of transformative methodologies like PoinTr and SnowflakeNet is examined, highlighting their contributions and potential impact on the field. The potential cross-disciplinary applications of these techniques are also discussed, underscoring the broad scope and impact of these developments. This review fills a knowledge gap by offering a focused and comprehensive synthesis of recent research on deep learning techniques for 3D point cloud data processing, thereby serving as a useful resource for both novice and experienced researchers in the field.

Keywords: 3D data; deep learning; mesh; point cloud; voxel

Citation: Ding, Z.; Sun, Y.; Xu, S.; Pan, Y.; Peng, Y.; Mao, Z. Recent Advances and Perspectives in Deep Learning Techniques for 3D Point Cloud Data Processing. *Robotics* **2023**, *12*, 100. https://doi.org/10.3390/robotics12040100

Academic Editors: Salvatore Livatino, Dario Calogero Guastella, Lucio Tommaso De Paolis and Daniele Ravi

Received: 28 May 2023
Revised: 4 July 2023
Accepted: 6 July 2023
Published: 11 July 2023

1. Introduction

Over the past decade, deep learning techniques have become increasingly central to the field of 3D computer vision. In particular, advancements in the handling of 3D point cloud data have revolutionized applications in robotics [1–3], autonomous vehicles [4,5], and other areas dependent on computer vision. Despite the explosive growth in this area, comprehensive reviews that address recent literature on the processing and application of 3D point cloud data are scarce [6–10].

This paper aims to fill this gap by reviewing the fundamental techniques and recent progress in deep learning applied to 3D point cloud data, with a focus on classification, tracking, pose estimation, segmentation, and completion. Here, we examine the mechanism, architecture, data types, and the current state of the art in each of these areas, including groundbreaking methodologies, like PoinTr and SnowflakeNet, for point cloud completion.

Moreover, we delve into both traditional and innovative techniques to shed light on the inherent challenges and potential solutions for processing 3D point cloud data. Our

paper aims to serve as a valuable reference for future research by summarizing mature methodologies and offering insights into burgeoning ones. Ultimately, this review strives to synthesize existing knowledge, identify gaps in current understanding, and, consequently, pave the way for further innovation in the rapidly evolving field of 3D point cloud data processing. Figure 1 shows the stages involved to process 3D point cloud data using deep learning. The first stage is collecting 3D point cloud data using sensors like LiDAR, depth cameras, or photogrammetry techniques. The acquired data represent the scene or objects in three-dimensional space. The second stage is data preprocessing which is performed to prepare the point cloud data for deep learning algorithms. Common preprocessing steps include data cleaning, noise removal, downsampling, and normalization to ensure consistency and improve data quality. Thirdly, the network architecture defines the structure and connectivity of the deep learning model. Different architectures can be employed, including Convolutional Neural Networks (CNNs) adapted for 3D data, Graph Neural Networks (GNNs), or hybrid models. Architectures are designed based on the specific task, such as classification, segmentation, object detection, or reconstruction. The next step is training the deep learning model using labeled or annotated point cloud data. This involves feeding the point cloud data into the network, computing the loss between predicted and ground truth labels, and updating the model parameters through backpropagation. Training is typically performed on a large dataset, and techniques like data augmentation may be used to improve generalization. Once the model is trained and evaluated, it can be deployed for inference on new, unseen point cloud data. At the last, post-processing steps can be applied to refine the output of the model.

Figure 1. The stages involved to process 3D point cloud data.

Currently, there are many reviews summarizing deep learning on point clouds, for example, Guo et al. [8] summarized different application in 3D tasks with various methods and the datasets can be used, but they did not introduce the datasets and the process of the deep learning. Moreover, Lu et al. [9] and Lahoud et al. [11] introduced the transformers in 3D point clouds and they both described the principle and the application in detail, but none of them covered deep learning. Furthermore, Xiao et al. [12] presented the unsupervised point cloud representation learning with deep neural networks. They provided an overview of common databases and gave an insight into how each approach worked, but they did not detail the various applications. We addressed four articles with their shortages and updated them in this review.

To perform the search better, we defined the years of the surveys and reviews from 2017 to 2023 and search the definitions and principles from 2005 to 2020 so that to make our articles more advanced. Moreover, we defined the key terms "3D points cloud, Transformers, deep learning, representation and application". Furthermore, after search the articles with the restrictions, we reviewed the papers and search the databases from these papers.

Our contribution are as follows, we concluded the advantage of each article and introduce the datasets, principles, different methods in 3D representation, transformers, and application as detailed as possible. Moreover, our review offers a critical assessment of the current limitations and challenges in the field, while identifying potential areas for future research. We also explored groundbreaking methodologies like PoinTr and SnowflakeNet, accentuating their contributions and potential impact. Additionally, we discussed the interdisciplinary applications of these techniques, underscoring their wide-ranging scope and significance.

2. Background

2.1. Basic Concepts

Three-dimensional single object tracking necessitates an abundance of datasets, typically collected from a myriad of experiments, for the assessment of performance and other characteristics. These datasets serve as a common benchmark for evaluating the efficiency of deep learning algorithms. Various 3D operations such as 3D shape classification, 3D object detection, and object tracking utilize the datasets listed in Table 1 [8]. There exist two categories of datasets for 3D shape classification, synthetic datasets [13,14] and real-world datasets [8,15,16]. Likewise, datasets for 3D object detection and tracking are divided into two types, indoor scenes [15,17] and outdoor urban scenes [8,18–21]. Various sensors, including Terrestrial Laser Scanners (TLS) [22], Aerial Laser Scanners (ALS) [23,24], RGBD cameras [15], Mobile Laser Scanners (MLS) [25–27], and other 3D scanners [8,28], are employed to gather these datasets for 3D point segmentation.

2.2. 3D Datasets

This section introduces several popular datasets used for evaluating and training unsupervised point clouds [29]. We classify the datasets into three categories: Detection and Tracking, Segmentation and Classification, and introduce them in this order. Since we have listed 26 kinds of datasets, it is very difficult to explain each dataset. We selected the most commonly used and representative datasets for introduction. To augment the existing work on learning unsupervised point cloud representations, two types of datasets are commonly used: real datasets for scenes and synthetic datasets for objects. Among the real datasets, ScanNet [15] and KITTI [30] are more practical compared to others. Regarding the synthetic object datasets, ModelNet [13] and ShapeNet [14] are the most widely used. There exist numerous datasets for specific tasks as well. For instance, the fine-tuning downstream models can employ the following datasets: For semantic segmentation, S3DIS [28], ScanNet [15] or Synthia 4D [31] are preferred. For object detection, indoor datasets such as SUN RGB-D [17] and ScanNet [15] along with outdoor datasets ONCE [32] are more appropriate. For point cloud classification, ModelNet40 [13], ScanObjectNN [16] and ShapeNet [14] can be used directly. For part segmentation, ShapeNetPart [14] is the best option [29]. The aforementioned datasets, along with their corresponding information, are represented in Tables 1–3.

Table 1. Datasets for 3D detection and tracking.

Name and Reference	Year	Scene Type	Sensors	Website
KITTI [30]	2012	Urban (Driving)	RGB and LiDAR	https://www.cvlibs.net/datasets/kitti/ (accessed on 4 July 2023)
SUN RGB-D [17]	2015	Indoor	RGB-D	https://rgbd.cs.princeton.edu/(accessed on 4 July 2023)
ScanNetV2 [15]	2018	Indoor	RGB-D and Mesh	http://www.scan-net.org/ (accessed on 4 July 2023)
H3D [33]	2019	Urban (Driving)	RGB and LiDAR	https://usa.honda-ri.com/h3d (accessed on 4 July 2023)
Argoverse [34]	2019	Urban (Driving)	RGB and LiDAR	https://www.argoverse.org/ (accessed on 4 July 2023)
Lyft L5 [35]	2019	Urban (Driving)	RGB and LiDAR	-
A*3D [36]	2019	Urban (Driving)	RGB and LiDAR	https://github.com/I2RDL2/ASTAR-3D (accessed on 4 July 2023)
Waymo Open [20]	2020	Urban (Driving)	RGB and LiDAR	https://waymo.com/open/ (accessed on 4 July 2023)
nuScenes [21]	2020	Urban (Driving)	RGB and LiDAR	https://www.nuscenes.org/ (accessed on 4 July 2023)

Table 2. Datasets for 3D point cloud segmentation.

Name and Reference	Year	RGB	Sensors	Website
Oakland [37]	2009	N/A	MLS	-
ISPRS [23]	2012	N/A	ALS	-
Paris-rue-Madame [26]	2014	N/A	MLS	https://people.cmm.minesparis.psl.eu/users/serna/rueMadameDataset.html (accessed on 4 July 2023)
IQmulus [38]	2015	N/A	MLS	-
ScanNet [15]	2017	Yes	RGB-D	http://www.scan-net.org/ (accessed on 4 July 2023)
S3DIS [28]	2017	Yes	Matterport	http://buildingparser.stanford.edu/dataset.html (accessed on 4 July 2023)
Semantic3D [22]	2017	Yes	TLS	http://www.semantic3d.net/ (accessed on 4 July 2023)
Paris-Lille-3D [27]	2018	N/A	MLS	-
SemanticKITTI [25]	2019	N/A	MLS	http://www.semantic3d.net/ (accessed on 4 July 2023)
Toronto-3D [39]	2020	Yes	MLS	-

Table 3. Datasets for 3D shape classification.

Dataset	Year	Type	Representation	Website
McGill Benchmark [40]	2008	Synthetic	Mesh	https://www.cim.mcgill.ca/~shape/benchMark/ (accessed on 4 July 2023)
Sydney Urban Objects	2013	Real-World	Point Clouds	-
ModelNet10 [13]	2015	Synthetic	Mesh	https://modelnet.cs.princeton.edu/ (accessed on 4 July 2023)
ModelNet40 [13]	2015	Synthetic	Mesh	https://modelnet.cs.princeton.edu/ (accessed on 4 July 2023)
ShapeNet [14]	2015	Synthetic	Mesh	https://shapenet.org/ (accessed on 4 July 2023)
ScanNet [15]	2017	Real-World	RGB-D	http://www.scan-net.org/ (accessed on 4 July 2023)
ScanObjectNN [16]	2019	Real-World	Point Clouds	-

S3DIS [28]: A widely used dataset for semantic segmentation and scene understanding in indoor environments. The Stanford Large-Scale 3D Indoor Space (S3DIS) dataset is an extensive collection of over 215 million points scanned from three office buildings, covering six large indoor areas totaling 6000 square meters. It constitutes a detailed point cloud representation, complete with point-by-point semantic labels from 13 object categories. S3DIS is a widely-used dataset for semantic segmentation and scene understanding in indoor environments. It provides realistic 3D spatial information, making it suitable for recreating real spaces in cloud-based applications.

ScanNet-V2 [15]: A large-scale dataset of annotated 3D indoor scenes. The ScanNet-V2 dataset emerges from the compilation of more than 2.5 million views from over 1500 RGB-D video scans. Primarily capturing indoor scenes, such as bedrooms and classrooms, this dataset enables annotation through surface reconstruction, 3D camera poses, and semantic and instance labels to facilitate segmentation. ScanNet-V2 is a popular dataset that enables rich scene understanding and reconstruction in indoor environments. It provides a large-scale and comprehensive dataset for cloud-based real-space recreation.

SUN RGB-D [17]: A dataset that focuses on indoor scene understanding and semantic parsing. The SUN RGB-D dataset comprises a collection of single-view RGB-D images harvested from indoor environments, encompassing residential and complex spaces. It features 10,335 RGB-D images and 37 categories of 3D oriented object bounding boxes. The KITTI dataset, a pioneer in outdoor datasets, provides a wealth of data, including over 200 k 3D boxes for object detection across more than 22 scenes, dense point clouds from LiDAR sensors, and additional modes, such as GPS/IMU data and frontal stereo images [41]. SUN RGB-D is a benchmark dataset for various tasks, including scene understanding and object recognition in indoor environments. Its comprehensive annotations make it useful for recreating accurate real spaces in the cloud.

ONCE [32]: A project focused on developing object-centric navigation algorithms using catadioptric omnidirectional vision. The ONCE dataset encompasses seven million corresponding camera images and one million LiDAR scenes. It contains 581 sequences, including

10 annotated sequences for testing and 560 unlabeled sequences for unsupervised learning, thereby offering a benchmark for unsupervised learning and outdoor object detection. ONCE dataset offers detailed annotations for part-level co-segmentation, making it valuable for cloud-based real-space recreation that requires accurate part-level understanding.

ModelNet10/ModelNet40 [13]: A benchmark dataset for 3D object classification and shape recognition. ModelNet, a synthetic object-level dataset designed for 3D classification, offers CAD models represented by vertices and faces. ModelNet10 provides 3377 samples from 10 categories, divided into 2468 training samples and 909 test samples. ModelNet40, around four times the size of ModelNet10, contains 13,834 objects from 40 categories, with 9843 objects forming the training set and the remainder allocated for testing. ModelNet10/ModelNet40 dataset focuses on object recognition and classification rather than scene-level understanding or space reconstruction. It might not be directly applicable to recreating real spaces in the cloud.

ScanObjectNN [16]: A dataset designed for object instance segmentation and semantic segmentation in large-scale 3D indoor scenes. ScanObjectNN is a real object-level dataset consisting of 2902 3D point cloud objects from 15 categories, created by capturing and scanning real indoor scenes. Distinct from synthetic object datasets, the point cloud objects in ScanObjectNN are not axis-aligned and contain noise. ScanObjectNN dataset specializes in object instance segmentation and might not be suitable for full scene reconstruction or real-space recreation in the cloud.

ShapeNet [14]: A large-scale dataset of 3D shape models covering a wide range of object categories. ShapeNet comprises 55 categories of synthetic 3D objects, collected from online open-source 3D repositories. Similar to ModelNet, ShapeNet is complete, aligned, and devoid of occlusion or background. ShapeNet dataset is valuable for object-level understanding, but they may not directly address full scene reconstruction or real-space recreation in the cloud.

ShapeNetPart [14]: A subset of the ShapeNet dataset that focuses on fine-grained object classification and semantic part segmentation. ShapeNetPart, an extension of ShapeNet, includes 16,881 objects from 16 categories, each represented by point clouds. Each object is divided into 2 to 6 parts, culminating in a total of 50 part categories in the datasets. ShapeNetPart dataset primarily focuses on part-level semantic segmentation of 3D models rather than real-world spatial reconstruction, limiting its applicability to cloud-based real-space recreation.

In summary, S3DIS, ScanNet-V2, and SUN RGB-D are datasets that are particularly well-suited for recreating real spaces in the cloud due to their realistic indoor scene captures and extensive annotations. ONCE dataset focuses on part-level co-segmentation, which can contribute to accurate 3D space recreation. However, datasets like ModelNet10/ModelNet40, ScanObjectNN, ShapeNet, and ShapeNetPart are more suitable for tasks like object recognition, instance segmentation, and part-level semantic segmentation in 3D models rather than full-scale real-space recreation.

Despite the numerous datasets available and their massive data volume, publicly accessible point cloud datasets are still limited. This is due to the countless scenes in life that cannot be entirely captured, regardless of dataset size. Consequently, the creation of large-scale, high-quality point cloud data with wide coverage remains a significant future research topic [29].

2.3. Point Clouds Imaging

In this part, the imaging resolutions of the three methods are introduced and compared which are LiDAR, Photogrammetry, and Structured Light.

LiDAR systems typically provide high-resolution point clouds. The resolution is determined by factors such as the laser pulse rate, laser beam width, and scanning pattern. Higher pulse rates and narrower beam widths generally result in higher imaging resolution. The benefit of LiDAR is that LiDAR point clouds have high accuracy and can capture detailed geometric information with fine resolution. They are particularly useful for

capturing complex scenes and structures. The limitations of LiDAR is that LiDAR systems can be expensive and require sophisticated equipment. They may have limitations in capturing color or texture information [42].

The imaging resolution in Photogrammetric point clouds is influenced by factors like camera sensor resolution, image overlap, and the quality of feature matching algorithms. Higher-resolution cameras and a larger number of high-quality images generally result in higher imaging resolution. The benefits of Photogrammetry are that Photogrammetry is a cost-effective technique, widely accessible through cameras and drones. It can provide detailed and accurate point clouds with good resolution, color information, and texture mapping. The limitations of Photogrammetry are that Photogrammetry may have challenges in capturing accurate depth information, especially in scenes with low texture or occlusions. It may require careful camera calibration and image processing [43].

Structured light systems project known patterns onto a scene and use cameras to capture the deformations. The imaging resolution depends on factors such as the number and complexity of projected patterns, camera sensor resolution, and the accuracy of calibration. Higher-resolution cameras and more detailed patterns can increase the imaging resolution. The benefits of Structured light are that Structured light techniques can provide accurate and detailed point clouds with relatively good resolution. They can capture color and texture information alongside geometric data. The limitations of Structured light are that Structured light requires careful system setup and calibration. The resolution and accuracy can be affected by factors like ambient lighting conditions and the presence of reflective or glossy surfaces [44,45].

2.4. Point Cloud Transformation Algorithms

In this part, 5 commonly used point cloud transformation algorithms and an overview of their computational efficiency are introduced and compared, which are Iterative Closest Point (ICP), Normal Distribution Transform (NDT), Moving Least Squares (MLS), Voxel Grid Downsampling, and Principal Component Analysis (PCA).

ICP is an iterative algorithm used for point cloud registration and alignment [46]. The computational time of ICP depends on the number of iterations required to converge and the complexity of distance calculations, typically between (N^2) and (N^3), where N is the number of points. ICP can be time-consuming, especially for large point clouds, and may require initial alignment estimates for convergence. However, there are variants and optimizations available, such as parallelization and approximate nearest neighbor search, to improve efficiency [47].

NDT is a technique used for point cloud registration by estimating a probability distribution of the point data. The computational time of NDT depends on the voxel grid resolution, typically between (N) and (N^2), where N is the number of points. NDT can be computationally efficient, especially for large point clouds, as it uses voxel grids to accelerate computations. However, higher grid resolutions increase memory requirements and may impact processing time [48].

MLS is a method used for point cloud smoothing and surface reconstruction. The computational time of MLS depends on the radius used for local computations and the number of neighbors, typically between $(NlogN)$ and (N^2), where N is the number of points. Efficiency Considerations: MLS can be relatively efficient, especially with optimized data structures like kd-trees for nearest neighbor searches. However, larger radii and denser neighborhood computations can impact processing time [49].

Voxel grid downsampling is a technique used to reduce the density of point clouds by grouping points within voxel volumes [50]. The computational time of voxel grid downsampling is typically (N), where N is the number of points. Voxel grid downsampling is efficient as it involves spatial partitioning, enabling faster processing of large point clouds. The processing time is influenced by the size of the voxel grid used.

PCA is a statistical method used for feature extraction and dimensionality reduction in point clouds [51]. The computational time of PCA depends on the number of dimensions

and the number of points, typically between (DN^2) and (DN^3), where D is the number of dimensions and N is the number of points. PCA can be computationally efficient for moderate-sized point clouds, but for high-dimensional data, dimensionality reduction techniques may be required to maintain efficiency [52].

3. The Representation of 3D Model

Pictures and videos usually use the arrangement and combination of pixels to convey information. Similarly, 3D models also need to realize the overall recognition by various means. Various representation methods emerge endlessly. In this section, the common representation are introduced for 3D data, and Figure 2 shows different representation.

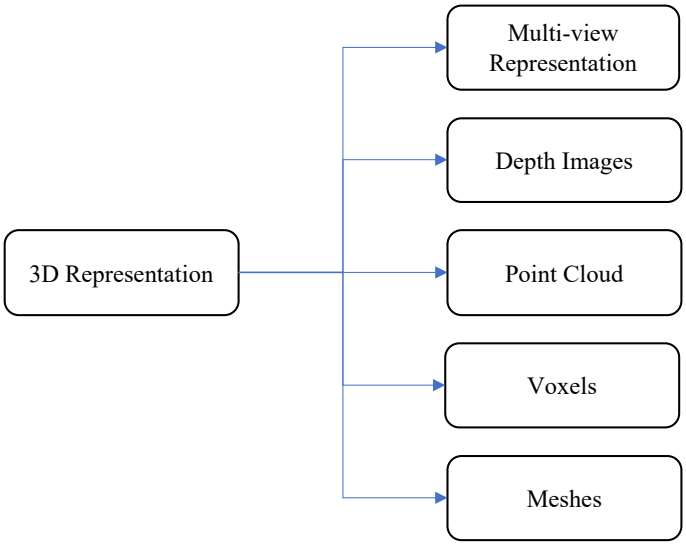

Figure 2. The 3D representation.

Multi-view Representation: This method is the simplest way to show a 3D model. As we know, 2D model have less representation, also it is easy for observers describing a 3D model with a single viewpoint. Therefore, a series of 2D capturing from different viewpoints can be used to show a 3D shape. Because of reducing one dimension, it is relatively convenient and efficient for the observers to record a 3D shape while shrinking the size of the data [11,53].

Depth Images: The use of depth images can provide the distance between the camera and the scene to each pixel. First, depth images can be obtained from multi-view or stereo images, where a disparity map is calculated for each pixel in the image, but we usually use the form of RGB-D data to represent such images. Because RGB-D data are composed of color images and corresponding depth images, depth sensors such as kinect can easily obtain RGB-D data [54]. Since the object can only be seen from one side, the depth image cannot describe the shape entirely because the depth image is captured from a viewpoint. Fortunately, thanks to huge advances in 2D processing, many 2D algorithms can use these data directly [55]. For instance, depth images can be used to enhance the performance of 2D semantic segmentation algorithms. Semantic segmentation is the task of classifying each pixel in an image into a category (like “car”, “tree”, “road”). By providing depth information as an extra channel of input data, along with the traditional RGB channels, the model can learn to better understand the spatial relations in the scene, which can improve segmentation results. There are numerous semantic segmentation models, such as U-Net, SegNet, and DeepLabv3+, that can benefit from the additional depth data. Also, object detection is another field where depth images can be directly used. Algorithms like R-CNN, YOLO (You Only Look Once), or SSD (Single Shot MultiBox Detector) can benefit from depth information to better locate and categorize objects. This becomes particularly useful in crowded or overlapping scenes where the depth information helps distinguish between

different objects. Last but not least, algorithms that deal with image restoration tasks, such as de-noising or super-resolution, can also benefit from depth information. This can be especially beneficial in scenarios such as restoring old photographs or improving the quality of images captured in poor lighting conditions. The depth information can provide additional context about the scene that can assist the model in reconstructing finer details.

Point Cloud: A point cloud is a group of unordered points in 3D space, which are represented by coordinates on the x, y, and z axes, and from which a specific 3D shape can be formed [56]. The coordinates of these points can be obtained from one or more views using a 3D scanner, such as the RGB-D cameras or LiDAR mentioned earlier. At the same time, RGB cameras can capture color information. These color information can be selectively superimposed on the point cloud as additional information to enrich the content expressed by the point cloud. A point cloud is an unordered set, so it differs from the image usually represented by a matrix. Therefore, a permutation invariant method is crucial for processing such data, so as to ensure that the results do not change with the order of the points in the cloud.

Voxels: For a picture, pixels are made up of small squares of an image. These squares have a clear position and specific color. The color and position of the small squares determine the appearance of the image. Therefore, we can also define a similar concept named "voxels" as the pixels. In 3D space, a voxel representation provides information on regular grid [57,58]. Voxels can be obtained from point clouds in the voxelization process, in which all features of 3D points within a voxel are grouped for subsequent processing. The structure of 3D voxels is similar to that of 2D. For example, convolution, in 2D convolution the kernel slides in 2D, while in 3D convolution the kernel slides in 3D instead of 2D as in 2D convolution. Since voxels contain a large number of empty volumes corresponding to the space around the object, in general, the voxel representation is relatively sparse. In addition, since most capture sensors can only collect information on the surface of an object, the interior of the object is also represented by empty volume.

Meshes: Unlike voxels, a mesh incorporates more elements and is a collection of vertices, edges, and faces (polygons) [59]. Its basic components are polygons and planar shapes defined by the connection of a set of 3D vertices. Point clouds, in contrast, can only provide vertex locations, but because grids incorporate more elements, they can contain information about the surface of an object. This way of representing 3D models is very common in computer graphics applications. Nonetheless, surface information is difficult to process directly using deep learning methods, and in order to transform the mesh representation into a point cloud, many techniques pursue sampling points from the surface [60].

4. 3D Transformer

4.1. 3D Transformer Architecture

The archetypal Transformer model, which employs an encoder–decoder framework, is illustrated here, where the encoder represents the upper module, and the decoder, the lower. This section provides a comprehensive introduction to both the encoder and decoder.

In the encoder, we can identify Ne identical blocks, while Nd identical blocks constitute the decoder. Each block within the encoder comprises a multi-head self-attention sublayer and a feedforward network. Through the feedforward network, the multilayer perceptron can efficiently transform the features of each input element. Conversely, the multi-head self-attention sublayer is capable of capturing the relations between varying input elements. Post each sublayer, a normalization operation and a residual connection are employed to further enhance the model's efficiency.

The decoder follows a similar pattern, with each block adding a multi-head cross-attention sublayer compared to its encoder counterpart. This decoder block includes a multi-head cross-attention sub-layer, multi-head self-attention sub-layer, and a feed-forward network. The multi-head self-attention sublayer is designed to capture the relationship between different decoder elements, while using the encoder output as the key and value

of the multi-head cross-attention sublayer to attend to the encoder output. Similarly to the encoder, a multilayer perceptron can transform the features of each input element via the feedforward network in the decoder. Moreover, a normalization operation and a residual connection follow each sublayer in the decoder, mirroring the encoder's structure.

The architecture of the model is illustrated in Figure 3.

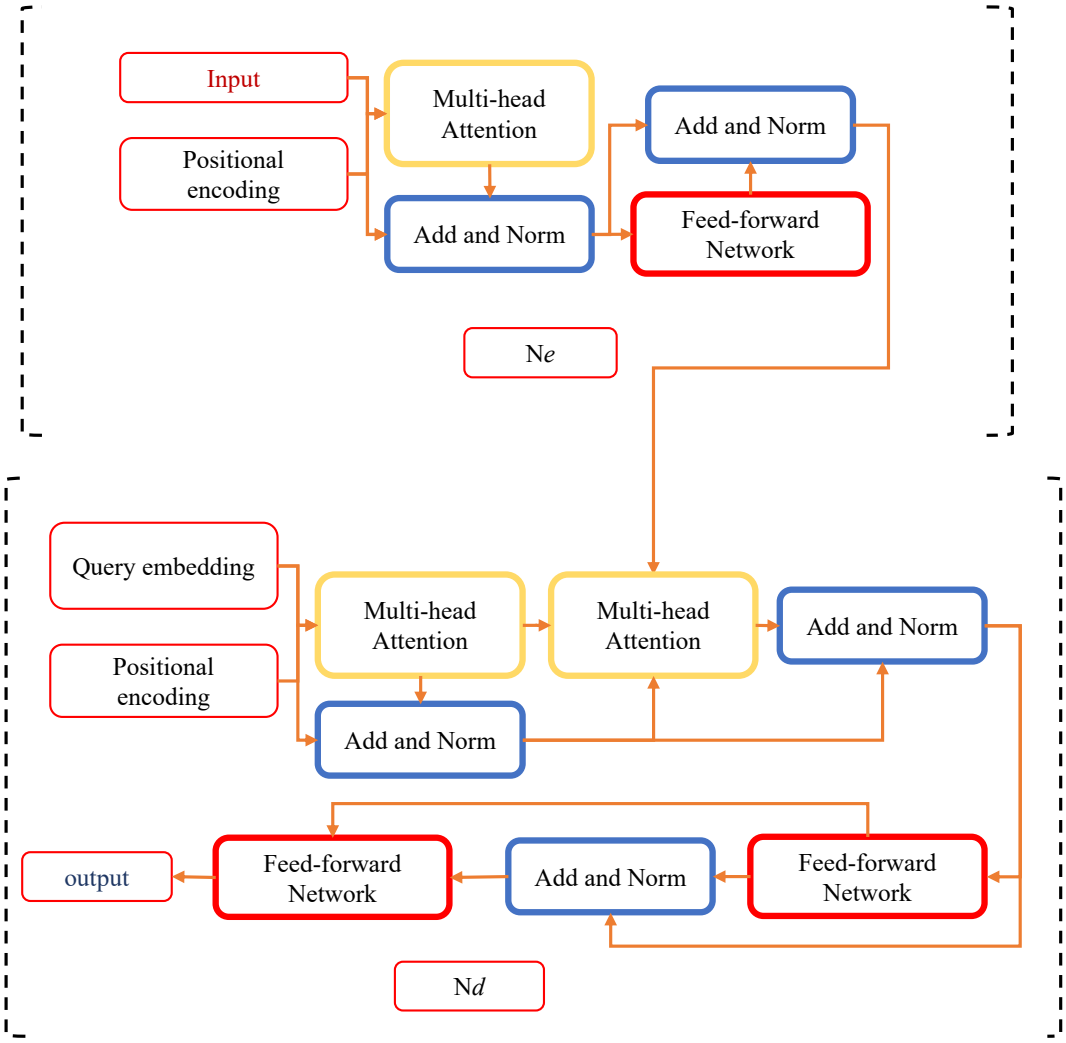

Figure 3. The transformer model.

4.2. Classification of 3D Transformers

In Section 3, we discussed various methods to represent 3D data, including Multi-view Representation, Depth Images, Point Clouds, Voxels, and Meshes. Each of these representations can serve as input to 3D transformers. Considering the intrinsic properties of voxels and points, these entities can be represented interchangeably and parently; that is, points can be either depicted by voxels or transformed into voxels. Consequently, some voxel methods can be applied to point clouds to fulfill the requirements of 3D transformers. Moreover, depending on the differing input formats, two classifications of methods are generated as illustrated in Figure 4, Voxel-based Transformers and Point-based Transformers [9]. Voxel-based Transformers contain the VoxSet [61], PVT [62], and SVT-Net [63]. Point-based Transformers contain two size Transformers. For Uniform Scale, there are PCTMA-Net [64], PCT [65], and Point-BERT [66]. For Multi-Scale, there are 3DCTN [67], PPT [68], PT [69], and Stratified Transformer [70]. In this section, we delve into these two types of transformers to introduce.

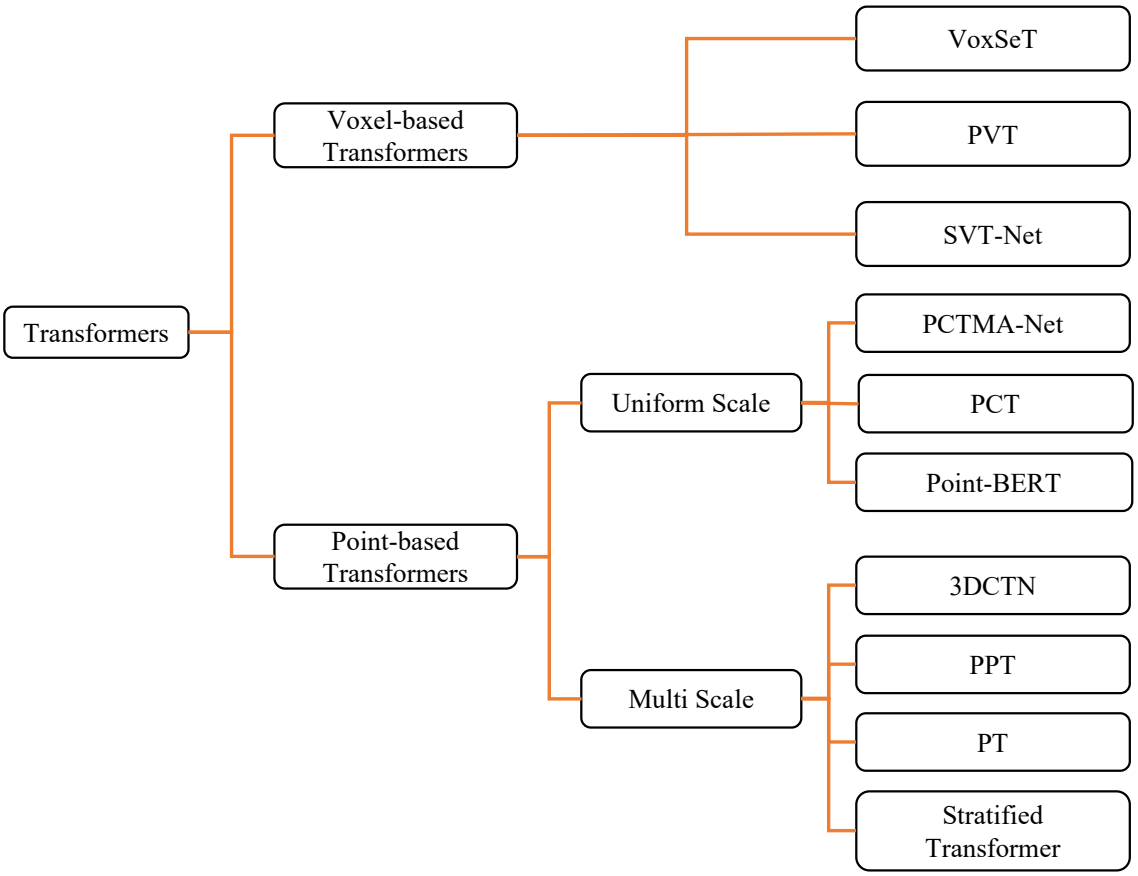

Figure 4. The classification of 3D transformers.

Point-based Transformers: Initially, it should be noted that points follow an irregular format, unlike the regular structure of voxels. Therefore, during the process of point-to-voxel conversion, due to the constraints imposed by this regimented format, geometric information may be inevitably lost to some extent [71,72]. Conversely, given that the point cloud is the most raw representation, formed by the aggregation of points, comprehensive geometric information is inherent in the point cloud. Consequently, the majority of Transformer-based point cloud processing frameworks fall under the category of point Transformer-based. Their architectures are usually bifurcated into two main categories, Multi-Scale architectures [67–70,73,74] and Uniform Scale architectures [64,65,75–78].

Voxel-based Transformers: 3D point clouds are typically unstructured, which starkly contrasts with the structure of images. Therefore, conventional convolution operators cannot process this kind of data directly. However, by simply converting the 3D point cloud into 3D voxels, this challenge can be easily addressed. The 3D voxel structure bears similarities to images. As such, many transformer works aim to convert 3D point clouds into voxel representations [61,62,70,79]. The most commonly employed voxelization method is outlined as follows [80]. Firstly, the bounding box of the point cloud is systematically divided into 3D cuboids via rasterization. The voxels containing the points are retained, thereby generating a voxel representation of the point cloud.

5. Applications

Deep learning has found widespread use in numerous 3D vision applications, mainly permeating newer domains, such as medical imaging [81], geometry coding [82], sound creation [83], robotic control and manufacturing [84,85], virtual reality [86], computational hallucination [87], deep learning-based actuators [88], autonomous vehicles [89], big data [90], and even in the study of COVID-19 during the pandemic [91] and forecasting weather patterns and valuable trends [92]. These domains are intimately connected with

data analytics, accumulation of experience, locational judgment, simulation, and computation [93,94]. In this section, we will explore automated applications from three perspectives, object classification, object detection, and pose estimation. To elucidate the applications of deep learning more effectively, we provide detailed examples that demonstrate the various application directions of deep learning:

Google's AlphaGo program triumphed over Lee Sedol in the Go game in 2016, attesting to deep learning's robust capacity to learn advanced strategies and memorize diverse paths to victory. Google's Deep Dream, apart from classifying images, can also generate bizarre, artificial drawings drawing upon its own knowledge base. Moreover, the "Lingjing" APP, which gained significant popularity in China in 2023, can create its own painting style based on the pictures uploaded by users, and generate new images as per the specifications provided by users [95]. ChatGPT, known globally, can engage in question-and-answer interactions with users utilizing its existing knowledge base, build a new knowledge system according to the information input by users, enrich its knowledge pool, and use it to respond to and infer more complex user requirements.

There are two types of tasks in application as depicted in Figures 5 and 6.

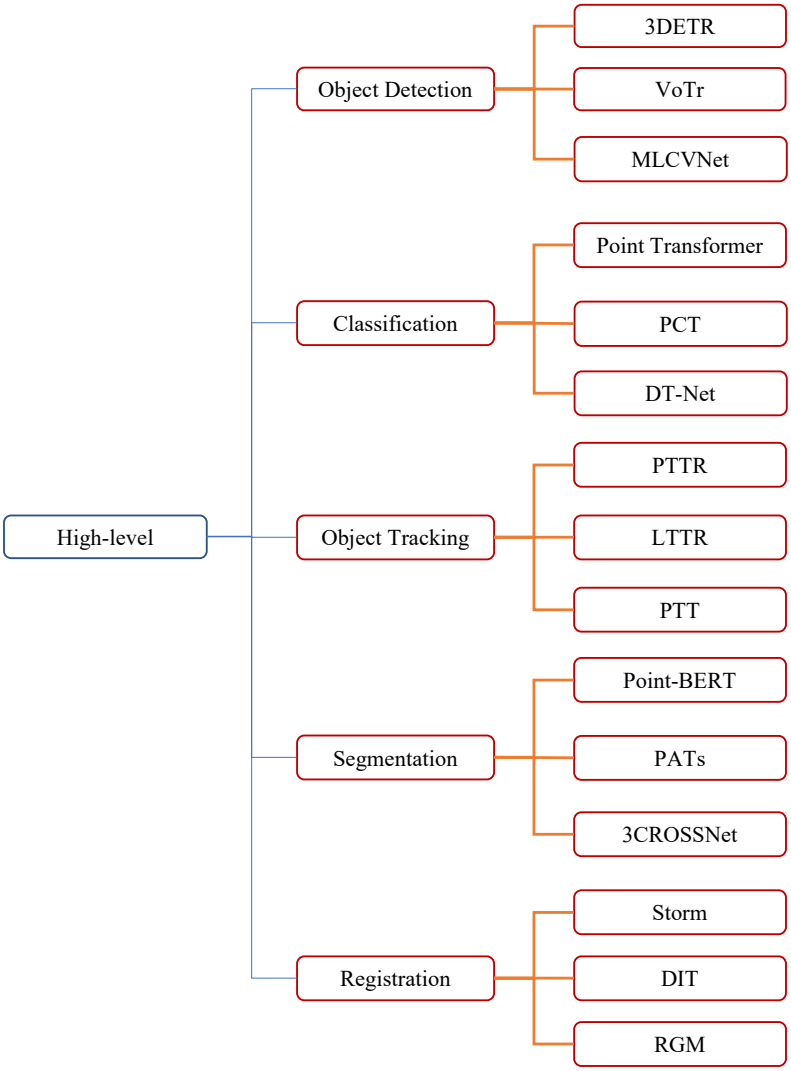

Figure 5. The classification of high-level tasks.

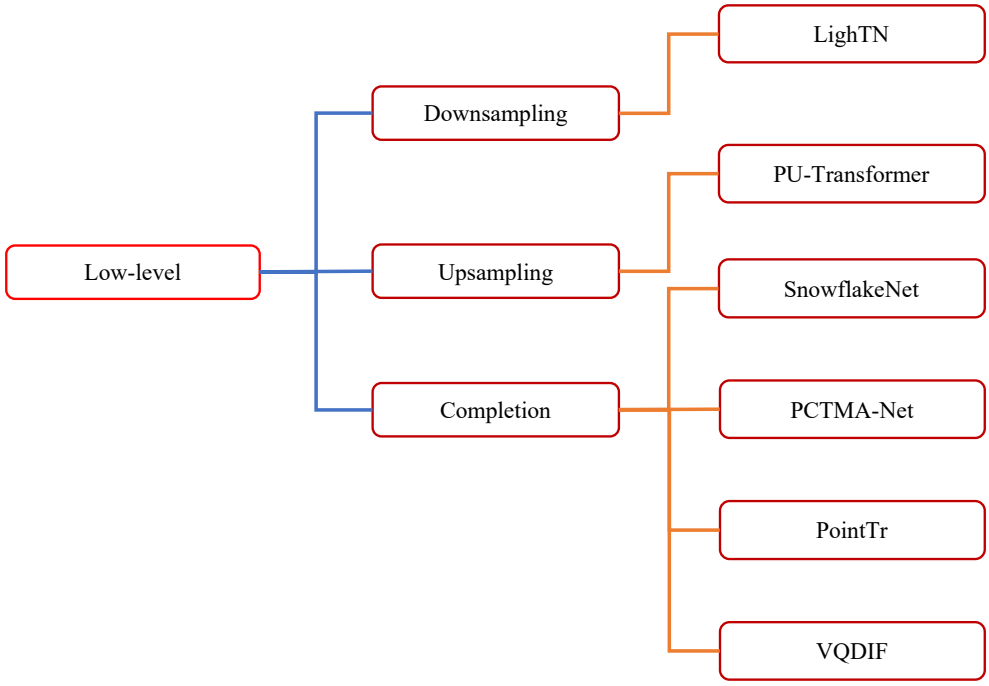

Figure 6. The classification of low-level tasks.

5.1. 3D Object Detection

The objective of 3D object detection is to predict the rotation bounding box of a 3D object [18,96–103]. Three-dimensional object detectors demonstrate distinct differences when compared to 2D detectors. For instance, Vote3Deep [104] leverages feature-centric voting [105] to efficiently process sparse 3D point clouds on evenly spaced 3D voxels. A unified feature representation can be produced through the combination of 3D sparse convolutions and 2D convolutions in the detection head, a requirement that necessitates VoxelNet [106] to use PointNet [72] within each voxel. Building upon this, SECOND [101] simplifies the VoxelNet process and makes the 3D convolution sparse [107]. To take it a step further, PIXOR [108], in order to eliminate costly 3D convolutions, projects all points onto a 2D feature map equipped with 3D occupancy and point intensity information. Moreover, to enhance backbone efficiency, PointPillars [97] replaces all voxel calculations with a columnar representation featuring one elongated voxel for each map location. By merging multi-view features, MVF [109] and pillar-od [110] can learn a more effective column representation [111].

Jean Lahoud et al. posit that 3D object detection can be divided into two components, indoor 3D object detection and outdoor 3D object detection, with different datasets being employed for each situation [11]. The SUN RGB-D dataset is one of the most commonly utilized datasets for indoor 3D object detection [17]. It comprises 10,335 RGB-D frames, each containing 37 oriented bounding boxes and patterns of object classes, with the test set consisting of 5050 frames and the training set comprising 5285 frames.

In the realm of outdoor 3D object detection, 3D object detection plays a pivotal role in autonomous driving. The KITTI dataset [18] is one of the most frequently used datasets in this field due to its precise and clear provision of 3D object detection annotations. The KITTI dataset encompasses 7518 test samples and 7481 training samples, with standard average precision being used for easy, medium, and hard difficulty levels. The KITTI dataset enables the use of either LiDAR or RGB as input, or both. As per Lahoud, methods utilizing LiDAR information tend to outperform those relying solely on RGB, given that the necessary 3D information is not contained in RGB, rendering it incapable of accurately placing the bounding box [112,113]. In contrast, Monoflex [114] falls short of the transformer-based architecture MonoDETR [115].

For the LiDAR input, PDV [116] demonstrates the best performance in the Easy car category. It employs 3D sparse convolution computation and uses a self-attention module to obtain long-range dependencies of grid points. In terms of moderate difficulty, the Voxel Transformer [79] and the Voxel Set Transformer [61] achieve the best performance. The methods for 3D object detection are depicted in Figure 7.

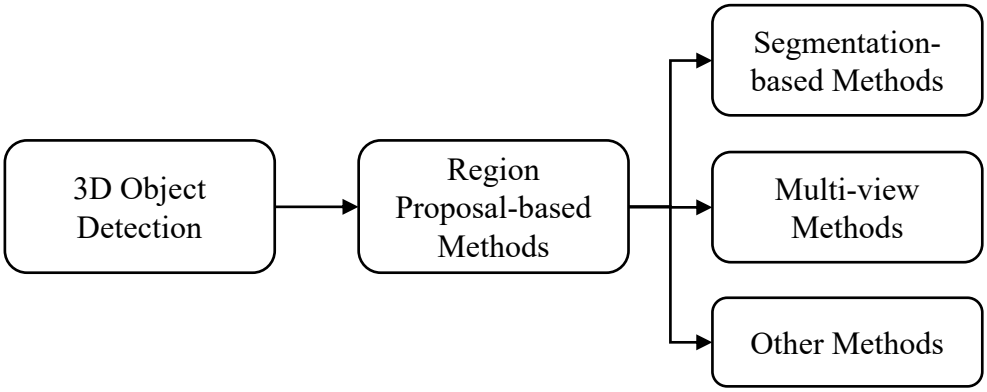

Figure 7. Methods for 3D object detection.

5.2. 3D Object Classification

Object classification in deep learning pertains to the identification of an object's category or class present in data sources such as images, videos, or other types of data [117]. This involves training a neural network model on a substantial dataset of labeled images, with each image being associated with a distinct object class. The trained model can subsequently be employed to predict the class of objects in novel, unseen images [72,118]. In a previous discussion, we introduced image classification [119,120], a task that Dening Lu described as being similar to point cloud classification [9]. However, point cloud classification does not deal with 2D or 3D images but rather with a set of points in space that represent a 3D object or scene [121,122].

Point cloud classification [65,69,123,124] strives to classify each point in the cloud into a predefined set of categories or classes [66,75,125]. This task frequently arises in the fields of robotics, autonomous vehicles, and other computer vision applications where sensor data are represented in the form of point clouds. In order to classify a given 3D shape into a specific category, certain unique characteristics must be identified. Each object possesses unique shape features that distinguish it from other objects [70]. For instance, a human would not categorize apples and pears as the same fruit; they apply the same principle [74,126]. Additionally, 3D classification presents different types. Indoor classifications typically involve small objects such as fruits, appliances, everyday items, and furniture. In contrast, outdoor applications commonly classify larger objects like cars, buildings, and people with varying characteristics moving on the street [73,127].

To perform point cloud classification, a deep learning model is trained on a substantial dataset of labeled point clouds, with each point in the cloud associated with a specific class. The model learns to extract relevant features from the point cloud data and classify each point into its respective category. The methods for 3D object detection are depicted in Figure 8.

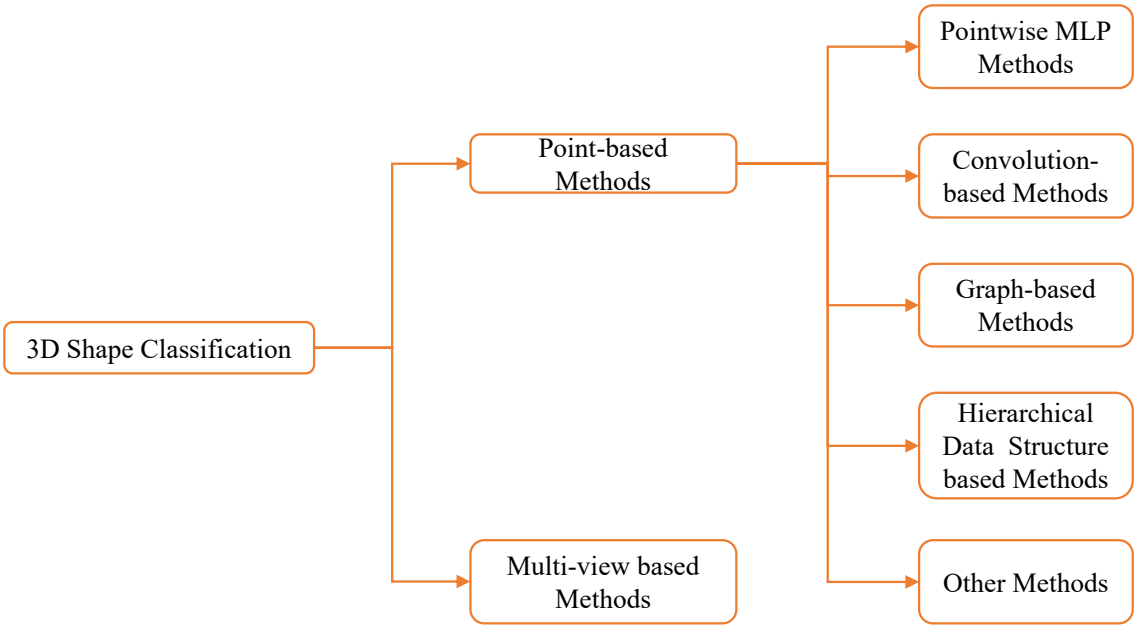

Figure 8. The methods can be used in 3D classification.

5.3. 3D Object Tracking

Three-dimensional object tracking in deep learning refers to the detection and tracking of the 3D position and movement of one or multiple objects within a scene over time. This process involves training a neural network model on an extensive dataset of labeled 3D objects or scenes, each annotated with its corresponding 3D position and movement over a period of time [128,129].

The purpose of 3D object tracking is to precisely track the movement of one or multiple objects in real-world environments, a crucial component in various computer vision applications, such as robotics, autonomous vehicles, and surveillance.

A deep learning model is trained on a large dataset of labeled 3D objects or scenes for 3D object tracking, with each object or scene annotated according to its respective 3D position and movement over time. The model learns to extract pertinent features from the 3D data and to track the object's or objects' movement in real time. During inference, the trained model is applied to new, unseen 3D data to track the object's or objects' movement in the scene over time. The model output comprises a set of 3D coordinates and trajectories, representing the movement of the object or objects in 3D space over time. Figure 9 illustrates various methods for 3D object tracking [130].

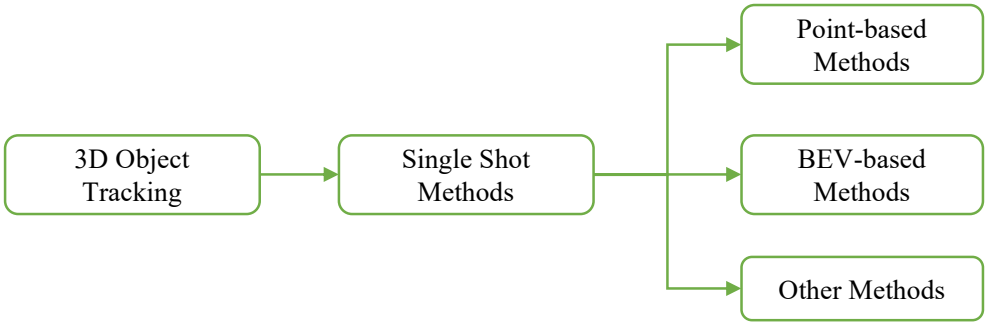

Figure 9. Various methods for 3D object tracking.

5.4. 3D Estimation

Three-dimensional pose estimation in deep learning pertains to estimating the 3D position and orientation of an object or scene from a 2D image or set of 2D images [131]. This process involves training a neural network model on an extensive dataset of labeled

images and their corresponding 3D poses, with each pose representing the position and orientation of the object or scene in 3D space [132].

Three-dimensional pose estimation aims to accurately estimate the 3D pose of an object or scene in real-world settings, a key aspect in various computer vision applications, such as robotics, augmented reality, and autonomous vehicles [133–135].

To perform 3D pose estimation, a deep learning model is trained on an extensive dataset of labeled images and their corresponding 3D poses [136]. The model learns to extract pertinent features from the 2D images and estimate the 3D pose of the object or scene. During inference, the trained model is applied to new, unseen images to estimate the 3D pose of the object or scene in real-time. The model output comprises a set of 3D coordinates and orientations, representing the position and orientation of the object or scene in 3D space [137].

5.5. 3D Segmentation

Three-dimensional segmentation [138,139] in deep learning involves dividing a 3D object or scene into meaningful parts or regions [118,140,141]. This process necessitates training a neural network model on an extensive dataset of labeled 3D objects or scenes, with each object or scene segmented into its constituent parts or regions. The trained model can then predict segmentation labels for new, unseen 3D data [72,142–144].

In point cloud 3D segmentation [145], the goal is to partition a 3D point cloud into distinct regions based on their semantic meaning. This task is vital in robotics, autonomous vehicles, and other computer vision applications where sensor data are represented in the form of point clouds [92,146–148]. For instance, point cloud segmentation can be employed to identify different parts of a car, such as wheels, doors, and windows [107,144,149].

To execute point cloud 3D segmentation, a deep learning model is trained on an extensive dataset of labeled point clouds, with each point in the cloud associated with a specific semantic label [150]. The model learns to extract pertinent features from the point cloud data and segment points into different regions based on their semantic meaning. The model output is a set of labels corresponding to different regions of the point cloud, which can be used for further analysis and processing [92,151,152]. Figure 10 showcases methods for 3D segmentation.

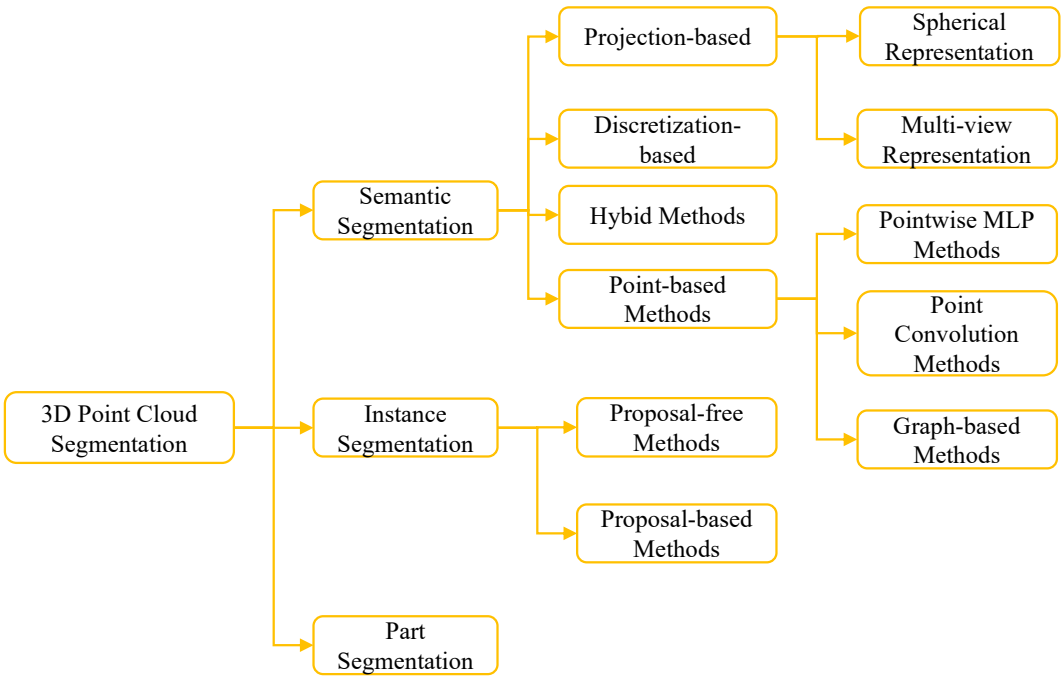

Figure 10. Methods for 3D segmentation.

5.6. 3D Point Cloud Completion

Three-dimensional point cloud completion in deep learning pertains to reconstructing missing or incomplete 3D point cloud data. This process involves training a neural network model on a comprehensive dataset of incomplete point clouds, where each point cloud lacks some points or possesses incomplete information. The trained model can then generate complete point clouds from new, incomplete point cloud data [151].

The purpose of 3D point cloud completion is to recover the missing information within the point cloud and create a comprehensive 3D representation of the object or scene. This task holds significant importance in robotics, autonomous vehicles, and other computer vision applications where sensor data may be incomplete or noisy. For instance, point cloud completion can generate a comprehensive 3D map of a scene, even when some parts of the scene are obscured or missing due to sensor limitations.

To perform 3D point cloud completion, a deep learning model is trained on an extensive dataset of incomplete point clouds, each paired with a corresponding complete point cloud. The model learns to extract relevant features from the incomplete point cloud data and generate missing or incomplete points to reconstruct a comprehensive 3D representation of the object or scene. The model's output is a complete point cloud, available for further analysis and processing.

PoinTr [153] introduces a novel perspective by transforming point cloud completion into a set-to-set translation task. In this approach, the input point cloud can be represented as a set of local points, termed "point proxies". Leveraging these point proxies along with intelligent prediction, missing parts of the point cloud can be generated, thereby accomplishing completion.

Conversely, Xiang et al. [154] proposed a distinct approach from PoinTr. Their primary concept perceives the point cloud completion task as a snowflake-like growth of 3D points, hence the introduction of SnowflakeNet. This approach focuses more on restoring fine geometric details of the complete point cloud, such as edges and surfaces. The Snowflake Point Deconvolution (SPD) is capable of generating multiple points from any given point, capturing contextual and spatial information from the aggregated points effectively.

6. Discussion and Conclusions

This review provides a comprehensive overview of the latest advancements in the deep learning-based processing and application of 3D point cloud data, with a particular emphasis on classification, tracking, pose estimation, segmentation, and completion. Through our synthesis of recent studies, we have highlighted the choice of model architectures, learning algorithms, and the diversity of application domains that have seen significant progress.

Three-dimensional point cloud data processing, underpinned by deep learning techniques, is advancing rapidly and has vast potential across multiple disciplines. From an industrial perspective, techniques such as point cloud completion and segmentation are increasingly applied in robotics, autonomous vehicles, and other computer vision applications, and they exhibit immense developmental prospects in commercial markets.

Recent literature emphasizes the importance of fine-tuning these techniques to address real-world challenges, for instance, recovering fine geometric details in point cloud completion or ensuring accurate pose estimation in complex environments. The introduction of transformative approaches like PoinTr and SnowflakeNet signal a new era in the field and provide promising directions for future research.

Further research in deep learning techniques for 3D point cloud data processing can focus on several areas. There is a need to improve the robustness of models to handle noisy and incomplete point cloud data, as well as challenging scenarios like occlusions and cluttered environments [155,156]. Techniques that enhance model resilience and enable reliable processing in real-world conditions are essential. In addition, transfer learning [157] and domain adaptation [158] methods can address the scarcity of labeled datasets by leveraging pre-trained models on large-scale 3D datasets and transferring the learned

knowledge to new tasks or domains with limited labeled data. This approach can enhance the efficiency and generalization capabilities of models. Furthermore, there is a demand for developing explainable [159] and interpretable [160] models to gain insights into the decision-making process of deep learning models for 3D point cloud data. Exploring techniques that provide explanations and interpretations of model outputs can enhance transparency and trustworthiness in critical applications. Therefore, further research in these areas will contribute to the advancement and practical applicability of deep learning techniques in 3D point cloud data processing.

The limitations of the research include possible biases in the literature selection due to chosen search literatures and databases, the challenge of covering all aspects of deep learning techniques for 3D point cloud data processing in a single review, and the subjective nature of the evaluation and identification of future research areas based on the author's perspective and interpretation of the literature.

The authors maintain a positive outlook on the future of this technology, especially in application areas demanding high-quality 3D perception, such as autonomous driving and robotics. This review, by summarizing the current knowledge and identifying areas that warrant further investigation, aspires to foster new ideas and facilitate the next wave of breakthroughs in the rapidly progressing domain of 3D point cloud data processing.

Author Contributions: Z.D. wrote the draft paper and designed the tables and figures. Y.S. and S.X. edited the tables, formulas, and illustrations. Y.P. (Yan Pan) edited the tables, formulas, and illustrations in the revision and polished the sentences. Y.P. (Yanhong Peng) developed the conception and coordinated the project. Z.M. acquired the funding and supervised the project. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Duan, H.; Wang, P.; Huang, Y.; Xu, G.; Wei, W.; Shen, X. Robotics dexterous grasping: The methods based on point cloud and deep learning. *Front. Neurorobot.* **2021**, *15*, 658280. [CrossRef]
2. Wang, Z.; Xu, Y.; He, Q.; Fang, Z.; Xu, G.; Fu, J. Grasping pose estimation for SCARA robot based on deep learning of point cloud. *Int. J. Adv. Manuf. Technol.* **2020**, *108*, 1217–1231. [CrossRef]
3. Peng, Y.; Yamaguchi, H.; Funabora, Y.; Doki, S. Modeling Fabric-Type Actuator Using Point Clouds by Deep Learning. *IEEE Access* **2022**, *10*, 94363–94375. [CrossRef]
4. Yue, X.; Wu, B.; Seshia, S.A.; Keutzer, K.; Sangiovanni-Vincentelli, A.L. A lidar point cloud generator: From a virtual world to autonomous driving. In Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, Yokohama, Japan, 11–14 June 2018; pp. 458–464.
5. Cui, Y.; Chen, R.; Chu, W.; Chen, L.; Tian, D.; Li, Y.; Cao, D. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 722–739. [CrossRef]
6. Srivastava, A.M.; Rotte, P.A.; Jain, A.; Prakash, S. Handling Data Scarcity Through Data Augmentation in Training of Deep Neural Networks for 3D Data Processing. *Int. J. Semant. Web Inf. Syst. IJSWIS* **2022**, *18*, 1–16. [CrossRef]
7. Lee, S.; Jeon, M.; Kim, I.; Xiong, Y.; Kim, H.J. Sagemix: Saliency-guided mixup for point clouds. *arXiv* **2022**, arXiv:2210.06944.
8. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4338–4364. [CrossRef] [PubMed]
9. Lu, D.; Xie, Q.; Wei, M.; Xu, L.; Li, J. Transformers in 3d point clouds: A survey. *arXiv* **2022**, arXiv:2205.07417
10. Zeng, C.; Wang, W.; Nguyen, A.; Yue, Y. Self-Supervised Learning for Point Clouds Data: A Survey. *arXiv* **2023**, arXiv:2305.11881.
11. Lahoud, J.; Cao, J.; Khan, F.S.; Cholakkal, H.; Anwer, R.M.; Khan, S.; Yang, M.H. 3d vision with transformers: A survey. *arXiv* **2022**, arXiv:2208.04309.
12. Xiao, A.; Huang, J.; Guan, D.; Zhang, X.; Lu, S.; Shao, L. Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**. [CrossRef]
13. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
14. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3d model repository. *arXiv* **2015**, arXiv:1512.03012.

15. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5828–5839.
16. Uy, M.A.; Pham, Q.H.; Hua, B.S.; Nguyen, T.; Yeung, S.K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1588–1597.
17. Song, S.; Lichtenberg, S.P.; Xiao, J. Sun rgb-d: A rgb-d scene understanding benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 567–576.
18. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
19. Li, G.; Jiao, Y.; Knoop, V.L.; Calvert, S.C.; van Lint, J.W.C. Large Car-following Data Based on Lyft level-5 Open Dataset: Following Autonomous Vehicles vs. Human-driven Vehicles. *arXiv* **2023**, arXiv:2305.18921.
20. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2446–2454.
21. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
22. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv* **2017**, arXiv:1704.03847.
23. Rottensteiner, F.; Sohn, G.; Jung, J.; Gerke, M.; Baillard, C.; Benitez, S.; Breitkopf, U. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *1*, 293–298. [CrossRef]
24. Varney, N.; Asari, V.K.; Graehling, Q. DALES: A large-scale aerial LiDAR data set for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 186–187.
25. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9297–9307.
26. Serna, A.; Marcotegui, B.; Goulette, F.; Deschaud, J.E. Paris-rue-Madame database: A 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In Proceedings of the 4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014, Angers, France, 6–8 March 2014.
27. Roynard, X.; Deschaud, J.E.; Goulette, F. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Int. J. Robot. Res.* **2018**, *37*, 545–557. [CrossRef]
28. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3d semantic parsing of large-scale indoor spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543.
29. Xiao, A.; Huang, J.; Guan, D.; Lu, S. Unsupervised representation learning for point clouds: A survey. *arXiv* **2022**, arXiv:2202.13589.
30. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
31. Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; Lopez, A.M. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3234–3243.
32. Mao, J.; Niu, M.; Jiang, C.; Liang, H.; Chen, J.; Liang, X.; Li, Y.; Ye, C.; Zhang, W.; Li, Z.; et al. One million scenes for autonomous driving: Once dataset. *arXiv* **2021**, arXiv:2106.11037.
33. Patil, A.; Malla, S.; Gang, H.; Chen, Y.T. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9552–9557.
34. Chang, M.F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. Argoverse: 3d tracking and forecasting with rich maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, California, CA, USA, 20–24 May 2019; pp. 8748–8757.
35. Geyer, J.; Kassahun, Y.; Mahmudi, M.; Ricou, X.; Durgesh, R.; Chung, A.S.; Hauswald, L.; Pham, V.H.; Mühlegg, M.; Dorn, S.; et al. A2d2: Audi autonomous driving dataset. *arXiv* **2020**, arXiv:2004.06320.
36. Pham, Q.H.; Sevestre, P.; Pahwa, R.S.; Zhan, H.; Pang, C.H.; Chen, Y.; Mustafa, A.; Chandrasekhar, V.; Lin, J. A 3D dataset: Towards autonomous driving in challenging environments. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 2267–2273.
37. Munoz, D.; Bagnell, J.A.; Vandapel, N.; Hebert, M. Contextual classification with functional max-margin markov networks. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 975–982.

38. Vallet, B.; Brédif, M.; Serna, A.; Marcotegui, B.; Paparoditis, N. TerraMobilita/iQmulus urban point cloud analysis benchmark. *Comput. Graph.* **2015**, *49*, 126–133. [CrossRef]
39. Tan, W.; Qin, N.; Ma, L.; Li, Y.; Du, J.; Cai, G.; Yang, K.; Li, J. Toronto-3D: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 202–203.
40. Siddiqi, K.; Zhang, J.; Macrini, D.; Shokoufandeh, A.; Bouix, S.; Dickinson, S. Retrieving articulated 3-D models using medial surfaces. *Mach. Vis. Appl.* **2008**, *19*, 261–275. [CrossRef]
41. Muro, M.; Maxim, R.; Whiton, J. *Automation and Artificial Intelligence: How Machines Are Affecting People and Places*; Brookings Institution: Washington, DC, USA, 2019.
42. Behroozpour, B.; Sandborn, P.A.; Wu, M.C.; Boser, B.E. Lidar system architectures and circuits. *IEEE Commun. Mag.* **2017**, *55*, 135–142. [CrossRef]
43. Mikhail, E.M.; Bethel, J.S.; McGlone, J.C. *Introduction to Modern Photogrammetry*; John Wiley & Sons: Hoboken, NJ, USA, 2001.
44. Bell, T.; Li, B.; Zhang, S. Structured light techniques and applications. In *Wiley Encyclopedia of Electrical and Electronics Engineering*; Wiley: Hoboken, NJ, USA, 1999; pp. 1–24.
45. Angelsky, O.V.; Bekshaev, A.Y.; Hanson, S.G.; Zenkova, C.Y.; Mokhun, I.I.; Jun, Z. Structured light: Ideas and concepts. *Front. Phys.* **2020**, *8*, 114. [CrossRef]
46. Chetverikov, D.; Svirko, D.; Stepanov, D.; Krsek, P. The trimmed iterative closest point algorithm. In Proceedings of the 2002 International Conference on Pattern Recognition, Quebec City, QC, Canada, 11–15 August 2002; Volume 3, pp. 545–548.
47. Zhang, J.; Yao, Y.; Deng, B. Fast and robust iterative closest point. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3450–3466. [CrossRef]
48. Biber, P.; Straßer, W. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 3, pp. 2743–2748.
49. Cheng, Z.Q.; Wang, Y.; Li, B.; Xu, K.; Dang, G.; Jin, S. A Survey of Methods for Moving Least Squares Surfaces. In Proceedings of the VG/PBG@ SIGGRAPH, Los Angeles, CA, USA, 10–11 August 2008; pp. 9–23.
50. Orts-Escolano, S.; Morell, V.; Garcia-Rodriguez, J.; Cazorla, M. Point cloud data filtering and downsampling using growing neural gas. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; pp. 1–8.
51. Abdi, H.; Williams, L.J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459. [CrossRef]
52. Ringnér, M. What is principal component analysis? *Nat. Biotechnol.* **2008**, *26*, 303–304. [CrossRef]
53. Li, Y.; Yang, M.; Zhang, Z. A survey of multi-view representation learning. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 1863–1883. [CrossRef]
54. Xiong, F.; Zhang, B.; Xiao, Y.; Cao, Z.; Yu, T.; Zhou, J.T.; Yuan, J. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 793–802.
55. Masoumian, A.; Rashwan, H.A.; Cristiano, J.; Asif, M.S.; Puig, D. Monocular depth estimation using deep learning: A review. *Sensors* **2022**, *22*, 5353. [CrossRef]
56. Han, X.F.; Jin, J.S.; Wang, M.J.; Jiang, W.; Gao, L.; Xiao, L. A review of algorithms for filtering the 3D point cloud. *Signal Process. Image Commun.* **2017**, *57*, 103–112. [CrossRef]
57. Ashburner, J.; Friston, K.J. Voxel-based morphometry—The methods. *Neuroimage* **2000**, *11*, 805–821. [CrossRef]
58. Ashburner, J.; Friston, K.J. Why voxel-based morphometry should be used. *Neuroimage* **2001**, *14*, 1238–1243. [CrossRef]
59. Tam, G.K.; Cheng, Z.Q.; Lai, Y.K.; Langbein, F.C.; Liu, Y.; Marshall, D.; Martin, R.R.; Sun, X.F.; Rosin, P.L. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graph.* **2012**, *19*, 1199–1217. [CrossRef]
60. Bassier, M.; Vergauwen, M.; Poux, F. Point cloud vs. mesh features for building interior classification. *Remote Sens.* **2020**, *12*, 2224. [CrossRef]
61. He, C.; Li, R.; Li, S.; Zhang, L. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8417–8427.
62. Zhang, C.; Wan, H.; Liu, S.; Shen, X.; Wu, Z. Pvt: Point-voxel transformer for 3d deep learning. *arXiv* **2021**, arXiv:2108.06076.
63. Fan, Z.; Song, Z.; Liu, H.; Lu, Z.; He, J.; Du, X. Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 551–560. [CrossRef]
64. Lin, J.; Rickert, M.; Perzylo, A.; Knoll, A. Pctma-net: Point cloud transformer with morphing atlas-based point generation network for dense point cloud completion. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September 2021; pp. 5657–5663.
65. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.J.; Martin, R.R.; Hu, S.M. Pct: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [CrossRef]

66. Yan, X.; Zheng, C.; Li, Z.; Wang, S.; Cui, S. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2022; pp. 5589–5598.
67. Lu, D.; Xie, Q.; Gao, K.; Xu, L.; Li, J. 3DCTN: 3D convolution-transformer network for point cloud classification. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24854–24865. [CrossRef]
68. Hui, L.; Yang, H.; Cheng, M.; Xie, J.; Yang, J. Pyramid point cloud transformer for large-scale place recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 6098–6107.
69. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF international Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 16259–16268.
70. Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; Jia, J. Stratified transformer for 3d point cloud segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8500–8509.
71. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5105–5114.
72. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
73. Yu, J.; Zhang, C.; Wang, H.; Zhang, D.; Song, Y.; Xiang, T.; Liu, D.; Cai, W. 3d medical point transformer: Introducing convolution to attention networks for medical point cloud analysis. *arXiv* **2021**, arXiv:2112.04863.
74. Han, X.F.; Jin, Y.F.; Cheng, H.X.; Xiao, G.Q. Dual transformer for point cloud analysis. *IEEE Trans. Multimed.* **2022**, 1–20. [CrossRef]
75. Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; Lu, J. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 19313–19322.
76. Mao, Z.; Asai, Y.; Yamanoi, A.; Seki, Y.; Wiranata, A.; Minaminosono, A. Fluidic rolling robot using voltage-driven oscillating liquid. *Smart Mater. Struct.* **2022**, *31*, 105006. [CrossRef]
77. Chen, G.; Wang, M.; Yue, Y.; Zhang, Q.; Yuan, L. Full transformer framework for robust point cloud registration with deep information interaction. *arXiv* **2021**, arXiv:2112.09385.
78. Gao, X.Y.; Wang, Y.Z.; Zhang, C.X.; Lu, J.Q. Multi-head self-attention for 3D point Cloud classification. *IEEE Access* **2021**, *9*, 18137–18147. [CrossRef]
79. Mao, J.; Xue, Y.; Niu, M.; Bai, H.; Feng, J.; Liang, X.; Xu, H.; Xu, C. Voxel transformer for 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 3164–3173.
80. Xu, Y.; Tong, X.; Stilla, U. Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry. *Autom. Constr.* **2021**, *126*, 103675. [CrossRef]
81. Shen, D.; Wu, G.; Suk, H.I. Deep learning in medical image analysis. *Annu. Rev. Biomed. Eng.* **2017**, *19*, 221–248. [CrossRef]
82. Guarda, A.F.; Rodrigues, N.M.; Pereira, F. Adaptive deep learning-based point cloud geometry coding. *IEEE J. Sel. Top. Signal Process.* **2020**, *15*, 415–430. [CrossRef]
83. Ghose, S.; Prevost, J.J. Autofoley: Artificial synthesis of synchronized sound tracks for silent videos with deep learning. *IEEE Trans. Multimed.* **2020**, *23*, 1895–1907. [CrossRef]
84. Pierson, H.A.; Gashler, M.S. Deep learning in robotics: A review of recent research. *Adv. Robot.* **2017**, *31*, 821–835. [CrossRef]
85. Peng, Y.; Li, D.; Yang, X.; Ma, Z.; Mao, Z. A Review on Electrohydrodynamic (EHD) Pump. *Micromachines* **2023**, *14*, 321. [CrossRef]
86. Kang, T.; Chae, M.; Seo, E.; Kim, M.; Kim, J. DeepHandsVR: Hand interface using deep learning in immersive virtual reality. *Electronics* **2020**, *9*, 1863. [CrossRef]
87. Yeh, A.H.W.; Norn, C.; Kipnis, Y.; Tischer, D.; Pellock, S.J.; Evans, D.; Ma, P.; Lee, G.R.; Zhang, J.Z.; Anishchenko, I.; et al. De novo design of luciferases using deep learning. *Nature* **2023**, *614*, 774–780. [CrossRef]
88. Mao, Z.B.; Asai, Y.; Wiranata, A.; Kong, D.Q.; Man, J. Eccentric actuator driven by stacked electrohydrodynamic pumps. *J. Zhejiang Univ. Sci. A* **2022**, *23*, 329–334. [CrossRef]
89. Kuutti, S.; Bowden, R.; Jin, Y.; Barber, P.; Fallah, S. A survey of deep learning applications to autonomous vehicle control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 712–733. [CrossRef]
90. Chen, X.W.; Lin, X. Big data deep learning: Challenges and perspectives. *IEEE Access* **2014**, *2*, 514–525. [CrossRef]
91. Awassa, L.; Jdey, I.; Dhahri, H.; Hcini, G.; Mahmood, A.; Othman, E.; Haneef, M. Study of Different Deep Learning Methods for Coronavirus (COVID-19) Pandemic: Taxonomy, Survey and Insights. *Sensors* **2022**, *22*, 1890. [CrossRef]
92. Liu, W.; Sun, J.; Li, W.; Hu, T.; Wang, P. Deep learning on point clouds and its application: A survey. *Sensors* **2019**, *19*, 4188. [CrossRef]
93. Shinde, P.P.; Shah, S. A review of machine learning and deep learning applications. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018; pp. 1–6.
94. Wang, Q.; Kim, M.K. Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Adv. Eng. Inform.* **2019**, *39*, 306–319. [CrossRef]

95. Gheisari, M.; Wang, G.; Bhuiyan, M.Z.A. A survey on deep learning in big data. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017; Volume 2, pp. 173–180.
96. Ding, Z.; Hu, Y.; Ge, R.; Huang, L.; Chen, S.; Wang, Y.; Liao, J. 1st Place Solution for Waymo Open Dataset Challenge–3D Detection and Domain Adaptation. *arXiv* **2020**, arXiv:2006.15505.
97. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 12697–12705.
98. Mao, Z.; Yoshida, K.; Kim, J.W. A micro vertically-allocated SU-8 check valve and its characteristics. *Microsyst. Technol.* **2019**, *25*, 245–255. [CrossRef]
99. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-task multi-sensor fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7345–7353.
100. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum pointnets for 3d object detection from rgb-d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 918–927.
101. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef]
102. Yang, Z.; Sun, Y.; Liu, S.; Jia, J. 3dssd: Point-based 3d single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2022; pp. 11040–11048.
103. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. Std: Sparse-to-dense 3d object detector for point cloud. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1951–1960.
104. Engelcke, M.; Rao, D.; Wang, D.Z.; Tong, C.H.; Posner, I. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1355–1361.
105. Wang, D.Z.; Posner, I. Voting for voting in online point cloud object detection. In Proceedings of the Robotics: Science and Systems, Rome, Italy, 13–15 July 2015; Volume 1, pp. 10–15.
106. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
107. Graham, B.; Engelcke, M.; Van Der Maaten, L. 3d semantic segmentation with submanifold sparse convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9224–9232.
108. Yang, B.; Luo, W.; Urtasun, R. Pixor: Real-time 3d object detection from point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7652–7660.
109. Zhou, Y.; Sun, P.; Zhang, Y.; Anguelov, D.; Gao, J.; Ouyang, T.; Guo, J.; Ngiam, J.; Vasudevan, V. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In Proceedings of the Conference on Robot Learning, PMLR, Auckland, New Zealand, 16–18 November 2022; pp. 923–932.
110. Wang, Y.; Fathi, A.; Kundu, A.; Ross, D.A.; Pantofaru, C.; Funkhouser, T.; Solomon, J. Pillar-based object detection for autonomous driving. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 18–34.
111. Yin, T.; Zhou, X.; Krahenbuhl, P. Center-based 3d object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11784–11793.
112. Doki, K.; Suzuki, K.; Torii, A.; Mototani, S.; Funabora, Y.; Doki, S. AR video presentation using 3D LiDAR information for operator support in mobile robot teleoperation. In Proceedings of the 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Herl'any, Slovakia, 21–23 January 2021; pp. 59–64.
113. Maeda, K.; Doki, S.; Funabora, Y.; Doki, K. Flight path planning of multiple UAVs for robust localization near infrastructure facilities. In Proceedings of the IECON 2018–44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 2522–2527.
114. Zhang, Y.; Lu, J.; Zhou, J. Objects are different: Flexible monocular 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3289–3298.
115. Zhang, R.; Qiu, H.; Wang, T.; Xu, X.; Guo, Z.; Qiao, Y.; Gao, P.; Li, H. Monodetr: Depth-aware transformer for monocular 3d object detection. *arXiv* **2022**, arXiv:2203.13310.
116. Hu, J.S.; Kuai, T.; Waslander, S.L. Point density-aware voxels for lidar 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8469–8478.
117. Socher, R.; Huval, B.; Bath, B.; Manning, C.D.; Ng, A. Convolutional-recursive deep learning for 3d object classification. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 656–664.
118. Grilli, E.; Menna, F.; Remondino, F. A review of point clouds segmentation and classification algorithms. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 339. [CrossRef]
119. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]

120. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
121. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
122. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
123. Xie, S.; Liu, S.; Chen, Z.; Tu, Z. Attentional shapecontextnet for point cloud recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4606–4615.
124. Mao, Z.; Shimamoto, G.; Maeda, S. Conical frustum gel driven by the Marangoni effect for a motor without a stator. *Colloids Surf. A Physicochem. Eng. Asp.* **2021**, *608*, 125561. [CrossRef]
125. Gao, Y.; Liu, X.; Li, J.; Fang, Z.; Jiang, X.; Huq, K.M.S. LFT-Net: Local feature transformer network for point clouds analysis. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 2158–2168. [CrossRef]
126. Qiu, S.; Anwar, S.; Barnes, N. Geometric back-projection network for point cloud classification. *IEEE Trans. Multimed.* **2021**, *24*, 1943–1955. [CrossRef]
127. Yang, J.; Zhang, Q.; Ni, B.; Li, L.; Liu, J.; Zhou, M.; Tian, Q. Modeling point clouds with self-attention and gumbel subset sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3323–3332.
128. Cui, Y.; Fang, Z.; Shan, J.; Gu, Z.; Zhou, S. 3d object tracking with transformer. *arXiv* **2021**, arXiv:2110.14921.
129. Funabora, Y. Flexible fabric actuator realizing 3D movements like human body surface for wearable devices. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1 October 2018; pp. 6992–6997.
130. Zhou, C.; Luo, Z.; Luo, Y.; Liu, T.; Pan, L.; Cai, Z.; Zhao, H.; Lu, S. Pttr: Relational 3d point cloud object tracking with transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8531–8540.
131. Li, Y.; Snavely, N.; Huttenlocher, D.P.; Fua, P. Worldwide pose estimation using 3d point clouds. In *Large-Scale Visual Geo-Localization*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 147–163.
132. Sanchez, J.; Denis, F.; Coeurjolly, D.; Dupont, F.; Trassoudaine, L.; Checchin, P. Robust normal vector estimation in 3D point clouds through iterative principal component analysis. *ISPRS J. Photogramm. Remote Sens.* **2020**, *163*, 18–35. [CrossRef]
133. Vock, R.; Dieckmann, A.; Ochmann, S.; Klein, R. Fast template matching and pose estimation in 3D point clouds. *Comput. Graph.* **2019**, *79*, 36–45. [CrossRef]
134. Guo, J.; Xing, X.; Quan, W.; Yan, D.M.; Gu, Q.; Liu, Y.; Zhang, X. Efficient center voting for object detection and 6D pose estimation in 3D point cloud. *IEEE Trans. Image Process.* **2021**, *30*, 5072–5084. [CrossRef]
135. Funabora, Y.; Song, H.; Doki, S.; Doki, K. Position based impedance control based on pressure distribution for wearable power assist robots. In Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, 5–8 October 2014; pp. 1874–1879.
136. Wu, W.; Wang, Z.; Li, Z.; Liu, W.; Fuxin, L. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *arXiv* **2019**, arXiv:1911.12408.
137. Zhou, J.; Huang, H.; Liu, B.; Liu, X. Normal estimation for 3D point clouds via local plane constraint and multi-scale selection. *Comput.-Aided Des.* **2020**, *129*, 102916. [CrossRef]
138. Xu, G.; Cao, H.; Zhang, Y.; Ma, Y.; Wan, J.; Xu, K. Adaptive channel encoding transformer for point cloud analysis. In Proceedings of the Artificial Neural Networks and Machine Learning–ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, 6–9 September 2022; Proceedings, Part III; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–13.
139. Wang, Z.; Wang, Y.; An, L.; Liu, J.; Liu, H. Local Transformer Network on 3D Point Cloud Semantic Segmentation. *Information* **2022**, *13*, 198. [CrossRef]
140. Malinverni, E.S.; Pierdicca, R.; Paolanti, M.; Martini, M.; Morbidoni, C.; Matrone, F.; Lingua, A. Deep learning for semantic segmentation of 3D point cloud. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W15*, 735–742. [CrossRef]
141. Nguyen, A.; Le, B. 3D point cloud segmentation: A survey. In Proceedings of the 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM), Manila, Philippines, 12–15 November 2013; pp. 225–230.
142. He, Y.; Yu, H.; Liu, X.; Yang, Z.; Sun, W.; Wang, Y.; Fu, Q.; Zou, Y.; Mian, A. Deep learning based 3D segmentation: A survey. *arXiv* **2021**, arXiv:2103.05423.
143. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. Segcloud: Semantic segmentation of 3d point clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 537–547.
144. Hackel, T.; Wegner, J.D.; Schindler, K. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 177–184. [CrossRef]
145. Wu, L.; Liu, X.; Liu, Q. Centroid transformers: Learning to abstract with attention. *arXiv* **2021**, arXiv:2102.08606.
146. Feng, M.; Zhang, L.; Lin, X.; Gilani, S.Z.; Mian, A. Point attention network for semantic segmentation of 3D point clouds. *Pattern Recognit.* **2020**, *107*, 107446. [CrossRef]
147. Zermas, D.; Izzat, I.; Papanikolopoulos, N. Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5067–5073.

148. Douillard, B.; Underwood, J.; Kuntz, N.; Vlaskine, V.; Quadros, A.; Morton, P.; Frenkel, A. On the segmentation of 3D LIDAR point clouds. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2798–2805.
149. Xie, Y.; Tian, J.; Zhu, X.X. Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 38–59. [CrossRef]
150. Liu, S.; Fu, K.; Wang, M.; Song, Z. Group-in-group relation-based transformer for 3d point cloud learning. *Remote Sens.* **2022**, *14*, 1563. [CrossRef]
151. Huang, Q.; Wang, W.; Neumann, U. Recurrent slice networks for 3d segmentation of point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2626–2635.
152. Gélard, W.; Herbulot, A.; Devy, M.; Debaeke, P.; McCormick, R.F.; Truong, S.K.; Mullet, J. Leaves segmentation in 3d point cloud. In Proceedings of the Advanced Concepts for Intelligent Vision Systems: 18th International Conference, ACIVS 2017, Antwerp, Belgium, 18–21 September 2017; Proceedings 18; Springer: Berlin/Heidelberg, Germany, 2017; pp. 664–674.
153. Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; Zhou, J. Pointr: Diverse point cloud completion with geometry-aware transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 12498–12507.
154. Xiang, P.; Wen, X.; Liu, Y.S.; Cao, Y.P.; Wan, P.; Zheng, W.; Han, Z. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 5499–5509.
155. Zeng, A.; Yu, K.T.; Song, S.; Suo, D.; Walker, E.; Rodriguez, A.; Xiao, J. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1383–1386.
156. Bassier, M.; Vergauwen, M.; Van Genechten, B. Automated classification of heritage buildings for as-built BIM using machine learning techniques. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 25–30. [CrossRef]
157. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In Proceedings of the Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; Proceedings, Part III 27; Springer: Berlin/Heidelberg, Germany, 2018; pp. 270–279.
158. Dutta, S. An overview on the evolution and adoption of deep learning applications used in the industry. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1257. [CrossRef]
159. Parascandolo, G.; Neitz, A.; Orvieto, A.; Gresele, L.; Schölkopf, B. Learning explanations that are hard to vary. *arXiv* **2020**, arXiv:2009.00329.
160. Li, X.; Xiong, H.; Li, X.; Wu, X.; Zhang, X.; Liu, J.; Bian, J.; Dou, D. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowl. Inf. Syst.* **2022**, *64*, 3197–3234. [CrossRef]