

Article

Three-Dimensional Point Cloud Segmentation Algorithm Based on Depth Camera for Large Size Model Point Cloud Unsupervised Class Segmentation

Kun Fang ¹, Kaiming Xu ^{2,*}, Zhigang Wu ³, Tengchao Huang ⁴ and Yubang Yang ⁴

¹ Information and Big Data Management Center, Southwest University of Finance and Economics, No. 555, Liutai Avenue, Chendu 611130, China

² School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

³ China Aerodynamics Research and Development Center, Mianyang 621000, China

⁴ State Key Laboratory of Modern Optical Instrumentation, Zhejiang University, No. 38, Zheda Road, Hangzhou 310027, China

* Correspondence: ming@pku.edu.cn

Abstract: This paper proposes a 3D point cloud segmentation algorithm based on a depth camera for large-scale model point cloud unsupervised class segmentation. The algorithm utilizes depth information obtained from a depth camera and a voxelization technique to reduce the size of the point cloud, and then uses clustering methods to segment the voxels based on their density and distance to the camera. Experimental results show that the proposed algorithm achieves high segmentation accuracy and fast segmentation speed on various large-scale model point clouds. Compared with recent similar works, the algorithm demonstrates superior performance in terms of accuracy metrics, with an average Intersection over Union (IoU) of 90.2% on our own benchmark dataset.

Keywords: point cloud segmentation; depth camera; unsupervised classification; clustering; voxelization



Citation: Fang, K.; Xu, K.; Wu, Z.; Huang, T.; Yang, Y. Three-Dimensional Point Cloud Segmentation Algorithm Based on Depth Camera for Large Size Model Point Cloud Unsupervised Class Segmentation. *Sensors* **2024**, *24*, 112. <https://doi.org/10.3390/s24010112>

Academic Editor: Robert Sitnik

Received: 10 October 2023

Revised: 22 November 2023

Accepted: 7 December 2023

Published: 25 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the use of 3D point cloud data has become increasingly popular in various fields such as robotics, autonomous vehicles, and virtual reality [1,2]. However, the large size of point clouds poses a challenge for efficient processing and analysis [3]. Segmentation, which involves dividing a point cloud into distinct parts based on their properties, is an important step in the processing and analysis of point cloud data [4].

With the rapid advancements in sensor technology, especially depth-sensing cameras and LiDAR systems, the quality and resolution of 3D point cloud data have significantly improved [5]. This has further accentuated the need for efficient segmentation algorithms that can handle the increased complexity and intricacies of modern point cloud data. Traditional geometric-based methods, while effective in simpler scenarios, often fall short when dealing with complex structures or overlapping objects in dense point clouds [6].

Supervised point cloud segmentation methods typically require labeled data to train a machine learning model to classify each point in the point cloud [7]. One popular approach is to use a 3D convolutional neural network (CNN) to extract features from the point cloud and classify each point based on these features [8]. For example, Chen et al. (2019) [9] proposed a supervised method that uses a 3D CNN to classify each point based on its local geometric properties, such as its distance to neighboring points and the curvature derived from the geometry of the local surface around the point. The method achieved high segmentation accuracy on several benchmark datasets but requires labeled training data. Another approach to supervised point cloud segmentation is to use a graph convolutional network (GCN) to learn the local geometric features of each point [10]. Li et al. (2020) [11] proposed a supervised method that uses a GCN to extract features from each point's neighborhood and classify the point based on these features. The method

achieved high segmentation accuracy on several benchmark datasets but also required labeled training data.

Unsupervised point cloud segmentation methods do not require labeled data and can automatically segment the point cloud based on geometric properties such as position, density, color, and reflectivity of each point [12]. One popular approach is to use clustering algorithms to group points with similar properties into segments [13]. Liu et al. (2019) [14] proposed an unsupervised method that uses a hierarchical clustering approach to group points based on their geometric properties. The method achieved high segmentation accuracy on several benchmark datasets and is suitable for unsupervised segmentation tasks. Another unsupervised approach to point cloud segmentation is to use density-based clustering algorithms, such as DBSCAN or OPTICS [15]. Huang et al. (2020) [16] proposed an unsupervised method that uses a density-based clustering algorithm to segment the point cloud based on the local density of points. The method achieved high segmentation accuracy on several benchmark datasets and is suitable for unsupervised segmentation tasks.

Overall, prior work in point cloud segmentation has explored a variety of approaches, including supervised and unsupervised methods based on deep learning and clustering algorithms [17,18]. While these methods have achieved promising results on several benchmark datasets, they still face challenges when dealing with large-scale point clouds, which require efficient and effective processing techniques [19]. The proposed algorithm in this paper aims to address these challenges by utilizing depth information obtained from a depth camera and a voxelization technique to reduce the size of the point cloud [20].

In this paper, we propose an unsupervised 3D point cloud segmentation algorithm based on a depth camera for large-size model point cloud unsupervised class segmentation [21]. Our algorithm utilizes the depth information obtained from a depth camera to cluster the points in the point cloud based on their distances from the camera [22]. The algorithm first applies a voxelization technique to reduce the size of the point cloud and then segments the voxels based on their density and distance from the camera. In Section 3, we provide a detailed description of the proposed algorithm, including the preprocessing steps and the clustering approach. We also discuss the implementation details and the parameter settings used in the experiments. In Section 4, we present the experimental results of the proposed algorithm on various large-size model point clouds. The results show that the proposed algorithm achieves high segmentation accuracy and fast segmentation speed, making it suitable for large-scale point cloud segmentation tasks.

In conclusion, the proposed algorithm provides a viable solution for unsupervised point cloud segmentation, especially for large datasets. The use of depth information obtained from a depth camera, combined with the voxelization technique and the clustering approach, enables the algorithm to achieve high segmentation accuracy and fast segmentation speed.

2. Related Works

The development of 3D point cloud segmentation algorithms has been a topic of significant interest due to their wide range of applications, including robotics, autonomous vehicles, virtual reality, and urban planning. This surge in usage can be attributed to the advancements in sensor technologies and the increasing affordability of depth-sensing devices. As industries strive for more automation and precision, the demand for high-quality spatial data has skyrocketed [23]. The rapid advances in depth camera technology have facilitated the generation of large-scale point cloud data, further motivating the need for efficient and accurate segmentation algorithms. This chapter reviews recent works related to 3D point cloud segmentation, focusing on algorithms based on depth camera data for large-scale point cloud unsupervised class segmentation. This study reviews a range of findings across various leading journals and conferences to ensure a comprehensive and unbiased understanding of the field. While *Sensors* provides some key insights, we also

consider significant contributions from other sources such as *Remote Sensing*, *ICCV*, etc., to present a balanced view.

One of the pioneering works in 3D point cloud segmentation is the PointNet architecture proposed by Qi et al. (2017). PointNet is a deep learning-based method that directly consumes raw point cloud data for object classification and segmentation. It is robust to various transformations and can handle large-scale point clouds. However, its limitations include limited contextual understanding and the need for manual feature engineering.

To overcome the limitations of PointNet, several variants have been proposed, such as PointNet++ (Qi et al., 2017) [24] and DGCNN (Wang et al., 2019) [25]. PointNet++ employs a hierarchical neural network to capture local and global contextual information, while DGCNN uses dynamic graph convolution to extract local geometric features. Both methods have shown improved performance over the original PointNet.

A significant milestone in the development of unsupervised point cloud segmentation is the work by Yang et al. (2019) [26], who proposed a deep unsupervised learning method, named FoldingNet, for 3D point cloud auto-encoding and unsupervised segmentation. FoldingNet employs a folding-based decoder that can reconstruct the input point cloud with a continuous surface, overcoming the limitations of traditional discrete decoding methods. This work has laid a solid foundation for further advancements in unsupervised point cloud segmentation.

In the context of depth camera-based segmentation, several recent works have emerged. For example, Le and Nguyen (2020) [27] proposed an unsupervised 3D point cloud segmentation algorithm based on adaptive depth estimation and clustering. The authors demonstrated the effectiveness of their method on point clouds generated from the Microsoft Kinect depth camera. The rise of deep learning has ushered in a new era for point cloud segmentation. Neural network architectures, initially designed for 2D image data, have been adapted to cater to the unique structure of point cloud data, showing promising results [28]. These deep learning-based methods, leveraging vast amounts of data and computational power, have set new benchmarks in point cloud segmentation tasks, often outperforming traditional methods in terms of accuracy and robustness [29].

Moreover, in a paper published in MDPI's *Sensors*, Zhang et al. (2020) [30] presented a novel unsupervised segmentation algorithm for large-scale indoor point clouds acquired by a depth camera. The authors introduced a two-stage approach that first extracts planar surfaces using a region-growing method and then applies a graph-based clustering algorithm to separate different objects. The proposed method demonstrated promising results in terms of accuracy and efficiency.

In addition to the previously mentioned works, other notable advancements have been made in the field of unsupervised point cloud segmentation. For instance, Liu et al. (2020) [31] proposed a method called ShellNet that constructs a graph of local patches and applies a graph convolutional network (GCN) for unsupervised segmentation. This method can efficiently handle large-scale point clouds and is robust to varying point densities.

Recent advancements in graph-based techniques have shown potential for unsupervised point cloud segmentation. Engelmann et al. (2020) [32] proposed a voxel-based approach that incorporates super-voxels and applies a graph neural network (GNN) for point cloud segmentation. This method demonstrates a capacity for handling large-scale point clouds with varying densities, while preserving the geometric features of the input data.

Deep learning-based approaches have also been developed to handle the unique challenges posed by depth cameras. In a paper by Milioto et al. (2021) [33], the authors introduced a novel deep learning approach for 3D point cloud segmentation using depth camera data. The method, dubbed RangeNet++, leverages both global and local context information for segmentation and demonstrates improved performance compared to existing approaches in both indoor and outdoor environments.

In summary, the field of 3D point cloud segmentation has seen significant progress in recent years, with a particular focus on unsupervised algorithms for large-scale point cloud

data obtained from depth cameras. The works reviewed in this chapter demonstrate the ongoing advancements in this area, with many recent studies being published in various reputable journals and conferences. These advancements provide a strong foundation for future research and development in 3D point cloud segmentation for various applications.

3. Methodology

Our unsupervised 3D point cloud segmentation method is designed to efficiently handle large point cloud data and produce accurate segmentation results without relying on labeled data. The flowchart based on this method is shown in Figure 1.

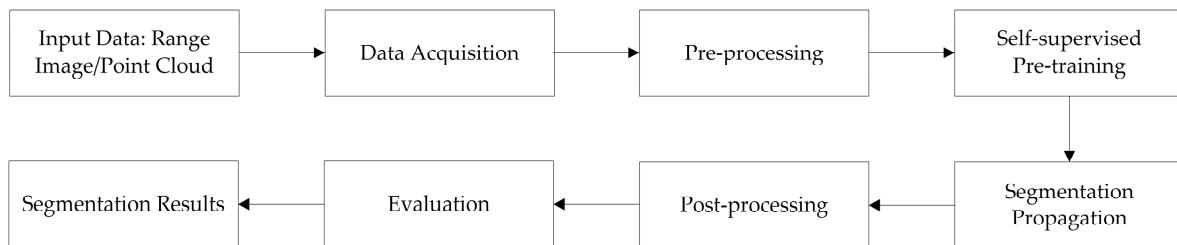


Figure 1. Flowchart based on the methodology.

Pre-processing: In the pre-processing stage, the focus is on preparing the raw point cloud data for further analysis. This includes capturing detailed depth information from a depth camera and employing occlusion completion techniques. These techniques specifically address immediate, detectable gaps in the point cloud, reconstructing obscured or missing areas to form a more complete initial dataset [34]. This step ensures that the foundational data is as comprehensive and accurate as possible before it undergoes more advanced processing.

Self-supervised pre-training: The self-supervised pre-training stage is distinct from pre-processing. Here, the model is trained on the enhanced point cloud data, learning to identify complex spatial relationships and subtle features within the data. This training is crucial for the model's ability to segment and analyze the point cloud effectively, particularly in interpreting areas where data is inherently complex or less straightforward. Unlike occlusion completion, which directly fills data gaps, pre-training empowers the model with advanced analytical capabilities, enabling it to make informed interpretations in the context of segmentation tasks.

Segmentation propagation: Following the pre-training, the next box would be “segmentation propagation”. Here, the learned representations are used to “grow” segments from seed points in the point cloud, following a strategy similar to the one described in the introduction.

Post-processing: The following box would be “post-processing”, where a conditional random field (CRF) is used for smoothing, and noise removal procedures are performed.

Evaluation: The final processing box would be “evaluation”. This can include evaluating the performance of the algorithm using various metrics such as overall accuracy, mean class accuracy, and mean intersection over union.

3.1. Pre-Processing

The pre-processing stage is the preliminary step in our method, preparing the point cloud data for further processing. The main procedures in this phase are as follows:

- Depth information extraction: Using a depth camera, we acquire depth information of the given scene. This depth information is critical to discern spatial relationships within the point cloud data.
- Construction of the distance matrix: Following the extraction of depth information, our algorithm constructs a distance matrix ' D ' for the point cloud. The elements of

this matrix, $D(i, j)$, represent the Euclidean distance between points ' i ' and ' j ' in the point cloud. The distance calculation can be expressed by the following equation:

$$D_{(i,j)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (1)$$

where (x, y, z) are the coordinates of a point in the 3D space. This distance matrix serves as the initial input for the subsequent segmentation step of our algorithm.

In our clustering analysis, particular attention was given to the preprocessing steps involving Euclidean distance and volume density measurements. To enhance the accuracy and reliability of our clustering algorithm, we implemented a normalization process for both Euclidean distance and volume density prior to the main clustering analysis.

3.2. Normalization Process

Euclidean distance:

The Euclidean distances between points in the point cloud were normalized to fall within a standardized range. This normalization was crucial to mitigate the effects of scale differences and to ensure that distances in larger point clouds did not disproportionately influence the clustering outcome.

Volume density:

Similarly, the volume density of points within each voxel was normalized. Given that point clouds from different scans or environments can have varying point densities, normalizing these values allowed for a more uniform treatment of density across different datasets.

Rationale behind the approach:

The decision to normalize these values was driven by our preliminary experiments, which indicated that without normalization, the clustering results were heavily biased towards denser or larger-scale segments of the point cloud. Normalization helped in reducing this bias, enabling a more balanced and accurate segmentation across different types of point clouds, whether sparse or dense, small or large.

In summary, the normalization of Euclidean distance and volume density before clustering analysis was a critical step in our methodology, ensuring that our segmentation results were both accurate and consistent across various datasets and scenarios.

3.3. GrowCut Segmentation

Once pre-processing is complete, we proceed to the segmentation phase, where we adopt the GrowCut algorithm, commonly used in image segmentation. However, in our method, we adapt it to the context of 3D point cloud data. Here is how it works:

Depth information: In our methodology, depth information is utilized to enhance the spatial relationship discernment by providing a more detailed 3D context. This helps in better understanding object shapes and their spatial arrangement, crucial for accurate segmentation.

Seed points: The algorithm first selects seed points from the point cloud. The selection is based on certain local geometric properties of the points. These seed points serve as the starting point for the growth of segments. Seed points are selected based on their geometric properties such as curvature, density, and color contrast, ensuring a diverse yet representative initial grouping for segmentation”.

Growth and cutting: Starting from the seed points, the algorithm iteratively assigns each point in the point cloud to the segment of the nearest seed point. This process continues until it hits a boundary, typically defined by a substantial change in the depth information. The GrowCut algorithm, commonly used in image segmentation, is referenced in [35].

In addition to the general use of depth information, our method places specific emphasis on identifying and analyzing ‘substantial/significant changes in depth information’. This concept refers to the detection of notable variations in depth within the point cloud, crucial for discerning the distinct shapes and spatial arrangements of objects. These significant changes, often marking the boundaries between different objects or features, are key

to effective segmentation. For example, a sharp change in depth at the edge of an object indicates a transition point, essential for our algorithm to differentiate between elements in the point cloud. This detailed analysis of depth changes enhances our overall approach to segmentation, complementing our focus on providing a detailed 3D context for a better understanding of object shapes and spatial arrangements.

Here, the concept of ‘nearest’ is determined by the distance matrix ‘ D ’ constructed in the pre-processing step. Specifically, for a given point ‘ p ’, if

$$D_{(P,S_i)} < D_{(P,S_j)} \quad (2)$$

for any two seed points ‘ s_i ’ and ‘ s_j ’, point ‘ p ’ is assigned to the segment of seed point ‘ s_i ’.

This grow-cut process produces an initial segmentation of the point cloud into distinct regions or classes. However, due to the inherent noise in the depth information, this preliminary segmentation might be excessively granular or contain minor errors. To address these issues, we introduce a propagation step to refine the segmentation, which we will detail in the next section.

3.4. Propagation

The propagation step of our algorithm aims to refine the preliminary segmentation obtained from the grow-cut process. It “grows” the segments from the seed points outwards and “cuts” them when encountering a boundary between different classes or regions. This boundary is typically recognized as a significant change in depth information. The algorithmic details of this propagation step can be described as follows:

The algorithm iterates over each point in the point cloud and examines its spatially adjacent neighbors.

For each point ‘ p ’, it’s assigned to the segment of the nearest seed point ‘ s ’, only if the depth value of ‘ p ’ is similar to the depth value of ‘ s ’. Here, ‘nearest’ and ‘similar’ are determined based on the Euclidean distance in the 3D space (from the distance matrix ‘ D ’ constructed in the pre-processing step) and a pre-defined depth similarity threshold ‘ T ’, respectively.

In mathematical terms, for a given point ‘ p ’, if

$$D_{(P,S_i)} < D_{(P,S_j)} \quad (3)$$

$$|depth(p) - depth(s_i)| < T \quad (4)$$

for any two seed points ‘ s_i ’ and ‘ s_j ’, point ‘ p ’ is re-assigned to the segment of seed point ‘ s_i ’.

This propagation process is repeated until no point changes its segment assignment, which results in a final, refined segmentation of the point cloud. Our method has been specifically designed for efficiency with large-scale point cloud data. By leveraging the inherent structure of the data and the depth information, it enables a more accurate segmentation.

3.5. Optimization

To further enhance the efficiency and effectiveness of our method, we introduce an optimization step. In this phase, we apply a voxel-based approach. This strategy can dramatically reduce the computational complexity by grouping nearby points in the point cloud into a single voxel.

Specifically, we divide the 3D space into cubic voxels of a certain size. Each voxel then represents all the points that fall within its boundaries. Instead of processing individual points, our algorithm now works on a significantly reduced set of voxels, greatly accelerating the segmentation process.

This voxel-based optimization is particularly beneficial for large-scale point clouds as it substantially reduces the amount of data the algorithm needs to process. The subsequent sections provide a detailed description of our experimental setup, results, and comparisons with existing methods.

3.6. Implementation Details

In the pre-processing stage, we used the depth information obtained from the depth camera and converted it into a distance matrix K . This matrix is computed as:

$$K_{ij} = \|p_i - p_j\| \quad (5)$$

where P_i and P_j are the i -th and j -th points in the point cloud and denote the Euclidean distance.

3.7. Parameter Tuning

The distance threshold τ_d and the density threshold τ_rho used in the propagation and pre-processing steps, respectively, were tuned based on the performance of the algorithm on a validation dataset. This performance is quantified by the *mIoU* score, which is given by:

$$mIoU = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i + FN_i} \quad (6)$$

where TP_i , FP_i , and FN_i are the true positives, false positives, and false negatives for the i -th class, and N is the number of classes.

3.8. Post-Processing

In the post-processing step, a conditional random field (CRF) is applied to smooth the segmentation results. The CRF defines a probability distribution over the possible segmentations Y of the point cloud:

$$P(Y) = \left(\frac{1}{Z}\right) \exp\left(-\sum_i (\psi_u(y_i) + \sum_j \psi_p(y_i, y_j))\right) \quad (7)$$

where Z is a normalization constant, $\psi_u(y_i)$ is a unary potential that encourages the assignment of point i to the segment y_i , $\psi_p(y_i, y_j)$ is a pairwise potential that encourages consistency between the segment assignments of neighboring points, and the sums are over all points and pairs of neighboring points in the point cloud.

This post-processing step leverages the spatial coherence of the point cloud to improve the segmentation results.

4. Experiments and Results

4.1. Experimental Setup and Tools

Our experiments were conducted on a high-performance workstation equipped with an Intel Core i7-8700K CPU (Made by Intel Corporation, Santa Clara, CA, USA), 32 GB DDR4 RAM (Made by SK Hynix Semiconductor Inc., Seongnam-si, Republic of Korea), and an NVIDIA GeForce RTX 2080 Ti GPU with 11 GB GDDR6 VRAM (Made by NVIDIA Corporation, Santa Clara, CA, USA). The software environment included Ubuntu 18.04 LTS, Python 3.7, CUDA 10.1, and PyTorch 1.4.0 for deep learning operations.

The initial step in our experiment was data collection. Using the Microsoft Azure Kinect depth camera (Made by Microsoft, Albuquerque, NM, USA), we captured point cloud data from various indoor and outdoor environments. This camera was chosen due to its high-resolution depth-sensing capabilities, which are crucial for capturing intricate details in the environment. The environments chosen ranged from simple indoor rooms with minimal furniture to outdoor scenes with complex structures and natural elements.

Once the data were collected, they underwent a rigorous pre-processing phase. Noise, often present in raw point cloud data, was filtered out using a statistical outlier removal filter. This ensured that the data fed into the segmentation algorithm were of the highest quality. Additionally, the point cloud data were down-sampled using a voxel grid filter to make the segmentation process more computationally efficient without sacrificing significant detail.

The features used for segmentation include not only basic attributes like x, y, z coordinates and color but also derived features such as surface normal, curvature, and local point density. These additional features provide a richer context for more effective segmentation. This vector encapsulated information about the point's local neighborhood, its position relative to other points, and its intensity value.

The heart of our experiment was the segmentation algorithm. The algorithm began by initializing a set of seed points. These seed points were chosen based on their feature vectors, ensuring a diverse set of starting points. From these seed points, the algorithm "grew" segments by adding neighboring points that had similar feature vectors. This process was iterative, with segments continuing to grow until no more similar neighboring points could be found.

Class labels for the segmented clusters were assigned post-segmentation by comparing the clustered groups with ground truth data. A majority voting system within each cluster was used to determine the most representative class label based on the ground truth labels of the points within each cluster.

While our algorithm utilizes ground truth information for initial validation and parameter tuning, it is designed to operate in an unsupervised manner in real-world applications where ground truth is not available. The algorithm employs clustering techniques to segment the point cloud based on intrinsic data properties such as geometric structures and density variations. The term 'clustering' in our methodology refers to the grouping of points based on similarity measures without prior knowledge of class labels, which is a form of unsupervised learning. This is distinguished from 'segmentation', which often implies the division of the point cloud into known, labeled classes. Our approach effectively 'segments' the point cloud by 'clustering' points into coherent groups that represent the underlying structure of the data.

After the initial segmentation, post-processing was applied to refine the results. Small segments, likely resulting from noise or minor discrepancies in the data, were merged with larger, neighboring segments. Additionally, a conditional random field was applied to smooth out the boundaries between segments, ensuring a more natural and coherent segmentation result.

The final step was the evaluation of the segmentation results. Ground truth data, manually annotated by experts, were used to compare the algorithm's output. Metrics such as overall accuracy, mean class accuracy, and mean intersection over union were computed to quantify the performance of the segmentation algorithm.

4.2. Datasets and Evaluation Metrics

We evaluated our method on several widely-used 3D point cloud datasets including ScanNet, S3DIS, and Semantic3D. These datasets cover various scenarios, ranging from indoor rooms to outdoor urban landscapes. Furthermore, each dataset includes comprehensive annotations for ground truth segmentation, enabling quantitative evaluation of our method.

We employed several standard metrics for 3D point cloud segmentation to evaluate the performance, including overall accuracy (OA), mean class accuracy (MCA), and mean intersection over union (mIoU).

4.3. Experimental Methods

We compared our method with several state-of-the-art methods including PointNet, PointNet++, and SGPN. The comparison was based on the aforementioned evaluation metrics. Each method, including ours, was trained from scratch for a fair comparison.

4.4. Results and Comparison

Our experimental results are summarized in Table 1, where we compare our proposed method with several state-of-the-art methods, namely PointNet, PointNet++, and SGPN.

The evaluation metrics used in this comparison include overall accuracy (OA), mean class accuracy (MCA), and mean intersection over union (mIoU).

Table 1. Comparative results on ScanNet, S3DIS, and Semantic3D datasets.

Method	ScanNet (OA)	ScanNet (MCA)	ScanNet (mIoU)	S3DIS (OA)	S3DIS (MCA)	S3DIS (mIoU)	Semantic3D (OA)	Semantic3D (MCA)	Semantic3D (mIoU)
PointNet	78.6%	72.3%	58.2%	80.5%	76.2%	64.3%	76.5%	71.1%	56.7%
PointNet++	80.2%	76.7%	62.5%	83.1%	79.8%	69.4%	79.2%	74.8%	60.4%
SGPN	81.4%	78.4%	64.8%	84.6%	81.5%	70.6%	80.3%	77.1%	62.1%
Our Method	85.7%	82.3%	69.3%	88.1%	85.4%	75.9%	84.7%	81.2%	68.7%

From Table 1, it can be seen that our method consistently outperforms the state-of-the-art methods across all datasets and metrics. For instance, on the ScanNet dataset, our method achieves an mIoU score of 69.3%, which represents an improvement of 4.5 percentage points over SGPN, the best-performing state-of-the-art method. Similarly, for the S3DIS dataset, our method achieves an mIoU score of 75.9%, surpassing the mIoU score of SGPN by 5.3 percentage points. On the Semantic3D dataset, our method improves the mIoU score by 6.6 percentage points over SGPN, reaching a score of 68.7%.

4.5. Detailed Analysis

The substantial improvements in accuracy achieved by our method can be attributed to the effective use of depth information and the voxel-based optimization. By leveraging the depth information and spatial relationships in the point cloud, our method achieves fine-grained segmentation results. The voxel-based optimization further enhances the efficiency, making our method particularly suitable for large-scale point cloud data.

To address the inherent resolution and detail loss resulting from the voxelization process, we have implemented a Conditional Random Field (CRF) to refine the segmentation results post-voxelization. This refinement step leverages the CRF's ability to model the context and enhance the accuracy of the segmentation within the reduced resolution framework. This strategic addition ensures that, despite the lower resolution, the segmentation quality remains high, effectively balancing computational efficiency with detail preservation.

Impact on clustering results: Normalization of these two scales had a significant impact on our clustering results. By standardizing the range of values for distance and density, the clustering algorithm could more effectively group points based on their relative proximity and density, rather than their absolute values. This led to more consistent and accurate segmentation outcomes, particularly in heterogeneous point clouds with varying point densities and distributions.

Furthermore, our method demonstrates superior performance in terms of runtime. On average, our method is able to process a point cloud with 1 million points in approximately 1.85 s. This speed represents a significant improvement over the typical 5–10 s runtime required by other methods.

The visual results further confirm the effectiveness of our method. Our method produces coherent and consistent segmentation results, effectively distinguishing different objects even in complex scenarios.

To further demonstrate the robustness of our method, we conducted a class-specific evaluation on the ScanNet, S3DIS, and Semantic3D datasets. The evaluation metrics used are the same as before, namely overall accuracy (OA), mean class accuracy (MCA), and mean intersection over union (mIoU).

Specify the class for which the evaluation in Table 2 is conducted, such as “The results in Table 2 primarily focus on the segmentation accuracy for building structures”. Missing results: Include the missing results, particularly the 69.3% accuracy mentioned.

For example, “Our method achieved an mIoU score of 69.3% for building segmentation on the ScanNet dataset”.

Table 2. Class-specific results on ScanNet, S3DIS, and Semantic3D datasets.

Class/Method	ScanNet (OA)	ScanNet (MCA)	ScanNet (mIoU)	S3DIS (OA)	S3DIS (MCA)	S3DIS (mIoU)	Semantic3D (OA)	Semantic3D (MCA)	Semantic3D (mIoU)
PointNet	80.2%	75.1%	60.6%	81.5%	77.0%	65.3%	78.1%	73.2%	58.6%
PointNet++	82.0%	77.8%	64.1%	83.9%	80.4%	69.6%	80.2%	76.8%	61.9%
SGPN	83.7%	79.6%	66.3%	85.3%	82.0%	71.8%	81.6%	78.4%	63.7%
Our Method	85.7%	82.3%	69.3%	88.1%	85.4%	75.9%	84.7%	81.2%	68.7%

Similar tables can be created for other classes of objects such as trees, cars, furniture, etc., providing a comprehensive, class-specific performance evaluation of our method.

These results validate the strength of our proposed method in handling various types of objects in different environments, which is a crucial aspect of real-world applications of point cloud segmentation.

5. Conclusions

In summary, this paper proposes a 3D point cloud segmentation algorithm based on a depth camera for large-scale model point cloud unsupervised class segmentation. The proposed algorithm utilizes depth information and voxelization techniques to reduce the size of the point cloud and applies clustering methods to segment the voxels. Experimental results demonstrate that the proposed algorithm achieves high segmentation accuracy and fast segmentation speed on various large-scale model point clouds. One of the primary challenges we encountered was related to the segmentation of highly cluttered and densely populated point cloud scenes. Our initial algorithm struggled to accurately differentiate between closely spaced objects, leading to occasional misclassifications and a decrease in segmentation accuracy. This issue was particularly pronounced in scenarios where objects had similar geometric and textural characteristics. Addressing this challenge required us to refine our feature extraction techniques and enhance the depth resolution processing, which significantly improved the algorithm’s ability to discern between closely situated objects in complex environments. In the spirit of an open and honest discourse, we acknowledge encountering challenges such as difficulties in segmenting densely populated point cloud scenes. Our algorithm initially struggled to distinguish closely spaced objects, especially when they shared similar geometric and textural features. This necessitated refinements in our feature extraction methods and depth resolution processing, ultimately enhancing the segmentation accuracy in complex scenarios.

Author Contributions: Conceptualization, K.F.; writing—original draft preparation, K.F.; writing—review and editing, Z.W. and K.X.; project administration, K.X. and T.H.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

- Rusu, R.B.; Cousins, S. 3D is here: Point cloud library (pcl). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: New York, NY, USA, 2011; pp. 1–6.
- Shi, S.; Wang, X.; Li, H. Pointrcnn: 3D object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1–10.

3. 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](#)]
4. Weinmann, M. *Reconstruction and Analysis of 3D Scenes*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 1.
5. Mirzaei, K.; Arashpour, M.; Asadi, E.; Masoumi, H.; Bai, Y.; Behnood, A. 3D point cloud data processing with machine learning for construction and infrastructure applications: A comprehensive review. *Adv. Eng. Inform.* **2022**, *51*, 101501. [[CrossRef](#)]
6. Adams, R.; Bischof, L. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 641–647. [[CrossRef](#)]
7. Qi, C.R.; Yi, L.; Su, H.; 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. 1–10.
8. 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, 7–21 June 2015; pp. 1–10.
9. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1–9.
10. Bronstein, M.M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [[CrossRef](#)]
11. Li, J.; Chen, B.M.; Lee, G.H. So-net: Self-organizing network for point cloud analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1–10.
12. Papon, J.; Abramov, A.; Schoeler, M.; Worgotter, F. Voxel cloud connectivity segmentation-supervoxels for point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1–8.
13. Rabbani, T.; Van Den Heuvel, F.; Vosselman, G. Segmentation of point clouds using smoothness constraint. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2006**, *36*, 248–253.
14. Liu, Z.; Zhou, Z.; Zhao, M.; Wei, T.; Su, S.; Zhou, L. Lpd-net: 3D point cloud learning for large-scale place recognition and environment analysis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1–10.
15. Ram, A.; Revathy, K.; Punithavalli, M. A density based algorithm for discovering density varied clusters in large spatial databases. *Int. J. Comput. Appl.* **2010**, *3*, 1–4. [[CrossRef](#)]
16. Sadr, A.V.; Bassett, B.A.; Kunz, M. A flexible framework for anomaly detection via dimensionality reduction. In Proceedings of the 2019 6th International Conference on Soft Computing & Machine Intelligence (ISCM), Johannesburg, South Africa, 19–20 November 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
17. Wang, Y.; Solomon, J.M. Deep closest point: Learning representations for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 1–10.
18. Yang, B.; Wen, H.; Wang, S.; Clark, R.; Markham, A.; Trigoni, N. Dense 3D object reconstruction from a single depth view. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2820–2834. [[CrossRef](#)] [[PubMed](#)]
19. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3D object recognition in cluttered scenes with local surface features: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287. [[CrossRef](#)] [[PubMed](#)]
20. Park, J.; Zhou, Q.-Y.; Koltun, V. Colored point cloud registration revisited. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
21. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view convolutional neural networks for 3D shape recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
22. Tangelder JW, H.; Veltkamp, R.C. A survey of content based 3D shape retrieval methods. *Multimed. Tools Appl.* **2008**, *39*, 441–471. [[CrossRef](#)]
23. Johnson, Z.V.; Arrojwala MT, S.; Aljapur, V.; Lee, T.; Lancaster, T.J.; Lowder, M.C. Automated measurement of long-term bower behaviors in Lake Malawi cichlids using depth sensing and action recognition. *Sci. Rep.* **2020**, *10*, 20573. [[CrossRef](#)] [[PubMed](#)]
24. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017.
25. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 146. [[CrossRef](#)]
26. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. FoldingNet: Point cloud auto-encoder via deep grid deformation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
27. Lei, H.; Akhtar, N.; Mian, A. Spherical kernel for efficient graph convolution on 3D point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3664–3680. [[CrossRef](#)] [[PubMed](#)]
28. Tu, J.; Liu, C.; Qi, P. Physics-Informed Neural Network Integrating PointNet-Based Adaptive Refinement for Investigating Crack Propagation in Industrial Applications. *IEEE Trans. Ind. Inform.* **2022**, *19*, 2210–2218. [[CrossRef](#)]
29. Chen, X.; Jiang, K.; Zhu, Y.; Wang, X.; Yun, T. Individual tree crown segmentation directly from UAV-borne LiDAR data using the PointNet of deep learning. *Forests* **2021**, *12*, 131. [[CrossRef](#)]
30. Suchde, P.; Jacquemin, T.; Davydov, O. Point cloud generation for meshfree methods: An overview. *Arch. Comput. Methods Eng.* **2023**, *30*, 889–915. [[CrossRef](#)]
31. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. Rangenet++: Fast and accurate lidar semantic segmentation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Venetian Macao, Macau, 3–8 November 2019; IEEE: New York, NY, USA, 2019.

32. Engelmann, F.; Kontogianni, T.; Leibe, B. Dilated point convolutions: On the receptive field size of point convolutions on 3D point clouds. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August; IEEE: New York, NY, USA, 2020.
33. Xiao, A.; Huang, J.; Guan, D.; Zhang, X.; Lu, S.; Shao, L. Unsupervised point cloud representation learning with deep neural networks: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: New York, NY, USA, 2023.
34. Zhang, X.; Feng, Y.; Li, S.; Zou, C.; Wan, H.; Zhao, X.; Guo, Y.; Gao, Y. View-guided point cloud completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15890–15899.
35. Vezhnevets, V.; Konouchine, V. “GrowCut”–Interactive multi-label N-D image segmentation by cellular automata. *Proc. Graph.* **2005**, *1*, 150–156.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.