



# Comparative Analysis of 3D Shape Recognition in the Presence of Data Inaccuracies

Mukhaimar, Ayman; Tennakoon, Ruwan; Lai, Chow Yin; Hoseinnezhad, Reza; Bab-Hadiashar, Alireza

<https://researchrepository.rmit.edu.au/esploro/outputs/conferenceProceeding/Comparative-Analysis-of-3D-Shape-Recognition/9921861182801341/filesAndLinks?index=0>

---

Mukhaimar, A., Tennakoon, R., Lai, C. Y., Hoseinnezhad, R., & Bab-Hadiashar, A. (2019). Comparative Analysis of 3D Shape Recognition in the Presence of Data Inaccuracies. *Proceedings of the 26th IEEE International Conference on Image Processing (ICIP 2019)*, 2471–2475.

<https://doi.org/10.1109/ICIP.2019.8803345>

Document Version: Accepted Manuscript

---

Published Version: <https://doi.org/10.1109/ICIP.2019.8803345>

Repository homepage: <https://researchrepository.rmit.edu.au>

© 2019 IEEE

Downloaded On 2024/05/01 05:19:32 +1000

# COMPARATIVE ANALYSIS OF 3D SHAPE RECOGNITION IN THE PRESENCE OF DATA INACCURACIES

Ayman Mukhaimar, Ruwan Tennakoon, Chow Yin Lai, Reza Hoseinnezhad and Alireza Bab-Hadiashar

School of Engineering, RMIT University, Melbourne, Australia.

## ABSTRACT

Classification of 3D shapes into physically meaningful categories is one of the most important tasks in understanding the immediate environment. Methods that leverage the recent advancements in deep learning have shown to outperform the traditional approaches. However, performances of those methods have only been analyzed with relatively clean data. Three-dimensional measurement sets (point clouds) produced by 3D scanners are rarely that accurate and often contain noise, outliers or missing points. This paper presents an extensive analysis of the robustness of the state-of-the-art neural network algorithms to realistic data inaccuracies. Our experiments show that the existence of these inaccuracies can significantly affect the performance of the deep learning-based algorithms.

**Index Terms**— 3D classification, neural networks, point cloud classification, robust 3D classification.

## 1. INTRODUCTION

Classification of 3D shapes into physically meaningful categories is one of the most important tasks in understanding the immediate environment. The availability of low-cost 3D sensors has increased the interest in 3D point cloud data utilization for understanding the surroundings. For instance, 3D shape classification is used in autonomous navigation [1][2], civil engineering [3], and robotics [4]. The state-of-the-art deep learning based 3D point cloud classification algorithms can be divided into three categories: Methods that use voxel grids of 3D objects [1], [5]–[7], methods that capture multiple views of the shape and use 2D CNNs [8], [9], [18], [19], [10]–[17], and methods directly working on point clouds [20]–[22].

Current challenges in using 3D scanners include the inaccuracies due to following input corruptions: noise, missing points, and outliers. These inaccuracies arise due to several conditions such as sensor type, or interference from the surrounding environment. Data inaccuracy can have significant impact on the performance of a method. For example, Figure 1 compares the performance of two methods with clean and perturbed data. The figure shows that the tested methods can correctly classify the point

clouds when data is clean. However, when noise or few outliers are added to the data, the performances deteriorates significantly.

A trivial solution to the above problem would be to use filters to remove these inaccuracies before classification. However, cleaning data is not easy, and input data filtering can remove features that are important for classification. In fact, this is a chicken-and-egg problem and we could clean the data properly if we knew the underlying object.

		A									
		Airplane		Car		Desk		TV stand		Bottle	
MVCNN		.99	.00	.99	0.0	.99	.00	.99	.00	.99	.00
Point Net ++		.99	.00	.99	.00	.99	.00	.99	.00	.99	.00
		B									
		Airplane		Car		Desk		TV stand		Bottle	
MVCNN		.00	.8	.00	.99	0.2	.77	.00	.66	.00	.00
Point Net ++		.99	.00	.00	.72	.00	.97	.00	.99	.00	.99

Figure 1: Classification success rate of MVCNN [23] & PointNet++[22]. (A) Clean data, (B) with inaccuracies. Data in different columns have additive noise, gross outliers, missing points, and pseudo outliers (part of the ground plane). The probability values are truncated to two decimal places for convenience.

Although a few 3D classification methods have been tested with unclear data, to the best of our knowledge, there is no comparative analysis of the robustness of point cloud classification algorithms. This work presents a systematic study of the performance of the state-of-the-art point cloud classification algorithms on data with above mentioned inaccuracies. Five competitively performing networks of three different approach categories (described earlier), covering majority of solutions, were compared. ModelNet40 dataset, introduced in PointNet [20], was elected and used to

compare the performance of chosen methods for different levels and types of inaccuracies.

## 2. BACKGROUND

Early machine learning algorithms for 3D shape classification such as Markov Networks [24] [25], support vector machine (SVM) [26], or cost function minimization [27] used frameworks with hand-tuned features. Several types of descriptors [28] were used to capture shape features along these classifiers. In contrast, deep learning networks that use machine learned features have delivered impressive performances in terms of classification accuracy and the number of recognized classes.

The use of deep learning networks for 3D object classification started when 2D image networks were well-developed and as such, already successful 2D techniques were chosen for solving 3D classification problem. Convolution neural networks (CNN) were one of those competitive 2D networks that were used for 3D classification. As it is not possible to feed 3D data directly to CNNs, different approaches were proposed such as the use of voxel geometries [1]. We briefly outline the main approaches here.

### 2.1 Volumetric CNN:

In Volumetric CNN based methods, the irregular format of the 3D point clouds is transformed into regular grid, making it possible to use volumetric or 3D CNNs. Wu et al. [5] used Volumetric CNN to classify and reconstruct objects from a single-view 2.5D depth map. They achieved 77% accuracy on ModelNet10 using a grid size of 30x30x30. Maturana and Sebastian [1] proposed VoxNet, a real time 3D CNN for object detection from RGB-D data. Zhou and Tuzel [6] proposed VoxelNet network for object detection on variable point density data such as those coming from LiDAR. The network directly receives raw point cloud data and partitions the space into voxels.

The drawback of using voxel grids is its computational cost and memory requirement that grow rapidly with the resolution. However the use of octree [29][7] helped generate grids with finer resolution.

### 2.2 Multi view CNN:

The other approach is to use 2D CNN on the 3D mesh/CAD objects by rendering them into 2D images [23]. These methods are also called Multi-view CNN because they use more than one view to capture 2D images of the object. Su et al. [23] develop a similar network for 3D shape recognition and reconstruction from 12 views. Each view is fed into a separate CNN layer. A Pooling layer compresses information from these CNN layers. In another work, instead of feeding all images to the network directly, a set of view pairs images was used [10]. A GPU accelerated MVCNN that processes shapes efficiently for classification and

retrieval in real-time was also proposed in [11]. Using panoramic view of shapes for classification and retrieval has also been proposed in conjunction with 2D CNNs [9][18].

The drawbacks of multi-view CNNs include the need to render the object into 2D images, and the use of CAD/Mesh shapes. Therefore, two pre-processing steps are required to use point cloud objects with this approach, and both processes are fairly time consuming.

### 2.3 Direct methods:

Direct methods use point cloud directly as an input to the network. An example of this approach is PointNet [20]. PointNet directly works on unsorted data where the input data is transformed using several perceptron layers. The outputs of the above layers are then aggregated using an order-less function (max pooling layer). PointNet++ [22] is an improved version of PointNet that learns local structures at different scales using a hierarchical neural network. They initially group points into sets and produce new sets from these groups with fewer elements by farthest point sampling. Following that, the number of points in the neighbourhood of centroid points within a distance K are found. Then all these information is fed into PointNet for capturing point-to-point relations in the local region.

Other networks which also implemented PointNet include SO-Net [30] which uses self-organized-maps (SOM) to gather spatial distribution of the 3D input data, followed by point-to-node k-nearest neighbour search.

Klokov and Lempitsky [21] proposed kd-trees neural network. The network uses the structure of the kd-tree to get a shape descriptor of objects. The neural network architecture computes the k-dimensional vectors describing the individual points in leaf nodes and non-leaf nodes.

Methods that work directly on point cloud require no additional pre-processing compared to previous two categories. Thus they are much faster and avoid any issues that might arise due to meshing or voxel generation procedures.

## 3. METHODOLOGY

Our aim is to analyse the robustness of state-of-the-art neural network based 3D point cloud classification techniques to noise, outliers, and missing points. To test each scenario, we created a new test set by adding the specified inaccuracy (with its relevant parameters) to the original Modelnet40 test data. The accuracy of all the chosen methods were then evaluated on the above datasets, and the classification accuracies are reported in the following section. The training data was not changed and the models were trained using the clean modelnet40 dataset. Several papers have mentioned that training the model under a particular inaccuracy type helps improve the performance of the model under the said inaccuracy type. However, we

believe that the training a model under specific inaccuracy is not ideal as by definition most inaccuracies are unforeseen (i.e. outliers), and being robust to unforeseen inaccuracies is needed in practical situations.

We selected five high performing methods from the networks reported in section 2, namely: MVCNN [23] for Multiview CNNs, Oct-Net [7] for volumetric CNNs, PointNet [20], PointNet++[22] for direct methods, as well as Kd-Net [35]. The selected methods cover most of the networks structures available in the literature, and provide classification accuracies as good as others in the same category.

### 3.1 Dataset

To simulate the sensor output, we will use ModelNet40 dataset, where every object consisted of 2048 point. The data set is designed to be between [-1 1] and it is used as is without any further normalization. The third and fourth categories of networks work directly on point cloud and require no further processing. We generated mesh from the point cloud for the first and second categories using a crust algorithm [1] that produces a set of triangles. Since the 3D shapes were generated by meshing the point cloud, it is expected that the shapes to be not perfect compared to the original CAD models of ModelNet40 dataset. Therefore, the accuracy of the used networks might be slightly less than what authors reported in their papers.

To test for the effect of noise, five test sets were created by adding Gaussian noise with zero mean and standard deviations 0.02, 0.04, 0.06, 0.08, and 0.10. To test for effect of outliers, five additional test sets were created by replacing 2-70% of data points in each model with points uniformly sampled from the cube between [-1, 1] in each dimension. To test for the effect of pseudo outliers (outliers with structures), a set of points forming a planar shape (simulating the capture of the ground plane underneath of an object by a scanner) were added to existing models as shown in Figure 1b-4. The number of points on those planes represent 10% of the total number of points. To test for the effect of missing points, random data reduction was applied to all shapes with percentages from 10 to 90.

## 4. RESULTS

### 4.1 Noise

Figure 2 shows the classification accuracy on the data perturbed with Gaussian noise. Overall, PointNet showed the best performance, followed by Oct-Net and Kd-Net. PointNet and PointNet++ showed only 2% drop in the accuracy at 0.02 noise level, but as the noise level increased to around 0.05, the performance of PointNet++ deteriorated significantly. At 0.02 noise level, Oct-Net and Kd-Net showed 6% drop in accuracy compared initial accuracy, while MVCNN's accuracy dropped by 20%.

### 4.2 Missing points

Figure 3 shows the behaviour of the networks when the percentage of points on an object is reduced. All networks showed almost no performance degradation until approximately 50% reduction, which is about 1000 outlier data points per instance. Above 50%, all methods showed the same behaviour except for PointNet++. Since we used mesh for both Oct-Net and MVCNN, mesh regions are expected to remain unchanged when points inside these regions are removed. However, when a large proportion of points are removed, the discriminative features of the model will vanish, and classification becomes difficult.

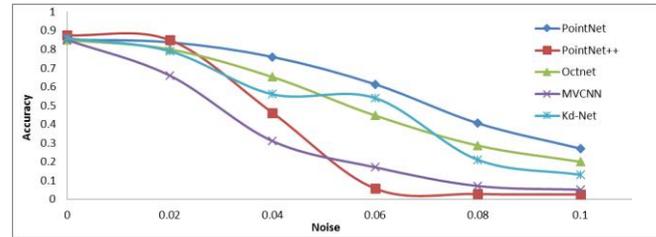


Figure 2: Variation of classification accuracy with different additive Gaussian noise.

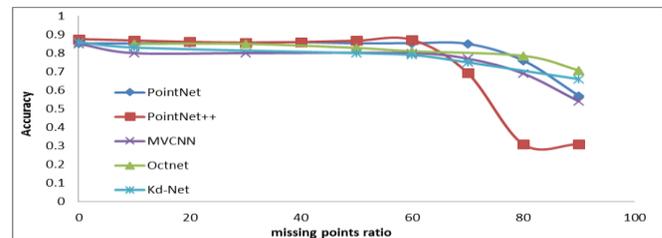


Figure 3: Variation of classification accuracy with different missing point ratios.

### 4.3 Outliers

Two types of outliers have been considered, the first type is the randomly distributed points as shown in the second object (car) in Figure 1b. Depiction of objects showed that at low outliers ratio, the mesh algorithm was able to eliminate most outliers and accurately represent the shape. However at higher outlier levels, the mesh algorithms produces shapes that are no longer close to the original model. Figure 4 shows that at 2% outliers ratio, Kd-Net showed only 2% drop in the performance. The accuracies of PointNet++, MVCNN, and Oct-Net dropped by 25%, while that of PointNet dropped by 70%. After 6% outliers, PointNet++ showed the best performance among all methods. PointNet++ is hierarchical learning network rather than point dependant, which could be the reason behind the improved performance compared to PointNet.

[1] <https://au.mathworks.com/matlabcentral/fileexchange/63731-surface-reconstruction-from-scattered-points-cloud-open-surfaces>

The second type of outliers are called pseudo outliers and are the results of introducing sets of points that form a shape by themselves. Here we generated a plane located near the object with random orientation, as shown in Figure 1b-4. This type of data appears when part of the object is reflective or scanner captures part of the ground underneath the object. Table 1 shows the performance of the networks with this type of outliers. The performance of both voxel network and MVCNN is still high, indicating that CNN filters still capture the object shape. However, in MVCNN, the accuracy dropped at a higher rate than voxel CNN, probably due to the fact that the plane could have blocked the object from being detected from certain views.

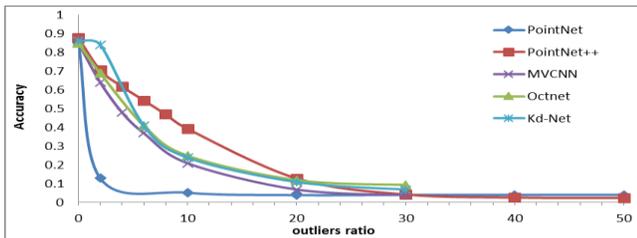


Figure 4: Variation of classification accuracy with different gross outliers ratios

Table 1: Classification accuracy for data corrupted with 10% pseudo outliers.

Method	MVCNN	Oct-Net	PointNet	PointNet ++	Kd-Net
Accuracy	74%	81%	3.7%	3.3%	75%

## 5. DISCUSSION

### 5.1 Using point clouds on mesh-based networks

We also investigated the performance of MVCNN and Oct-Net networks without the mesh generation stage. Here, the point clouds are directly used to generate voxels or 2D rendering [1] using Phong shading. The trained Oct-Net showed impressive performance in the presence of outliers, as seen in Figure 5. The network accuracy dropped to 70% at 2% outliers ratio, but the accuracy hardly changes after that. Training MVCNN on point clouds, however, did not resolve the problem of outliers and the accuracy dropped to 25% accuracy at only 2% outliers ratio, far worse than the mesh case. One reason could be that each rendered view in MVCNN accumulate all the outliers, so the rendered images have more levels of outliers than before. We then trained the network on top view only. This showed slightly better performance than using all 12 views, however the accuracy is still worse than the mesh case. For pseudo outliers, the accuracy increased from 75% to 86% for MVCNN, and this could be because the shape can be seen through the pseudo outliers. Also, the accuracy increased from 81% to 86% for

Oct-Net, meaning that the accuracy was not affected much by this type of outliers.

Figure 6 shows the performance of the MVCNN and Oct-Net networks trained on point clouds or mesh in the presence of noise. It can be seen that using point clouds leads to better performance. Also, Oct-Net with point cloud input exceedingly performed better than all other networks. On the other hand, Figure 7 shows that the two networks using point cloud perform worse than their mesh based counterparts when the data is corrupted by missing points.

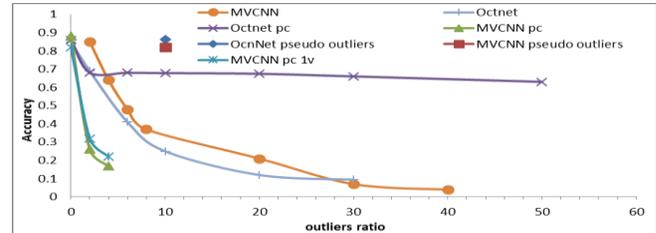


Figure 5: Variation of classification accuracy versus outliers for networks trained on point clouds. pc means network use point cloud. 1v means network trained on one view.

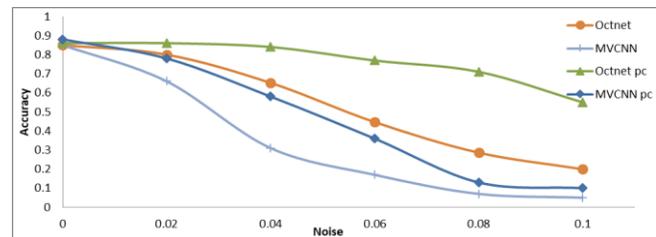


Figure 6: Variation of classification accuracy versus noise for networks trained on point cloud. pc means network use point cloud.

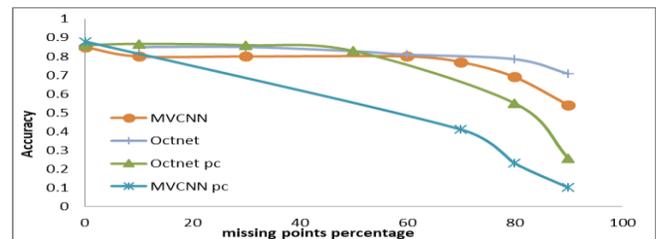


Figure 7: Variation of classification accuracy versus missing points for networks trained on point clouds. pc means network use point cloud.

## 6. CONCLUSION

We tested recent network algorithms for 3D shape with the presence of the data inaccuracies. Three types of data inaccuracies are assumed to appear with 3D scanners, noise, missing points, and outliers. The results showed that all the tested networks can handle significant amount of missing points and the performance degrades gradually with additive Gaussian noise. However, no method was robust to outliers and showed steep deterioration in performance when outliers were introduced. The results also show that Oct-Net without the mesh generation step is fairly robust to outliers.

[1] [https://github.com/jongchysisu/mvcnn\\_pytorch](https://github.com/jongchysisu/mvcnn_pytorch)

## 7. REFERENCES

- [1] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition," *IEEE Intell. Robot. Syst.*, pp. 922–928, 2015.
- [2] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 163–168.
- [3] E. Özdemir and F. Remondino, "SEGMENTATION OF 3D PHOTOGRAMMETRIC POINT CLOUD FOR 3D BUILDING MODELING," 2018.
- [4] U. Weiss and P. Biber, "Plant detection and mapping for agricultural robots using a 3D LIDAR sensor," *Rob. Auton. Syst.*, vol. 59, no. 5, pp. 265–273, May 2011.
- [5] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07–12–June, pp. 1912–1920, 2015.
- [6] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," pp. 4490–4499, 2018.
- [7] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning Deep 3D Representations at High Resolutions," Nov. 2016.
- [8] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 945–953, 2015.
- [9] B. Shi, S. Bai, Z. Zhou, and X. Bai, "DeepPano: Deep Panoramic Representation for 3-D Shape Recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2339–2343, 2015.
- [10] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise Decomposition of Image Sequences for Active Multi-View Recognition."
- [11] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki, "GIFT: A Real-time and Scalable 3D Shape Search Engine."
- [12] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and Multi-View CNNs for Object Classification on 3D Data."
- [13] K. Sfikas, T. Theoharis, and I. Pratikakis, "Exploiting the PANORAMA Representation for Convolutional Neural Network Classification and Retrieval," *Eurographics Work. 3D Object Retr.*, no. May, 2017.
- [14] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant Set Clustering and Pooling for Multi-View 3D Object Recognition."
- [15] P. Zanuttigh and L. Minto, "Deep learning for 3D shape classification from multiple depth maps," in *Proceedings - International Conference on Image Processing, ICIP*, 2018, vol. 2017–Sept, pp. 3615–3619.
- [16] F. Gomez-Donoso, A. Garcia-Garcia, J. Garcia-Rodriguez, S. Orts-Escolano, and M. Cazorla, "LonchaNet: A sliced-based CNN architecture for real-time 3D object recognition," in *Proceedings of the International Joint Conference on Neural Networks*, 2017, vol. 2017–May, pp. 412–418.
- [17] A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum, "Synthesizing 3D Shapes via Modeling Multi-View Depth Maps and Silhouettes with Deep Generative Networks."
- [18] K. Sfikas, I. Pratikakis, and T. Theoharis, "Ensemble of PANORAMA-based convolutional neural networks for 3D model classification and retrieval," *Comput. Graph.*, vol. 71, pp. 208–218, Apr. 2018.
- [19] A. Kanezaki, Y. Matsushita, and Y. Nishida, "RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints," 2016.
- [20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," Dec. 2016.
- [21] R. Klokov and V. Lempitsky, "Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017–Octob, pp. 863–872, 2017.
- [22] C. R. Q. Li, Y. Hao, S. Leonidas, and J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space."
- [23] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition."
- [24] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," *2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009*, vol. 2009 IEEE, pp. 975–982, 2009.
- [25] J. Behley, V. Steinhage, and A. B. Cremers, "Performance of histogram descriptors for the classification of 3D laser range data in urban environments," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 4391–4398, 2012.
- [26] M. Himmelsbach, T. Luettel, and H. J. Wuensche, "Real-time object classification in 3D point clouds using point feature histograms," *2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009*, pp. 994–1000, 2009.
- [27] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing Objects in Range Data Using Regional Point Descriptors," pp. 224–237, 2004.
- [28] X. Hana, J. Jin, J. Xie, M. Wang, W. J. preprint arXiv, and undefined 2018, "A comprehensive review of 3D point cloud descriptors," *arxiv.org*.
- [29] D. Meagher, "Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer," *Rensselaer Polytech. Inst.*, no. Technical Report IPL-TR-80-111, 1980.
- [30] J. Li, B. M. C. Gim, and H. Lee, "SO-Net: Self-Organizing Network for Point Cloud Analysis."