Final Report: EE199

Akash Gokul
University of California, Berkeley
Berkeley, CA 94709
akashgokul@berkeley.edu

Oladapo Afolabi
University of California, Berkeley
Berkeley, CA 94709
oafolabi@berkeley.edu

Abstract

Recent advances in neural network based generative models have paved the way for robust 3D reconstruction techniques based on shape models from CAD databases. However, one shortcoming of such approaches is that they produce incomplete results when under-constrained. Such scenarios occur with significant occlusions in indoor scenes. One approach to ameliorate this situation is to make use of repeated objects in the scene to provide more data and constraints. In this work, we investigate the first steps to this approach, namely, detecting repeated objects in a scene. We report our experiments and findings and present suggestions for even further improvements.

1 Introduction

The increase in both the demand and supply of commodity 3D sensors such as the Microsoft Kinect, has made it relatively easy to obtain scans of indoor scenes. Such scans find applications in areas such as robotics, gaming, virtual and augmented reality and real estate. In many of these applications, scans are converted to 3D models that can be interacted with.

One pipeline for this process involves the use of Simultaneous Localization and Mapping algorithms to estimate sensor pose and the 3D location of sparse keypoints in the scene. This step is followed by a volumetric fusion of depth data from the 3D sensor using algorithms such as [1]. A final step to extract a surface triangle mesh is carried out using variants of the marching cubes algorithm [2].

While this approach is very general and will work in most use cases, it can be superfluous in its representation and has no reliable way of dealing with missing data. One way to improve on this approach is to make use of Computer Aided Design (CAD) models to represent some of the objects in the scene. This approach provides a way to deal with occlusions and is compact since each model may be represented using an index in a database. This approach was shown to be viable in [3]

Unfortunately, this approach may sacrifice the accuracy of reconstructed models for completion and compactness. This occurs since, CAD models may not exactly match the data that is captured.

Neural network based generative models may help solve this issue. Approaches such as [4, 5, 6, 7] present neural network generative models for 3D shape. In particular [4], present an any-resolution approach to generating 3D models using signed distance functions. With this approach it is also possible to perform inference on dense and partial observations of an object, once its class is known. During inference, the closest matching 3D model from the space of shapes is found as well as a compact latent vector representation for the shape. While this approach helps with shape completion it may fail in cases of severe occlusions, producing unrecognizable and distorted shapes.

However, we observe that for indoor scenes, objects are typically found in pairs or tuples. One may make use of this fact to aid inference in cases of severe occlusion by combining information from repeated instances of the same object. A crucial task for this paradigm is to be able to detect when multiple instances of an object are present. In this work, we investigate approaches to detecting multiple instances of an object in a scene. We focus on chairs due to their ubiquitous presence in

indoor scenes. We formulate the problem as a classification task, where we map each 3D scan of a chair to a CAD model. Multiple instances of a chair can then be inferred if multiple 3D scans map to the same CAD model.

My contribution to this work includes:

- Data collection of both 3D scans and CAD models from ShapeNet and ScanNet datasets [8, 9]
- Data preprocessing to extract clean and meaningful information as well as labels from the Scan2CAD dataset [3].
- Problem formulation and model exploration
- Implementation, fine-tuning and evaluation of models

2 Related Work

This section reviews the tools and datasets used for our pipeline as well as those from preceding works.

2.1 Detecting Repeated Objects

In our line of work, we are interested in determining whether or not two objects are the same type of object. While much of the work in detection of symmetry and object detection through point cloud data revolve around the symmetries of each object individually [10][11][12], our efforts deal with the overall scene. Moreover, work such as Similarity Group Proposal Network (SGPN) [13] similar to the aforementioned works [10][11] [12] groups objects on a point-wise basis. As a result, as seen in [13], it fails to accurately detect repeated objects and instead aims to optimize object segmentation in scenes through grouping of objects and their components (e.g. a chair and each of its legs).

Additionally, there is extensive work in the direction of detecting similarity between point clouds such as [14][15] [16]. [14] and [15] rely on traditional methods which revolve around generating a signature or computing the spin image to determine alike point clouds and points. Similarly, [16] determines the extent to which pointclouds are alike through computing the similarity between normals, curvature, etc. Through the course of this work, we compare our learning based approach to such handcrafted techniques.

2.2 PointNet

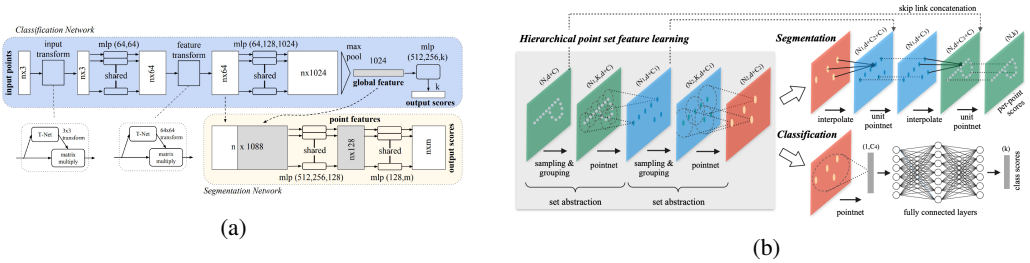
We began with trying simpler end to end methods, and as a result we used PointNet [17]. PointNet is a deep learning model which takes in pointcloud data as input and can be used for tasks such as classification [17]. Due to its popularity and ease of use, we began our work by using PointNet, and later on PointNet++ [18], as our primary classifier and spent our time augmenting and improving it for our task.

2.2.1 Datasets

In order to get objects from 3D scenes for the purposes of training and testing our classifier we used the ScanNet dataset [9] and the Scan2CAD dataset [3] to gather identification labels for said objects. These labels could be used to classify on a per-object basis or to determine similarity between objects by grouping based on same identifier. Additionally, our object point clouds came through the cropping of meshes in Scan2CAD.

3 Problem Statement and Approach

Our work aims to improve DeepSDF's [4] inference and reconstruction abilities in noisy environments, where data may be incomplete, through the use of a classifier which determines repeated objects. Such repeated objects could then be used to augment the data and provide multiple views of said object to enhance inference and reconstruction.



(a) (b)

Figure 1: (a) The architecture of the PointNet model as provided by [17]. (b) The architecture of Pointnet++ [18] In both models we modified the number of classes, k, to be 316, the number of unique CAD ObjectIDs corresponding to the chairs in our dataset.

Concretely, given a set of 3D meshes of objects, we would like to develop an algorithm to predict which of these meshes correspond to multiple instances of the same shape.

Our approach is to formulate this problem as a classification problem. If we can correctly classify the class of shape each mesh falls into, then all instances of a particular shape should share the same label. Consequently, we may use the class labels to identify multiple instances of the same object. We believe this is a reasonable approach, given our limited time, since it allows us take advantage of mature 3D classification techniques. Furthermore, we restrict our focus to 3D meshes of indoor chairs since this would allow us focus on our approach and minimize the amount of time spent preprocessing other types of shapes.

We have made used the PointNet and PointNet++ as 3D shape classifiers. We make use of the implementation provided by NVIDIA's Kaolin library [19].

4 Data Pre-processing and Implementation details

Most of our efforts this semester, were in designing the tools to use and train PointNet to map each chair's point cloud to the chair's CAD ObjectID as given by [3], as depicted in Figure 1. Code for our work can be found at the following fork of Kaolin, `https://github.com/akashgokul/kaolin`.

As a preprocessing step, we sample ScanNet scenes for scenes with chairs and then crop the areas corresponding to the chairs. This provides us with relatively clean 3D mesh data that we can use as input to our algorithm. In addition to provide class labels for each cropped scan, we have made use of the Scan2CAD dataset [20] which provides approximate correspondences for each cropped object to a cad model in ShapeNet [8]. We make use of this class-ID as labels for each cropped chair scan. In total we were able to identify 316 unique classes.

We also normalize each chair by enforcing that it is centered and lies within the unit sphere. We also ensure that all chairs are roughly facing the same direction. To enable better generalization, we augment our dataset by adding small random noise to points on each shape. This was suggested in [18] and helps improve generalization to unseen objects.

5 Experiments

To evaluate the effectiveness of our approach, we carry out two main experiments on our models.

5.1 Classification

We first test the ability of our models to classify 3D chairs as belonging to the set of ShapeNet [8] model classes used in training. We report the results of our work using both PointNet and PointNet++ in Table 1.

	Pointnet	Pointnet ++
Accuracy	0.194	0.454

Table 1: Classification accuracy of our models.

5.2 Multiple Instance Detection

The main goal of our work is to be able to detect multiple instances of an object in a scene. Note that this is not equivalent to classification. Consequently, we carry out an experiment to measure the ability of our methods to predict is 3D shapes in a scene correspond to the same object. The accuracy of a classifier may be low, but if it consistently predicts the same class for similar shapes, then it will still be adequate for our task. To provide a quantitative measure of our results we make use of precision and recall metrics. Where for each scene, a true positive occurs if our model predicts that two 3D objects are multiple instances of the same shape and our labels also imply that this is the case. We compute average precision and recall across all test scenes and report the results in Table 2. We compare our results using PointNet++ to the baseline of using OUR-CVFH [21] as a classifier.

OUR-CVFH [21] features are handcrafted features meant to provide a robust global descriptor of a 3D shape. We have computed features for both ShapeNet models and test chair scans. For each chair scan we search all the ShapeNet chair models provided as a class for our classifier and select the one with closest descriptor match. We consider the result to be correct if the match corresponds to the label provided for the chair scan.

	Precision	Recall
PointNet++	0.951	0.388
OUR-CVFH	0.927	0.257

Table 2: Precision and recall across test scenes.

Finally, we also provide some qualitative results of our experiments in Figs. 2, 4,and 5. We have placed side by side, a 3D scan of a scene an our prediction of multiple instances of an object using PointNet++. Objects predicted to be multiple instances of the same shape share the same color.

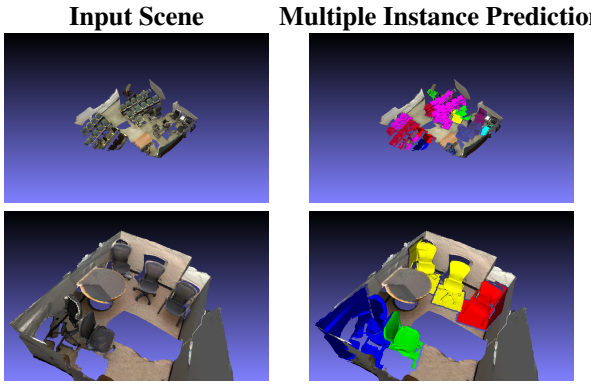
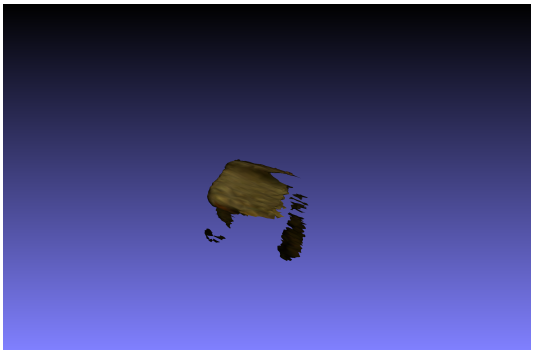


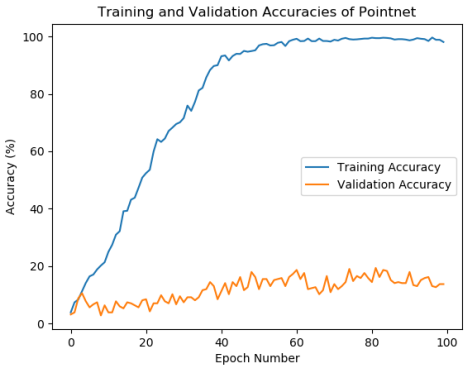
Figure 2: Qualitative results on ScanNet test set. Left: Input scene, Right: Color labelled Output, objects of same shape share same colors

6 Results

Our classifier shows marked improvement in detecting repeated objects in comparison to a a traditional method which uses handcrafted features such as OUR-CVFH. From Table 2 and the figures below, one can see that the high precision indicates that the classifier is correctly able to identify multiple instances of the same object. On the other hand, the recall of 0.388 indicates that the model favors the implicit grouping of objects as alike even when this is not the case.



(a)



(b)

Figure 3: A visualization and depiction of the sources of errors and faults of our model. The inputs are often: (a) structure less amalgamation of points, lacking much resemblance to a chair. (b) Our Pointnet model overfits in regards to classifying the CAD ObjectID of the chairs in the dataset.

6.1 Analyzing Potential Errors

While our classifier is able to perform well with respect to our greater task, it was unable to classify the CAD ObjectIDs of the chairs correctly as depicted in Figure 5b.

We attribute the cause of such lackluster performance to be the data we were using. Upon further analysis, many of the chairs in the scenes from the ScanNet are incomplete as depicted in Figure 5a. As a result, these chairs bare little to no resemblance to the intended structure nor do they resemble other complete chairs corresponding to the same CAD ObjectID. Furthermore, within the Scan2CAD dataset, there are many instances of chairs, which in reality are alike, but are mapped to different CAD ObjectIDs. This in turn may have added further confusion to the learning purpose and served as a hindrance in our model's performance in our overarching task.

7 Conclusion and Future Work

We have presented a model that shows marked improvement in detecting repeated objects in a 3D scene in comparison to traditional methods which rely upon handcrafted features. Through the use of Pointnet [17], Pointnet++ [18], the ScanNet dataset [9] and the Scan2CAD dataset [3], we were able to construct a classifier that originally mapped point clouds of chairs to the CAD ObjectIDs. Following, this classifer would be applied to each chair in a scene and we would group repeated chairs based on alike output CAD ObjectIDs.

Despite the significant improvement with respect to our greater goal, our classifier failed to correctly classify CAD ObjectIDs correctly. As a result, we believe that a better approach would be to use a siamese network like approach to determine whether two chairs, in the form of point clouds, are alike. Such a model could then be applied to each pair of chairs in a given scene. We find such a method to be promising due to the nature of our main problem revolving around detecting a binary output. As a result, given the aforementioned errors in the given datasets, we believe that such a model could perform better at our task given the simplicity of the reformulation.

8 Acknowledgements

We would also like to thank our partner on this project, Arbaaz Muslim.

9 Additional Figures

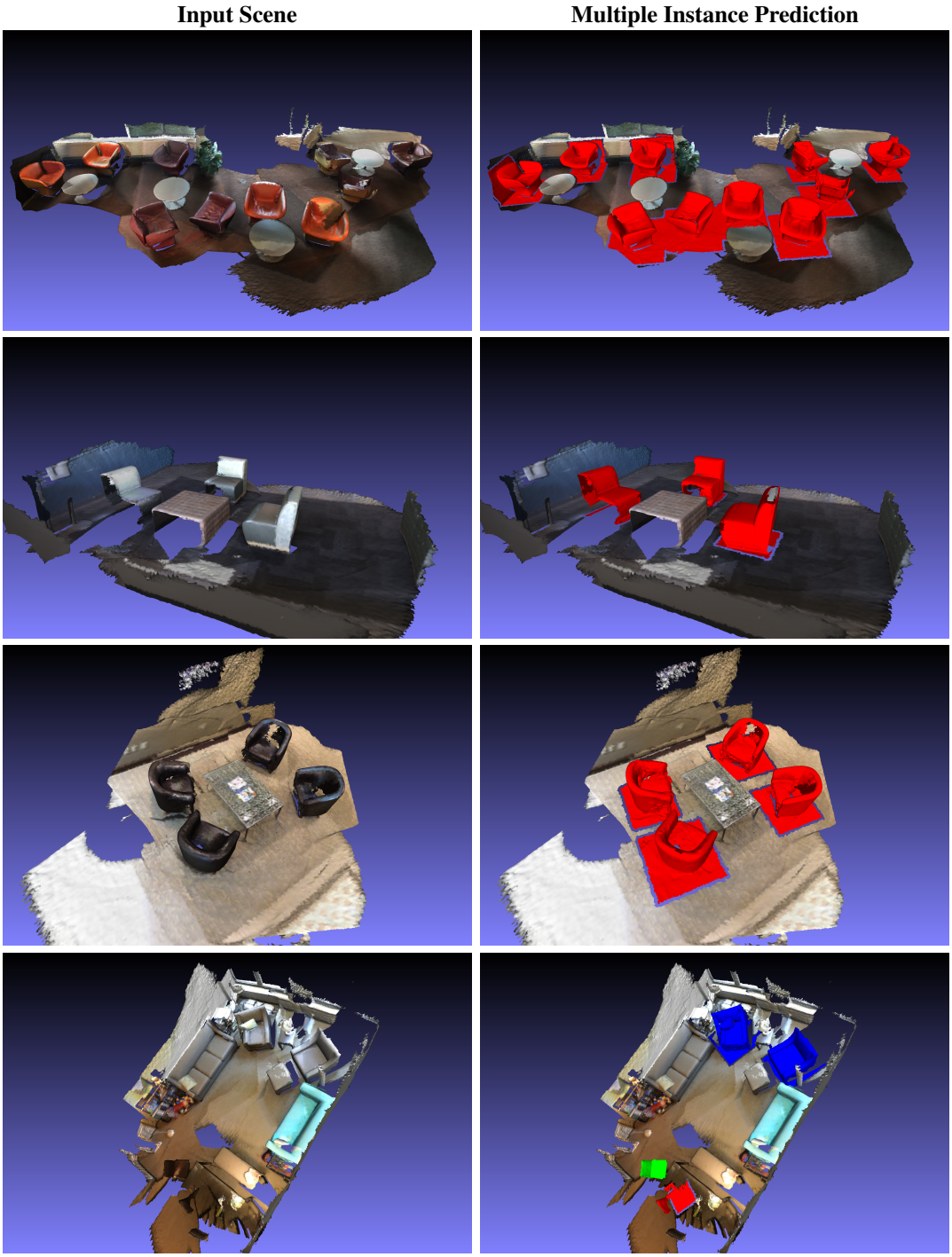


Figure 4: Qualitative results on ScanNet test set. Left: Input scene, Right: Color labelled Output, objects of same shape share same colors

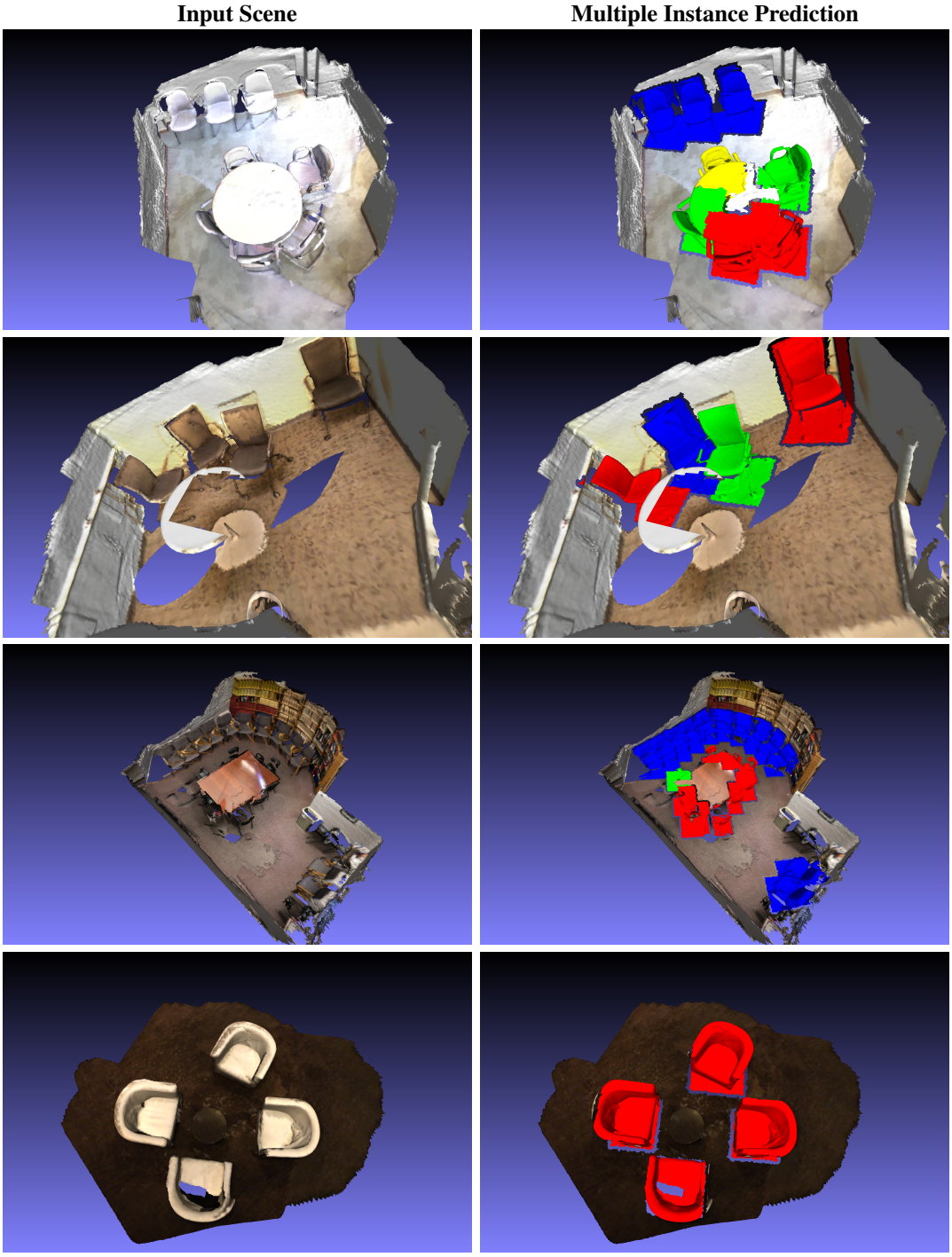


Figure 5: Qualitative results on ScanNet test set. Left: Input scene, Right: Color labelled Output, objects of same shape share same colors

References

[1] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[2] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[3] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2609–2618, 2019.

[4] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017.

[6] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[7] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 490–500, 2019.

[8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[9] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017.

[10] Andrej Karpathy, Stephen Miller, and Li Fei-Fei. Object discovery in 3d scenes via shape analysis. In *2013 IEEE International Conference on Robotics and Automation*, pages 2088–2095. IEEE, 2013.

[11] S. Thrun and B. Wegbreit. Shape from symmetry. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1824–1831 Vol. 2, 2005.

[12] Aleksandrs Ecins, Cornelia Fermüller, and Yiannis Aloimonos. Seeing behind the scene: Using symmetry to reason about objects in cluttered environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7193–7200. IEEE, 2018.

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

[14] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[15] Andrew Edie Johnson and Martial Hebert. Recognizing objects by matching oriented points. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 684–689. IEEE, 1997.

[16] J. Huang and S. You. Point cloud matching based on 3d self-similarity. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 41–48, 2012.

[17] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[18] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[19] Krishna Murthy J., Edward Smith, Jean-Francois Lafleche, Clement Fuji Tsang, Artem Rozantsev, Wenzheng Chen, Tommy Xiang, Rev Lebaredian, and Sanja Fidler. Kaolin: A pytorch library for accelerating 3d deep learning research. *arXiv:1911.05063*, 2019.

[20] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019.

[21] Aitor Aldoma, Federico Tombari, Radu Bogdan Rusu, and Markus Vincze. Our-cvfh–oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. In *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*, pages 113–122. Springer, 2012.