Poster: AutoPCD: Learning-Augmented Indoor Point Cloud Completion

Pingping Cai; Edward M Sitar; Sanjib Sur
Department of Computer Science and Engineering; University of South Carolina, Columbia, USA
{pcai, esitar}@email.sc.edu; sur@cse.sc.edu

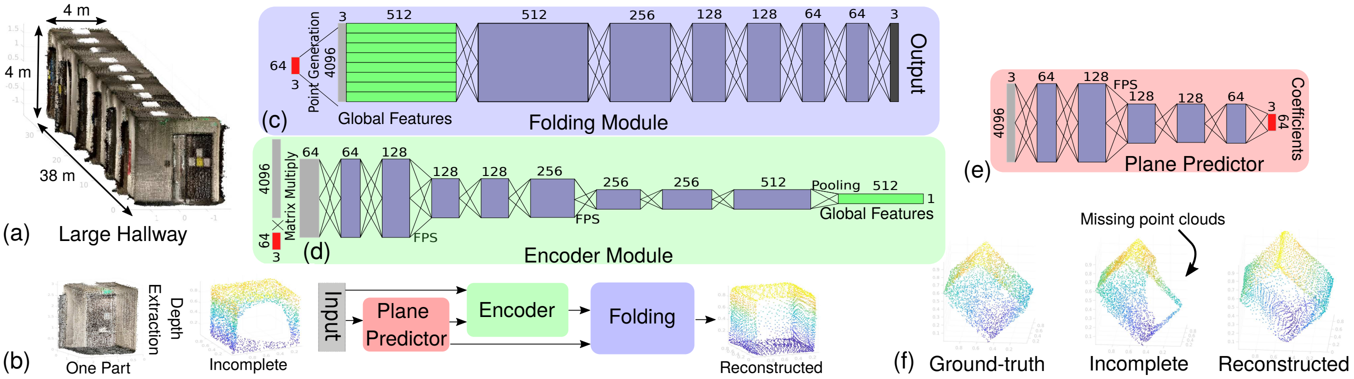


Figure 1: (a) Point Cloud (PCD) of a large indoor hallway; (b) The overall network structure of *AutoPCD*; (c–e) Folding, Encoder, and Plane Predictor modules; (f) An example result of PCD reconstruction.

ABSTRACT

3D Point Cloud (PCD) is an efficient machine representation for surrounding environments and has been used in many applications. But a fast reconstruction of complete PCD for large environments remains a challenge. We propose *AutoPCD*, a machine-learning model that reconstructs complete PCDs, under sensor occlusion and poor lighting conditions. *AutoPCD* splits the PCD into multiple parts, approximates them by several 3D planes, and independently learns the plane features for reconstruction. We have experimentally evaluated *AutoPCD* in a large indoor hallway environment.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**; • **Computing methodologies → Machine learning approaches**.

KEYWORDS

Point Cloud Data; Graph Convolution; Multi-Layer Perceptrons

ACM Reference Format:
Pingping Cai; Edward M Sitar; Sanjib Sur, Department of Computer Science and Engineering; University of South Carolina, Columbia, {pcai, esitar}@email.sc.edu; sur@cse.sc.edu. 2021. Poster: AutoPCD: Learning-Augmented Indoor Point Cloud Completion. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers (UbiComp-ISWC '21 Adjunct), September 21–26, 2021, Virtual, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3460418.3479309

1 INTRODUCTION

Understanding and interpreting the surrounding 3D environment is an important machine perception problem, and such perception enables many ubiquitous sensing applications in surface and underwater robotics, drones, autonomous driving, and augmented or extended reality (AR/XR). 3D Point Cloud (PCD) is one of the efficient machine representations of the environments. A PCD usually consists of depth and/or RGB information, represented by voxel intensity (see Figures 1[a–b]), and are used in many research and commercial applications: Mobile robot simultaneous localization and mapping, 3D object tracking for AR applications, Real-time mapping of floors and surfaces during construction, *etc.*

However, complete PCD reconstruction of an indoor environment faces three major challenges: (1) It requires a lot of time and effort from human/machine to scan a large area; (2) It requires precise planning of the scan trajectories; and (3) It requires powerful, long-range camera and/or LiDAR-based depth sensors. Even solving these major challenges may not ensure a complete reconstruction due to the sensor occlusion and poor lighting conditions. The collected PCDs could be sparse and incomplete, missing important geometric information of the environment. Thus, estimating the complete PCD from an incomplete one is of vital importance.

Researchers have proposed to estimate complete PCDs based on Convolutional Neural Networks (CNN) [1] or Multi-Layer Perceptrons (MLP) [2] or both [3]. Although these methods are effective, they are designed for the reconstruction of small 3D objects, such

UbiComp-ISWC '21 Adjunct, September 21–26, 2021, Virtual, USA

ACM ISBN 978-1-4503-8461-2/21/09.
https://doi.org/10.1145/3460418.3479309

as tables, cars, bikes, *etc.*, where PCDs comprise thousands of voxels only. Besides, they focus on the object's finer-grained, local features reconstruction. But PCD reconstruction of a large indoor environment requires an emphasis on both the local and global structures. What's more, indoor PCDs consist of millions of voxels; so, existing models prohibit reconstruction due to memory limits [4], and local geometric information could get buried under global features. Thus, training existing networks with millions of voxels as input not only would be cumbersome but also may often fail to converge.

In this paper, we propose *AutoPCD*, a learning-augmented PCD completion model that overcomes the challenges. *AutoPCD* uses two key intuitions: (1) Instead of considering the PCD as a collection of a random, unordered set of points, *AutoPCD* uses the observations that indoor buildings consist of simple geometric structures, such as straight walls, smooth floors, *etc.*; so, many points could be combined and approximated as 3D planes. (2) A large environment could be split into multiple parts, and each part could be predicted independently and later merged to reconstruct the full environment; this ensures that the reconstructed PCDs can preserve accurate local structures, and our model can converge during the training.

To this end, *AutoPCD* takes the following steps. *First*, *AutoPCD* trains a combination of graph convolution and MLP framework by showing several examples of partial and incomplete PCDs and their corresponding complete PCDs: We use high-end depth sensors to manually scan indoor environments for several minutes and randomly introduce incompleteness by post-processing them. *Next*, the disjoint set of 3D voxels are approximated as 3D planes and represented by their coefficients; the framework learns such coefficients to fill the gaps in the incomplete PCDs. *Finally*, during the run-time, when the model has been trained appropriately, *AutoPCD* can estimate complete PCDs from incomplete ones. We have experimentally evaluated the efficacy of *AutoPCD* in reconstructing partial, incomplete PCDs of a large indoor hallway environment.

2 *AUTOPCD* DESIGN

Figure 1(b) shows the network structure of *AutoPCD*. It consists of 3 modules: *Encoder* for feature extraction, *Plane Predictor* for plane coefficients prediction, and *Folding* for the final reconstruction.

Global Feature Extraction: Since a point in PCD may exist in any 3D location sparsely, traditional convolution may fail to identify the global features associated with this point. *AutoPCD*, thus, employs the Encoder module for feature extraction, leveraging a graph convolution network, built atop the existing PointNet++ [5]. But the global features are highly sensitive to the small rotation of the input PCDs. To make the feature extraction robust, *AutoPCD* transforms the original points from Cartesian coordinates into learned planes' coordinates: The Encoder module leverages the plane coefficients, predicted by the Plane Predictor, and multiply them with Cartesian point coordinates to achieve the transformation.

Plane Coefficients Prediction: *AutoPCD* then approximates the PCD into several 3D planes and learn the plane coefficients. The idea shares a similar spirit to a recently proposed work on sparse depth reconstruction [6]. However, different from the existing approach using compressed sensing and geometrical modeling, *AutoPCD* leverages machine-learning to learn the coefficients. In Cartesian coordinate space (x, y, z), a plane can be expressed as:

$$a*x+b*y+c*z=d; \quad a'*x+b'*y+c'*z=1; \quad w\cdot p=1 \quad (1)$$

where $\cdot$ is the element-wise multiplication, $w=[a', b', c']$ is the plane coefficients, $a'=a/d; b'=b/d; c'=c/d$, and $p=[x, y, z]$. Similar to the Encoder module, *AutoPCD* uses one PointNet++ block as the backbone to extract features and learn the coefficients.

Folding: Finally, to reconstruct the complete PCD from the abstract global features and plane coefficients, *AutoPCD* uses a similar technique described in FoldingNet [7]. It shows that we can reconstruct 3D shapes from 2D grids by two steps of folding: The first folding step transforms the 2D grids into points in 3D space, and the second folding step transforms 3D points into target point clouds. Different from the existing method, we can jump to the second step directly and start folding from points in 3D space; this is because, given the set of predicted plane coefficients, we can generate points on 3D space directly. These points are then merged with the global features and fed into an MLP network to reconstruct the final PCD.

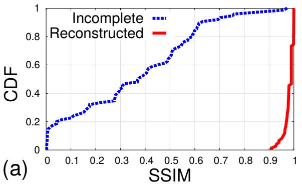
Loss Function: The network blocks rely on a loss function to appropriately tune the weights and train themselves. We use Chamfer Distance (ChD) [8] between the ground-truth and predicted PCDs as the loss function, which is defined as:

$$L_{ChD}(S_1, S_2)=\sum_{x\in S_1}\min_{y\in S_2}||x-y||_2^2+\sum_{y\in S_2}\min_{x\in S_1}||x-y||_2^2 \quad (2)$$

where S_1 and S_2 are the point sets. *AutoPCD* also uses a loss function for training the Plane Predictor to learn the plane coefficients w. Given M planes, *AutoPCD* uses the mean square error as the plane loss: $L_P(w, v)=\frac{1}{M}\sum_i ||w_i-v_i||_2^2$, where v is the ground-truth coefficients and is calculated from the point normal. We also manually set $M=64$. To train all network components simultaneously, we set the overall loss function for *AutoPCD* as: $L=\lambda_1*L_{ChD}+\lambda_2*L_P$, where λ_1 and λ_2 are the hyper-parameters that balances the ChD and plane loss. We explore the effect of different hyper-parameters combination and found that the network performed much better when the ratio between λ_1 and λ_2 is close to 2, *e.g.*, (λ_1, λ_2)=(1, 0.5).

3 PRELIMINARY RESULTS

To evaluate *AutoPCD*, we test our network in an indoor hallway (Figure 1[a]). We first split the full PCD into smaller pieces, rotate in azimuth and elevation, and generate the training and testing set. Specifically, we use 375 training samples and 75 testing samples. Furthermore, to generate the incomplete PCDs, we randomly remove about 15–30% points in chunks from the pieces. Figure 1(f) shows an example PCD depth reconstruction result: Clearly, *AutoPCD* is able to fully reconstruct the missing parts of the input. For statistical evaluation, we calculate the average Chamfer Distance and Structural Similarity (SSIM) of the reconstructed and incomplete PCDs *w.r.t.* the ground-truth across 75 test cases. Figure 2(a) shows the CDF of SSIM results for both the incomplete and reconstructed PCD, and *AutoPCD* improves the median SSIM from 0.36 to 0.98. Furthermore, Figure 2(b) shows that the average ChD is 0.003, which is approximately 4× better than the incomplete ones.



(b)

Chamfer Distance	smaller is better
Reconstructed	**0.0030**
Incomplete	0.0121
SSIM	larger is better
Reconstructed	**0.9823**
Incomplete	0.3617

Figure 2: (a) CDF of SSIM for incomplete and reconstructed PCDs; (b) Average values of Chamfer Distance and SSIM.

4 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we propose and evaluate *AutoPCD* that combines machine-learning and geometrical modeling for the fast reconstruction of large indoor PCDs. *AutoPCD* uses graph convolution and MLP based networks to extract the plane information and significantly improves the quality of the reconstructed PCDs. In the future, we propose to explore two avenues. (1) We currently split the environment into smaller pieces manually and process them individually. We propose to design a framework to automatically split, reconstruct, and merge them to ensure our model converges and preserves the local structures. (2) Our model currently reconstructs only the depth and geometric information in the PCD. However, color or RGB information is another important feature for 3D applications. So, we will improve our network to reconstruct not only the geometric structures but also the color characteristics of PCDs.

5 ACKNOWLEDGEMENT

We sincerely thank the reviewers for their comments. This work is partially supported by the NSF under grant CNS-1910853.

REFERENCES

[1] Dai, A. and Ruizhongtai Q., C. and Niessner, M., "Shape Completion Using 3D-Encoder-Predictor CNNs and Shape Synthesis," in *IEEE/CVF CVPR*, 2017.

[2] W. Yuan and T. Khot and D. Held and C. Mertz and M. Hebert, "PCN: Point Completion Network," in *International Conference on 3D Vision (3DV)*, 2018.

[3] Xie, Haozhe and Yao, Hongxun and Zhou, Shangchen and Mao, Jiageng and Zhang, Shengping and Sun, Wenxiu, "GRNet: Gridding Residual Network for Dense Point Cloud Completion," in *Computer Vision – ECCV*, 2020.

[4] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," in *Springer MICCAI*, 2016.

[5] C. R. Qi and L. Yi and H. Su and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *NIPS*, 2017.

[6] F. Ma and L. Carlone and U. Ayaz and S. Karaman, "Sparse Depth Sensing for Resource-Constrained Robots," *IJRR*, vol. 38, no. 8, 2019.

[7] Yang, Y. and Feng, C. and Shen, Y. and Tian, D., "FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation," in *IEEE/CVF CVPR*, 2018.

[8] Fan, H. and Su, H. and Guibas, L., "A Point Set Generation Network for 3D Object Reconstruction from a Single Image," in *IEEE/CVF CVPR*, 2017.