Point Cloud Geometry Prediction Across Spatial Scale using Deep Learning

Anique Akhtar*, Wen Gao†, Xiang Zhang†, Li Li*, Zhu Li*, Shan Liu†
University of Missouri-Kansas City*, Tencent†
emails: aniqueakhtar@mail.umkc.edu, {wengao, xxiangzhang}@tencent.com, {lil1, lizhu}@umkc.edu, shanl@tencent.com

***Abstract*—A point cloud is a 3D data representation that is becoming increasingly popular. Due to the large size of a point cloud, the transmission of point cloud is not feasible without compression. However, the current point cloud lossy compression and processing techniques suffer from quantization loss which results in a coarser sub-sampled representation of point cloud. In this paper, we solve the problem of points lost during voxelization by performing geometry prediction across spatial scale using deep learning architecture. We perform an octree-type upsampling of point cloud geometry where each voxel point is divided into 8 sub-voxel points and their occupancy is predicted by our network. This way we obtain a denser representation of the point cloud while minimizing the losses with respect to the ground truth. We utilize sparse tensors with sparse convolutions by using Minkowski Engine with a UNet like network equipped with inception-residual network blocks. Our results show that our geometry prediction scheme can significantly improve the PSNR of a point cloud, therefore, making it an essential post-processing scheme for the compression-transmission pipeline. This solution can serve as a crucial prediction tool across scale for point cloud compression, as well as display adaptation.**

***Index Terms*—Point Cloud, Denoising, Dilated Convolutions, Residual Learning**

I. Introduction

Point clouds are being readily used in augmented and virtual reality experiences, as well as 3D sensing for smart cities, robotics, and automated driving applications [1]. Therefore, point cloud capturing, transmission, and processing are essential for these use cases. However, point cloud representation requires a large amount of data which is not always feasible for transmission. Efficient compression technologies are in high demand to make point cloud transmission, storage, and processing more proficient [2]. Therefore, in 2017 MPEG issued a call for proposals on Point Cloud Compression (PCC), and since then MPEG has been evaluating and improving the performances of the proposed technologies [3].

For natural captured 3D sensor signals, scene geometry needs an efficient representation that is scalable in Level-of-Detail (LoD) as well as efficient in compression. MPEG has selected two technologies for PCC: Geometry-based PCC (G-PCC) for dynamically acquired LiDAR point cloud data and for static point cloud data, and video-based point cloud compression (V-PCC) for dynamic content [3]. G-PCC employs octree in its coding scheme, whereas, V-PCC projects point cloud into 2D cube surfaces and then uses state-of-the-art HEVC video encoding to encode dynamic point clouds. Octree has been widely used in processing as well as compression of point clouds [4], [5]. In Octree a node is subdivided into eight child-nodes and the occupancy of each child-node is decided by whether it has points or not. A linear model based PCC approach has been proposed in [6].

(a) (b)

Fig. 1: (a) Original (uncompressed) point cloud, (b) Reconstructed point cloud suffering from quantization noise.

Deep learning for point cloud solutions have also matured with PointNet [7] among the earlier works utilizing fully connected layers. This work was further extended into PointNet++ [8] by introducing hierarchical feature learning. Octree based voxelized deep learning solutions have also been proposed that remained state-of-the-art in the past [9]. Recently, sparse tensors and sparse convolutions have been explored in point cloud deep learning [10]. Sparse convolution leverages the inherent sparsity of point cloud which makes them memory efficient and enables deeper architecture to be built for point cloud learning. Submanifold sparse convolutional network [11] was the first to use sparse convolutions followed by Minkowski Engine [12].

The current compression and transmission schemes often suffer from quantization noise resulting in a lower LoD reconstructed point cloud as shown in Fig. 1. Due to quantization,

978-1-7281-8068-7/20/$31.00

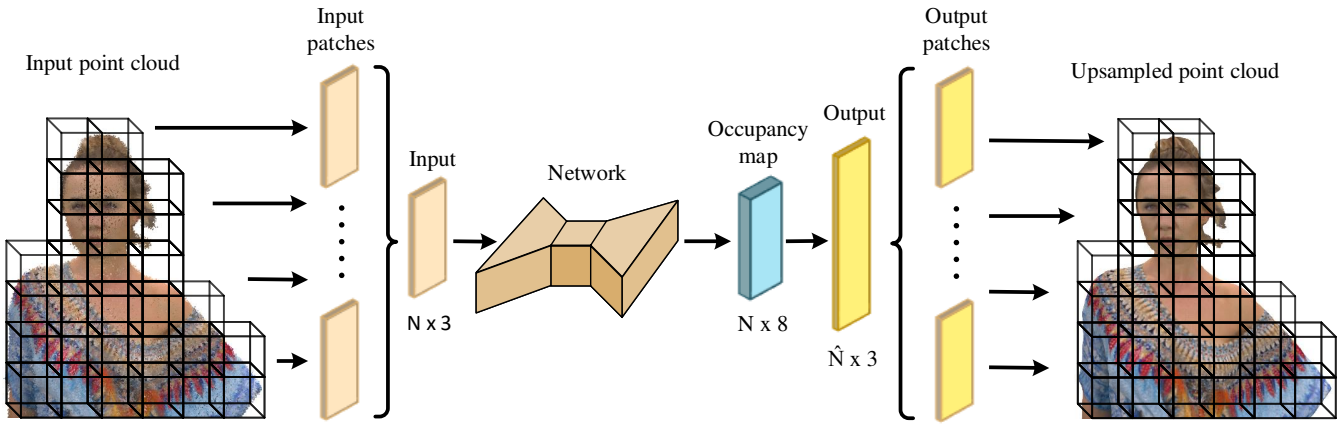
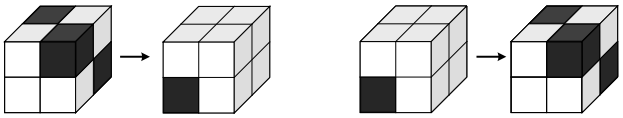


Fig. 2: System Model.



(a) (b)

Fig. 3: (a) Voxel merging due to quantization. (b) Upsample using voxel prediction.

the neighboring points in a voxelized point cloud are merged to form a single voxel resulting in a coarser point cloud with fewer points as shown in Fig. 3a. Leveraging this fact, we use octree voxel subdivisions to predict the occupancy of the empty neighboring voxels with a deep learning model as shown in Fig. 3b. This makes our architecture a point cloud geometry prediction scheme to upsample a lower Level-of-Detail (LoD) point cloud into a higher LoD point cloud without any overhead to the compression-transmission pipeline. We use sparse convolution by employing Minkowski Engine with a UNet like structure employing inception-residual network blocks. To the best of our knowledge, this is the only work on point cloud upsampling that specifically targets the quantization loss during the compression-transmission pipeline.

Both the objective and subjective results show that we significantly improve the quality of the point cloud. Since our technique is a post-processing step, there is no transmission overhead or a bit rate cost to achieve this gain. Another use case for this technique is in display adaptation, when zooming in a point cloud this technique can help super resolve details for display adaptation.

II. System Model

A. Problem Formulation

Quantization is a necessary step in most compression-transmission pipelines. As a consequence of quantization, the neighboring voxelized points get merged into one voxel. Depending on the compression rate, the quantization step-size (qs) can determine the number of points lost and the LoD of the reconstructed point cloud. The quantization loss is modeled by:

$$\hat{X} = \left\lfloor \frac{X}{qs} \right\rfloor \times qs \tag{1}$$

where qs is the quantization step-size, X is the original point cloud and $\hat{X}$ is the quantized point cloud. This quantization results in duplicate points that are removed during the compression process. One example of $qs = 2$ is shown in Fig. 3a. Our goal is to reproduce these lost points by predicting the occupancy of the neighboring empty voxels given the coarser low LoD point cloud. An example of how each voxel would be upsampled using voxel prediction is shown in Fig. 3b.

B. Network Architecture

Our system model is shown in Fig. 2. Generally, a point cloud can be large with millions of points. To feed the point cloud to the network and make our system scalable, we subdivide the point cloud into smaller cube patches and feed each cube patch to the network. The input patch is a voxelized geometry and is of dimension $N \times 3$, where N is the number of voxels in the input cube patch and 3 are the x,y,z coordinates. The output of our network is a $N \times 8$ occupancy map for the $2 \times 2 \times 2 = 8$ voxels encompassing the input voxel. We use this occupancy map to generate a denser point cloud $\hat{N} \times 3$, where $N \leq \hat{N} \leq N \times 8$. It should be noted that our output predicted occupancy map can be more than 8 voxels depending on the qs and amount of upsampling needed. e.g. For $qs = 4$ we can employ an occupancy map of $4 \times 4 \times 4 = 64$. We aggregate the output patches back together to form the upsampled point cloud.

We use sparse tensors and sparse convolution using Minkowski Engine [12]. We employ UNet type architecture [13] with three Inception-residual network blocks (IRB) [14]

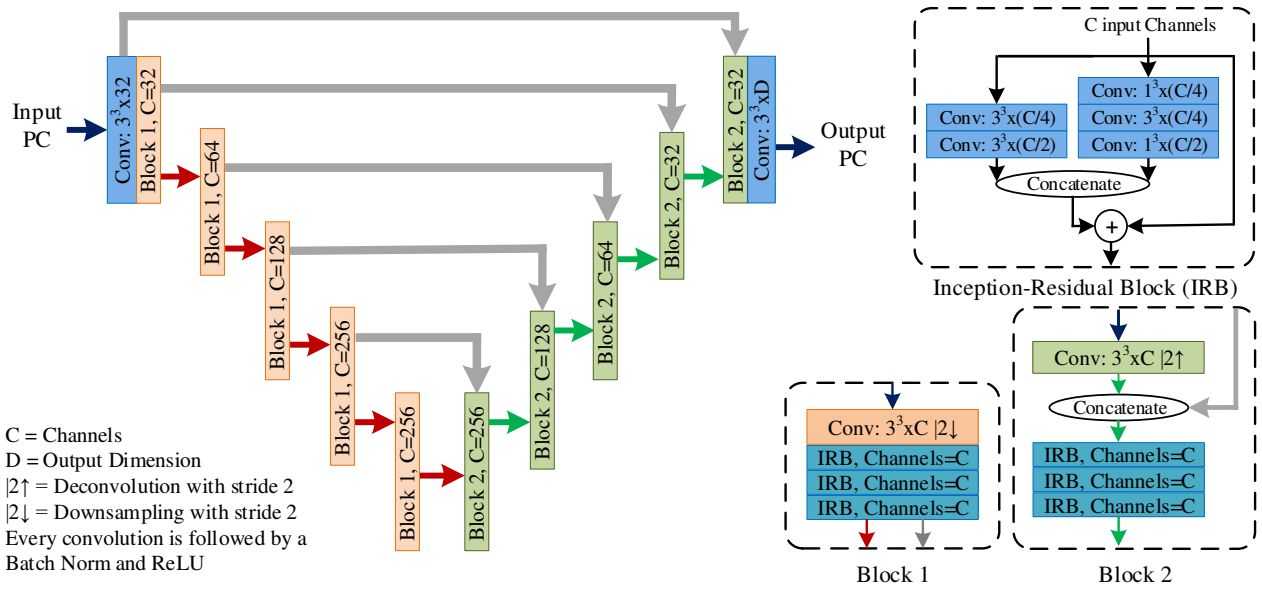


Fig. 4: Network Architecture.

(a) (b) (c)

Fig. 5: (a) Original point cloud, (b) Quantized point cloud with qs = 2, (c) Upsampled point cloud.

per layer as shown in Fig. 4. We use a binary cross-entropy classification loss to compare the occupancy map prediction from the network and the ground truth (original) point cloud.

III. Simulation Results

An input cube patch size of $128 \times 128 \times 128$ voxels is used. **Dataset:** the system model is simulated on 8i voxelized full bodies dataset [15] that is being used in the MPEG standardization. The training is performed on two sequences (longdress, loot) and testing on the 3 sequences (redandblack, soldier, queen). Each sequence has hundreds of point clouds with each point cloud having up to a million points each.

We perform both objective and subjective evaluations. We ran experiments for three $qs = 4/3, 2, 4$. These qs are being used in both MPEG PCC. For $qs = 4/3$ and $qs = 2$, we predict 8 neighboring voxels. However, for $qs = 4$, we increased our receptive field to include $4 \times 4 \times 4 = 64$ neighboring voxels. Which means the output of the network for $qs = 4$ is $N \times 64$. We use D1 geometry PSNR quality metric that is adopted by MPEG [16].

The results of the simulations are shown in Table I. *Input PC* is the reconstructed point cloud after compression pipeline with a specific quantization step and the *Output PC* is the output of our network. As can be seen from the table, we see a significant improvement in the PSNR of the point cloud. Since our method is a post-processing and adds no overhead on the compression-transmission pipeline. We get on average a 8.8678 dB improvement in the quality of a reconstructed point cloud of $qs = 2$ without any bit-rate cost.

(a) (b) (c)

Fig. 6: (a) Original point cloud, (b) Quantized point cloud with qs = 4, (c) Upsampled point cloud.

TABLE I: Average PSNR (dB) results.

qs	Input PC	Output PC	Difference
4/3	64.6646	73.8630	9.1984
2	63.2080	72.0758	8.8678
4	58.0077	65.1890	7.1813

The visual results are shown in Fig. 5 and Fig. 6 for $qs = 2$ and $qs = 4$ respectively. As can be seen our approach significantly improves the quality of the point cloud both objective as well as in subjective evaluations.

IV. Conclusion

Point cloud compression is a necessary step for point cloud transmission, storage, and processing. However, during compression and transmission, point cloud suffers from quantization noise which results in lower Level-of-Detail (LoD) point clouds. In this paper, we propose a deep learning-based point cloud geometry prediction scheme that takes a lower LoD point cloud and upsamples it into a higher LoD point cloud. We use octree to encompass each voxel and its neighboring voxels from the lower LoD point cloud into 8 voxels (or more). Then we learn an occupancy map for each of these voxels using a deep learning architecture. Based on the occupancy map, we generate a higher LoD point cloud by populating the empty voxels. The simulation results show that our method significantly improves the PSNR of the reconstructed point cloud geometry without adding any transmission overhead to the compression-transmission pipeline. This makes our method highly efficient and ideal post-processing step in decoding, as well as super-resolving point cloud for display adaptation.

References

[1] Anique Akhtar, Junchao Ma, Rubayet Shafin, Jianan Bai, Lianjun Li, Zhu Li, and Lingjia Liu, "Low latency scalable point cloud communication in vanets using v2i communication," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[2] Anique Akhtar, Birendra Kathariya, and Zhu Li, "Low latency scalable point cloud communication," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2369–2373.

[3] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al., "Emerging mpeg standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2018.

[4] Ruwen Schnabel and Reinhard Klein, "Octree-based point-cloud compression.," *Spbg*, vol. 6, pp. 111–120, 2006.

[5] X. Zhang, W. Gao, and S. Liu, "Implicit geometry partition for point cloud compression," in *Data Compression Conference (DCC)*, 2020, pp. 73–82.

[6] X. Zhang, W. Gao, and S. Liu, "Linear model based geometry coding for lidar acquired point clouds," in *Data Compression Conference (DCC)*, 2020, pp. 406–406.

[7] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[8] 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*, 2017, pp. 5099–5108.

[9] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–11, 2017.

[10] Benjamin Graham and Laurens van der Maaten, "Submanifold sparse convolutional networks," *arXiv preprint arXiv:1706.01307*, 2017.

[11] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9224–9232.

[12] Christopher Choy, JunYoung Gwak, and Silvio Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[14] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[15] M Krivokuća, PA Chou, and P Savill, "8i voxelized surface light field (8iVSLF) dataset," in *ISO/IEC JTC1/SC29/WG11 MPEG, input document m42914*, 2018.

[16] Sebastian Schwarz, Gaëlle Martin-Cocher, David Flynn, and Madhukar Budagavi, "Common test conditions for point cloud compression," *Document ISO/IEC JTC1/SC29/WG11 w17766, Ljubljana, Slovenia*, 2018.