Improved Deep Point Cloud Geometry Compression

Maurice Quach*, Giuseppe Valenzise*, Frederic Dufaux*

*Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes
91190 Gif-sur-Yvette, France

Abstract—Point clouds have been recognized as a crucial data structure for 3D content and are essential in a number of applications such as virtual and mixed reality, autonomous driving, cultural heritage, etc. In this paper, we propose a set of contributions to improve deep point cloud compression, i.e.: using a scale hyperprior model for entropy coding; employing deeper transforms; a different balancing weight in the focal loss; optimal thresholding for decoding; and sequential model training. In addition, we present an extensive ablation study on the impact of each of these factors, in order to provide a better understanding about why they improve RD performance. An optimal combination of the proposed improvements achieves BD-PSNR gains over G-PCC trisoup and octree of 5.50 (6.48) dB and 6.84 (5.95) dB, respectively, when using the point-to-point (point-to-plane) metric. Code is available at https://github.com/mauricequach/pcc_geo_cnn_v2.

Index Terms—point clouds, compression, neural networks, geometry, octree

I. INTRODUCTION

Due to recent advances in visual capture technology, point clouds have been recognized as a crucial data structure for 3D content. In particular, point clouds are essential for numerous applications such as virtual and mixed reality, sensing for autonomous vehicle navigation, architecture and cultural heritage, etc. Point clouds are sets of 3D points identified by their coordinates, which constitute the geometry of the point cloud. In addition, each point can be associated with attributes like colors, normals and reflectance. Point clouds can have a massive number of points, especially in high precision or large scale captures. This entails a huge storage and transmission cost. As a result, Point Cloud Compression (PCC) is fundamental in practice.

The Moving Picture Experts Group (MPEG) is planning to release two PCC standards [1]: Geometry-based PCC (G-PCC) and Video-based PCC (V-PCC). G-PCC approaches PCC from a 3D perspective and compresses point clouds in their native form using 3D data structures such as octrees. On the other hand, V-PCC approaches PCC from a 2D perspective, projects 3D data onto a 2D plane and makes use of video compression technology. In order to evaluate test models, common test conditions (CTCs) [2] were designed. In this context, the point-to-point (D1) and the point-to-plane [3] quality metrics (D2) are used for quantitative evaluation. Recently, deep point cloud compression (DPCC) methods have been proposed and shown to provide significant coding gains compared to traditional methodologies [4], [5].

In this paper, we focus on lossy compression of static point cloud geometry using deep convolutional networks. Specifically, we propose a set of contributions to improve RD performance and accelerate model training. We then present an ablation study identifying key performance factors for DPCC. In particular, we start from a baseline DPCC model [4] and we consider the following improvements:

- **Entropy modeling**: we consider a hyperprior model to improve entropy coding.
- **Deeper transforms**: that compensate downsampling with progressively higher numbers of filters.
- **Changing the balancing weight in the focal loss**: similar to [4], we cast PCC decoding as an unbalanced classification problem by optimizing a focal loss [6]. Hence, we study the RD performance impact of the focal loss α parameter.
- **Optimal thresholding for decoding**: in order to classify voxels as occupied or not, we propose an optimal thresholding approach that minimizes a given distortion metric (instead of a fixed threshold as in [4]).
- **Sequential training**: in order to reduce the computational complexity of training a network for each RD tradeoff, we propose a sequential training procedure. That is, we train a network corresponding to a given RD point by fine tuning the network trained from the previous RD point. This makes training times up to 8 times faster compared to training independently and improves RD performance.
- **Ablation study**: An extensive ablation study evaluating the impact of each factor mentioned above on RD performance. The evaluated conditions are detailed in Table I.
- **Octree partitioning**: An efficient octree partitioning algorithm that is significantly faster compared to recursive octree partitioning.

II. RELATED WORK

Our research is related most closely to three research areas: static point cloud geometry compression, deep image compression and deep point cloud compression.

Static point cloud geometry compression methods are usually based on the octree structure [7]. Indeed, octrees provide an efficient way of partitioning the 3D space and representing point clouds. In particular, they are especially suitable for lossless coding in combination with octree entropy models [8]. However, lossy compression using octrees alone has poor performance as pruning octree levels decreases the number of points exponentially resulting in significant distortion. To alleviate this issue, many solutions have been proposed such

Funded by ANR ReVeRy national fund (REVERY ANR-17-CE23-0020).
as triangle [9] surface models, planar [10] surface models, graph-based enhancement layers [11] and volumetric functions [12]. The core idea is that by encoding approximations along a coarse octree, we can alleviate the shortcomings of the octree structure. Different from previous work in this area, we study learned approximation models based on deep neural networks.

Deep image compression considers the use of deep neural networks for image compression. An end-to-end image compression solution with joint RD optimization along with a learned entropy model has been proposed in [13], which also replaces (non-differentiable) quantization with uniform noise at training time. As a follow-up of that work, a scale hyperprior model has been proposed in [14]. The scale hyperprior enables the modeling of spatial correlations in the latent space; for each element, it uses a Gaussian distribution whose standard deviation is predicted by a dedicated network. We design models for PCC using these learning-based entropy modeling techniques.

DPCC is a recent research avenue exploring the use of deep neural networks for PCC. For lossy geometry coding, voxel-based DPCC methods have been shown to outperform traditional methods significantly [4], [5], [15]. For lossless geometry coding, deep neural networks have been used to improve entropy modeling [16]. Also, DPCC for attributes has been explored by interpreting point clouds as a 2D discrete manifold in 3D space [17]. Closely related to our study, the behavior and performance of DPCC methods has been investigated in [5]. However, this particular study investigates the characteristics and RD impact of the latent space. In contrast, we seek to understand and identify key performance factors for rate-distortion (RD) performance on a larger scale.

### III. Proposed Improvements

In this section we present different strategies to improve DPCC. We consider as baseline the network proposed in our preliminary work [4] (denoted as c1 in the following). In that work, we relied on shallow transforms to compress entire point clouds at once. However, this has a fundamental limitation in terms of memory usage, as it does not allow to compress large point clouds as those commonly used in MPEG CTCs. Therefore, in this work we make use of octree partitioning to partition point clouds into blocks of size $64 \times 64 \times 64$ voxels, which we have found to be a good trade-off between memory usage and coding performance. In the rest of the paper, we denote the different considered improvements with c2, ..., c6, which are summarized in Table I.

| Name | Model | Transforms | $\alpha$ | Threshold | Training |
|------|-------|------------|---------|-----------|----------|
| c1   | Baseline | Shallow    | 0.90    | Fixed     | Independent |
| c2   | Hyperprior | —         | —       | —         | —        |
| c3   | —      | Deep       | —       | —         | —        |
| c4   | —      | —          | 0.75    | —         | —        |
| c5   | —      | —          | —       | Optimal   | —        |
| c6   | —      | —          | —       | —         | Sequential |

TABLE I: Experimental conditions evaluated in this study.

Fig. 1: Entropy models considered in this work. The $f$ functions are learned transforms, $Q$ refers to quantization and $AC$ refers to arithmetic coding with its associated density model.

A. Entropy modeling (c2)

We consider models that take the voxelized point clouds $x$ and $\tilde{x}$ as input and output. In particular, we consider a baseline model (Fig. 1a) and an hyperprior model (Fig. 1b).

The baseline model is based on an autoencoder architecture with an analysis $f_a$ and a synthesis transform $f_s$ [13]. $y$ is modeled using a learned entropy model for each feature map. The baseline model is expressed as follows

$$y = f_a(x) \quad \tilde{y} = Q(y) \quad \tilde{x} = f_s(\tilde{y}).$$

We consider a scale hyperprior model [14] as a better entropy model for $\tilde{y}$. Specifically, we model $y$ with a zero-mean gaussian density model $\mathcal{N}(0, \tilde{y})$ where standard deviations $\tilde{\sigma}^2$ are predicted from $y$ with $\tilde{\sigma} = f_{hs}(Q(f_{h0}(y)))$. As a result, the spatial dependencies can be modeled better compared to the learned entropy model. The hyperprior model is expressed as follows

$$y = f_a(x) \quad \tilde{y} = Q(y) \quad \tilde{x} = f_s(\tilde{y})$$

$$z = f_{hs}(y) \quad \tilde{z} = Q(z) \quad \tilde{\sigma} = f_{hs}(\tilde{z})$$

where $z$ is modeled with a learned density model for each feature map.

The compression model is trained using joint RD optimization with the loss function $R + \lambda D$. For each RD tradeoff, we train a model with the corresponding $\lambda$ value resulting in transforms and entropy models specialized for this particular tradeoff. The entropy $R$ is computed on $\tilde{y}$ and $\tilde{z}$ for the hyperprior model, using their associated entropy models. Since the quantization operation $Q$ is not differentiable, we use additive uniform noise during training in place of quantization as originally proposed in [13].

B. Deeper transforms (c3)

We compare shallow and deep transforms for analysis and synthesis, as illustrated in Fig. 2. Specifically, we focus on
The capacity at a given layer $W$ is decreased by downsampling along the spatial dimensions. In that way, the number of filters should compensate the loss due to downsampling. This choice is that the number of filters should compensate the loss due to downsampling. The more spatial dimensions are downscaled, the more filters are needed to maintain the same capacity.

Most of the space is empty (usually more than 95%). This large class imbalance between occupied voxels and unoccupied voxels is a barrier to effective training. Indeed, without any countermeasures, the network would converge towards empty outputs only. In order to resolve this class imbalance issue, we adopt the focal loss [6] as our distortion loss.

The focal loss is well suited for point clouds since it addresses the class imbalance issue with $\alpha$-balancing. Moreover, the focal loss differentiates between easy and hard examples using the $\gamma$ parameter. Specifically, the higher $\gamma$ is, the more hard examples are emphasized. With $\gamma = 0$, the focal loss becomes equivalent to the weighted binary cross-entropy.

For conciseness, we adopt the following notation. If $x = 1$, then $x_t = x$, $\alpha_t = \alpha$ and $\tilde{x}_t = \tilde{x}$; otherwise $x_t = 1 - x$, $\alpha_t = 1 - \alpha$ and $\tilde{x}_t = 1 - \tilde{x}$. We then define the focal loss as

$$FL(x, \tilde{x}) = \alpha_t x_t (1 - \tilde{x}_t)^\gamma \log(\tilde{x}_t).$$

We study the impact of the focal loss $\alpha$ parameter on RD performance. The $\alpha$ parameter governs the attention given to occupied voxels and empty voxels. A high $\alpha$ value makes marking occupied voxels as empty more costly than marking empty voxels as occupied and results in denser reconstructions. Originally, we picked the same $\alpha$ value (0.90) as in [4]. This was motivated by the fact that point clouds are often comprised of more than 95% of empty space.

However, we found experimentally that lower $\alpha$ values can actually provide better coding gains. We hypothesize that this is due to the fact that the default $\gamma = 2$ in the focal loss emphasizes hard examples (occupied voxels) more than easy examples (empty voxels). Thus, $\gamma = 2$ already alleviates the class imbalance issue to some extent which explains this phenomenon.

D. Optimal thresholding for decoding (c5)

For each block, after decoding $y$ and $z$ for the hyperprior model into $\tilde{x}$, we need to convert $\tilde{x}$ into binary values in order to obtain the decompressed point cloud. The baseline method (c1) employs a fixed threshold $t = 0.5$. In contrast, we perform this conversion by finding optimal thresholds for each block of voxels. This threshold is transmitted as side information in the bitstream with a small overhead in terms of bitrate.

We formulate optimal thresholding as the problem of finding an optimal threshold $t^*$ such that

$$t^* = \arg \min_t d(x, H(\tilde{x} - t))$$

where $d$ is a distortion metric and $H(x)$ is the heaviside step function (equal to 1 when $x \geq 0$ and 0 otherwise).

E. Sequential training (c6)

We train compression models for each RD tradeoff using a corresponding $\lambda$ value. This allows for transforms and entropy models to be specialized for this particular tradeoff resulting in better RD performance. Unfortunately, using this independent training scheme, we need to train one model for each tradeoff.

To alleviate this issue, we propose a novel sequential training scheme that speeds up training significantly and improves RD performance. The core idea of this scheme is to use previously trained neural network weights as a starting point for new neural networks. Essentially, given a set of $\lambda$ tradeoffs,
TABLE II: RD performance for each experimental condition. We specify BD-PSNR values (dB) compared to G-PCC trisoup and G-PCC octree in each cell (trisoup BD-PSNR / octree BD-PSNR). The best values for trisoup and octree are indicated in bold and the second best in italic. c6 consistently outperforms all other conditions.

TABLE III: Impact of the focal loss $\alpha$ parameter on RD performance. We specify BD-PSNR values (dB) compared to G-PCC trisoup for different $\alpha$ values. The best values are indicated in bold and the second best in italic. $\alpha = 0.75$ outperforms all other $\alpha$ values.

TABLE IV: Ablation study.

In this subsection, we present BD-PSNR values when compared to G-PCC trisoup. The hyperprior model (c2) results in an improvement of 0.47 dB for D1 and 0.77 dB for D2 compared to c1. Adding deep transforms (c3) further improves D1 by 2.04 dB and D2 by 2.81 dB compared to c2.

In Table III, we observe that setting $\alpha = 0.75$ for D1 and $\alpha = 0.50$ for D2 increases RD performance significantly for all point clouds. The average BD-PSNR for $\alpha = 0.75$ is 3.71 dB for D1 and 3.69 dB for D2. Also, the average BD-PSNR for $\alpha = 0.50$ is 3.70 dB for D1 and 6.07 dB for D2. Indeed, higher $\alpha$ values lead to denser reconstructions which are favored by D1 and lower $\alpha$ values to sparser ones which are favored by D2. We select $\alpha = 0.75$ (c4) as we have found experimentally that it performs better when associated with optimal thresholding. Compared to $\alpha = 0.90$ (c3), $\alpha = 0.75$ brings an improvement of 1.92 dB for D1 and 2.49 dB for D2.
Fig. 3: RD curves for each condition in Table II. \( c6 \) consistently outperforms G-PCC trisoup and G-PCC octree.

Fig. 4: Qualitative evaluation on “soldier_vox10_0690”. For \( c6 \) and G-PCC Trisoup, we show the decompressed point cloud and its D1 squared errors. The errors are displayed according to the color scale on the right and are truncated to the 99th percentile (3.0). In parentheses, we specify the D1 PSNR along with the number of bits per input point (bpp).
V. CONCLUSION

We propose a set of key performance factors for DPCC and we present an extensive ablation study on the individual impact of these factors. More precisely, we provide insights on the individual impact of scale hyperprior models, deep transforms, the focal loss $\alpha$ value, optimal thresholding and sequential training. We analyze each of these factors in order to provide a better understanding about why they improve RD performance. The final model (c6) outperforms G-PCC trisoup with an average BD-PSNR of 5.50 dB on D1 and 6.48 dB on D2 and outperforms G-PCC octree with an average BD-PSNR of 6.84 dB on D1 and 5.95 dB on D2.

REFERENCES

[1] S. Schwarz, M. Preda, V. Barontini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuca, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahana, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, “Emerging MPEG standards for point cloud compression,” pp. 1–1. [Online]. Available: https://ieeexplore.ieee.org/document/8571288/

[2] “Common test conditions for point cloud compression,” in ISO/IEC JTC1/SC29/WG11 MPEG output document N19084.

[3] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, “Geometric distortion metrics for point cloud compression,” in 2017 IEEE International Conference on Image Processing (ICIP). IEEE, pp. 3460–3464. [Online]. Available: http://ieeexplore.ieee.org/document/8296925/

[4] M. Quach, G. Valenzise, and F. Dufaux, “Learning convolutional transforms for lossy point cloud geometry compression,” in 2019 IEEE International Conference on Image Processing (ICIP), pp. 4320–4324, ISSN: 1522-4880.

[5] A. F. R. Guarda, N. M. M. Rodrigues, and F. Pereira, “Deep learning-based point cloud coding: A behavior and performance study,” in 2019 8th European Workshop on Visual Information Processing (EUVIP), pp. 34–39, ISSN: 2164-974X.

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007, ISSN: 2380-7504.

[7] R. Schnabel and R. Klein, “Octree-based point-cloud compression,” in Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics, ser. SPBG’06. Eurographics Association, pp. 111–121. [Online]. Available: http://dx.doi.org/10.2312/SPBG/SPBG06/111-120.

[8] D. C. Garcia and R. L. d. Queiroz, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1807–1811.

[9] A. Dricot and J. Ascenso, “Adaptive multi-level triangle soup for geometry-based point cloud coding,” in 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6, ISSN: 2163-3517.

[10] ——, “Hybrid octree-plane point cloud geometry coding,” in 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5, ISSN: 2219-5491.

[11] P. de Oliveira Rente, C. Brites, J. Ascenso, and F. Pereira, “Graph-based static 3d point clouds geometry coding,” vol. 21, no. 2, pp. 284–299.

[12] M. Krivokuča, M. Koroteev, and P. A. Chou, “A volumetric approach to point cloud compression.” [Online]. Available: http://arxiv.org/abs/1810.00484

[13] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” in 2017 5th International Conference on Learning Representations (ICLR).

[14] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, “Variational image compression with a scale hyperprior,” in 2018 6th International Conference on Learning Representations (ICLR).

[15] J. Wang, H. Zhu, Z. Ma, T. Chen, H. Liu, and Q. Shen, “Learned point cloud geometry compression.” [Online]. Available: http://arxiv.org/abs/1909.12037

[16] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, “OctSqueeze: Octree-structured entropy model for LiDAR compression.” [Online]. Available: http://arxiv.org/abs/2005.07178

[17] M. Quach, G. Valenzise, and F. Dufaux, “Folding-based compression of point cloud attributes,” in 2020 IEEE International Conference on Image Processing (ICIP), to be published. [Online]. Available: http://ieeexplore.ieee.org/document/8571288/

[18] ——, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1807–1811.

[19] A. Dricot and J. Ascenso, “Adaptive multi-level triangle soup for geometry-based point cloud coding,” in 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6, ISSN: 2163-3517.

[20] ——, “Hybrid octree-plane point cloud geometry coding,” in 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5, ISSN: 2219-5491.

[21] ——, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1807–1811.

[22] ——, “Learned point cloud geometry compression.” [Online]. Available: http://arxiv.org/abs/1909.12037

[23] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, “OctSqueeze: Octree-structured entropy model for LiDAR compression.” [Online]. Available: http://arxiv.org/abs/2005.07178

[24] M. Quach, G. Valenzise, and F. Dufaux, “Folding-based compression of point cloud attributes,” in 2020 IEEE International Conference on Image Processing (ICIP), to be published. [Online]. Available: http://ieeexplore.ieee.org/document/8571288/

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image compression with a scale hyperprior,” in 2017 26th IEEE International Conference on Image Processing (ICIP), pp. 1021–1025.

[26] ——, “Deep residual learning for image compression with a scale hyperprior,” in 2017 27th European Signal Processing Conference (EUSIPCO), pp. 1–6, ISSN: 2219-5491.

[27] ——, “End-to-end optimized image compression,” in 2017 5th International Conference on Learning Representations (ICLR).

[28] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, “Variational image compression with a scale hyperprior,” in 2018 6th International Conference on Learning Representations (ICLR).

[29] J. Wang, H. Zhu, Z. Ma, T. Chen, H. Liu, and Q. Shen, “Learned point cloud geometry compression.” [Online]. Available: http://arxiv.org/abs/1909.12037

[30] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, “OctSqueeze: Octree-structured entropy model for LiDAR compression.” [Online]. Available: http://arxiv.org/abs/2005.07178

[31] M. Quach, G. Valenzise, and F. Dufaux, “Folding-based compression of point cloud attributes,” in 2020 IEEE International Conference on Image Processing (ICIP), to be published. [Online]. Available: http://ieeexplore.ieee.org/document/8571288/