Comparative Study of 3D Point Cloud Compression Methods

Mai Bui[1], Lin-Ching Chang[1], Hang Liu[1], Qi Zhao[2], Genshe Chen[2]
[1]Department of Electrical Engineering and Computer Science, The Catholic University of America, Washington, DC, USA
[2]Intelligent Fusion Technology, Inc., 20271 Goldenrod Lane, Germantown, MD, USA
{buim, changl, liuh}@cua.edu, {qi.zhao, gchen}@intfusiontech.com

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671822

***Abstract* — 3D sensors such as LiDAR, stereo cameras, and radar have been used in many applications, for instance, virtual or augmented reality, real-time immersive communications, and autonomous driving systems. The output of 3D sensors is often represented in the form of point clouds. However, the massive amount of point cloud data generated from 3D sensors poses big challenges in data storage and transmission. Therefore, effective compression schemes are needed for reducing the bandwidth of wireless networks or storage space of 3D point cloud data. Several point cloud compression (PCC) algorithms have been proposed using signal processing or neural network techniques. In this study, we investigate four state-of-the-art PCC methods using two different datasets with various configurations. The objective of this study is to provide a comprehensive understanding of various approaches in PCC. The results of this paper will be helpful in developing an adaptive 3D point cloud stream compression benchmark that is efficient and benefited from different PCC techniques.**

Keywords — point cloud, data compression, performance assessment, evaluation metrics.

I. Introduction

A point cloud (PC) is a collection of an enormous number of measurements that can be used in 3D modeling. PCs can carry both geometry information and associate attributes (e.g., color, reflectance, and intensity, etc.), plus temporal changes [1]. PCs are widely used in various applications including virtual reality, augmented reality, real-time immersive communications, biomedical imagery, and autonomous driving systems [2-5]. PCs are commonly generated using stereo cameras, Light Detection and Ranging (LiDAR) sensors, or 3D laser scanners. Latest advancements in 3D data acquisition technology offer point cloud representations to be effective, high precision, reliable, and real-time. However, the volume of point cloud data generated by 3D sensors is massive. For example, a 64-line Velodyne LiDAR sensor continuously scanning a given scene generates over 1 billion points in twenty minutes; or a point cloud with 0.7 million points per 3D frame at 30 frames per second (fps) needs a bandwidth around 500 Megabyte per second (MB/s) to transmit [6, 7]. The huge amount of data poses big challenges in both data storage and transmission. For example, it required more space to store the point cloud data locally on a device and is more difficult to share the data with other network nodes (i.e., transmit the data in wireless networks), or to manipulate and analyze the data. In recent years, several point cloud compression (PCC) schemes have been developed [8-15]. Efforts have also been made in evaluating the performance of existing PCC algorithms, but most of them focus on a smaller set of compression techniques with a dataset and specific setting [16-21]. In this paper, four state-of-the-art PCC methods will be investigated using two different public domain datasets and with various configurations (e.g., lossless vs lossy). The result of this paper is essential in developing an adaptive 3D point cloud stream compression benchmark that can take advantages from different 3D PCC techniques under different scenarios, requirements, limitations, and constraints to intelligently address the bandwidth bottleneck of wireless networks.

II. 3D PCC Methods

In this section, we provide brief descriptions of the PCC methods that were selected and evaluated in this study.

G-PCC [8] is a geometry-based technique proposed by Apple Inc. G-PCC encodes the geometry positions directly in 3D space using the coordinated retrieved from octree representation. Geometry positions can be encoded using two approaches and the attributes of point cloud can be encoded using three different methods. The final bitstream can be produced from both geometry bitstream and color bitstream.

V-PCC [8] adopts a projection-based coding principle and is also proposed by Apple Inc. V-PCC decomposes the point cloud data into a set of patches. The 3D patches are generated from several orthographic directions and projected onto a 2D plane. These 2D patches are then processed using 2D video encoder technique. Both depth and attribute information can be retained in the resulting projection images.

Draco [9] is an open-source library developed by Google for compressing and decompressing 3D geometry meshes and point clouds. The main idea behind Draco is using KD tree. After KD tree formation, Draco encodes the data by entropy encoding tools. There is a trade-off between the compressed file size and the visual quality of point cloud depending on users' needs.

GeoCNNv2 [13] is an improved architecture from GeoCNNv1 [12]. GeoCNNv1 architecture is a 3D convolutional auto-encoder (CNN-AE) composed of 3 layers of analysis transform, followed by a uniform quantizer module and 3 layers of synthesis transform. GeoCNNv2 uses GeoCNNv1 as a baseline model and then added several new implementations including entropy modeling, deeper transform, changing the balancing weight in the focal loss, optimal thresholding in decoding, and sequential training.

We summary the differences among these four PCC methods in Table 1. Note that VPCC does not support a compression setting with geometric information only and GeoCNNv2 does not support point cloud data with color and lossless compression.

978-1-6654-3902-2/21/$31.00 ©2021 IEEE

TABLE I. SUMMARY OF POINT CLOUD COMPRESSION METHODOLOGIES

Compression Methods	Scheme	Support Color	Support Lossless	Support Lossy
GPCC [8]	Octree	✓	✓	✓
VPCC [8]	Video	✓	✓	✓
Draco [9]	KD-tree	✓	✓	✓
GeoCNNv2 [13]	CNN-AE	×	×	✓

III. EVALUATION METHODOLOGY

A. Datasets

Two public domain datasets, 8iVFB [22] and MVUB [23], were used for compression performance evaluation. Both datasets contain color information but can be excluded in testing.

8i Voxelized Full Bodies (8iVFB) Dataset is provided by 8i Labs and can be retrieved from the public JPEG Pleno Database. The dataset has four PC sequences known as longdress, redandblack, loot, and soldier with depth 10. This dataset contains a total of 1200 frames and the average size per frame is 23.1 MB.

Microsoft Voxelized Upper Bodies (MVUB) Dataset is provided by Microsoft and can be also retrieved from the public JPEG Pleno Database. The dataset has five PC sequences known as Andrew, David, Phil, Ricardo, and Sarah with 2 spatial resolutions, depth 9 and 10. This dataset contains a total of 1202 frames and the average size per frame is 35.7 MB.

B. Evaluation Metrics

Six quality assessment metrics were used to compare the performance of different compression methods.

Encoding and Decoding Times are the time (in seconds) needed to encode an original PC to binary bitstream and the time needed to decode the binary bitstream to reconstruct to PC data respectively.

Compression Ratio (CR) is the relative reduction of file size after compression. CR is computed as the ratio of the original file size divided by the compressed file size. The higher the compression ratio is, the more effective the PCC is.

Bits per point (bpp) or bitrate is the number of bits needed to store an individual point in a single input PC. Bpp is computed as the ratio of total number of bits divided by total number of input points.

Peak Signal-to-Noise Ratio (PSNR) [24] can be used to compare the data quality between the original PC and the reconstructed PC. PSNR is defined as in equation (1):

$$PSNR = 10\, log_{10} \frac{p^2}{e_{A,B}} \quad (1)$$

, where A and B are the two sets of input and output points (i.e., original PC and reconstructed PC), p is signal peak of original point cloud, and $e_{A,B}$ is the error between all points in A and B. Here we use two different types of PSNR (i.e., MSE-PSNR and HD-PSNR).

Mean Squared Error Peak Signal-to-Noise Ratio (MSE-PSNR) can be computed by replacing the error term in (1) using the mean squared error $e_{MSE\ A,B}$ that is defined as in equation (2):

$$e_{MSE\ A,B}^{p2point} = \frac{1}{N_A} \sum_{\forall a_j \in A} ||a-b||_2^2 \quad (2).$$

Hausdorff Distance Peak Signal-to-Noise Ratio (HD-PSNR) can be computed by replacing the error term in (1) using the Hausdorff distance error $e_{HD\ A,B}$ defined as in equation (3):

$$e_{HD\ A,B}^{p2point} = \max_{a \in A,\, b \in B} \left(\max \left(\min ||a-b||_2^2 \right) \right) \quad (3).$$

Equations (2) and (3) compute the point-to-point (p2point) distance between points in the original PC and reconstructed PC. The point-to-plane (p2plane) distance can be computed by replacing $||a-b||_2^2$ with $((a-b) \cdot N_j)_2$, where the normal vector N_j on point a_j is in the original point cloud A.

For each dataset, we randomly select 100 frames for testing. Each evaluation metric was computed by averaging the result from 100 frames. Various configurations were assessed: lossless vs lossy compression and with or without color attribute. Voxelization was first applied to all point clouds into 10-bit depth and normalization was also performed as a pre-processing step. All experiments were carried out on a computer with a 2.8 GHz Intel Core i7-7700HQ Quad-Core, NVIDIA GeForce GTX 1060 (6GB GDDR5) with 16GB DDR4 RAM.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Table II shows the decoding and encoding times and bits per point (bpp) for the three compression methods that support lossless compression. Draco outperforms G-PCC and V-PCC for both encoding and decoding times but has a much higher bpp. V-PCC requires a relatively high encoding time and GPP has a reasonable encoding time. It also has a much lower bpp than Draco.

TABLE II. THE COMPARISON OF ENCODING/DECODING TIME AND BITS PER POINT FOR LOSSLESS COMPRESSION METHODS

Lossless Condition	G-PCC	V-PCC	Draco
Encoding Time (s)			
8i with color	5.38	134.53	0.31
MVUB with color	8.73	216.10	1.03
8i w/o color	2.00	-	0.32
MVUB w/o color	3.97	-	0.28
Decoding Time (s)			
8i with color	2.83	1.70	0.10
MVUB with color	4.57	2.48	0.15
8i w/o color	0.49	-	0.08
MVUB w/o color	0.91	-	0.08
Bits per point (bpp)			
8i with color	15.63	13.13	108.72
MVUBC with color	14.75	13.86	107.04
8i w/o color	2.28	-	96.00
MVUB w/o color	2.49	-	96.00

Fig 1 shows the MSE-PSNR curves versus bitrate for lossy PCC methods with and without color attribute using 8iVFB dataset. V-PCC outperforms G-PCC and Draco when the PC contains color information while GeoCNNv2 outperforms when the PC contains geometric information only. Similar results were obtained when using MVUB dataset (data not shown).

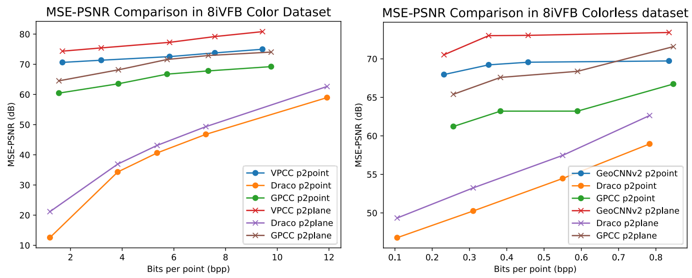

Fig 1. Comparison of MSE-PSNR for lossy compression methods with or without color information using 8iVFB dataset.

Fig 2 shows the HD-PSNR curves versus bitrate for lossy PCC methods with and without color attribute using MVUB dataset. G-PCC outperforms V-PCC and Draco disregarding the PCC contains color information or not when HD error was used to compute PSNR. GeoCNNv2 has the lowest HD-PSNR for both p2point and p2plane when using MVUB dataset without color information. Surprisingly, this is inconsistent with the results in Fig. 1. Further investment is needed to get a deeper understanding on using the neural network approach for point cloud compression under different scenarios.

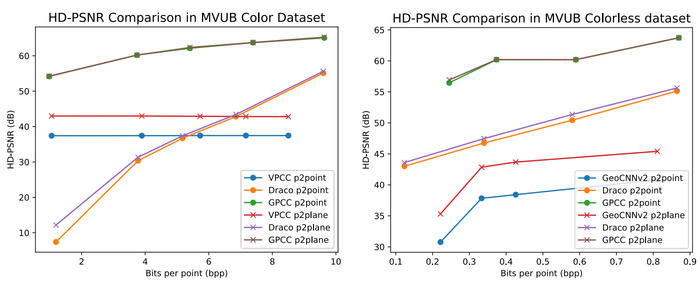

Fig 2. Comparison of HD-PSNR for lossy compression methods with or without color information using MVUB dataset.

V. Conclusion

In this paper, four state-of-the-art 3D PCC methods were investigated using two different datasets with various settings to provide a comprehensive understanding of these existing compression techniques. We developed a framework to objectively compare the efficiency of these PCC algorithms and observed some of their advantages and limitations. For instance, Draco has a limitation to achieve results in low bitrates; GPCC outperforms other two PCC algorithms at all bitrates in terms of HD-PSNR metric; GeoCNNv2 outperforms others when PC with geometric information only and in terms of MSE-PSNR metric. We conclude that despite great achievements of 3D PCC have been made in recent years, there are still a large space for further improvement especially on how to select or combine different 3D PCC techniques under constraints in various applications such as to intelligently resolve the bandwidth bottleneck of wireless networks for massive point cloud data.

VI. References

[1] E. Alexiou et al, “Towards neural network approaches for point cloud compression”, Proceedings of SPIE - The International Society for Optical Engineering, Volume 11510, 2020.

[2] E. Alexiou et al, “Towards subjective quality assessment of point cloud imaging in augmented reality”, 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), 2017, pp. 1-6, doi: 10.1109/MMSP.2017.8122237.

[3] R. Mekuria et al, "Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 4, pp. 828-842, April 2017, doi: 10.1109/TCSVT.2016.2543039.

[4] N. Farahani et al, “Three-dimensional Imaging and Scanning: Current and Future Applications for Pathology.”, Journal of pathology informatics, 2017.

[5] A. Geiger et al, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354-3361, doi: 10.1109/CVPR.2012.6248074.

[6] S. Biswas et al, “MuSCLE: Multi Sweep Compression of LiDAR using Deep Entropy Models”, Image and Video Processing, 2020.

[7] C. Cao et al, “3D Point Cloud Compression: A Survey”, The 24th International Conference on 3D Web Technology 2019, 1-9. 10.1145/3329714.3338130.

[8] D. Graziosi et al, “An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC).”, APSIPA Transactions on Signal and Information Processing, 2020, 9, E13. doi:10.1017/ATSIP.2020.12.

[9] Draco authors. (2020) Draco 3D graphics compression. https://google.github.io/draco/

[10] D.T. Nguyen et al, “Learning-Based Lossless Compression of 3D Point Cloud Geometry”, ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4220-4224.

[11] D.T. Nguyen et al, “Multiscale deep context modeling for lossless point cloud geometry compression.”, ArXiv, abs/2104.09859.

[12] M. Quach et al, “Learning convolutional transforms for lossy point cloud geometry compression,” in 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019, pp. 4320–4324.

[13] M. Quach et al, “Improved deep point cloud geometry compression,” in 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP). IEEE, 2020, pp. 1–6.

[14] J. Wang et al, “Lossy point cloud geometry compression via end-to-end learning,” IEEE Transactions on Circuits and Systems for Video Technology, 2021.

[15] J. Wang et al, “Multiscale point cloud geometry compression,” in 2021 Data Compression Conference (DCC). IEEE, 2021, pp. 73–82.

[16] J. Zhang et al, “A subjective quality evaluation for 3D point cloud models”, in International Conference on Audio, Language and Image Processing (ICALIP), Shanghai, China, 2014.

[17] H. Liu et al, “A Comprehensive Study and Comparison of Core Technologies for MPEG 3D Point Cloud Compression”, IEEE Transactions on Broadcasting, 2019.

[18] E. Alexiou et al, “A comprehensive study of the rate-distortion performance in MPEG point cloud compression.” APSIPA Transactions on Signal and Information Processing, vol. 8, 2019.

[19] S. Perry et al, “Quality evaluation of static point clouds encoded using mpeg codecs.”, IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020.

[20] C.H. Wu et al, “PCC arena: a benchmark platform for point cloud compression algorithms.”, Proceedings of the 12th ACM International Workshop on Immersive Mixed and Virtual Environment Systems, 2020.

[21] C.H. Wu et al, “Technical Report: Quantitative Comparison of Point Cloud Compression Algorithms.”

[22] E. d'Eon et al, "8i Voxelized Full Bodies - A Voxelized Point Cloud Dataset," ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006, Geneva, January 2017.

[23] C. Loop et al, “Microsoft voxelized upper bodies - a voxelized point cloud dataset,” in ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012. May 2016.

[24] D. Tian et al, "Geometric distortion metrics for point cloud compression," 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3460-3464, doi: 10.1109/ICIP.2017.8296925.