




Transform Decomposition Switching for Efficient Attribute Compression of 3D Point Clouds Using Neural Networks

Reetu Hooda  *[†], W. David Pan  *[§] and Bernard Benson  †

*Department of Electrical & Computer Engineering, University of Alabama in Huntsville, AL 35899, USA

†McLeod Software Corporation, Birmingham, AL 35242, USA

†rh0059@uah.edu, §pand@uah.edu, †bernard.benson@mcleodsoftware.com

Abstract—An adaptive technique to switch between RAHT and Dyadic RAHT using 3D Sobel filter has been found to improve the compression in 3D point clouds by offering substantial cumulative compression gains. However, the drawback of this switching scheme is its need for tuned thresholds. To this end, we propose to use neural networks to resolve the threshold dependency issue so that the switching becomes truly adaptive. Two publicly available point cloud datasets were used to test the effectiveness of the proposed method. We achieved significant gains on MVUB and minor gains on 8iVFB dataset over all Dyadic approach.

Index Terms—Attribute compression, Neural networks, Point clouds, RAHT, Dyadic RAHT.

I. INTRODUCTION

AFTER 3D meshes, point cloud (PCs) are the most advanced media format used in data representation. They comprise of scattered points in space where each point is represented using a spatial coordinate (x, y, z) called Geometry with color and/or reflectance information related to it called Attributes [1]. Although a mesh provides far more complex geometry and sub-metric inspection of an object, point clouds are more widely used due to some limiting factors of meshes, mainly linked to complexity. Moreover, dense PCs are used in the development of 3D meshes to generate their finely detailed faces, edges and vertices. Therefore, PCs can also be perceived as building blocks of a mesh. Due to the popularity of PCs in recent years, they are employed in various applications such as immersive media, medical tomography, autonomous driving, augmented reality (AR), robotics, etc.

With increasing usage of inexpensive 3D scanners and modern multibeam echosounders, there has been a rise in generation of very high volume of dense point cloud datasets [2]. Because of the unstructured nature of these PCs, unlike traditional 2D images/videos, PC compression can be extremely challenging [3]. Therefore, in many practical applications such as smooth streaming with limited bandwidth, efficient PC coding solutions become essential [4].

Paramount efforts have been made by researchers to improve the compression efficiency of point clouds. Moving Picture Expert Group (MPEG) has been conducting meetings towards standardization of compression technologies for point cloud compression (PCC), which is now widely used as a benchmark in academic and non-academic research [5]. The

two distinct technologies are Geometry-based PCC (G-PCC) and Video-based PCC (V-PCC). From the details of the codec architecture mentioned in [1], it can be concluded that these technologies mainly comprises of 3D to 2D projection and rule-based traditional approaches [6].

In addition to the conventional coding solutions implemented on PCs, deep learning (DL) has also made its way in advance media compression with impressive preliminary results [7]. Most of the DL coding solutions are 3D CNN-based autoencoders (AE) [8], [9] with few fully-connected neural network (FCNN) approaches [10], [11] and even fewer recurrent neural network (RNN) [12] based techniques. Among the limited neural network based solutions, only a handful of them are end-to-end.

In [13], one of the first few end-to-end framework for lossy attribute coding using AE is introduced. There are also few partitioning-based methods such as [14] which segments the PC into fine-grained patches, whereas [15] uses kd-tree based decomposition to efficiently divide the color distribution. The coding gains of the above mentioned approaches are reported to be comparable and in some cases outperforms the MPEG-anchor. However, these approaches need to train large models to generate rate-distortion curves and they are also data dependent [16].

In G-PCC, attributes are encoded using Region Adaptive Hierarchical Transform (RAHT), separately from geometry [17]. Due to the effectiveness of Dyadic decomposition [18], Blackberry proposed to replace RAHT with Dyadic RAHT which was later adopted in the codec [19]. Then, it was observed in [20] that switching between the two types of decomposition was found to be more effective instead of using only one type of transform throughout the PC. But the approach was threshold dependent, thereby hindering its practical usability.

In this paper, we address the threshold tuning problem of the technique in [20] by training neural networks to learn the switch between RAHT and Dyadic RAHT. Experimental results show considerable compression gains while successfully eliminating the threshold dependency.

The remainder of the paper is organized as follows: Section II provides the problem statement with details related to data preprocessing. Section III describes the shallow neural

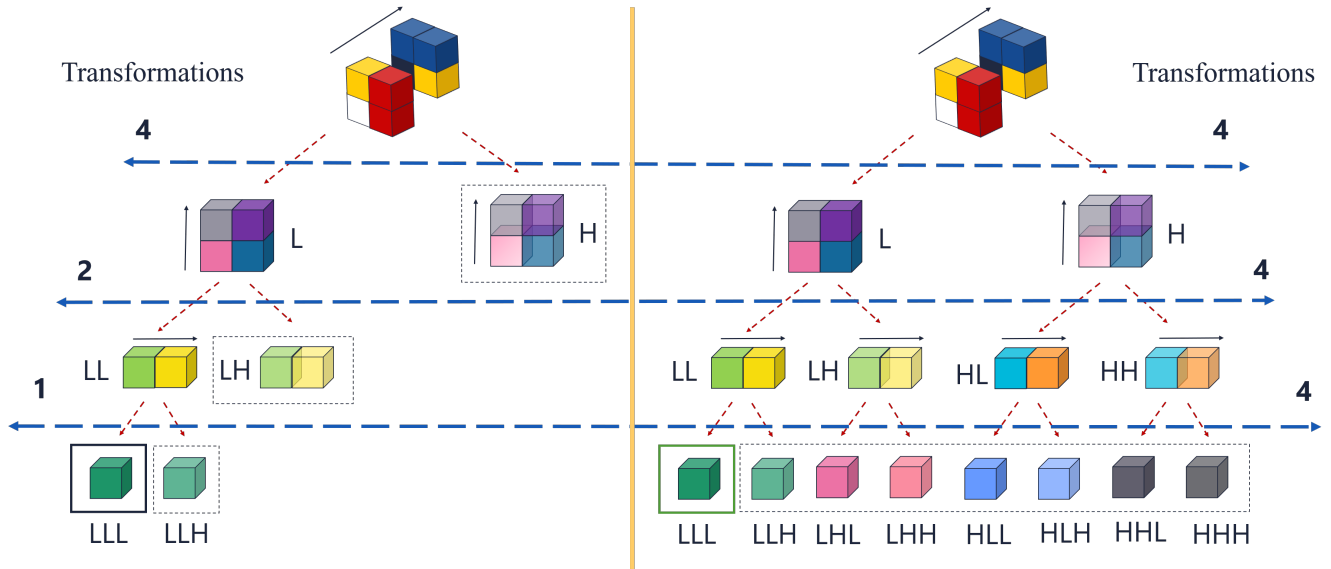


Fig. 1. Two types of decomposition (RAHT Vs Dyadic RAHT).

network (SNN) coding solution. Section IV discusses the BD-rate performance of the proposed technique. Section V concludes the paper.

II. PROBLEM STATEMENT

One of the options to encode the attribute values of a PC is RAHT, which is an adaptive variation of a Haar wavelet transform introduced in [5]. It is based on the hierarchical structure of the occupancy map called Octree. In G-PCC codec, the transform is applied in three steps as shown in Fig. 1. Let us consider the 8 blocks at the top. They represent the attribute values that are first decomposed in the z -direction to generate the low-pass (L) and high-pass (H) components performing four transformations in step 1 (High-pass components are shown in dashed lines). In the second step, only low-pass coefficients from the first step are decomposed to output LL and LH performing two transformations in y -direction. Finally, LL is transformed to generate LLL and LLH performing only one transformation in x -direction. In Dyadic decomposition shown on the right, high-pass components at each stage are also decomposed performing four transformations in each step.

position showed $\sim 2.3\%$ average gain on the entire MPEG dataset and hence was adopted in the G-PCC codec.

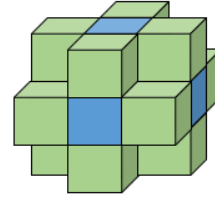


Fig. 3. 18-neighbourhood.

It was recently found that switching between these two transforms based on the characteristics of the neighboring blocks offered improvements in rate-distortion sense. From the experimental analysis in [20], it was concluded that RAHT is beneficial for flat regions and Dyadic RAHT provides better gains when applied on discontinuous regions. The idea was to study the nature of 18 neighboring blocks (shown in Fig. 3) using a 3D Sobel filter kernels (S_x, S_y and S_z) to output the strength of the edge (∇f) defined as follows:

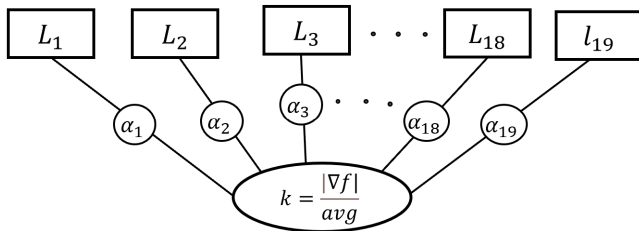


Fig. 2. 3D edge detection scheme (The original switching which depends on the k value).

RAHT was later replaced by Dyadic decomposition. Changing the fundamental structure from RAHT to Dyadic decom-

$$\nabla f(x, y, z) = G(x, y, z) = [G_x G_y G_z]^T \quad (1)$$

Where G_x, G_y and G_z are gradients in x, y and z direction respectively. The magnitude was approximated using the absolute values for faster computation as defined below:

$$|(\nabla f)| \approx |G_x| + |G_y| + |G_z| \quad (2)$$

The magnitude of the edge ($|\nabla f|$) was normalized using the average (avg) of its neighbouring blocks to compute normalized magnitude of the edge called k as shown in Fig. 2. Thresholding on k was performed to interpret the continuity in the central block. A uniform block is indicated if k exceeds

a certain tuned threshold and hence RAHT decomposition is used, otherwise Dyadic is used.

This process can be perceived as the luma values (represented as L_i where $1 \leq i \leq 19$) of 18 neighbors and central block multiplied with constant weights (α 's) as shown in Fig. 2, where α 's represents the fixed weights of the Sobel filter. Normalized value k was used to make a binary decision based on a threshold. The disadvantage of the proposed method is its threshold dependency on k , which was selected by trial and error for each point cloud. Therefore, threshold dependency hinders the general applicability of this switching scheme.

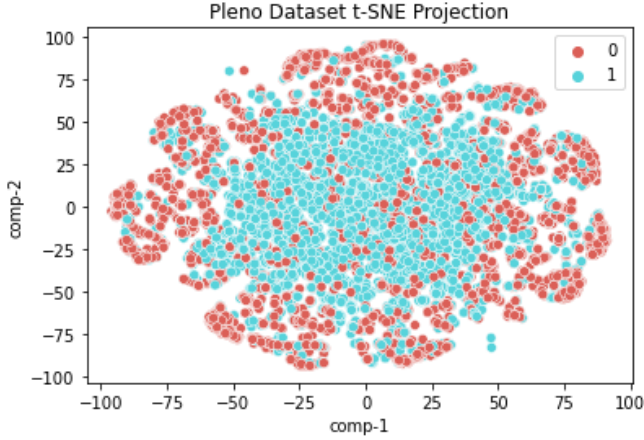


Fig. 5. T-SNE visualization for Pleno data.

Neural network (NN) based compression methods for PC have emerged recently with comparable compression gains, albeit with the necessity to store multiple trained models to generate different rate-distortion trade-offs [21]. In this paper, we address the threshold tuning problem by replacing the original 3D edge filter scheme with a shallow neural network. The problem now becomes a pattern classification task with two output classes (RAHT and Dyadic RAHT).

Data Preprocessing: Only Luma values of 18 neighboring blocks with the central block was used in the original scheme and tested for threshold values of $T = 0.2, 0.4, 0.6$ and 0.8 . The threshold with maximum cumulative gain was selected. The

19 features with the transform chosen (0 for RAHT and 1 for Dyadic) based on the manual tuning was written in a data file to prepare the training dataset. Data cleaning is performed by first separating the 0 (RAHT) samples with 1 (Dyadic RAHT) samples to study their distribution. Duplicate data samples were dropped from both the classes and concatenated together followed by random shuffling.

Data Visualization: We use t-SNE to visualize data in two dimensions. A random sample of 15000 data points is used to create the embedding which are then projected onto a two dimensional plane to make it easier for visualization [22]. Here we observe more of Dyadic points compared to RAHT making the data biased as shown in Fig 5. Since performing Dyadic transform a majority of the time and using RAHT only for very weak edges was found to be beneficial in [20], it was expected for the data samples to be more biased towards the Dyadic class.

III. NEURAL NETWORK STRUCTURE

The PC geometry is encoded using the octree approach, where the PC is enclosed in a 3D volume of $D \times D \times D$ voxels. The 3D volume is divided into 8 sub-cubes of size $D/2 \times D/2 \times D/2$. Only occupied voxels are divided further and represented by '1', and '0' otherwise. This process is repeated until the dimension reduces to $1 \times 1 \times 1$. Since the occupancy information is required for the attribute compression method chosen by the user, the geometry is encoded first.

For attribute compression using either RAHT or Dyadic RAHT, the octree representation is also considered. Let us consider a certain region in a 3D PC. The block to be transformed is highlighted in orange as shown in Fig. 4 referred as central block. Now, similar to the prediction scheme [23], the original scheme of 3D edge detection also uses 18 neighbors around the central block. 19 Luma values (18 neighbors plus the central block) were used in detecting the edges in the central block to decide the type of decomposition to be used. These 19 values form features for the target class based on the original scheme used in [20].

Although most deep learning based coding solutions for PC compression uses CNN-based architecture to retain 3D correlations and maintain lower complexity via weight sharing.

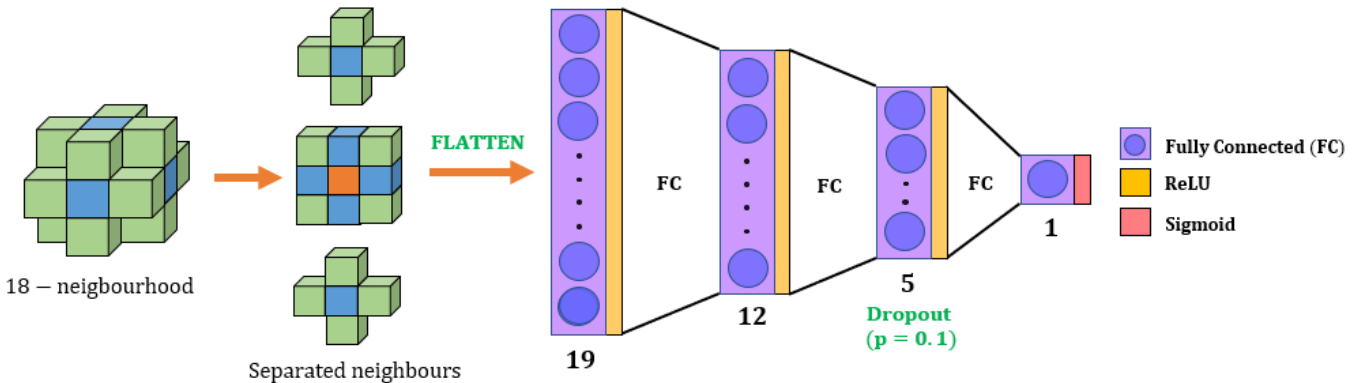


Fig. 4. Fully connected neural network (FCNN) architecture.

They need to store multiple models to obtain different RD curves with extensive running time required in training large networks [11]. In our problem, we encounter a very localized region in the PC where at a time, a maximum of 19 Luma values are processed. Therefore, we have opted for a fully connected structure as shown in Fig. 4. Instead of using a set of 3D blocks, directly fed to the neural network, we flatten the 18 neighbors and the central block and use 19-tuple feature vectors. The architecture has an input layer of 19 neurons with an output layer of 1 neuron for the target label ('0' is used for RAHT and '1' is used for Dyadic RAHT), which is essentially a binary classification problem. We have used 2 small hidden layers of 12 and 5 neurons respectively, making it a total of 4-layer architecture including the input and output layers. All the layers are fully connected. A sigmoid function was used for the final layer, whereas the ReLU activation function was used for the remaining three layers. The FCNN architecture as shown in Fig. 4 is used for training with the data split of 70%, 15%, 15% to divide it into training, validation, and testing set respectively. The model was trained for 1000 iterations with the binary cross entropy (BCE) loss function. Adam optimizer with a learning rate of 0.01 and weight decay of 1×10^{-6} was used for fast convergence. Regularization with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and dropout with probability of 0.1 in the third layer was used to avoid over-fitting and to improve generalization on the biased data. Finally, the trained model was imported into the MPEG-GPCC codec replacing the original 3D edge detection scheme which was threshold dependent.

IV. RESULTS

This section presents the performance assessment of the proposed method to eliminate the threshold dependency. Learning based coding solutions are generally most effective for dataset that share some similarity that uses the adaptation from the training data onto the testing data. In this context, we used Microsoft Voxelized Upper Bodies (MVUB) dataset [24] from the open source JPEG Pleno database, which is a dynamic point cloud dataset publicly available. Each sequence consists of multiple frames that share correlations between the frames within a sequence. We trained the neural network shown in Fig. 4 with specifications provided at the end of the previous section.



Fig. 6. MVUB dataset (from left to right): Andrew, Phil, Ricardo, Sarah and David.

The model was trained using only the first frame of each of the five sequences and tested on ten random frames. Fig. 6 shows the first frames of Phil, Ricardo, Sarah and Andrew sequence. Accuracy of 92.78%, 92.54%, 92.38% was achieved on the training, validation and testing data respectively. The

TABLE I
BD-RATE GAINS FOR PROPOSED SCHEME OVER DYADIC RAHT ON MICROSOFT VOXELIZED UPPER BODY (MVUB) DATASET.

Test Sequences	No. of Points	BD-rate			Cumulative Gain
		Luma	Cb	Cr	
Andrew	277038	-0.9%	-8.4%	-3.6%	-12.9%
Phil	336323	0.2%	-5.4%	-2.1%	-7.3%
Ricardo	952178	-1.2%	-3.8%	-3.8%	-8.8%
Sarah	304528	-0.9%	-4.9%	-4.4%	-10.2%
David	302584	-2.9%	-3.7%	-2.9%	-9.5%

loss curve is shown in Fig. 8 and accuracy curve is shown in Fig. 7 for training and validation set.

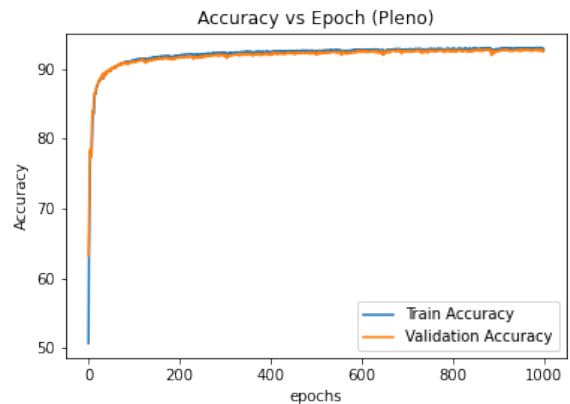


Fig. 7. Training vs validation accuracy.

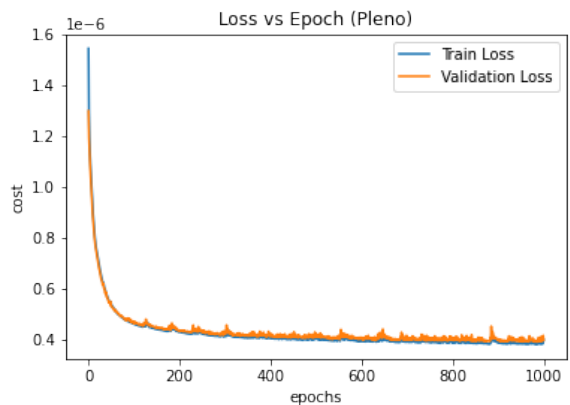


Fig. 8. Training vs validation loss.

The Dyadic RAHT approach was used as the benchmark to assess the performance of the proposed technique and RD curves were used as the performance metrics to observe the gain across the three channels (Luma, Cb, Cr). Table I shows the gains over each channels for a random frame that achieved the highest gain. The confusion matrix of the

TABLE II
CUMULATIVE BD-RATE GAINS FOR PROPOSED SCHEME OVER DYADIC RAHT ON 10 RANDOM FRAMES ON MVUB DATASET.

Test Sequences	Cumulative Gain										Average
	1	2	3	4	5	6	7	8	9	10	Gain
Andrew	-12.9%	-12.7%	-10.0%	-9.5%	-8.3%	-8.0%	-6.6%	-6.3%	-5.7%	-5.3%	-8.53%
Phil	-7.3%	-5.9%	-5.6%	-3.6%	-3.4%	-3.0%	-2.7%	-2.3%	-2.0%	-1.7%	-3.75%
Ricardo	-8.8%	-8.4%	-6.8%	-5.2%	-4.2%	-4.1%	-3.9%	-3.5%	-3.0%	-2.7%	-5.06%
Sarah	-10.2%	-8.5%	-8.2%	-6.6%	-5.6%	-4.9%	-4.4%	-4.2%	-2.4%	-2.2%	-5.72%
David	-9.5%	-8.2%	-7.6%	-7.3%	-5.7%	-5.4%	-5.2%	-4.6%	-2.7%	-2.5%	-5.87%

classification results are shown in Fig. 10. The instances or counts in the confusion matrix can also be expressed in terms of percentages. The proposed scheme achieved 92.37% of accuracy with high sensitivity, specificity and precision of 95.88%, 88.88% and 89.69% respectively, showing the accuracies obtained are not skewed by uneven test data.



Fig. 9. 8iVFB dataset (from left to right): Soldier, Loot, Long Dress and Red & Black.

TABLE III
BD-RATE GAINS FOR PROPOSED SCHEME OVER DYADIC RAHT ON 8i VOXELIZED FULL BODIES (8iVFB) DATASET .

Test Sequences	Cumulative Gain				Average
	1	2	3	4	Gain
Soldier	-4.8%	-4.0%	-3.8%	-3.4%	-4.0%
Loot	-8.5%	-6.9%	-5.5%	-3.9%	-6.2%
Long dress	-1.3%	-1.0%	-0.6%	-0.4%	-0.825%
Red and Black	-3.0%	-1.9%	-1.6%	-1.5%	-2.0%

To summarize the RD performance of the proposed scheme, we use the cumulative gain, which is calculated by simply adding the gain or loss across the three channels. The cumulative gain on 10 random frames from the five dynamic sequences is tabulated in Table II arranged in decreasing order. The proposed scheme provided an average cumulative gain of 8.53%, 3.75%, 5.06%, 5.72% and 5.87% for Andrew, Phil, Ricardo, Sarah and David sequence respectively over the Dyadic RAHT approach. In order to verify the robustness

of the proposed scheme, we also tested it on the 8iVFB (8i Voxelized Full Bodies) JPEG Pleno dataset shown in Fig. 9. Our method not only provided an average cumulative gain of 4.0%, 6.2%, 0.825% and 2.0% for Soldier, Loot, Long Dress and Red & Black sequence, respectively (as summarized in Table III), but also eliminates the need to tune the threshold as in the original switching scheme.

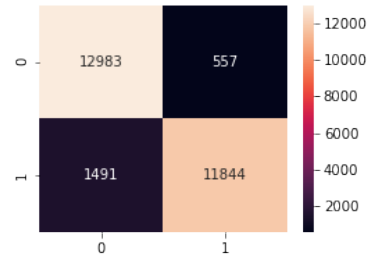


Fig. 10. Confusion matrix of classification results.

V. CONCLUSION

In this paper, we present a new neural network based coding approach that focuses on compression of static point cloud attributes. More precisely, we address the threshold dependency problem to enable the generalized applicability of the transform switching technique based on the characteristics of different regions in a point cloud. The proposed neural network technique comprises of three main steps: collecting data from the 3D edge detection scheme, using the data to train a fairly simple shallow neural network and finally deploying the trained network to replace the original switching scheme. We have demonstrated the efficiency of proposed method for point cloud attribute compression in terms of RD-performance, by comparing it to MPEG-GPCC standardized method that uses only Dyadic transform throughout the point cloud. Average cumulative gains of over 3% was achieved on MVUB dataset, with only minor gains attained on 8iVFB dataset. In this research work, we have only used the Luma values for classification. In the future work, feature size could be increased to obtain higher training, validation and testing accuracy. We will study the effect of these changes on the attribute compression of point clouds.

REFERENCES

- [1] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "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*, vol. 9, p. e13, 2020.
- [2] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, "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, 2019.
- [3] S. Zhang, W. Zhang, F. Yang, and J. Huo, "A 3D Haar Wavelet Transform for Point Cloud Attribute Compression Based on Local Surface Analysis," in *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1–5.
- [4] G. P. Sandri, "Compression of Point Cloud Attributes," Ph.D. dissertation, University De Brasilia, Dec 2019.
- [5] R. L. de Queiroz and P. A. Chou, "Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, 2016.
- [6] L. Gao, T. Fan, J. Wan, Y. Xu, J. Sun, and Z. Ma, "Point cloud geometry compression via neural graph sampling," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 3373–3377.
- [7] A. F. R. Guarda, N. M. M. Rodrigues, and F. Pereira, "Point cloud coding: Adopting a deep learning-based approach," in *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1–5.
- [8] M. Quach, G. Valenzise, and F. Dufaux, "Learning convolutional transforms for lossy point cloud geometry compression," *CoRR*, vol. abs/1903.08548, 2019. [Online]. Available: <http://arxiv.org/abs/1903.08548>
- [9] J. Wang, H. Zhu, Z. Ma, T. Chen, H. Liu, and Q. Shen, "Learned point cloud geometry compression," 2019. [Online]. Available: <https://arxiv.org/abs/1909.12037>
- [10] W. Yan, Y. Shao, S. Liu, T. H. Li, Z. Li, and G. Li, "Deep autoencoder-based lossy geometry compression for point clouds," *CoRR*, vol. abs/1905.03691, 2019. [Online]. Available: <http://arxiv.org/abs/1905.03691>
- [11] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Interpretable unsupervised learning on 3d point clouds," *CoRR*, vol. abs/1712.07262, 2017. [Online]. Available: <http://arxiv.org/abs/1712.07262>
- [12] C. Tu, E. Takeuchi, A. Carballo, and K. Takeda, "Point cloud compression for 3d lidar sensor using recurrent neural network with residual blocks," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3274–3280.
- [13] X. Sheng, L. Li, D. Liu, Z. Xiong, Z. Li, and F. Wu, "Deep-pcac: An end-to-end deep lossy compression framework for point cloud attributes," *IEEE Transactions on Multimedia*, vol. 24, pp. 2617–2632, 2022.
- [14] B. Zhao, W. Lin, and C. Lv, "Fine-grained patch segmentation and rasterization for 3-d point cloud attribute compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4590–4602, 2021.
- [15] H. Liu, H. Yuan, Q. Liu, J. Hou, H. Zeng, and S. Kwong, "A hybrid compression framework for color attributes of static 3d point clouds," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1564–1577, 2022.
- [16] R. Hooda, W. D. Pan, and T. Syed, "A survey on 3D point cloud compression using machine learning approaches," in *SoutheastCon 2022 (SoutheastCon22)*, Mobile, USA, Mar. 2022.
- [17] G. P. Sandri, P. A. Chou, M. Krivokuća, and R. L. de Queiroz, "Integer Alternative for the Region-Adaptive Hierarchical Transform," *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1369–1372, 2019.
- [18] E. Peixoto, "Intra-Frame Compression of Point Cloud Geometry Using Dyadic Decomposition," *IEEE Signal Processing Letters*, vol. 27, pp. 246–250, 2020.
- [19] J. Taquet and S. Lasserre, "G-PCC On dyadic RAHT," *ISO/IEC JTC1/SC29/WG11 MPEG/m53557*, 2020.
- [20] R. Hooda and W. D. Pan, "Early termination of dyadic region-adaptive hierarchical transform for efficient attribute compression of 3d point clouds," *IEEE Signal Processing Letters*, vol. 29, pp. 214–218, 2022.
- [21] A. F. R. Guarda, N. M. M. Rodrigues, and F. Pereira, "Deep learning-based point cloud geometry coding: Rd control through implicit and explicit quantization," in *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2020, pp. 1–6.
- [22] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: <http://www.jmlr.org/papers/v9/vandemaaten08a.html>
- [23] D. Flynn and S. Lasserre, "G-PCC CE13.18 report on upsampled transform domain prediction in RAHT," *ISO/IEC JTC1/SC29/WG11 MPEG/m49380*, July 2019.
- [24] C. Loop, Q. Cai, S. O. Escolano, and P. A. Chou, "Microsoft Voxelized Upper Bodies – A Voxelized Point Cloud Dataset," *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012*, May 2016. [Online]. Available: <http://plenodb.jpeg.org/pc/microsoft/>