TOWARDS ACCURATE RATE ESTIMATION FOR 3D POINT CLOUD COMPRESSION BY TSPLVQ

Amira Filali, Vincent Ricordel, Nicolas Normand

TOWARDS ACCURATE RATE ESTIMATION FOR 3D POINT CLOUD COMPRESSION BY TSPLVQ

Amira Filali, Vincent Ricordel and Nicolas Normand

LS2N Laboratory, Nantes University, Polytech, rue Christian Pauc, 44306 Nantes, France

ABSTRACT

Point clouds are widely emerged as a promising 3D visual representation model for immersive and high quality experiences using dense point cloud. However, as they are usually made up of thousands up to billions of points, advanced techniques of data compression are essential to store and transmit this type of data. This paper improves prior work, which provided a new top-down hierarchical geometry representation based on adaptive Tree-Structured Point-Lattice Vector Quantization (TSPLVQ), to make the point cloud geometry compression more adaptive to the input point cloud characteristics. In the paper, more robust Rate-Distortion optimization process is introduced to perform efficient and accurate rate-aware splitting decisions when building and coding the tree structure. Experimental results in geometry point cloud compression, considering the tree structures and the leaves coding, show that the solution takes advantage of well-established principles that have been paramount to reach higher levels of point cloud geometry compression performance.

Index Terms— 3D point cloud, compression, rate-distortion optimization, G-PCC

1. INTRODUCTION

Among various newly acquired content types, 3D point cloud (PC) appears to be one of the most efficient representation of immersive media content thanks to the fast development of 3D scanning techniques, establishing a convergence between real and virtual realities and enabling more sophisticated experiences applications.

Characterized by geometry and multiple associated attributes, point cloud forms a spatially discrete set of points in a 3D geometric coordinate system. The development of even more precise capturing devices and the increasing requirements to realistically represent and to vividly render the 3D scenes, inevitably not only induce thousands up to billions of points, but also cause high complexity in the scattered random distribution of the spatial distribution, which brings great challenges to the storage consumption and transmission system. Thus, more advanced compression techniques are in urgent demand to make point clouds useful in practice.

The Moving Picture Experts Group (MPEG) is leading the process of seeking technologies and building an open standard for point cloud compression (PCC) [1]. The targeting standard adresses two main classes of solutions dealing with PCC: V-PCC which takes advantage of the usage of well-known 2D video technologies by projecting the point characteristics onto 2D frames, and a second class called G-PCC, for geometry-based compression of static point clouds. G-PCC is more appropriate for the context of this paper, as the G-PCC uses native 3D data structures and has large potential of improvements.

Both G-PCC and V-PCC are based on conventional models, such as octree decomposition [2], triangulated surface model, region-adaptive hierarchical transform [3], [4], and plane projection [5]. Other explorations related to the PCC relied on graphs [6], binary tree embedded with quadtree [7], or recently volumetric model [8]. To compare the proposed compression solutions, MPEG provided quality evaluation metrics for PCC leading to the selection of the point-to-point and point-to-plane as baseline metrics [9].

The rest of the paper is organized as follows. Section 3 describes briefly the adaptive TSPLVQ method. Details on the new tree rate estimation are given in Sections 3.1 and 3.2. In Section 4 we present and analyze the coding performance of the proposed approach. Finally, conclusions are drawn in section 5.

2. RELATED WORK

Actual researches in point cloud compression field can be classified as point cloud geometry compression and attributes compression [10,11]. In the present paper, our work is mainly related to point cloud geometry compression. Many solutions for point cloud geometry compression have been explored using the octree structure to split the whole point cloud into smaller voxel volumes to better structure and represent the 3D point cloud [12] [13]. *Cohen et al.* [14, 15] extended the well-known shape adaptive Discrete Cosine Transform (SA-DCT) [16], to the voxelized 3D point clouds. Authors in [17] employed intra prediction and binary tree structure for effectively partitioning unorganized points into block structure in a lossless point cloud geometry compression approach. A

lossless intra encoder of voxelized point clouds is introduced in [18] that views the point cloud geometry as an array of bi-level images using a dyadic decomposition instead of the popular octree decomposition.
Ricordel et al. have introduced in [19, 20] a Vector Quantizer (VQ) based on truncated cubic lattices embedding in the context of classical image and video encoding according to a trade-off between rate and distortion using the Lagrangian method.
Inspired by the formulation of quantization in [19, 20], we proposed in [21] to significantly expand it by using new tools, different rate-distortion optimizations and several practical adaptations to the case of G-PCC.

The present paper builds on the work proposed in [21] to introduce a more acurate rate optimization for the purpose of efficient 3D point cloud geometry coding. Exactly, we improve the adaptive Tree-Structured Point-Lattice Vector Quantization (TSPLVQ) by using two splitting schemes (2×2×2 or 3×3×3) of a cubic Voronoï cell, and by adapting the distortion versus rate trade-off. Thus, the purpose of the present work is to reduce more the amount of data of a 3D point set while preserving as much information as possible by considering the distortion in the rendered 3D content from the decoded point cloud. In this work, we consider more accurate rate computation based on the entropic cost estimation of the tree. It is therefore better for rate-distortion optimization during the TSPLVQ procedure. Additionally, in contrast to the MPEG G-PCC reference model, the optimized TSPLVQ method produces higher resolution point clouds for lower bitrates.

3. POINT CLOUD COMPRESSION BY TSPLVQ

Our prior work in [21] is at the crossroads of static G-PCC, vector quantization of 3D data and rate-distortion optimization driven compression. The TSPLVQ approach, based on the embedding of truncated cubic lattices, permits hierarchical description of the 3-D PC through an unbalanced tree structure. The tree growing structure is achieved by using an iterative process such as, at each loop the best choice has to be done, between the node to split and according to two splitting schemes (2×2×2 versus 3×3×3) to better map the PC splitting. This choice is based on a rate-distortion criterion, the optimization is then performed locally, where the Lagrange multiplier λ controls the trade-off between rate and geometric distortion. The rate considered in [21], is either a constant cost of node splitting (8 bits if the splitting is 2×2×2, 27 bits if the splitting is 3×3×3), or a basic entropy estimation taking into account the entropy of the population in relation to node points.

3.1. Optimized estimation of the tree rate cost

Entropy coding is critical for source compression to exploit statistical redundancies. Theoretically, the entropy bound of the source symbol (e.g., splitting bitstream in our case) is closely related to its probability distribution, and accurate rate estimation plays a major role in rate-distortion optimization driven compression.
We propose to use a more accurate estimation of the entropic cost of the node splitting during the tree growing process. For each node splitting is computed, the increase in rate calculated in terms of entropic encoding cost of the node splitting, and the decrease in geometric distortion.

At this level of process, our objective is to code PC geometry stored in its 3D volumetric representation, this is referred as the *voxelization*. Considering the geometry of PC, a voxel V in the 3D representation at (i, j, k) position, corresponds also to a node (n_j) in the tree structure, is set to 1, e.g., $V(i, j, k) = 1$, if it is occupied (contains one or more PC points) and split, and $V(i, j, k) = 0$ otherwise. Each splitting scheme partitions the 3D cube associated to a node n_j either into 8 or 27 embedded smaller cubes (so children nodes of n_j). For each branch connecting n_j to its child, one bit is used, let's define it as the splitting state. Splitting state is a bitstream associated with each voxel node. Each voxel node is divided into several voxels (children) depending on the quantization scheme being considered (namely, 2×2×2 or 3×3×3) and every leaf node has in turn an associated voxel value (1 or 0). Note also that each node indicates by a position index where it lies inside its parent's splitted 3D cube (for instance when considering 2×2×2 splitting case, the position index are simply listed from 1 up to 8). If after a loop, a child node undergoes splitting (namely, its corresponding cube splitting), we update the splitting state of its parent node. The splitting state determines using 0s and 1s the children nodes that are split, e.g. a given splitting by 2×2×2, could have $splittingstate = 01001101$. So children nodes with position index 1, 4, 5, 7 are split and children nodes with position index 0, 2, 3, 6 are not split. At each level of quantization step, every splitting state has associated occurency probability which gives how likely the splitting state occurs in the tree. In other words, the occurency probability of a splitting state is the ratio between the number of time this splitting state occurs in the tree, and the number of splitted nodes. We can then approximate the actual rate of every splitting state by computing its entropy.
For instance, the rate $R(s_j)$, used to calculate λ score in equation (4) in [21], is equal to the entropy of the splitting states of the tree nodes at the $j-th$ loop in the TSPLVQ growing tree s_j.
The splitting states entropies of the tree nodes has to be updated dynamically after each loop of the TSPLVQ splitting process, by taking into account of the incrementing and

decrementing of the splitting states counters.
In order to count and to store the probabilities of all the splitting states, we use a hash table. Advantage of this strategy is that splitting state of any length can be dynamically stored in the table to avoid storing all the possible combinatories of all the splitting states (2^8 for 2×2×2 splitting and 3^{27} for 3×3×3 splitting). Due to this, we are able to run a hybrid quantization method using 2×2×2 and 3×3×3 in competition with each other and locating probability occurency of any state becomes fast due to hashing.

3.2. Final Bit-stream and attribute Compression

In the tree, each occupied leaf node corresponds exactly to one representant point which 3D location is set to the average value of the initial cloud points contained within the cube associated to the leaf. The mean colour is used to represent the colour information of all the cloud points inside the leaves cell. To further compress the corresponding reproduction vectors geometry and colour at the leaves level, we propose in this work to perform range coding by using Lempel–Ziv–Markov chain algorithm (LZMA) [22] for a lossless compression.
In other context, we could consider the voxels centers, instead of the mean points coordinates, to represent the input point cloud geometry and colour using the optimized TSPLVQ. In this case, we do not need to encode any explicit geometry information at nodes level, only build the tree structure by using recursive splittings described arithmetically in the bit stream when traversing the tree in different orders. Thus to encode the point cloud geometry : either we proceed progressively by using the scalable descriptions obtainable at the tree nodes (for instance by scanning its successive depth levels) the PC points in each leaf node are replaced by the corresponding center point, either we consider directly and only the final points positions and colours associate to the tree leaves.
For the purposes of this paper we aim to encode and decode only the point cloud thus the geometry and colour information at the leaves level are enough.

4. EXPERIMENTAL RESULTS

4.1. Experimental setup

We used our optimized TSPLVQ, relied on Point-to-Point geometry distortion, to encode 3D point clouds. We compared our approach against the MPEG octree-based reference test model (lossy G-PCC model) [23]. All parameters in the G-PCC test model are kept unchanged for fair comparisons. We selected four static point clouds from people object dataset: $Soldier$, $LongDress$, $Loot$ and $LongDress$ suggested by MPEG-3DG group [24] as a test dataset. The performance is then measured in terms of the point-to-plane symmetric PSNR metric given in [25].

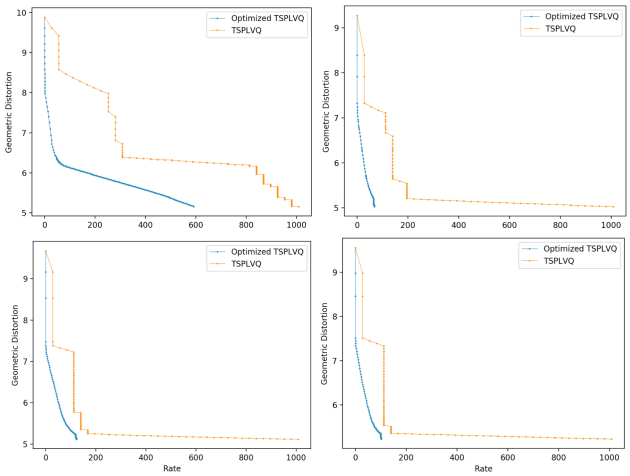

Fig. 1. RD curves showing the performance of the different steps of the optimized TSPLVQ. (first row): $Soldier$ and $RedandBlack$, (second row): $Loot$ and $LongDress$. Distortion is computed using the Point-to-Point geometric distortion [21] and the rate is the entropic cost of the splitting scheme.

4.2. Results

A selection of results is presented in Table 1. The analysis of the PSNR metric of the four point clouds, $Soldier$, $LongDress$, $Loot$ and $Redandblack$, shows that our method when considering the leaves cost only, obviously outperforms the reference model on all the testing point clouds in term of PSNR and Bitrate. Considering these 4 points clouds, the reference model has an average bitrate of 0.03 $bpov$ (bits per occupied voxel) and an average PSNR of 45.93 dB while our method has an average bitrate of 0.0002 $bpov$ and an average PSNR of 59.90 dB. It is worth observing that the entropy cost of the built trees of all point clouds, arithmetically encoded, is greater than the reference model and our method considering the leaves encoding. Moreover, we should take into consideration the colour information that also has to be transmitted, it is why we add this information (the colours) to the leaves, when we consider the leaves cost in Table 1.

We compute RD curves for each sequence of the test point clouds. We compared both rate-distortion driven TSPLVQ: the adaptive TSPLVQ [21] and the rate-adaptive TSPLVQ. The performance comparisons in term of rate and distortion between the 2 methods are reported in Fig. 1. The optimized method outperforms the previous TSPLVQ on all point clouds providing a maximal decrease in distortion, and a minimal increase in rate.
In Fig. 2, we show examples on the four selected 3D objects. These particular examples show that our method produces more points than the reference model at lower bitrates. The MPEG G-PCC model does not show details in complex regions. It also leads to halo artifacts and the resulting point

Table 1. Comparison of symmetric PSNR and Bitrate metrics results between the optimized TSPLVQ based on Point-to-Point distortion and MPEG reference model.

Point Cloud	Number of points	MPEG G-PCC			Our approach			
		Number of points	PSNR	Bitrate (bpov)	Number of points	PSNR	Leaves cost (bpov)	Tree nodes cost (bpov)
Soldier	1089091	20065	52,79	0.029	73103	58.86	0.0002	0.1473
LongDress	857966	15685	52.78	0.031	48869	59.74	0.0002	0.0418
Loot	805285	14828	52.78	0.029	47616	60.08	0.0003	0.0444
Redandblack	757691	13832	25.40	0.032	45034	60.08	0.0003	0.3203

Fig. 2. Rendering results for original point clouds (first row), compressed point clouds using MPEG lossy PCC (second row): (a), (b), (c), (d) and using our optimized TSPLVQ (third row): (e), (f) (g), (h).

cloud requiring therefore a post-processing. For instance, for the *Soldier* point cloud, the MPEG reference model cannot show details in the complex and smooth regions inducing a great loss of visual details (see Fig. 2 (a), (b), (c), (d)) while with our method using a lower bitrate for the leaves (0.0002 bpov), reasonable quality was achieved on all point clouds, as shown in 2(e), (f), (g), (h). Hence, the optimized method can preserve the details and the global look of the original point cloud.

In addition, we could obtain very close rendering to the original point cloud when we produce more points thanks to the proposed multiscale approach with low bitrate. Our visual rendering seems qualitatively very close to the original rendered point clouds compared to the reference test model. The benefits of the optimized method over the reference model in terms of both objective and subjective quality are easily observable.

5. CONCLUSION

In this paper, we introduced a rate-optimized TSPLVQ method for lossy compression of the 3D point cloud geometry. Our method takes advantage of well-established principles that have been paramount to reach higher levels of point cloud compression performance.

6. REFERENCES

[1] S. Schwar et al., "Emerging mpeg standards for point cloud compression," 2019.

[2] D. Meagher, "Geometric modeling using octree encoding," Computer graphics and image processing, vol. 19, no. 2, pp. 129–147, 1982.

[3] 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, Aug. 2016.

[4] R. L. de Queiroz and P. A. Chou, "Transform coding for point clouds using a gaussian process model," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3507–3517, July 2017.

[5] E. Lopes, J. Ascenso, C. Brites, and F. Pereira, "Adaptive plane projection for video-based point cloud coding," IEEE International Conference on Multimedia and Expo (ICME), pp. 49–54, July 2019.

[6] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," IEEE Transactions on Image Processing, vol. 25, no. 4, pp. 1765–1778, April 2016.

[7] B. Kathariya, L. Li, Z. Li, J. Alvarez, and J. Chen, "Scalable point cloud geometry coding with binary tree embedded quadtree," IEEE International Conference on Multimedia and Expo (ICME), p. 1–6, July 2018.

[8] M. Krivokuća, P. A. Chou, and M. Koroteev, "A volumetric approach to point cloud compression–part ii: Geometry compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 2217–2229, December 2020.

[9] S. Schwarz, G. Cocher, D. Flynn, and M. Budagavi, "Common test conditions for point cloud compression," inISO/IECJTC1/SC29/WG11 MPEG output document N17766, July 2018.

[10] X. Yiqun and et al., "Rate-distortion optimized scan for point cloud color compression," Visual Communications and Image Processing (VCIP), 2017.

[11] C. Zhang, D. Florencio, and C. Loop, "Point cloud attribute compressionwith graph transform," International Conference on ImageProcessing (ICIP), pp. 2066–2070, 2014.

[12] R. Schnabel and R. Klein, "Octree-based point-cloud compression," Eurographics Symposium on Point-Based Graphics, pp. 111–120, 2006.

[13] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," International Conference on Robotics and Automation, Shanghai, 2011.

[14] R.A. Cohen, D. Tian, and V. Vetro, "Point cloud attribute compression using 3-D intra prediction and shape-adaptive transforms," Data Compression Conference (DCC), pp. 141–150, 2016.

[15] R.A. Cohen, D. Tian, and V. Vetro, "Attribute compression for sparse point clouds using graph transforms," International Conference on Image Processing (ICIP), Phoenix, AZ, USA, pp. 1374–1378, Sept. 2016.

[16] T. Sikora and B. Makai, "Shape-adaptive dct for generic coding of video," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 59–62, 1995.

[17] W. Zhu, Y. Xu, L. Li, and Z. Li, "Lossless point cloud geometry compression via binary tree partition and intra prediction," IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Luton, pp. 1–6, Oct. 2017.

[18] E. Peixoto, "Intra-frame compression of point cloud geometry using dyadic decomposition," IEEE Signal Processing Letters, vol. 27, pp. 246–250, 2020.

[19] V. Ricordel and C. Labit, "Tree-structured lattice vector quantization," European Signal Processing Conference (EUSIPCO), Italy, Sept 1996.

[20] V. Ricordel and C. Labit, "Vector quantization by packing of embedded truncated lattices," in *Proceedings., International Conference on Image Processing, Washington, DC, USA*, 1995, vol. 3, pp. 292–295 vol.3.

[21] A. Filali, V. Ricordel, N. Normand, and W. Hamidouche, "Rate-distortion optimized tree-structured point-lattice vector quantization for compression of 3D point clouds geometry," International Conference on ImageProcessing (ICIP), sept. 2019.

[22] A. Akoguz, S. Bozkurt, A. Gozutok, G. Toprakci, M. Bogaz E. Turan, and S. Kent, "Comparison of open source compression algorithms on VHR remote sensing images for efficient storage hierarchy," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, July 2016.

[23] K. Mammou, P. A. Chou, D. Flynn, M. Krivokuća, O. Nakagami, and T. Sugio, "G-PCC codec description v1," ISO/IEC JTC1/SC29/WG11 N18015, October 2018.

[24] C. Tulvan, R. Mekuria, and Z. Li, "Draft dataset for point cloud coding (PCC)," ISO/IEC JTC1/SC29/WG11N16333, June 2016.

[25] Julien Ricard, "PCC CE 0.3 on new metrics," ISO/IEC JTC1/SC29/WG11 N18032, October 2018.