Geometry Compression of 3D Static Point Clouds based on TSPLVQ

Amira Filali, Vincent Ricordel, Nicolas Normand

Fifth Sino-French Workshop on Information and Communication Technologies
SIFWICT 2019 - June 21, 2019, Nantes, France

Geometry Compression of 3D Static Point Clouds based on TSPLVQ

Amira Filali
LS2N Laboratory
Nantes University
Rue Christian Pauc
44306 Nantes, France
amira.filali@univ-nantes.fr

Vincent Ricordel
LS2N Laboratory
Nantes University
Rue Christian Pauc
44306 Nantes, France
vincent.ricordel@univ-nantes.fr

Nicolas Normand
LS2N Laboratory
Nantes University
Rue Christian Pauc
44306 Nantes, France
nicolas.normand@univ-nantes.fr

***Abstract*—In this paper, we address the challenging problem of the 3D point cloud compression required to ensure efficient transmission and storage. We introduce a new hierarchical geometry representation based on adaptive Tree-Structured Point-Lattice Vector Quantization (TSPLVQ). This representation enables hierarchically structured 3D content that improves the compression performance for static point cloud. The novelty of the proposed scheme lies in adaptive selection of the optimal quantization scheme of the geometric information, that better leverage the intrinsic correlations in point cloud. Based on its adaptive and multiscale structure, two quantization schemes are dedicated to project recursively the 3D point clouds into a series of embedded truncated cubic lattices. At each step of the process, the optimal quantization scheme is selected according to a rate-distortion cost in order to achieve the best trade-off between coding rate and geometry distortion, such that the compression flexibility and performance can be greatly improved. Experimental results show the interest of the proposed multi-scale method for lossy compression of geometry.**

***Index Terms*—3D point cloud geometry, lattice vector quantization, rate-distortion optimization**

I. Introduction

Due to the increased advances in 360-degree video, virtual and augmented reality fields, the interest in capturing the real world in multiple dimensions and in presenting it to users in a fully immersive environment has never been higher. Many fast and reliable cameras, laser scanners, etc. are rapidly spreading as devices for capturing the 3D data in different ways providing high definition 3D content [1]. It becomes also feasible to capture a 3D object using dense point cloud. 3D point cloud is a *de facto* standard for Computer Graphic and 3D modeling. Technically, 3D point clouds are referred as *unordered* sets of points in a 3D coordinate system which indicates the location of each point showing the external surface of an object, along with one or more attributes (such as color, normal, transparency, etc.) associated with each point. They have recently emerged as representations of the real world enabling immersive forms of interaction, navigation, and communication. Unfortunately, such representations require a large amount of data, not feasible for common processing such as storage or transmission. This large amount of point cloud data provides a significant barrier for mass market applications and presents new challenges to the signal processing and compression research community. Efficient compression technologies well adapted to the content chain are thus in high demand and are key components to democratize the immersive content of augmented and virtual reality applications.

In 2017, the Moving Picture Experts Group (MPEG) as one of the main standardization groups dealing with multimedia, identified the trend, and started recently the process of seeking technologies and building an open standard for compactly representing 3D point clouds, and since then it has been evaluating and improving the performance of the proposed technologies [2]. The targeting standard addresses two classes: V-PCC which consists in using the well-known 2D video technologies by projecting the point characteristics onto 2D frames, a second class is more appropriate for the context of this paper, called G-PCC, for geometry-based methods. G-PCC consists in decomposing the 3D space into a hierarchical structure of cubes and encoding each point as an index of the cube it belongs to.

Several methods of point cloud representation have been developed for geometry compression purpose. *Schnabel and Klein* introduced in [3] the octree structure to partition the whole point cloud into small voxels. Recently, the Point Cloud Library (PCL) [4], which provides a complete set of processing methods for point cloud, has been applied as the basic infrastructure for point cloud compression. In this framework, octree structure is adopted to organize and represent the 3D point cloud, serving as a primary data structure of point cloud codec. The principle consists in building a large cuboïd to contain all the point cloud data in a 3-D space. Then, the algorithm divides the large cuboïd into small 3D voxels with the same size. For each voxel, the algorithm calculates the mean value of the 3-D coordinates of all the points in that voxel. The mean point is used to represent all the points that lie in that voxel. In this way, the point cloud can be compressed while preserving the geometric details. Based on the graph based transform [5], in [6], *Zhang et al.* proposed an efficient point cloud compression, which compacts the energy of color attributes to achieve higher compression ratio. *Cohen et al.* [7], [8] extended concepts used to code arbitrarily shaped regions in image and video the well-known shape adaptive Discrete Cosine Transform (SA-DCT) [9] to the voxelized 3D point

clouds. Authors in [10], have introduced an adaptive scanning scheme to improve color attribute compression performance. The best scheme is subsequently selected in the sense of rate-distortion optimization based on the Lagrange multiplier. In the context of classical image and video encoding field, *Ricordel et al.* [11], [12] have proposed a new Vector Quantizer (VQ) based on truncated lattice embedding. They have investigated the VQ complete design with: the lattice truncation, the multi-stage procedure of quantization, the unbalanced tree-structured codebook design according to a trade-off between distortion and rate.

In this paper, we propose to adapt this method for the multi-scale representation of 3D point cloud geometry. We innovate by introducing adaptive Tree-Structured Point-Lattice Vector Quantization (TSPLVQ) with two partitioning methods (2x2x2 or 3x3x3) of a cubic Voronoï cell, and by adapting the distortion versus rate trade-off. Thus, the purpose of our method is to reduce 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.

The remainder of the paper is organized as follows. Section III describes the TSLVQ method, while the proposed quantization decision scheme is detailed in Section IV. In Section V we present and analyze the coding performance of the proposed approach. Finally, Section VI concludes this paper.

II. TREE-STRUCTURED LVQ (TSLVQ)

Tree-Structured LVQ (TSLVQ) is a gathering of many quantization approaches where the quantization is processed through a decision tree. Its benefits are reduced computation complexity with the use of simpler sub-codebooks, and a structure adapted to progressive representation.

TSLVQ [11] aims at using a hierarchical set of embedded lattices which is achieved such as it is possible to embed a lower scale truncated lattice into a cell of the next higher scale truncated lattice. So a scaling factor b between successive truncated lattices of the hierarchy has to be set (see 1). The use of the cubic lattice permits optimal embedding and fast quantization. As the tree growing approach [13] is considered, the tree structure is achieved by using an iterative process such as, at each loop, an individual leaf partitioning is applied. The partitioned leaf is chosen according to a rate-distortion criterion.

III. THE PROPOSED ADAPTIVE TSPLVQ

The point cloud is highly complex and unstructured, such that an organization and representation strategy of the point cloud would directly affect the processing and coding performance. In [12], *Ricordel et al.* made an attempt to improve the coding efficiency by projecting vectors directly into 3x3x3 truncated lattices based on the TSLVQ. The octree structure is employed in many previous works [4] where the whole point cloud is partitioned only into 2x2x2 small voxels. Nevertheless, one single partitioning scheme is insufficient to account for the characteristics of all kinds of point clouds for best fitting of the content, due to the diverse point cloud structures. The proposed approach, based on the embedding of truncated lattices, permits hierarchical description of the 3-D point cloud through an adaptive tree-structure codebook simplifying its representation and compression. Instead of a single partitioning scheme, in this work two partitioning schemes (2x2x2 versus 3x3x3) are employed to better map the point cloud to 3-D adaptive truncated lattices. This greatly increases the flexibility in mapping the 3D point cloud data into tree-structure for efficient compression.

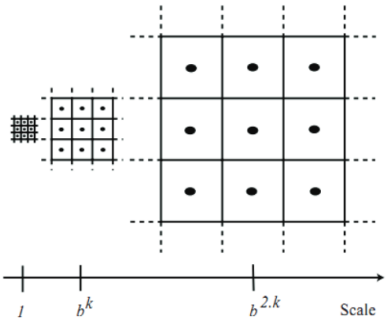

Fig. 1. Hierarchy principle of cubic lattice. In 2D with scaling factor $b = 3$ such as a cubic Voronoï is partitioned in (3x3) [12].

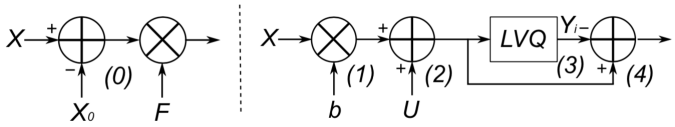

Fig. 2. Initialization (left) and basic procedure of TSPLVQ (right).

The basic quantization procedure used for partition a cube based on TSPLVQ is illustrated in 2 in five main steps:

(0): Initialization: The point X is first normalized to fit inside the root Voronoï cell by centering it around the mean point X_0 and then by scaling data with the factor: $F = \frac{1}{\sqrt{E_{max}}}$, where E_{max} is the maximal euclidean distance between a point and the center.

(1) and (2): A 3D point X (characterized by its coordinates) inside a given cubic Voronoï is scaled by factor b and then shifted, the shifting U and factor b set up the splitting of the cube either in 2x2x2 (b=2, U=(1/2,1/2,1/2)) or in 3x3x3 (b=3, U=(0,0,0)).

(3): The fast quantization algorithm [14] is then used to produce by rounding the corresponding reproduction vector Y_i.

(4): The output vector is centered to permit the next quantization level.

So to initialize the construction of the tree-structured codebook that will represent the 3D point cloud geometry, all the cloud points are projected into a first cubic Voronoï (namely the tree root). Next, the basic quantization structure is repeated iteratively from each point of the cloud. At each step of the greedy process, the cube to partition and the adapted partitioning method, have to be chosen according to a rate-distortion criterion. Exactly a Lagrangian optimization is employed, the

multiplier associated to each Voronoï (namely tree node) is computed, such as when a node is partitioned, the increase in rate is calculated in terms of tree coding cost (here: the encoding cost estimation based on the LZIP algorithm of the nodes is employed), and the decrease in distortion in terms of geometric distortion.

To assess objectively this distortion, we put in competition 2 metrics [15]:

Point-to-point metric $D_{P2Point}$, as the square distance between the lattice point Y_i to all assigned cloud points X inside the corresponding Voronoï cell C_i. So the total P2Point distortion associated to the cell C_i is:

$$D_{P2Point} = \sum_{X \in C_i} d(X, Y_i) \tag{1}$$

where d is square distance between the original point X and the representing point Y_i of the C_i i-th Voronoï cell.

Point-to-plane metric [16] $D_{P2Plane}$, as the square distance between the lattice point Y_i to normals $\overrightarrow{N}_X$ to the plane associated to the cloud points X inside the corresponding cell C_i. So the total P2Plane distortion associated to the cell C_i is:

$$D_{P2Plane} = \sum_{X \in C_i} || \overrightarrow{V_d}(X, Y_i) \cdot \overrightarrow{N}_X || \tag{2}$$

To compute the P2Plane, we need an unit normal vector $\overrightarrow{N}_X$ for each X which is estimated using the least-square plane fitting method described in [17]. Hence the distance vector between X and Y_i is then calculated as $\overrightarrow{V_d}(X, Y_i)$ which is projected on the normal vector direction to finally compute the P2Plane distance.

At each loop of the growing tree process, the best choice has to be done between the node (or leaf) to partition and according to 2 partitioning schemes. A Lagrangian optimization is then employed at this level.

IV. Quantization Scheme Decision

Similar to the classic image coding, the geometry compression of point cloud is to convey the geometry information with minimum possible distortion within constraint bit rate. To achieve more efficient compression, a crucial issue is to determine the partitioning scheme that reaches the best balance between rate and distortion. This can be cast into the classical rate-distortion framework, which is formulated as pursuing the best quality under the limitation of a given target rate [18], [19].

$$\min_{\{s_j\}} D(s_j) \text{ subject to } R(s_j) \leq R_T \tag{3}$$

where R_T denotes the target rate, $D(s_j)$ and $R(s_j)$ indicate the distortion and the rate associated with the partitioning scheme s_{j-th} at the $j-th$ loop of the growing process. This constrained problem can be converted into the Lagrangian rate-distortion optimization problem as,

$$\min_{\{s_j\}} L(s_j) \text{ where } L(s_j) = D(s_j) + \lambda \cdot R(s_j) \tag{4}$$

where λ is the Lagrange multiplier which controls the trade-off between rate and distortion. In our context, the BFOS algorithm [20] could be adapted and considered as using a Lagrangian optimization approach that aims to minimize L. The rate-distortion optimization is then performed locally. For a given loop of the tree growing process, it aims at choosing among all the possible partitioned nodes n_i (or leaves), the one which offers the maximal decrease in distortion (ΔD), and the minimal increase in rate (ΔR), so the maximal lambda λ is given by:

$$\max_{\{s_j\}} \lambda(n_i) \text{ where } \lambda(n_i) = \frac{\Delta D}{\Delta R} = \frac{D_{i+1}(s_j) - D_i(s_j)}{R_{i+1}(s_j) - R_i(s_j)} \tag{5}$$

The codebook process is stopped for instance when the bit budget is reached, and the final tree-structure is then unbalanced. Each occupied leaf node corresponds to a point output. The location of the output point is set to the average value of the PC points contained within the leaf node.

V. Experimental Results

A. Experimental setup

For the experimental results, we used our multiscale TSPLVQ to encode 3D point clouds. We compared our approach, based on different tree partitioning strategies: for the first case only 2x2x2 partitioning scheme (octree) is used, for the second case only 3x3x3 scheme, and for the third case the two partitioning (hybrid) schemes (2x2x2 vs. 3x3x3) have been put in competition, against the MPEG reference test model [21]. We selected three static point clouds from people object dataset: Soldier, Long dress and Loot suggested by MPEG-3DG group [22] as a reference test dataset. For fair comparison, we set up our method to obtain decoded point clouds with the same number of points as the point cloud decoded with MPEG-PCC model. The performance is first measured in terms of MSE metric given in [23].

B. Results

A selection of results is presented in I. The analysis of the MSE metric of $LongDress$ and $Loot$ shows that our hybrid method, relied on point-to-point distortion optimization, can outperform the reference method, while the result concerning $Soldier$ is greater than the result obtained by the reference method. However, we could obtain obviously better results for the $Soldier$ when we use only 2x2x2 partitioning. In the case of $LongDress$ PC, we note that MPEG model outperforms our approach with only 2x2x2 partitioning and we explain this phenomenon by the existing «holes» in some smooth areas of the rendered result. It is worth to note that the 3 schemes lead to different performance. The 2x2x2 partitioning is more progressive and requiring more loops for the same number of points compared to the 2 other schemes while the 3x3x3 partitioning scheme is faster. Thus, the hybrid scheme is proposed to control the partitioning effects. The benefits of the proposed method over the reference model in terms of visual quality are clearly visible and in line with

(a) (b) (c) (d) (e) (f) (g)

Fig. 3. Rendering results for compressed point clouds: (a) original point cloud with 1 089 091 points, (b) using MPEG lossy PCC, (c), (d), (e), (f) using our approach with same number of points $\approx$ 20 000 as (b) but with different schemes: (c) only 2x2x2 partitioning, (d) only 3x3x3 partitioning, (e) hybrid partitioning based on Point-to-Plane distortion, (f) hybrid partitioning based on Point-to-Point distortion and (g) hybrid partitioning with $\approx$ 30 000 points.

the objective evaluation results. For instance, for the *Soldier* point cloud, the MPEG reference method cannot show details in the complex and smooth regions inducing a great loss of visual details (see 3(b)) while with the same number of points, reasonable quality was achieved as shown in 3(c),(d) and (f). Our proposed method, using any partitioning scheme where the distortion is assessed according to point-to-plane metric, preserves more visual details in the complex areas compared to the reference method, whereas it produces less points to represent smooth surfaces as seen in 3(e). Hence our proposed adaptive method using point-to-plane distortion seems more linked with the subjective quality of the rendered decoded object compared to the reference method. Thanks to our multiscale approach, the achieved quality was clean globally and very close to the original data when we use more points as shown in 3(g).

TABLE I
COMPARISON OF SYMMETRIC MSE METRIC BETWEEN PROPOSED METHODS BASED ON POINT-TO-POINT AND POINT-TO-PLANE DISTORTION (IN **BOLD**) AND MPEG REFERENCE MODEL.

PC Images	**MPEG G-PCC**	**Our approach**		
		3x3x3	**2x2x2**	**Hybrid**
Soldier	16.48	18.72 **19.02**	11.2816 **14.32**	18.70 **14.45**
Long Dress	16.53	18.74 **19.89**	9.28 **61.08**	15.89 **17.74**
Loot	16.54	15.33 **17.65**	18.77 **10.49**	15.36 **19.13**

VI. CONCLUSION AND OUTLOOK

This paper addressed the geometric data of 3-D point cloud compression problem. We have proposed a lossy compression method based on TSLVQ. Our method innovates because it introduces two partitioning schemes (2x2x2 versus 3x3x3) for the cubic Voronoï cell, and by the use of two metrics (point-to-point vs. point-to-plane) to estimate the geometric distortion. The developed TSPLVQ algorithm is based on a multiscale approach where the optimal scheme is iteratively chosen according to rate-distortion optimization criterion. Experimental results show promising performances of TSPLVQ using the two quantization schemes with different qualities of the rendered contents, their analysis explains the TSPLVQ coding improvement in this context. The subjective evaluation showed conclusive results.

As future work, we would like to include more accurate geometric primitives (plane estimation) and other attributes (as colour) to better characterize the content of a Voronoï cell, giving rise to a more intuitive and semantically meaningful representation process, in order to better render the decoded 3D objects. This paper focuses on encoding static point clouds. For sequences of point clouds, we would need to further explore the temporal relationship between point clouds in neighboring frames.

REFERENCES

[1] G. Apostolopoulos et al. John, "The road to immersive communication," in *Proceedings of the IEEE 100.4*. IEEE, 2012, pp. 974–990.

[2] S. Schwar et al., "Emerging mpeg standards for point cloud compression," *EEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2019.

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

[4] "3d is here: Point cloud library (pcl)," in *International Conference on Robotics and Automation, Shanghai*. IEEE, 2011.

[5] G. Shen, W.-S. Kim, S. K. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *In Picture Coding Symposium (PCS)*. IEEE, 2010, pp. 566–569.

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

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

[8] R.A. Cohen, D. Tian, and V. Vetro, "Attribute compression for sparse point clouds using graph transforms," in *International Conference on ImageProcessing (ICIP)*. IEEE, 2016, pp. 1374–1378.

[9] 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.

[10] X. Yiqun, W. Shanshe, Z. Xinfeng, W. Shiqi, Z. Nan, M. Siwei, and G. Wen, "Rate-distortion optimized scan for point cloud color compression," in *Visual Communications and Image Processing (VCIP)*. IEEE, 2017.

[11] V. Ricordel and C. Labit, "Tree-structured lattice vector quantization," in *European Signal Processing Conference(EUSIPCO)*, 1996.
[12] V. Ricordel and C. Labit, "Vector quantization by packing of embedded truncated lattices," in *International Conference on Image Processing (ICIP)*. IEEE, 1995.
[13] E.A. Riskin and R.M Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizers," *IEEE Transactions on Signal Processing*, vol. 39, no.11, pp. 2500–250, 1991.
[14] J. H. Conway and N. J. A. Sloane, "Sphere packings lattices and groups," *Springer-Verlag*, 1993.
[15] D. Tian, H. Ochimizu, C. Feng, R. A. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in *International Conference on Image Processing (ICIP)*. IEEE, 2017.
[16] A. Javaheri, C. Brites, F.D. Pereira, and J. Ascenso, "Subjective and objective quality evaluation of 3d point cloud denoising algorithms," in *International Conference on Multimedia Expo Workshops (ICMEW)*. IEEE, 2017, pp. 1–6.
[17] P. Podulka, "Selection of reference palen by the least squares fitting methods," *Advances in Science and Technology Research Journal*, vol. 10(30), pp. 164–175, 2016.
[18] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 74–90, 1998.
[19] X. Zhang, S. Wang, K. Gu, W. Lin, S. Ma, and W. Gao, "Just-noticeable difference-based perceptual optimization for jpeg compression," *IEEE Signal Processing Letters*, vol. 24, pp. 96–100, 2017.
[20] E. Riskin, "Optimal bit allocation via the generalized bfos algorithm," *IEEE Transactions on Information Theory*, 1991.
[21] K. Mammou, P. A. Chou, D. Flynn, M. KrivokuĂĞa, O. Nakagami, and Sugio T., "G-pcc codec description v1," in *ISO/IEC JTC1/SC29/WG11 N18015*. MPEG, October 2018, Macau, China.
[22] C. Tulvan, R. Mekuria, and Li Z., "Draft dataset for point cloud coding (pcc)," in *ISO/IEC JTC1/SC29/WG11N16333*. MPEG, June 2016, Geneva.
[23] Julien Ricard, "Pcc ce 0.3 on new metrics," in *ISO/IEC JTC1/SC29/WG11 N18032*. MPEG, October 2018, Macau, China.