ADAPTIVE BLOCK PARTITIONING OF POINT CLOUDS FOR VIDEO-BASED COLOR COMPRESSION

Camilo Dorea and Ricardo L. de Queiroz

Dept. of Computer Science
University of Brasilia, DF, Brazil
Emails: camilodorea@unb.br, queiroz@ieee.org

ABSTRACT

Point cloud compression presents novel and challenging demands. Video-based solutions can harness existing video codec technology for efficient compression, however, the cloud's 3D geometry and attributes must be presented within a compatible, regular 2D grid. We propose a method for generating equally-sized, square-shaped video blocks containing point cloud color information. Our video blocks may be readily tiled into an image for efficient color compression through a traditional video codec. The focus is on the development of a computationally simple, voxel-to-image projection methodology. Results provide evidence of competitive performance, with average PSNR gains of 0.68dB and attribute (color) bitrate savings of -15.10%, with respect to MPEG's G-PCC (TMC13 v5.1).

***Index Terms*—** Video-based point cloud compression, compression of color attributes.

1. INTRODUCTION

Among the novel 3D representations for imaging systems, point clouds (PCs) constitute a geometrically simple yet versatile alternative, offering relative independence from acquisition and rendering procedures. They are a set of points $\{p = (p_x, p_y, p_z)\}$ in 3D space with associated attributes, such as color or reflectancy, for example. In voxelized clouds, the points assume integer coordinate values on a regular 3D grid. Points within such a grid are called voxels and may be occupied or not.

The large amount of data generally involved in this 3D representation requests compression. The Moving Pictures Experts Group (MPEG) has recently launched efforts to exceed compression capability of previous work and standardize technology [1]. Therein, two approaches are currently being discussed: geometry- and video-based point cloud compression (G-PCC and V-PCC, respectively).

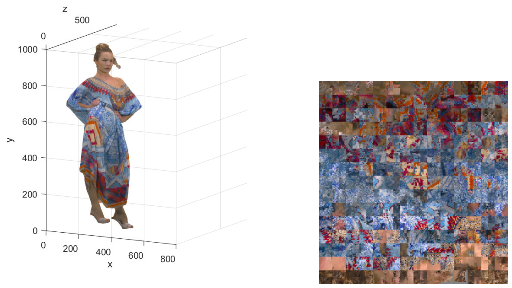
G-PCC addresses compression of PCs with static object content as well as dynamically acquired scenes. Geometry, or point positions, are coded first, generally through an octree coding mechanism [2]. Both lossless and lossy coding are contemplated. Attribute coding depends on the decoded geometry and, typically, uses the Region Adaptive Hierarchical Transform (RAHT) [3] or an interpolation-based hierarchical nearest-neighbor prediction with an update/lifting step (Lifting Transform).

V-PCC focuses on compression of PC sequences depicting dynamic objects. One of its assets is leveraging existing video coding technology, such as H.265/MPEG-H HEVC [4], to achieve PC compression efficiency. Essentially, geometry and attribute (texture) information are projected onto 2D grids to form video sequences who are then compressed through a traditional video codec [5]. Projection aims at maintaining spatial and temporal coherency while respecting reconstruction constraints. In the process, the PC is decomposed into patches whose points are associated to the planes of the PC's bounding box. A padding strategy is adopted to best fit extracted patches onto a 2D grid, called an atlas, while minimizing unused space.

Our purpose is to investigate an alternative, different from that adopted by V-PCC, for mapping voxels to images in order to use in video-based PC coders. The proposal focuses on the advancement of the mapping technique rather than the development of a complete coder. In spite of room for improvements, our projective video-based approach has shown to be competitive when coding PC color attributes.

Specifically, the PC is quickly partitioned into fixed-size voxel sets. Each voxel in the set is associated to a point on a square-shaped grid pertaining to its best-fitting plane. Such squares define *video blocks* which, internally, respect spatial coherency of color and, due to their standard size and shape, may be easily assembled into 2D images for video coding. In our current proposal, geometry is losslessly encoded and a reverse procedure is used to reconstruct decoded PC color. The partitioning and projection procedures are computationally simple and our coding efficiency compares favorably to that of RAHT, adopted by state-of-the-art G-PCC as a color compressor and, similarly to our implementation, also tested under lossless geometry for static object PC content [6].

This work was partially supported by CNPq grant 301647/2018-6 and FAPDF.



(a) (b)

Fig. 1: (a) An example point cloud *Longdress* [7] and (b) corresponding image on 2D grid composed of extracted video blocks.

The general, fixed-size, PC partitioning scheme and the generation of video blocks is described in Sec. 2. An adaptive partitioning improvement is detailed in Sec. 3. Experimental results and conclusions are discussed in Secs. 4 and 5.

2. VIDEO BLOCKS

We describe the PC partitioning algorithm for generating geometrically compact, fixed-size voxel sets in Sec. 2.1. Next, we define the procedure for mapping voxel color onto a square-shaped regular 2D grid, forming video blocks. Resulting video blocks are tiled into an image for subsequent video coding, as illustrated in Fig. 1.

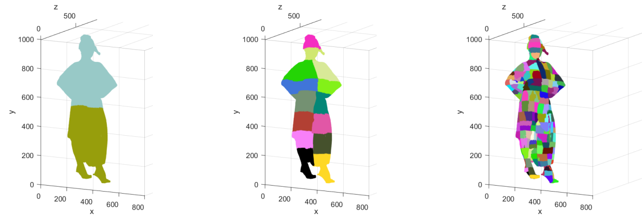
2.1. Initial partition

Partitioning of the PC is efficiently achieved through successive bi-sectioning of an initial voxel set P. At each iteration, the procedure devides the previous voxel set into two new sets, P' and P'', using as seperation criterion the median voxel position along the widest axis of occupied voxel space. For the voxel space whose elements' coordinates are listed in column vectors $\bar{p}_x$, $\bar{p}_y$ and $\bar{p}_z$, the widest axis is given by

$$i^* = \underset{i \in \{x,y,z\}}{\arg\max} (\ \|\bar{p}_i\|_{max} - \|\bar{p}_i\|_{min}\ )$$

where $\|\cdot\|_{max}$ and $\|\cdot\|_{min}$ denote, respectively, maximum and minimum values among vector elements. P' and P'' are formed by all voxels p such that $\{p \mid p_{i^*} \leq m_{i^*}\}$ and $\{p \mid p_{i^*} > m_{i^*}\}$, respectively, where m_{i^*} is the median of $\bar{p}_{i^*}$.

As an example, the voxels of Fig. 1(a) are split into two sets: one with elements whose z coordinate value is above the median z of the initial occupied space and another with elements below the median. The result is two equally-sized voxel sets as shown in Fig. 2(a). Partitioning is terminated when a minimum set size N^2 is reached. The inital voxel set may be zero-padded to guarantee equally-sized partitioning results. A final voxel partition is depicted in Fig. 2(c).



(a) (b) (c)

Fig. 2: (a) Result from the 1st iteration of PC partitioning of *Longdress*, (b) the 10th iteration and (c) the final partition with 256 voxel sets each of size $N^2 = 4096$ (in pseudo colors).

2.2. Best-fitting quadrangle

Each voxel set resulting from partitioning is used to form a corresponding video block of dimensions $N \times N$ on a regular 2D grid. Geometrical compactness of the voxel set and one-to-one correspondence with respect to video block elements then allow for coherent mapping of color attributes as described next.

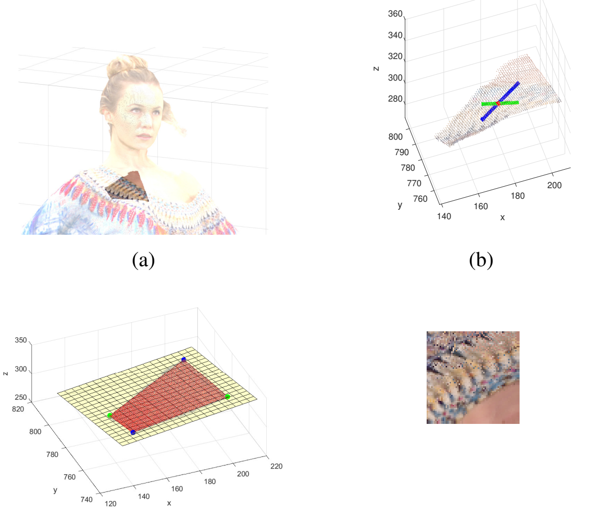
First, the eigen-decomposition of the voxel set covariance matrix is used to define a best-fitting plane. Let Σ be the covariance matrix of voxel set $P = [\bar{p}_x\ \bar{p}_y\ \bar{p}_z]$. Covariance matrix eigen-vectors are $V = [\bar{v}_1\ \bar{v}_2\ \bar{v}_3]$, listed in descending order of eigen-value magnitude. The best-fitting plane is defined by its normal vector $\bar{v}_3$ corresponding to the eigenvector with smallest eigen-value magnitude. Figs. 3(a)-(c) provide illustrations.

Next, anchor points corresponding to voxels located at coordinate extrema along the axes of largest variances, indicated by $\bar{v}_1$ and $\bar{v}_2$, are orthogonally projected onto the best-fitting plane. More precisely, anchor points are defined in terms of the basis-transformed coordinates $[\bar{p}_1\ \bar{p}_2\ \bar{p}_3] = PV$ as

$$\begin{aligned} p^A &= \{p \mid p_1 = \|\bar{p}_1\|_{max}\}, \\ p^B &= \{p \mid p_1 = \|\bar{p}_1\|_{min}\}, \\ p^C &= \{p \mid p_2 = \|\bar{p}_2\|_{max}\} \text{ and} \\ p^D &= \{p \mid p_2 = \|\bar{p}_2\|_{min}\}. \end{aligned}$$

The projected anchor points define the vertices of a quadrangle on the plane, as exemplified in Fig. 3(c). The quadrangle edges are linearly sampled into N points and a total of $N \times N$ samples are formed through linear interpolations between points on opposing edges.

Lastly, for each voxel in the given set, a nearest neighbor is identified among the samples of the quadrangle. The color attributes of a voxel are thus assigned to a nearest point among the quadrangle samples. A one-to-one correspondence is guaranteed by removing a matched pair from subsequent nearest neighbor search. Once voxel colors have been assigned, the linearly-spaced $N \times N$ quadrangle points are re-shaped into a square, regularly-spaced, 2D grid, or video block, as represented in Fig. 3(d). The collection of



(a) (b) (c) (d)

Fig. 3: (a) Detail crop of *Longdress* highlighting a voxel set, (b) scaled eigen-vectors for the voxel set ($\bar{v}_1$, $\bar{v}_2$ and $\bar{v}_3$ in blue, green and red, respectively), (c) best-fitting plane (in yellow) and anchor points (in blue and green) defining a quadrangle and its samples (in red) and (d) corresponding video block.

video blocks is easily tiled into a rectangular image with zero-padding, if necessary, as pictured in Fig. 1(b).

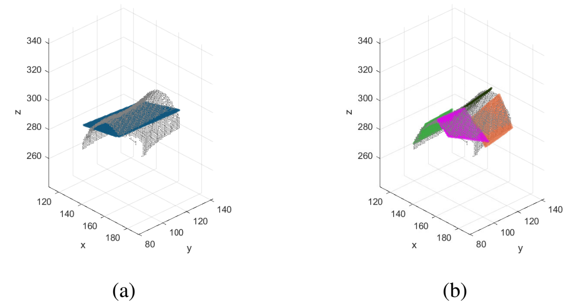
3. ADAPTIVE PARTITIONING

The initial partitioning procedure described in Sec. 2, although fast, may not guarantee sufficient geometric compactness of a resulting voxel set. A set may contain disconnected components, for example, which distort voxel to video-block mapping and compromise consistency of color distribution. Furthermore, a fixed N^2 size may be inadequate for certain geometric structures, in particular, those of greater curvature. See, as an example, Fig. 4(a). For these cases an adaptive partitioning step is applied to improve color consistency within video blocks.

Adaptive partitioning recursively sub-divides voxel sets deemed as insufficiently flat into 4 sub-sets. Each recursion consists of two voxel set bi-section operations as explained in Sec. 2. Flatness f is measured as a relative difference between eigen-vector magnitudes

$$f = (\lambda_2 - \lambda_3)/\lambda_3$$

where λ_2 and λ_3 are the second-smallest and smallest magnitude eigen-values of covariance Σ. Voxel sets whose flatness is below a threshold are recursively partitioned into 4 sub-sets until a minimum n^2 size is reached. Upon termination, the resulting sets are mapped to corresponding quadrangles, now on better-fitting planes, as defined in Sec. 2.2. Adaptive partitioning reduces plane-fitting errors, as shown in Fig. 4(b).



(a) (b)

Fig. 4: (a) Example of a voxel set with significant curvature and its quadrangle (in pseudo color) on the best-fitting plane and (b) best-fitting quadrangles associated to voxel set sub-divisions resulting from an adaptive partition recursion.

Video blocks of dimension $N \times N$are then assembled from the quadrangles corresponding to the voxel set sub-divisions.

Assuming that geometry used for encoding is available for decoding, i.e., through lossless geometry transmission, the partitioning procedures may be replicated and the decoded video block color attributes mapped back to their voxel positions.

4. EXPERIMENTAL RESULTS

Our tests were conducted on the publicly available PC data sets *Longdress*, *Loot*, *Red and Black* and *Soldier*, all with 10 bit geometry precision (*vox10*) [7], as well as *Egyptian Mask* with 12 bit (*vox12*). Frames 1300, 1200, 1550 and 690 were used, respectively, for the first four data sets as recommended in [8]. A conversion from RGB space to YUV space is conducted using ITU-R BT.709.

All experiments used minimum sizes of $N = 64$ and $n = 4$ for initial and adaptive partitioning, respectively, and flatness threshold of 1.0. Although agnostic to video codec, our results are based on H.265/MPEG-H HEVC [4] from the x265 version 3.0 implementation [9]. Encoding parameters are preset to placebo option defined therein and QP points are from range $\{22, 27, ..., 47\}$. Note that choice of 64×64 video block dimensions correspond to the maximum dimensions of the coding tree units of HEVC.

Color compression results are reported in terms of bitrate, specified in bits per input point [bpip], for the YUV color channels. Distortion is calculated from the MSE of the Y-channel with MPEG metric software [8] and reported as PSNR-Y. As set forth in Sec. 1, our results are contrasted against the color compression results of MPEG's G-PCC using the RAHT color compressor. These results are published in [6] and correspond to software implementation TMC13 version 5.1. Geometry is losslessly encoded and its bitrate is not considered.

Rate-distortion (R-D) curves for *Longdress* are presented in Fig. 5. As with other data sets, PSNR-Y differences be-

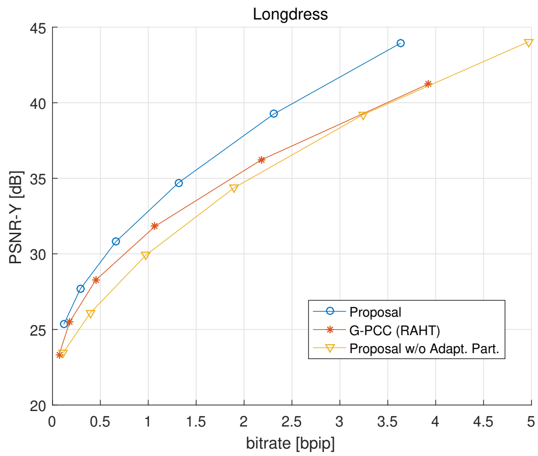


Fig. 5: R-D performance comparison between the proposal and G-PCC's RAHT color compressor for *Longdress*, *vox10*, frame 1300. Also shown is proposal performance without Adaptive Partitioning.

tween the proposal and G-PCC (RAHT) are more significant at higher bitrates. Visual quality improvements are also more perceivable at these rates. Also presented in the figure is the R-D curve of our proposal without the improvements of Adaptive Partitioning, introduced in Sec. 3. In this case, all voxel partitions are of size N^2 regardless of flatness and results are inferior to those of our complete proposal as well as G-PCC (RAHT).

Table 1 presents Bjontegaard delta (BD) [10] bitrate savings or the equivalent PSNR-Y gains for all tested sequences. Our proposal outperforms G-PCC (RAHT) for all the data sets except for slightly inferior BD PSNR-Y in the case of *Egyptian Mask*. Average BD PSNR-Y gains and BD bitrate savings are 0.68dB and -15.10%, respectively.

Table 1: BD PSNR-Y gains and bitrate savings of proposal relative to G-PCC (RAHT) codec for various data sets.

Data Set	BD PSNR-Y [dB]	BD bitrate [%]
Longdress	1.42	-25.60
Loot	0.25	-6.54
Soldier	0.40	-9.58
Red and Black	1.55	-33.61
Egyptian Mask	-0.21	-0.18
Average	0.68	-15.10

5. CONCLUSIONS

We proposed a video-based solution for PC color attribute compression. Our proposal relies on simple yet efficient partitioning and 2D grid mapping procedures for generating video blocks. These are readily tiled into an image and submitted for compression through traditional video codecs. Experimental results show the proposal can outperform MPEG's G-PCC RAHT color compressor for the tested data set.

Future works include the introduction of lossy geometry coding within the framework; further reduction of mapping distortions for improvement of color coherency within video blocks and extension of the proposal to compression of dynamic PCs, exploiting temporal correlations among frames of video blocks.

6. REFERENCES

[1] S. Schwarz, *et al.*, "Emerging MPEG standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Syst.*, vol. 9, no. 1, Mar. 2019.

[2] G-PCC, "G-PCC codec description v3," in *ISO/IEC JTC1/SC29/WG11 N18486*, Geneva, Switzerland, Mar. 2019.

[3] R. L. de Queiroz and P. A. Chou, "Compression of 3D point clouds using a region-adaptive hierarchical transform," *IEEE Trans. Image Process.*, vol. 25, no. 8, Aug. 2016.

[4] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Overview of the high efficiency video coding HEVC standard," *Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, Dec. 2012.

[5] V-PCC, "V-PCC codec description," in *ISO/IEC JTC1/SC29/WG11 N18674*, Gothenburg, Sweden, Aug. 2019.

[6] G-PCC, "G-PCC performance evaluation and anchor results," in *ISO/IEC JTC1/SC29/WG11 w18476*, Geneva, Switzerland, Mar. 2019.

[7] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i voxelized full bodies - a voxelized point cloud dataset," in *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m40059/M74006*, Geneva, Switzerland, Jan. 2017.

[8] 3DG, "Common test conditions point cloud compression," in *ISO/IEC JTC1/SC29/WG11 w18474*, Geneva, Switzerland, Mar. 2019.

[9] videoLAN x265, *http://ftp.videolan.org/pub/videolan/x265/*.

[10] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," in *doc VCEG-M33, ITU-T SG16/Q6*, Austin, TX, USA, Apr. 2001.