OCCUPANCY-MAP-BASED RATE DISTORTION OPTIMIZATION FOR VIDEO-BASED POINT CLOUD COMPRESSION

Li Li[*] *Zhu Li*[*‡] *Shan Liu*[†] *Houqiang Li*[⋆]

[*] University of Missouri Kansas City
[†] Tencent America
[⋆] University of Science and Technology of China
[‡] Peng Cheng Lab

ABSTRACT

The state-of-the-art video-based point cloud compression scheme projects the 3D point cloud to 2D patch by patch and organizes the patches into frames to compress them using the efficient video compression scheme. Such a scheme shows a good trade-off between the number of points projected and the video continuity to utilize the video compression scheme. However, some unoccupied pixels between different patches are compressed using almost the same quality with the occupied pixels, which will lead to the waste of lots of bits since the unoccupied pixels are useless for the reconstructed point cloud. In this paper, we propose to consider only the rate instead of the rate distortion cost for the unoccupied pixels during the rate distortion optimization process. The proposed scheme can be applied to both the geometry and attribute frames. The experimental results show that the proposed algorithm can achieve an average of 11.9% and 15.4% bitrate savings for the geometry and attribute, respectively.

Index Terms— Occupancy map, Point cloud compression, Rate distortion optimization, Sample adaptive offset, Video-based point cloud compression

1. INTRODUCTION

Point cloud is a set of 3D points with each point associated with some attributes such as color, reflection, and so on. As point cloud has the capability to render an object or scene, the point cloud can be used in many scenarios [1] such as 3D immersive telepresence [2], virtual reality, and some other applications. However, the large size for the point cloud especially the dynamic point cloud (DPC) is preventing the usage of such a promising media format. For example, for a DPC with 300 frames, each frame with one million points, each point with 30 and 24 bits to represent the geometry and attribute, the original size can be as large as 2GBytes. Therefore, there is an urgent need to compress the DPC efficiently.

In addition to its large size, the isolated points also make the point cloud difficult to be compressed due to the irregular patterns. The current methods used for DPC compression can be roughly divided into two groups: the 3D-based method and the video-based method. As its name implies, the 3D-based method compresses the point cloud in the original domain. Usually, the geometry of the first frame will be compressed using the octree tree [3] or binary tree combined with the quadtree [4]. Some transforms such as Graph Fourier Transform (GFT) [5] and region-adaptive hierarchical transform (RAHT) [6] or layer based prediction [7] are used to compress the attributes. For the inter frames, Queiroz and Chou [8] proposed to divide the frames into multiple cubes and perform the translational motion model based motion estimation to find the corresponding cube. Mekuria *et al.* [9] further introduced the iterative closest points instead of the translational motion model to improve the coding efficiency. However, the 3D-based schemes are unable to fully exploit the correlations among various views, which make the compression ratio unsatisfiable.

The video-based method which projects the 3D point cloud into 2D space to fully utilize the intercorrelations using the mature video compression schemes such as High Efficiency Video Coding (HEVC) [10]. For example, Schwarz *et al.* [11] and He *et al.* [12] proposed to use cylinder or cube projection to project the point cloud to a 2D video, and the 2D video is compressed using HEVC. However, such kind of methods may lose lots of points due to the points occlusion in 3D space. To solve this problem, Mammou *et al.* [7] [13] proposed to project the point cloud to 2D video patch by patch to increase the number of points projected. The projected video is then compressed using HEVC and wins the competition of the MEPG call for proposal for the dynamic point cloud compression [14]. Such a video-based method will be called as patch-based point cloud compression in the following sections. However, many unoccupied pixels of the projected video are not beneficial for the quality of the reconstructed video, which will lead to a serious waste of the bits if they are encoded in the same quality with the occupied pixels. The unoccupied pixels are even not continuous in the temporal domain, which may lead to even more bits waste.

In this paper, we propose to use the occupancy-map-based

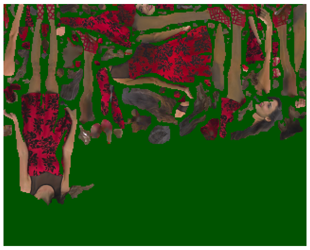


Fig. 1. Frame with unoccupied pixels set as 0 from redAndBlack, POC 16.

rate distortion optimization (RDO) to solve this problem. When the whole or just a part of the block is unoccupied, only the rate instead of the rate distortion (RD) cost of the unoccupied pixels is taken into consideration in the RDO process. Such a scheme is applied not only to the mode decision process but also to the sample adaptive offset (SAO) [15] process. Under the proposed scheme, the unoccupied pixels will be coded with very bad quality to save the bitrate, while the occupied pixels are encoded in the same quality with the original scheme to guarantee the quality of the reconstructed point cloud. As far as we can see, this is the first work trying to design specific RDO algorithm to improve the efficiency of the point cloud compression.

This paper is organized as follows. We will give a brief introduction of the patch-based point cloud compression method and the motivation of the proposed algorithm. We will describe the proposed occupancy-map-based RDO algorithm in Section 3. In Section 4, we will show the detailed experimental results. Section 5 concludes the whole paper.

2. MOTIVATION

Under the patch-based point cloud compression method, the 3D point cloud is first projected to its bounding box to generate several patches. Then all the patches are packed properly to form the geometry and attribute frames. The patch information including occupancy map, patch position in the 2D frame, and some other information is first transmitted to the decoder, and then the geometry and attribute video are encoded using HEVC. In the lossy point cloud compression, the occupancy map is losslessly encoded as 4×4 blocks to obtain a better trade-off between the bitrate and accuracy.

Fig. 1 gives a typical frame with unoccupied pixels set as 0. As we can obviously see from Fig. 1 that there are many unoccupied pixels among different patches. Except those unoccupied pixels in the bottom of the frame which are easy to be encoded using intra prediction, the unoccupied pixels among different patches may cost lots of bits. As these unoccupied pixels will have no influences on the quality of the reconstructed point cloud, the bits spent are totally useless which may seriously degrade the compression performance. In addition, under the patch-based point cloud compression scheme, the occupancy map can be encoded before the geometry and attribute videos, which makes it possible for us to use such kind of information to improve the geometry and attribute compression. Therefore, we propose to use an occupancy-map-based RDO to solve the problem of the unoccupied pixels in this paper.

3. PROPOSED ALGORITHM

In the default encoder of HEVC reference software, the encoding parameters P including mode, motion and residue of a block are determined by the RD cost J.

$$\min_{P} J = \sum_{i=1}^{N} D_i + \lambda R \tag{1}$$

where D_i is the distortion of each pixel, R is the bitrate of a block, and N is the number of pixels in a block. During the RDO process, the D_i is the square difference between the original pixel and the reconstructed pixel for a specified position i. However, such an optimization problem treats the occupied pixels and the unoccupied pixels with equal importance, which may degrade the performance of the point cloud compression significantly.

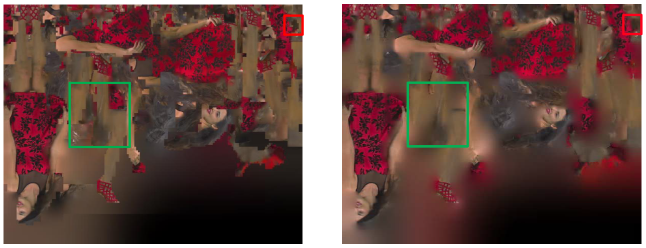
In the proposed solution, a distortion mask is added to (1) to handle the unoccupied pixels in the RDO process.

$$\min_{P} J = \sum_{i=1}^{N} D_i \times M_i + \lambda R \tag{2}$$

where M_i is 1 when the current pixel is occupied and M_i is 0 when the current pixel is unoccupied. We then apply such a formula in the intra mode decision, inter mode decision, and SAO process to obtain good performance.

For intra prediction, the mode decision process can be divided into 3 stages: rough intra mode selection, intra mode decision, intra residue quadtree decision. In the first stage, the sum of the absolute transformed difference (SATD) between the original signal and the prediction signal is used as the distortion D. The bitrate of the Luma intra prediction direction is used as the bitrate R. In this stage, the aim is to obtain several candidates which can provide more accurate prediction blocks. As the residue bits are not considered in this stage, we cannot consider only part of the pixels to calculate the distortion as the distortion can be considered as an indicator of the residue bitrate. Without considering the total residue bitrate, we may select the prediction mode candidates with too many residue bits which is not beneficial for the RD performance. In the second and third stages, as the total bitrate is taken into consideration, we apply (2) to both the mode decision and residue quadtree processes.

For inter prediction, the mode decision can be divided into two modes: $2N \times 2N$ merge/skip and other modes including $2N \times 2N$ advanced motion vector prediction mode and



(a) Occupancy-map-based RDO (b) Original RDO

Fig. 2. Reconstructed frame comparison from redAndBlack, POC 16.

all the other partitions. For the $2N \times 2N$ merge/skip mode, all the merge candidates are considered using the full RDO, which calculates the sum of the squared difference (SSD) between the original signal and the reconstructed signal as the distortion and the overall bits including the header and residue as the bitrate. Therefore, Eq. (2) is applied directly for the $2N \times 2N$ merge/skip mode. For the other modes, the RDO process can be roughly divided into two stages: the integer/fractional motion estimation (ME) or merge to determine the prediction and the residue determination. In the integer ME process, the sum of absolute difference (SAD) between the original signal and prediction signal is used as the prediction, and the bits of the motion vector (MV) are used as the bitrate. In the fractional ME or merge process, the SATD between the original signal and prediction signal is used as the prediction, and the bits of the MV or merge index are used as the bitrate. As in both cases, the residue bitrate is not taken into consideration, we cannot consider only part of the pixels to calculate the distortion. In the residue determination, as the full RDO is used, Eq. (2) is used in the RDO process.

In the SAO process, the offsets between the original pixels and the reconstructed pixels are first counted for different kinds of edge offsets and the band offset. Then for each CTU, we will select from these kinds of offsets using RDO to determine the SAO type. From the above descriptions, we know that the offset for different types of SAO is the key to influence the RDO process. However, during the mode decision process, the occupancy-map based RDO encodes some of the unoccupied pixels with very small bitrate and bad quality. If we statistic the offsets among all the pixels, the unoccupied pixels will take the majority of the offsets since their distortions are much larger. In this way, the offset for each type of SAO will be quite inaccurate for the occupied pixels but quite accurate for the unoccupied pixels, which may lead to serious bits cost to recover the unoccupied pixels while degrading the quality of the occupied pixels during the RDO process for each CTU. Therefore, we do not explicitly change the RDO process of SAO but change the statistic process to get the offset. We will calculate the offsets according to only the occupied pixels between the original signal and the reconstructed signal. The RDO process will then automatically choose to use the offset for the occupied blocks while not to use the offset for the unoccupied blocks.

Fig. 2 shows the reconstructed frame comparison between the occupancy-map-based RDO and the original RDO. From Fig. 2 and Fig. 1, we can see that most occupied pixels are kept the same, which means the reconstructed point clouds under the proposed algorithm will be with almost the same quality as the original RDO. From the red rectangles in Fig. 2, we can see that some of the occupied pixels can be reconstructed with better quality since we can focus on finding a more suitable MV for the occupied pixels. From Fig. 2 and Fig. 1, we can also see that many unoccupied pixels are encoded with very bad quality from the green rectangles. Some of them are just copied directly from the temporal co-located position to save the bitrate. Since they will not have any influences on the reconstructed quality of the point cloud, saving some bitrate is beneficial for the overall performance.

4. EXPERIMENTAL RESULTS

We implement the proposed algorithm in the video-based point cloud compression (V-PCC) reference software [16] and the corresponding HEVC reference software [17] to compare with the V-PCC anchor. We test all the five DPCs defined in the V-PCC common test condition [18]. We test both the lossy geometry lossy attribute in all intra and random access cases to show the benefits of the proposed algorithm in intra and inter cases. We test all the DPCs for 32 frames, which can be a good representative of the whole sequence, to save some encoding time. We also follow the rate points defined in the V-PCC CTC and the BD-rate is employed for comparisons of various algorithms. For the BD-rate of the geometry, we report the results with both the point-to-point error (D1) and point-to-plane error (D2) as the quality metrics. For the attributes, the results for Luma, Cb, and Cr are provided. Both the encoder and decoder complexities of the V-PCC and HEVC reference softwares are provided.

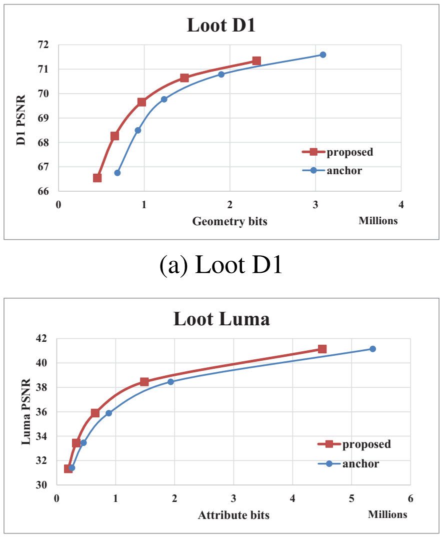
Table 1 shows the performance of the proposed occupancy-map-based RDO compared with the V-PCC anchor in random access case. From table 1, we can see that the proposed occupancy-map-based RDO can lead to an average of 10.9% and 11.3% performance improvements compared with the V-PCC anchor for the D1 and D2, respectively. We can also see that the proposed algorithm can save averagely 15.0%, 9.2%, and 10.4% bitrates for the Luma, Cb and Cr, respectively. The proposed algorithm obviously demonstrates that the proposed algorithm can lead to significant bitrate savings by ignoring the distortions of the unoccupied pixels. Sometimes, the proposed algorithm can also lead to slightly better predictions for the occupied pixels since only the distortions of the occupied pixels are considered when searching through the candidates of the merge candidate list. We also show two example R-D curves in random access case for both the ge-

Table 1. Performance of the proposed occupancy-map-based RDO compared with the V-PCC anchor in random access case

Test	Geom.BD-Rate		Attr.BD-Rate		
point cloud	D1	D2	Luma	Cb	Cr
Loot	–16.3%	–16.4%	–24.3%	–18.2%	–19.3%
RAB	–6.6%	–7.2%	–12.2%	–9.8%	–12.3%
Solider	–15.8%	–16.0%	–16.8%	–9.4%	–9.0%
Queen	–13.4%	–13.2%	–15.7%	–11.2%	–10.5%
LD	–7.5%	–7.8%	–7.9%	–7.7%	–7.2%
Avg.	–11.9%	–12.1%	–15.4%	–11.3%	–11.7%
Enc. self			101%		
Dec. self			88%		
Enc. child			99%		
Dec. child			88%		

Table 2. Performance of the proposed occupancy-map-based RDO compared with the V-PCC anchor in all intra case

Test	Geom.BD-Rate		Attr.BD-Rate		
point cloud	D1	D2	Luma	Cb	Cr
Loot	–3.4%	–3.5%	–1.4%	–0.5%	–0.9%
RAB	–2.7%	–3.1%	–1.1%	–0.9%	–1.4%
Solider	–2.9%	–3.2%	–1.1%	0.7%	1.2%
Queen	–2.6%	–2.5%	–1.2%	–1.3%	–2.0%
LD	–2.7%	–2.9%	–0.7%	–0.7%	–0.8%
Avg.	–2.9%	–3.0%	–1.1%	–0.5%	–0.8%
Enc. self			101%		
Dec. self			94%		
Enc. child			98%		
Dec. child			88%		



(a) Loot D1

(b) Loot Luma

Fig. 3. Typical R-D curves for geometry and attribute.

ometry and attribute as shown in Fig. 3. From Fig. 3, we can also obviously see that the proposed algorithm can bring very significant bitrate savings for both the geometry and attribute.

The proposed algorithm will not have significant influences on the V-PCC encoder and decoder themselves since we only output the occupancy map information from the encoder. However, for the HEVC encoder, the proposed algorithm can lead to significant time savings for the encoder and decoder since the motion estimation and motion compensation processes are both significantly simplified. For the encoder, there is no need to search over too many merge candidates and MVs especially for the unoccupied pixels thus the encoding complexity can be reduced. For the decoder, more blocks will choose zero motion vectors, which lead to fewer interpolation operations and save the decoding complexity.

Table 2 shows the performance improvement of the proposed occupancy-map-based RDO compared with the V-PCC anchor in all intra case. From Table 2, we can see that the proposed occupancy-map-based RDO algorithm can achieve an average of 1.9% and 2.1% performance improvements for the D1 and D2, respectively. The proposed algorithm can achieve averagely 0.9%, 0.2%, and 0.3% performance improvements for the Luma, Cb and Cr, respectively. The performance of the proposed algorithm is quite obvious but not as significant as the inter case. Since we cannot change the rough intra prediction direction process as mentioned in Section 3, what we have mainly changed is only the final intra prediction direction and residue quadtree. Compared with the inter case where we can influence the MV in the merge candidates and the partition determination, the changes of the proposed algorithm in intra are much smaller.

5. CONCLUSION

In this paper, to prevent the wasting of the bitrate on the unoccupied pixels in the video-based point cloud compression (V-PCC) scheme, we propose to consider only the rate instead of the rate distortion cost for the unoccupied pixels during the rate distortion optimization process. The proposed idea is applied to intra prediction, inter prediction and sample adaptive offset processes. The proposed algorithm is implemented in the V-PCC and the corresponding High Efficiency Video Coding reference softwares. The experimental results show that the proposed algorithm can lead to averagely 11.9% and 15.4% bitrate savings for the geometry and attribute, respectively. In the future, we will further consider the blocks in the patch boundary with part of blocks occupied while the other part of blocks unoccupied to further improve the performance.

Acknowledgement

This work was supported in part by 973 Program under Contract 2015CB351803, and a gift grant from Tencent.

References

[1] Christian Tulvan, Rufael Mekuria, Zhu Li, and Sebastien Laserre, "Use cases for point cloud compression (PCC)," ISO/IEC JTC1/SC29/WG11 MPEG2015/N16331, Geneva, CH, Jun. 2016.

[2] H. Fuchs, A. State, and J. Bazin, "Immersive 3d telepresence," *Computer*, vol. 47, no. 7, pp. 46–52, July 2014.

[3] Ruwen Schnabel and Reinhard Klein, "Octree-based point-cloud compression.," *Spbg*, vol. 6, pp. 111–120, 2006.

[4] Birendra Kathariya, Li Li, Zhu Li, Jose Alvarez, and Jianle Chen, "Scalable point cloud geometry coding with binary tree embedded quadtree," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[5] C. Zhang, D. Florncio, and C. Loop, "Point cloud attribute compression with graph transform," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 2066–2070.

[6] 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.

[7] Khaled Mammou, Alexis M. Tourapis, David Singer, and Yeping Su, "Video-based and Hierarchical Approaches Point Cloud Compression," Document ISO/IEC JTC1/SC29/WG11 m41649, Macau, China, Oct. 2017.

[8] R. L. de Queiroz and P. A. Chou, "Motion-compensated compression of dynamic voxelized point clouds," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3886–3895, Aug 2017.

[9] R. Mekuria, K. Blom, and P. Cesar, "Design, implementation, and evaluation of a point cloud codec for tele-immersive video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828–842, April 2017.

[10] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[11] S. Schwarz, M. M. Hannuksela, V. Fakour-Sevom, N. Sheiki-Pour, V. Malamalvadakital, and A. Aminlou, "Nokias response to CfP for Point Cloud Compression (Category 2)," Document ISO/IEC JTC1/SC29/WG11 m41779, Macau, China, Oct. 2017.

[12] L. He, W. Zhu, and Y. Xu, "Best-effort projection based attribute compression for 3d point cloud," in *2017 23rd Asia-Pacific Conference on Communications (APCC)*, Dec 2017, pp. 1–6.

[13] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al., "Emerging mpeg standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2018.

[14] Marius Preda, "Report on PCC CfP answers," Document ISO/IEC JTC1/SC29/WG11 w17251, Macau, China, Oct. 2017.

[15] Chih-Ming Fu, Elena Alshina, Alexander Alshin, Yu-Wen Huang, Ching-Yeh Chen, Chia-Yang Tsai, Chih-Wei Hsu, Shaw-Min Lei, Jeong-Hoon Park, and Woo-Jin Han, "Sample adaptive offset in the hevc standard," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 22, no. 12, pp. 1755–1764, 2012.

[16] "Point Cloud Compression Category 2 reference software, TMC2-3.0," http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc2.git, Accessed: 2018.

[17] "High Efficiency Video Coding test model, HM-16.18+SCM8.7," https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/, Accessed: 2018.

[18] Sebastian Schwarz, Galle Martin-Cocher, David Flynn, and Madhukar Budagavi, "Common test conditions for point cloud compression," Document ISO/IEC JTC1/SC29/WG11 w17766, Ljubljana, Slovenia, Jul. 2018.