MPEG Video-based Point Cloud Coding based on JPEG

Tianyu Dong, So Myung Lee, and Euee S. Jang

Department of Computer and Software, Hanyang University, Seoul, Korea
Keywords: MPEG, V-PCC, JPEG, HEVC, Video Codec

ABSTRACT

In this paper, we proposed a method to design MPEG Video-based point cloud compression (V-PCC) based on JPEG. We chose JPEG for its simplicity, low computational complexity, and ubiquitous support of encoder and decoder. For performance evaluation, we compared the proposed method with the HEVC-based V-PCC reference software.

1 INTRODUCTION

Recently, we observe that new augmented reality/virtual reality (AR/VR) applications are now on the verge of becoming the next in multimedia. These new media require not only traditional 2D video, but also 3D objects. Point clouds are one of emerging 3D objects.

Usually, a point cloud is comprised of a huge amount of points where each point has multiple attributes including geometry coordinates, color, normal, and so on. For storage and transmission purposes, the need to compress point cloud data has become overwhelming in related industries.

Started with a call for proposals on point cloud compression from 2017, MPEG is now in the final stage to produce video-based point cloud compression (V-PCC) standard [1] to compress dynamic point clouds. One unique feature of V-PCC is the use of a video codec to compress point clouds after converting 3D point cloud structure into 2D patch video. Currently, V-PCC uses the HEVC codec as a base video codec. The use of video codec for PCC is found to be beneficial not only in compression efficiency, but also in the fast market penetration of V-PCC thanks to the availability of HEVC in most mobile devices.

VPCC is designed with a “codec-agnostic” approach, which means that any video codec can be used for V-PCC. For example, the V-PCC profiles under discussion include the possible use of MPEG-4 Part 10 AVC [2] and AV1 [3]. Although it is claimed to be codec agnostic, it requires further study to use other video codecs than HEVC for V-PCC.

In this paper, we proposed a JPEG-based V-PCC solution. We selected JPEG for its low computational complexity and ubiquitous support on a wide range of multimedia devices [4]. The proposed JPEG-based V-PCC provides an extreme example since its compression efficiency is expected to be worse than any existing video codecs, but the computational complexity is only better than any existing video codecs. In this paper, we present an exemplar analysis on how a video codec can be applied to V-PCC.

This paper is organized as follows. In Section 2, we explained the background of V-PCC. We discuss the design issue with the new codec and their configuration in Section 3. In Section 4, the experimental results with analysis are provided. Finally, we conclude the paper in the last section.

2 BACKGROUND

In V-PCC, the encoding process is as shown in Figure 1. The V-PCC encoder includes patch generation, patch packing, video compression and other process such as auxiliary information coding. First, the 3D geometry coordinates of an input point cloud are projected into 2D patches, then the corresponding color attributes are also been projected into 2D patches. While these patches are packed into 2D images, a 2D occupancy map indicating the location of patches is also been generated. Finally, the video coder compresses these images into bitstream.

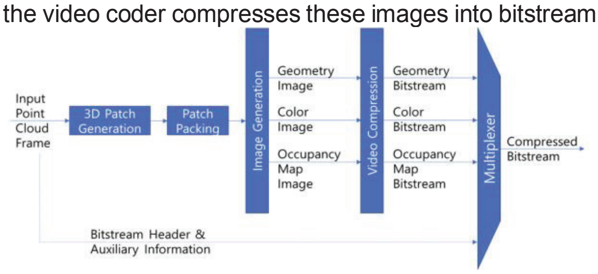

Fig.1 V-PCC Encoder Diagram

For example, Figure 2. is a captured image frame of a point cloud object with 765,821 points and 10 bits per axis for its geometry precision. Three intermediate images shown in Figure 3 were generated during the image generation and will be compressed. Image (a) showing the geometry patches in gray scale image, and image (b) showing the texture patches. 2D occupancy map is image (c).

Fig.2 A Capture of one Point Cloud

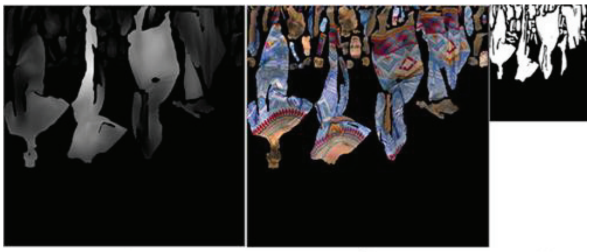

Fig.3 Intermediate 2D Images

On the decoder side of the V-PCC, the demultiplexer splits the bitstream for geometry, color attribute, and occupancy map. Through decoding the reconstructed geometry, color, and occupancy map, the data will be used to convert patches to 3D point cloud representation as shown on Figure 4.

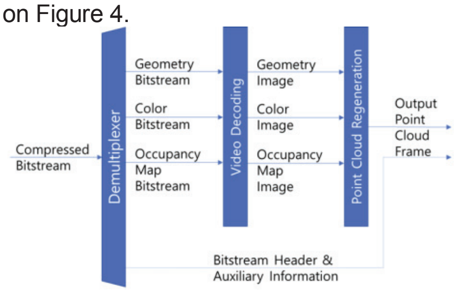

Fig.4 V-PCC Decoder Diagram

Presently, HEVC is used in V-PCC as a base video codec for its coding efficiency on sequential 2D Video [5]. However, the high complexity computation process of encoding high definition video is still a challenging work [6]. A statistic based on encoding a point cloud sequence named "red and black" with random access configuration in our research shows that up to 60% of total time was spent on video compression as shown on Figure 5. This measurement was taking on an Intel i7-7700K processor based on Microsoft Windows 7 operating system with single thread setting.

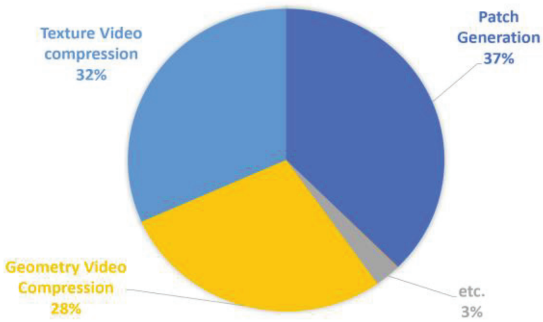

Fig.5 Encoding Complexity Compare on Patch Generation and Video Encoding

The V-PCC standard [7] defined several profile and levels to regulated conformance point as the V-PCC profiles. In addition, the basic support of multiple video codec is also mentioned. In the Base profile, it suggests supporting some of the AVC or HEVC profile. Thus, the problem on how to apply video codecs into V-PCC is unclear, which needs more study.

Furthermore, the bit rate related characteristics such as maximum bitrate per video and overall bitrate are suggested in the requirement and profile, which was presented by study on the HEVC. Profile setting of the other codecs needs to be researched.

3 PROPOSED METHOD

3.1 Design Issue on Video Codec Selection

In V-PCC, the converted geometry, texture and occupancy map (OM) 2D images are in different color space (i.e., geometry in grey scale, texture in 8-bit RGB, and occupancy in black and white). The color conversion process is necessary to compress such data by video codec. Take the V-PCC base profile as an example. The bit-depth and chroma sampling are limited because the bit depth is 8-bit on geometry, attribute and occupancy layers, and YUV Chroma sampling is YUV-420.

The extensive support of one video codec also needs to be considered. Well implemented software or hardware embedded processing unit will let the V-PCC developer deploy their application more efficiently. As a result, the V-PCC base profile suggest using AVC and HEVC.

Focusing within the codec, while multiple point cloud frames have been compressed as a group of frames (GOF). The accessibility on each frame is also very important inside the GOF. By inherited the coding structure of video codec, all-intra, random-access and low-delay profiles in video coding provide different behavior characteristics that may be suitable for different scenarios.

Moreover, the accessibility inside the frame such as partial decoding is also needed. Techniques including dividing one frame into tile or slice partitions that already used in the video codec would help the implement on V-PCC.

3.2 Design Issue to Use JPEG In V-PCC

To use JPEG as a base video codec for V-PCC, we need to consider the design issues mentioned in the previous section including color space, frame structure design and other details.

The geometry layers can use grey color space with the value from 0~255, and the texture layers are in RGB color space. The geometry image can be converted into YUV420-8-bit monochromatic only color space; and the texture image can be converted into YUV420 color space. In this process, Common test condition (CTC) [8] recommended HDR converter applied with ITU-R standard color space matrix representation is used. However, for JPEG image coding, instead of YUV raw video format we use the RGB Raw image format.

Due to the lack support on pixel format of MPEG recommended HDR converter, JPEG reference software required interleaved RGB format that was not readily available. We use ffmpeg to convert the un-interleaved RGB format into interleaved portable pixmap format (PPM) to intermediately store image file and feed it to JPEG reference software as input.

In the V-PCC, for one GOF, the bitstream multiplexer combine single geometry and texture video bitstreams into GOF bitstream with a bitstream size syntax in the header. But for the JPEG, multiple coded jpeg files need to be processed. So, we use 7z to patch these JPG files into one archive.

On the decoding side of our design, geometry and texture 7z files are split into several bitstreams. Then, we un-compress each jpeg bitstream to restore the corresponding layer patch. Finally, the V-PCC decoder uses this patch information to reconstruct the point cloud frames.

In CTC, 5 rate points is used for different compression ratio points. For these compression rate points, different HEVC compression parameter was set separately for geometry, attribute, and occupancy. These parameters from HEVC was called Quantization parameter (QP), which means that a lower QP results in a better image quality. However, the JPEG uses compression quality factor (QF) instead, the higher quality needs the higher setting. Thus, the conformance point for the base profile need to be measured.

4 EXPERIMENT AND RESULT

Based on these issues with V-PCC, we implemented our design with Test Model Category 2(TMC2) version 2.0 [9]. We merged the JPEG reference software with TMC2 2.0. In TMC2 2.0, the occupancy layer was not processed as image but in text mode. So, this part of video compression was not implemented in this experiment.

The developing environment is Visual Studio 2015, and the test was performed on Microsoft Windows 7 SP1 64-bit with multi thread enabled. The hardware specification is Intel i7-7700K @ 4.0GHz with system memory of 16GB, and all 8 cores were used in patch generation process.

In this experiment, 4 sequences of dynamic object category were used, the size of them were shown in Table 1. All 4 of these sequences are in 300 frames, the frame rate is 30fps and the bit precision is 10 bits, which also include RGB attribute for texture information.

Test Sequence	Point / Frame	Raw Data Size
loot	~780,000	5,144,378,340 bytes (4.79GB)
red_and_black	~700,000	4,699,844,836 bytes (4.37GB)
soldier	~1,500,000	7,150,911,747 bytes (6.65GB)
long_dress	~800,000	5,685,631,637 bytes (5.29GB)

Table 1. Details of sequences of Category 2

After the patch generation, the 2D images raw size is still very huge. For example, typical Category 2 geometry images are 0.91GB and color attribute images are 2.75GB. We take the “long_dress” test sequence to show the rate distortion (RD) curve in Figure 6. In this figure, peak signal to noise ratio (PSNR) and bitstream size in bytes are used. The proposed method takes JPEG QF from 20 to 90 with step of 10, and anchor using the CTC configuration. Through the graph we can observe that, due to the low compression rate of JPEG, proposed method with the highest QF could not to CTC’s R5 quality on geometry.

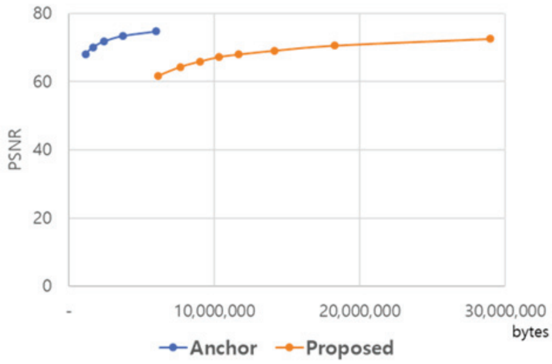

Figure 6. Geometry RD of long_dress Sequence

As geometry quality would also affect the texture PSNR. In order to compare the RD cost, and provide standard point for selecting texture quality of the corresponding profile. A fixed geometry quality using JPEG QF 90 was applied, then the experiment was performed while changing the JPEG QF from 10 to 90 with the step size = 20.

First, we compared the execution time on encoding of V-PCC with HEVC and JPEG. In Table 2, the minimum and maximum times are showed for V-PCC anchor, and an average time for proposed encoder is also listed. The encoding time differences for proposed method with different quality factor is within 10 seconds. The proposed method reduced 27% ~ 57% on total encoding time.

	HEVC	JPEG	Time Reduced
red_and_black	17162 ~ 20897	11011	35%~47%
loot	18033 ~ 21440	11892	34%~45%
soldier	21965 ~ 27398	16085	27%~41%
long_dress	21050 ~ 28476	12332	41%~57%

Table 2. Execution Time in seconds of Anchor and Proposed Method

Focus on the 2D video compression, the video compression time is significantly reduced as shown in Figure 7. A typical encoding time for 300 frames sequence is around 28~37 seconds, which means JPEG is 297 times faster than HEVC Intra profile on average.

Secondly, we compare the texture attribute and total bit size, the PSNR and byte size of all four sequences are shown in Figure 8. The proposed method with higher QFs could achieve similar quality as anchor, but the size is obviously larger than anchor. As expected, JPEG-based V-PCC coder is less efficient in compression efficiency.

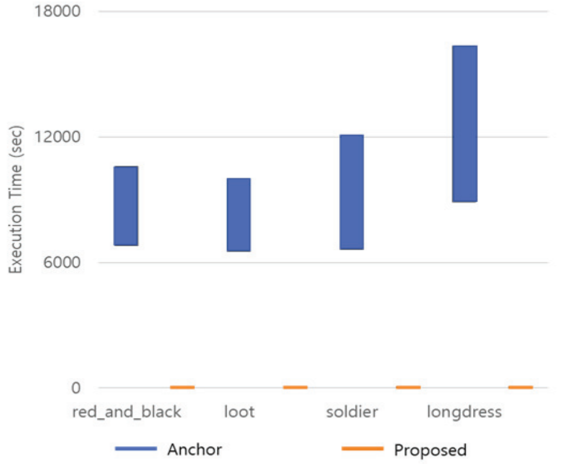

Figure 7. Execution Time of Video Compression

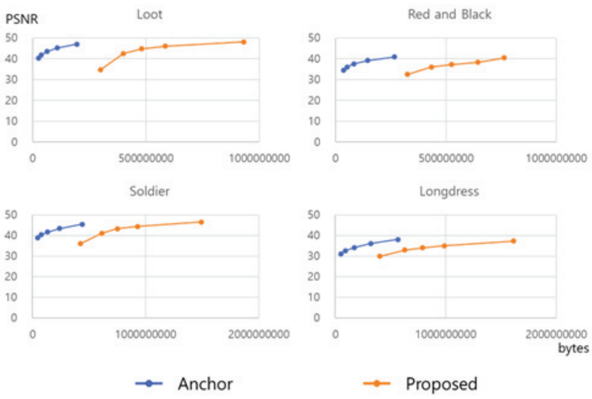

Figure 8. RD of color attribute

The original and reconstructed point cloud use anchor and proposed method are shown in Figure 9. The anchor using HEVC with texture QP setting 42 and proposed method using QF 50 shares a similar subjective visual quality.

Figure 9. Original and reconstructed point cloud

Through the experiment the proposed method reduced huge encoding time, it also provides 77 times compression ratio compared to raw data size on average. The gap of compression rate between the JPEG and HEVC is still very large, selecting a proper video codec to suit this gap and enrich related application will be a future work.

5 CONCLUSIONS

By reviewing the V-PCC codec structure and analysis the video compression details, we proposed our method of applying JPEG based V-PCC. With implementation of the proposed method, the proposed V-PCC encoder reduces significantly the amount of time, which also provides a low complexity approach on V-PCC development. We discovered that JPEG provides a good foundation to analysis connection between video codec and V-PCC. The study of selecting video codec for V-PCC is still an interesting research field that need to be focused.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1F1A1061737).

REFERENCES

[1] 3DG, "Text of ISO/IEC DIS 23090-5 Video-based Point Cloud Compression", ISO/IEC JTC 1/SC 29/WG 11, N18670 (2019).

[2] Soon-kak K, A. Tamhankar, K.R. Rao., "Overview of H.264/MPEG-4 part 10," Journal of Visual Communication and Image Representation, 17(2): 186-216 (2006).

[3] P. de Rivaz, J. Haughton, "AV1 bitstream and decoding process specification", [online] Available: https://aomediacodec.github.io/av1-spec/av1-spec.pdf (2019).

[4] Hudson, G., Léger, A., Niss, B., Sebestyén, I., & Vaaben, J., "JPEG-1 standard 25 years: past, present, and future reasons for a success," Journal of Electronic Imaging, 27(4), 040901, (2018).

[5] Sullivan G J, Ohm J R, Han W J, et al., "Overview of the high efficiency video coding (HEVC) standard", IEEE Transactions on circuits and systems for video technology, 22(12): 1649-1668 (2012).

[6] Frank B., "HEVC Complexity and Implementation Analysis", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, (2012).

[7] 3DG, "Working Draft 1.0 of V-PCC Profiles", ISO/IEC JTC 1/SC 29/WG 11, N18519 (2019).

[8] 3DG, "Common test conditions for point cloud compression", ISO/IEC JTC 1/SC 29/WG 11, N18474 (2019).

[9] 3DG, Test Model for Category 2, ISO/IEC JTC 1/SC 29/WG 11, [online] Available: http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc2 (2019).