Real-Time Video-based Point Cloud Encoding System on a Distributed Platform

Yura Kim, Yong-Hwan Kim
Intelligent Image Processing Research Center
Korea Electronics Technology Institute (KETI)
Seongnam-si, Korea
keyura@keti.re.kr, yonghwan@keti.re.kr

***Abstract*—In 2021, the MPEG introduced the video-based point cloud compression(V-PCC) standard, achieving an excellent 3D point cloud data compression ratio. However, the high computational complexity made real-time encoding impossible. To address this issue, we designed and implemented the V-PCC encoding system in real-time, leveraging the capabilities of a distributed platform. We achieve over 30 frames per second when encoding point cloud data with the proposed V-PCC real-time system. In this paper, we describe the architecture of the proposed real-time V-PCC encoding system and present demonstration results.

Keywords—Point Cloud Compression, V-PCC, distributed 3D compression system

I. INTRODUCTION

With the increasing use of 3D point cloud data in AR/VR applications, the need to compress point cloud data has become more apparent. In response, the ISO/IEC MPEG-I group established the video-based point cloud compression (V-PCC) standard in 2021 after conducting various experimental activities since 2017 [1-3]. In the V-PCC standard, a method of reconstructing 3D point clouds from atlas data and 2D images is defined. Among them, 2D images are compressed using the existing 2D codec, so they are named V-PCC [2]. Because it works based on legacy video codecs that are already widely used and installed on many devices, commercialization of V-PCC technology is expected to be achievable in a short time frame.

Although the V-PCC standard achieved a great compression ratio, the encoding process can be computationally demanding due to the need to convert 3D data into atlas data and 2D images, also encoding three videos simultaneously – an occupancy map, geometry, and attribute. To develop a real-time V-PCC encoding system, we utilized a distributed platform and customized various encoding tools. This approach enabled us to distribute the encoding workload effectively and achieve real-time V-PCC encoding performance.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (No. 2020-0-00231-004, Development of Low Latency VR·AR Streaming Technology based on 5G edge cloud)

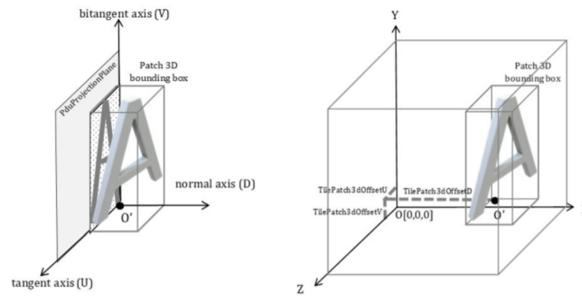

Figure 1. Patch definition on V-PCC [1]

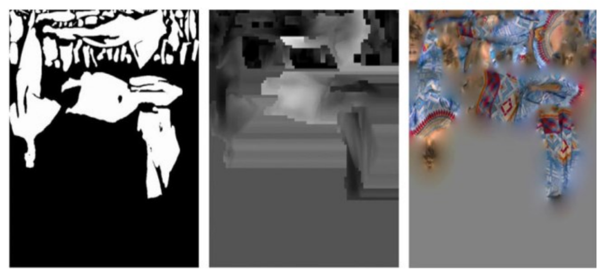

Figure 2. Example of patch packing

II. VIDEO-BASED POINT CLOUD COMPRESSION

A. Core concept

The 3D point cloud is compressed in the following order in the V-PCC encoder.

First, each point's projection plane index (PPI) is determined for the entire point cloud. The projection planes are basically six planes similar to the bounding box: up, down, right, left, front, and back. After determining the PPI for each point, the points with the same PPI are grouped into a segment. The included points in each segment are orthogonally projected onto a 2D plane to generate a patch. "Fig. 1" shows an example of the segment mapped into the 2D image with U and V values through projection [1]. The patch includes information such as the width, height, and depth values of the points projected onto 2D and the offset in 3D coordinates. Next, the patches are arranged in a 2D image, as shown in "Fig. 2." The packed 2D image sequences are compressed by a legacy video codec. Finally, the compressed video bitstream and atlas data of patches are encoded into a visual volumetric video-based coding (V3C) syntax format bitstream as "Fig.3."

When the point cloud is compressed using Test Model Candidate software (TMC2) of the V-PCC [4], the average bits per input point(bpip) value is 0.14 (C2AI condition: 0.11, C2RA condition: 0.17). This means that 0.14 bits represent information about one 3D point. Since the point cloud data consists of tens of thousands to millions of points, the V-PCC results in a great compression rate.

However, since the TMC2 encoder shows a speed of about 0.01 fps when using 16 threads for a point cloud video with an average of 1 million points per frame, real-time encoding is not possible. Of course, the TMC2 encoder has not been optimized and is reference software used for research purposes only. Of course, the TMC2 encoder has not been optimized and is reference software used for research purposes.

In this paper, the KETI V-PCC encoder was implemented by optimizing various parts based on the TMC2 encoder. Representatively, by replacing the video codec of the HEVC test model (HM) with x265, applying high-speed algorithms to various parts of the 3D to 2D conversion process [5], optimizing memory usage and access, and utilizing single instruction multiple data (SIMD) instruction, a speedup of more than 100 times was achieved. Due to various high-speed technologies applied to the KETI V-PCC encoder, the compression rate is somewhat lower than that of the TMC2 encoder, but the compressed bitstream can be transmitted in real time within 10Mbps.

Table 1 shows the result of comparing the encoding time of TMC2 and KETI V-PCC encoder. We tested C2AI and C2RA cases MPEG-I V-PCC CTC (common test condition) [6] for 32 frames in the same PC environment.

Table 1. The V-PCC Encoding time comparison[32-frame]

Sequence	Geometry bit-depth [bit]	C2AI		C2RA	
		TMC2 (16 threads) [sec]	KETI V-PCC (16 threads) [sec]	TMC2 (16 threads) [sec]	KETI V-PCC (16 threads) [sec]
Loot	10	1287.8	21.8	3383.4	25.0
Red and Black		1334.8	22.0	3997.7	24.7
Soldier		1519.8	29.8	3622.4	33.4
Queen		1559.4	27.9	3555.5	31.2
Long dress		1403.6	26.2	4382.3	28.4
Avg. encoding time		***1421.1***	***25.5***	***3788.3***	***28.5***
Basketball_ player	11	4133.4	110.0	14096.9	112.3
Dancer		3951.6	93.0	13183.9	96.2
Avg. encoding time		***4042.5***	***101.5***	***13640.4***	***208.6***

B. Group of Frame and V3C bitstream structure

Fig. 3" provides an example of V3C format. Each V3C sample stream unit contains the data of one Group of Frames (GOF), a bundle of multiple frames. Therefore, several frames

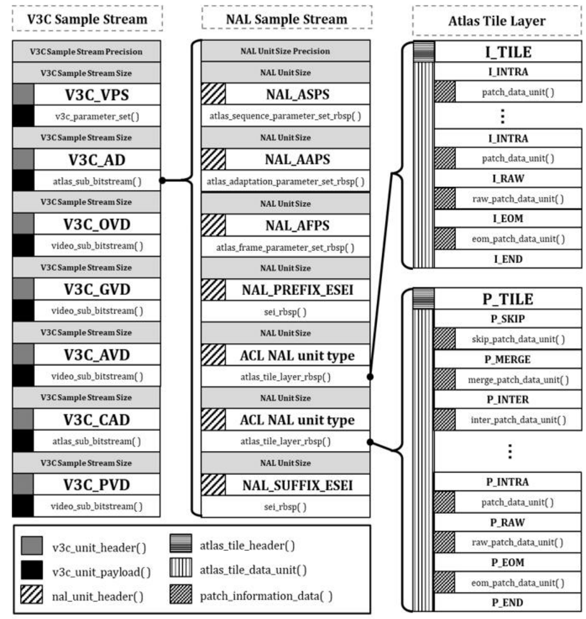

Figure 3. Overview of V3C bitstream structure [1]

of a GOF unit are inputted into the V-PCC encoder to generate one V3C sample stream. This input-output structure can be improved speed by performing distributed processing to encode multiple GOFs simultaneously.

III. Overview of Real-time System

The Apache Spark platform integrates clusters for fast and simple distribution of tasks and real-time data delivery to handle large-scale data processing [7]. Previously, we introduced a

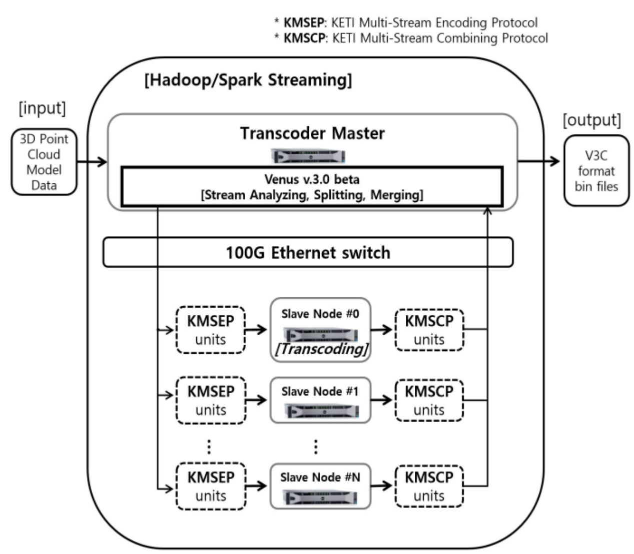

Figure 4. Overall work flow of system
(Hadoop 3.1.4, Spark Streaming 2.4.8)

distributed system for ultra-high-resolution video transcoding in [8], and now we have added the KETI V-PCC encoder to the system.

The overall system design is shown in "Fig. 4." The system is composed of a master node and 12 slave nodes, which are connected through a 100G Ethernet network and configured into a cluster.

At the master node, the input point cloud data is divided into GOF units and sent to each slave node in the KETI Multi-Stream Encoding Protocol (KMSEP) format using Spark streaming engine via a custom TCP/IP socket. The slave nodes receiving the KMSEP format then encode the allocated GOF point cloud data using the V-PCC encoder and generate the V3C bitstream. The slave node sends the V3C bitstream to the master node using the KETI Multi-Stream Combining Protocol (KMSCP) after completing the task.

At the master node, KMSCP packets are received from each slave node, and the V3C data is sorted and merged in chronological order to create the final bitstream.

IV. Demonstration Result

The experimental hardware environment is shown in Table 2. The input data consists of some test point cloud sequences used in MPEG-I V-PCC CTC. Originally, these test sequences were provided with only 250-300 frames, but we prepared them by repeating them to 5472 frames to verify the distributed processing effect. In addition, since the size of raw point cloud data is too large, we compressed it using Google Draco [9], which can perform near-lossless encoding of point clouds to reduce transmission overload.

Table 2. Test hardware environment

TYPE	CPU	RAM	OS	EA
Master	Dual AMD EPYC 74F3 CPU@3.19	DDR4 256GB	Windows Server 2019	1
Slave#1	Dual AMD EPYC 74F3 CPU@3.19		Window 10 Pro	4
Slave#2	Dual Xeon Gold 6242R CPU@3.10GHz,3.09GHz			8

Table 3 provides detailed information on the experimental sequences. The KETI V-PCC encoder in the slave node is set to encode according to the lossy random access and rate3 test condition of CTC. The KETI V-PCC encoder on the slave node has been optimized using x265, single instruction multiple data (SIMD) instruction, multi-threading and various fast techniques.

Table 3. Test sequence encoded by DRACO

Sequence	Frame rate	Frame count	Avg. Point Count per frame	Geometry Precision	Bitrate [Mbps]
Loot	30	5472	~780,000	10 bit	716
Red and Black	30		~700,000		721
Soldier	30		~1,500,000		1042
Queen	50		~1,000,000		896
Long dress	30		~800,000		904

Table 4 shows optimized standalone KETI V-PCC encoding speed and distributed encoding speed. While a standalone V-PCC encoder could not encode in real-time, real-time encoding was achieved through distributed processing.

Table 4. Experimental result

Sequence	Bit-rate [Mbps]	Standalone Encoding speed (16 threads)[FPS]	Distributed Encoding speed [FPS]
Loot	3.41	1.38	**40.23**
Red and Black	5.94	1.42	**40.46**
Soldier	5.42	0.97	**28.11**
Queen	4.12	1.23	**34.67**
Long dress	10.16	1.26	**34.66**
Avg. Encoding Speed		***1.25***	***35.62***

V. Conclusion

In this demonstration, we implemented distributed V-PCC encoder compressing point cloud over 30FPS in real-time. This system is developed based on the Hadoop/Spark platform. Especially, the Spark streaming scheme provides real-time memory-based data delivery. However, it has since been deprecated and replaced by a new and improved method called Spark Structured Streaming. So the system needs to be upgraded to Spark structured streaming. Also, since it is a distributed system, we will develop it so that it encodes multiple 3D objects at once.

References

[1] Coded representation of immersive media - Part 5:Visual volumetric video-based coding(V3C) and video-based point cloud compression(V-PCC), ISO/IEC 23090-5:2021.

[2] T. Dong, K. Kim, and E. S. Jang, "Performance Evaluation of the Codec Agnostic Approach in MPEG-I Video-Based Point Cloud Compression," in *IEEE Access*, vol. 9, pp. 167990-168003, 2021, doi: 10.1109/ACCESS.2021.3137036.

[3] C. Guede, P. Andrivon, J. -E. Marvie, J. Ricard, B. Redmann and J. -C. Chevet, "V-PCC Performance Evaluation of the First MPEG Point Codec," in *SMPTE, Motion Imaging Journal*, vol. 130, no. 4, pp. 36-52, May 2021.

[4] ISO/IEC JTC1/SC29, "Reference Sofrware for V-PCC", ISO/IEC 23090-19:2023.

[5] Y. Kim and Y. -H. Kim, "Low Complexity Fast Grid-Based Refining Segmentation in the V-PCC encoder," *Proc. IEEE Int. Conf. on Consumer Electronics-Asia (ICCE-Asia)*, Yeosu, Korea(South), 2022, pp. 1-4, doi: 10.1109/ICCE-Asia57006.2022.9954713.

[6] ISO/IEC MPEG 3DG group, *Common Test Conditions for V3C and V-PCC*, Standard ISO/IEC JTC1/SC29/WG11 MPEG2020/N19518, July 2020.

[7] Spark, https://spark.apache.org/

[8] J. Huh, Y. -H. Kim and J. Jeong, "Ultra-High Resolution Video Distributed Transcoding System Using Memory-based High-speed Data Distribution Method," *Proc. 34th Int. Technical Conf. on Circuits/Systems, Computers and Communications (ITC-CSCC)*, Jeju, Korea (South), 2019, pp. 1-4, doi: 10.1109/ITC-CSCC.2019.8793379.

[9] DRACO, 3D Data compression, https://google.github.io/draco/.