

# 포인트 클라우드 기반의 볼류메트릭 비디오 플레이어 구현 및 성능평가

김아영, 안은빈, 서광덕  
연세대학교 소프트웨어학부  
kdseo@yonsei.ac.kr

## Implementation and Performance Evaluation of a Point Cloud-based Volumetric Video Player

A-Young Kim, Eun-Bin An, Kwang-Deok Seo  
Division of Software, Yonsei University

### 요 약

본 논문에서는 사용자가 보다 간편하게 볼류메트릭 비디오를 소비할 수 있도록 볼류메트릭 비디오 기본 플레이어를 구현하고, 구현한 플레이어에 대하여 성능평가를 진행한다. 본 논문에서 구현한 볼류메트릭 비디오 플레이어는 Draco 와 V-PCC 를 복호화기로 지원하며, 압축 전의 포인트 클라우드 데이터와 Draco 와 V-PCC 로 압축한 비트스트림에 대하여 성능 평가를 진행하였다. 플레이어의 성능을 평가한 결과를 통해 초기 충분한 량의 프레임을 버퍼에 확보할 만큼의 초기 지연시간을 설정하지 않는 이상, 볼류메트릭 비디오를 30fps 이상으로 소비하기에는 어려움이 있음을 확인하였다. 이를 토대로 현재 볼류메트릭 비디오 재생을 위한 기술적 한계를 살펴보고, 볼류메트릭 비디오 플레이어의 성능 향상을 위한 향후 연구개발 방향에 대하여 논의한다.

### 1. 서론

지난 1 월에 국내에서 최초로 볼류메트릭(Volumetric) 기술을 도입한 뮤지컬 영상이 공개되었다<sup>[1]</sup>. 볼류메트릭 비디오는 6-DoF(Degree of Freedom) 환경에서 소비할 수 있는 차세대 미디어로써 기존의 비디오의 관념에서 한 차원 공간성이 확장된 비디오이다. 따라서 사용자는 자신이 원하는 방향에서 자유롭게 영상을 소비할 수 있으며, 이는 볼류메트릭 비디오와 같은 실감미디어(Immersive Media)가 사용자에게 보다 흥미롭게 접근할 수 있는 하나의 요소가 된다.

볼류메트릭 비디오는 3 차원 기하구조(Geometry)로 구성되며, 포인트 클라우드(Point Cloud) 또는 메쉬(Mesh) 등의 표현 방식이 있다. 특히 포인트 클라우드로 구성된 볼류메트릭 비디오는 각 포인트의 3 차원 좌표 정보 및 색상정보로 구성되기

때문에 방대한 용량을 가진다. 따라서 이를 압축 및 전송하고, 효율적으로 재생하기 위한 기술에 대한 요구가 증가하고 있다

2021 년 6 월, ISO MPEG(Moving Picture Expert Group)은 HEVC(High Efficiency Video Coding)나 VVC(Versatile Video Coding)와 같은 종래의 비디오 압축 기술을 활용하여 포인트 클라우드를 압축하기 위한 표준 V-PCC(V-PCC(Video-based Point Cloud Compression))를 제정하였다<sup>[2]</sup>. 포인트 클라우드 데이터는 크게 정적 객체(Static Object), 동적 객체(Dynamic Object), 동적 획득(Dynamic Acquired) 데이터로 분류할 수 있으며, 그 중 동적 객체의 압축을 위한 기술을 V-PCC 에서 다루고 있다. 또한 2021 년 10 월에는 동적 메쉬 압축을 위한 표준화 논의를 위하여 CfP(Call for Proposal)가 발행되었다<sup>[3]</sup>. 나아가 포인트 클라우드를 압축하기 위한 연구와 함께 포인트 클라우드를 효율적으로 전송하기 위한 많은 연구가 진행되고 있다. DASH-PC<sup>[4]</sup>와 PCC-DASH<sup>[5]</sup>는 MPEG-DASH(Dynamic

Adaptive Streaming over HTTP)를 통해 포인트 클라우드를 적응적으로 전송하기 위한 포인트 클라우드 서브 샘플링 알고리즘 및 MPD(Media Presentation Description) 구조에 대해 논의하였다. 또한 전송 대역폭을 효율적으로 활용하면서 네트워크 지연을 최소화하기 위한 클라우드 기반의 전송 기법도 연구되었다<sup>[6][7]</sup>.

앞서 살펴본 바와 같이 포인트 클라우드 기반의 볼류메트릭 비디오의 소비 플랫폼을 구축 및 활성화하기 위하여 압축과 전송 단계의 논의가 적극적으로 이루어지고 있다. 하지만 볼류메트릭 비디오를 재생하기 위한 플랫폼은 일반적으로 게임 엔진을 기반으로 구현되기 때문에 사용자가 가볍게 접근하기에는 다소 부담이 될 수도 있다. 따라서 본 논문에서는 사용자가 볼류메트릭 비디오를 소비할 수 있는 간단한 재생 플랫폼을 구현하고, 볼류메트릭 비디오의 안정적인 소비를 확인하기 위한 성능 평가를 진행한다.

## 2. 관련 연구

볼류메트릭 비디오를 구성하는 단일 포인트 클라우드 객체는 평균 100 만개의 포인트로 구성된다. 이때, 볼류메트릭 비디오를 30 fps 로 10 초 길이를 제공한다면, 해당 비디오의 용량은 대략 4.19 GB 가 된다. 이는 포인트 클라우드 객체의 각 포인트가 색상 정보를 비롯한 3 차원 위치 좌표 정보로 구성되기 때문이다. 따라서 방대한 크기의 볼류메트릭 비디오를 무리 없이 소비하기 위해서는 이에 대한 효율적인 압축이 선행되어야 한다. 현재 포인트 클라우드 데이터를 압축하기 위한 다양한 코덱이 존재하고 있으며, 본 논문에서는 V-PCC 와 Draco 를 활용하였다.

Draco 는 Google 사에서 제공하는 3D 그래픽 압축 도구이며, 단일 정적 객체 압축에 높은 성능을 보인다<sup>[8]</sup>. 웹 상에서 활용할 수 있도록 자바스크립트 기반의 API 를 제공하며, C++ 기반의 라이브러리도 활용 가능하다. Draco 는 k-d 트리를 활용한 공간적 압축 기법을 제공하지만, 시간적 압축은 아직 고려되지 않고 있다. 따라서 본 논문에서는 단일 포인트 클라우드 객체 300 장에 대하여 공간적 압축을 진행하였다.

V-PCC 는 단일 포인트 클라우드 동적 객체의 시간적, 공간적 중복성을 압축하기 위한 표준이며, 그림 1 은 V-PCC 부호화기의 가장 핵심이 되는 내용을 보여주는 개념도이다. V-PCC 부호화기는 영상의 압축 전에 3 차원의 포인트 클라우드 데이터를 2 차원으로 투영(Projection)하는 과정을 거쳐 패치(Patch)를 생성한다. 이때 하나의 포인트 클라우드 데이터에

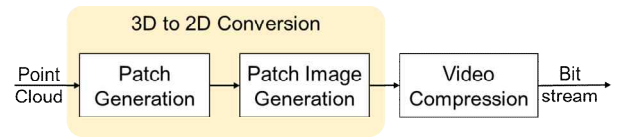


그림 1. V-PCC 부호화기의 개념도

대하여 점유 지도(Occupancy Map), 기하 영상(Geometry Map) 그리고 속성(Attribute) 정보가 생성되며 각 정보는 기존의 비디오 코덱으로 압축할 수 있다. V-PCC 에서 지원하는 내장 코덱은 AVC(Advanced Video Coding)와 HEVC 그리고 VVC 등이 있다.

인텔에서 개발한 3D 그래픽 렌더링 라이브러리인 Open 3D 는 C++과 Python 을 지원하며, Open GL 을 기반으로 GUI 제작의 편의성을 갖추으로써 보다 간편하고 빠르게 3D 그래픽 데이터를 다룰 수 있도록 개발되었다<sup>[9]</sup>. 포인트 클라우드의 렌더링을 지원하는 라이브러리에 비하여 상대적으로 써드 파티(Third Party)의 의존성이 적으며, 사용자 시점 카메라의 이동을 지원하기 때문에 6-DoF 환경에서의 미디어 소비가 가능하다.

## 3. 볼류메트릭 비디오 플레이어의 구현

본 논문에서 구현한 볼류메트릭 비디오 플레이어의 구조도는 그림 2 와 같다. 플레이어는 압축 전의 원시 포인트 클라우드와 압축된 비트스트림 모두를 재생할 수 있다. 저장소로부터 미리 저장된 파일을 읽고, 데이터의 종류에 따라서 바로 재생하기 위해 랜더러 버퍼에 데이터를 저장하거나 복호화기를 통해 복호화 과정을 거친다. 복호화기는 V-PCC 의 테스트 모델인 TMC2 15.0 버전과 Draco 1.5.0 버전을 선택할 수 있도록 구성되어 있으며, 파일을 읽어올 때 압축 비트스트림을 통해 복호화기를 선택하게 된다. 복호화 과정을 거친 포인트 클라우드 데이터는 재생을 위해 Open 3D 의 포인트 클라우드 구조로 변환되는 과정을 거친다. 이는 각 복호화기를 거쳐서 나온 포인트 클라우드의 데이터 구조와 Open 3D 에서 제공하는 포인트 클라우드 데이터 구조에 차이가 있기 때문에 필요한 과정이다. 구조 변환 과정에서 포인트의 좌표 정보를 3 차원 벡터로 구성하여 저장하고, RGB 로

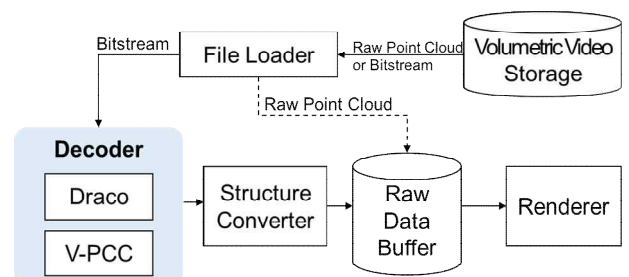


그림 2. 볼류메트릭 플레이어 구조도

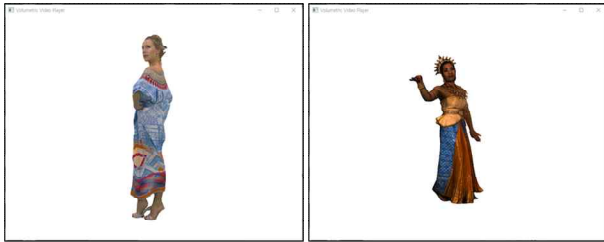


그림 3. 볼류메트릭 비디오 재생 화면

표현된 색상 정보는 0 에서 1 사이의 값으로 정규화 하는 등의 동작이 수행된다. 구조 변환이 완료된 포인트 클라우드 데이터는 렌더러를 통해 화면에 출력되며, 이때 렌더러는 마우스나 키보드의 움직임을 통해 전방위에서 볼류메트릭 비디오를 소비할 수 있다. 그림 3 은 구현된 볼류메트릭 비디오 플레이어를 통하여 다양한 시퀀스를 소비하는 장면이다.

#### 4. 실험 및 성능 평가

본 논문에서는 구현한 볼류메트릭 비디오 플레이어가 소비자의 몰입감을 보장하기 위하여 최소 프레임 레이트로써 30 fps 를 설정하고 정상적인 재생이 가능한지에 대한 성능평가를 진행하였다. 성능 평가 환경은 CPU Intel i7-10700K, 64 GB RAM 그리고 그래픽 카드로 NVIDIA GTX 1080 을 사용하였다. 또한 GCC C++17 기반으로 구현하였으며, 사용된 복호화기와 라이브러리의 버전 정보는 표 1 과 같다. 성능 평가에 활용한 데이터 셋은 8i 사에서 제공하는 데이터 셋 중 redandblack 시퀀스를 사용하였다<sup>10)</sup>.

표 1. 복호화기 및 라이브러리 정보

GCC	C++ 17
V-PCC	15.0 (내장 코덱: HM-16.20+SCM-8.8 & VTM 13.0)
Draco	1.5.0
Open3D	0.14.1

사용자의 몰입감을 저해하지 않는 선에서 볼류메트릭 비디오를 제공하려면 영상의 프레임 레이트는 최소한 30 fps 이상으로 제공되어야 하기 때문에 한 프레임이 복호화되어 화면에 출력되기까지 평균적으로 소요되는 시간은 약 33ms 가 소요된다. 따라서 정밀한 성능평가를 위해 플레이어가 파일을 읽고 화면에 출력하기까지 걸리는 시간을 단계 별로 나누어 측정하였다. 원시 포인트 클라우드는 복호화 및 구조 변환 과정을 거치지 않기 때문에 오직 파일을 읽어오는 시간만을 측정하였다. 압축 비트스트림의 경우에는 전체 과정 중 소모되는 시간이 가장 긴 복호화 및 구조 변환에 걸리는 시간을 프레임 단위로

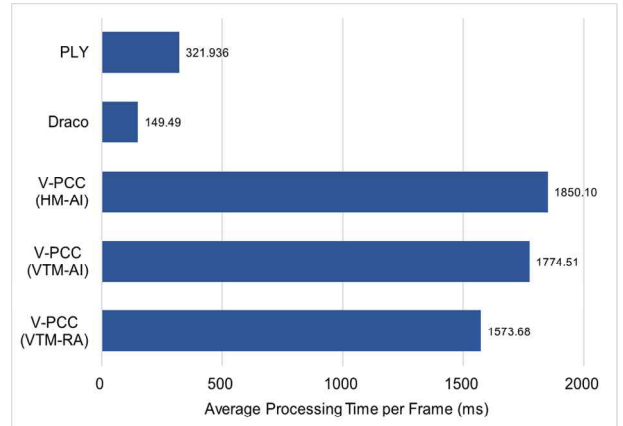


그림 4. 한 프레임 당 평균 처리 시간(ms)

측정하여, 포인트 클라우드 데이터가 처리되는 과정에서 시간 소모가 큰 부분을 확인할 수 있었다.

그림 4 는 원시 포인트 클라우드 데이터와 Draco 압축 비트스트림, V-PCC 압축 비트스트림에 대하여 플레이어에서 각각의 한 프레임을 처리하기 위해 평균적으로 걸리는 시간을 나타낸다. V-PCC 의 경우, 다양한 내장 코덱을 활용하여 HM 기반의 압축 비트스트림과 VTM 기반의 압축 비트스트림의 결과를 확인할 수 있다.

압축하지 않은 포인트 클라우드 데이터의 경우, 복호화와 구조 변환에 걸리는 시간은 발생하지 않고 파일을 읽어오는 시간만 소모되며, 이 경우 한 프레임을 읽는데 약 321.94 ms 가 소모된다. 반면 Draco 와 V-PCC 를 통해 부호화 된 비트스트림을 읽기 위해서 약 4 ms 의 시간이 소요되어 상대적으로 매우 적은 시간이 소모되었다. 또한 복호화 후 Open 3D 의 데이터 구조로 변환하는데 소모되는 시간은 약 45ms 로 모든 압축 비트스트림에 대하여 비슷한 시간이 걸렸다. Draco 압축 비트스트림은 한 프레임이 처리되는데 평균 149.49 ms 가 소모되어 원시 포인트 클라우드 파일에 비하여 약 50%의 시간이 단축된 것을 확인할 수 있다. 반면 V-PCC 압축 비트스트림을 처리하기 위해 소모되는 시간은 상대적으로 약 5 배 증가하였다. 구조 변환 시간이 Draco 와 V-PCC 의 구분 없이 유사한 것을 고려하였을 때, 전체적으로 복호화에 소모되는 시간이 더 큰 것을 확인할 수 있으며, 목표로 하는 수치에 미치지 못함을 확인할 수 있다. 따라서 볼류메트릭 비디오를 사용자의 몰입감 저하 없이 재생하기 위해, 초기 지연 시간을 두어 충분한 양의 프레임을 버퍼에 확보해야 할 것이다.

V-PCC 압축 비트 스트림의 경우, 평균적인 프레임 처리 시간이 매우 높은 편이다. 하지만 시간적 압축을 고려하기 때문에 Draco 에 비하여 압축률이 상당히 높은 편이다. 압축 전의 포인트 클라우드 데이터가 약 4.45 GB 였음을 고려하였을 때, V-PCC 의 압축 비트스트림은 HM-AI(All-Intra)의 경우 9.310 KB,

VTM-AI 와 RA(Random Access)는 각각 7,905KB 와 4,242 KB 이다. Draco 의 압축 비트스트림이 541MB 임을 고려하였을 때 매우 높은 압축 성능을 보여주는 것을 확인할 수 있다.

## 5. 결론

본 논문에서는 볼류메트릭 비디오 플레이어 구현하고 성능 평가를 진행하였다. V-PCC 의 경우 활용하는 내장 코덱과 모드에 따라 파일의 압축률과 평균 프레임 처리 시간을 확인했다. 전체적으로 평균 프레임 처리 시간은 높은 편이었으나, 매우 높은 압축 성능 또한 확인할 수 있다. 따라서 전체적인 전송까지 고려하였을 때 V-PCC 의 활용을 배제할 수는 없다. Draco 의 경우 압축 비트스트림을 처리하기 위한 시간이 전체적으로 제일 낮게 측정되었음에도 불구하고, 약 6 fps 의 재생속도에 불과하였으며, 다수의 복호화를 병렬로 사용할 경우에는 평균 15 fps 까지 증가였지만, 사용자의 몰입감을 보장하기에는 다소 낮은 수치였다. 따라서 추후 연구를 통해 볼류메트릭 비디오 플레이어의 성능이나 복호화기의 성능을 개선할 것이며, 나아가 현재 표준화 작업이 시작된 DMC(Dynamic Mesh Coding)에 대하여도 지원을 할 예정이다.

## ACKNOWLEDGMENT

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(NRF-2021R1F1A1048404).

## 참 고 문 헌

- [1] 연합뉴스, "'이퀄', 뮤지컬 최초로 볼류메트릭 기술 도입", <https://www.yna.co.kr/view/AKR20220117112400005>
- [2] ISO/IEC 23090-5:2021, "Coded representation of immersive media – Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)"
- [3] CfP for Dynamic Mesh Coding, ISO/IEC JTC1/SC29/WG7 N231, Online, October, 2021.
- [4] M. Hosseini, C. Timmerer, "Dynamic adaptive point cloud streaming", Proceedings of the 23rd Packet Video Workshop, pp. 25-30, 2018.
- [5] J. van der Hooft, T. Wauters, F. De Turck, C. Timmerer, H. Hellwagner, "Towards 6DoF HTTP adaptive streaming through point cloud compression", Proceedings of the 27th ACM International Conference on Multimedia, pp. 2405-2413, 2019.
- [6] S. Gül, D. Podborski, T. Buchholz, T. Schierl, C. Hellge, "Low-latency cloud-based volumetric video streaming using head motion prediction", Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 27-33, 2020.
- [7] S. Gül, D. Podborski, J. Son, G. S. Bhullar, T. Buchholz, T. Schierl, et al., "Cloud rendering-based volumetric video streaming system for mixed reality services", Proceedings of the 11th ACM Multimedia Systems Conference, pp. 357-360, 2020.
- [8] Google, "Draco: 3D Data Compression", 2018. <https://github.com/google/draco>
- [9] Zhou, Qian-Yi, Jaesik Park, and Vladlen Koltun. "Open3D: A modern library for 3D data processing." arXiv preprint arXiv:1801.09847, 2018
- [10] Maja Krivokuća, Philip A. Chou, and Patrick Savill, "8i Voxelized Surface Light Field (8iVSLF) Dataset," ISO/IEC JTC1/SC29 WG11 (MPEG) input document m42914, Ljubljana, July 2018.