

포인트 클라우드 기반의 볼류메트릭 비디오 플레이어 설계 및 구현

김아영*, 안은빈*, 서광덕^o

Design and Implementation of a Point Cloud-Based Volumetric Video Player

A-young Kim*, Eun-bin An*, Kwang-deok Seo^o

요약

포인트 클라우드로 표현되는 3차원의 볼류메트릭 비디오는 사용자가 원하는 시점에 따라 자유롭게 소비할 수 있기 때문에 높은 자유도와 몰입감을 제공한다. 볼류메트릭 비디오는 한 개 이상의 동적 객체로 구성되며, 동적 객체는 포인트 클라우드와 같은 3차원 기하구조 데이터로 표현할 수 있다. 동적 객체는 여러 개의 정적 객체가 시간적 연관성을 가지고 구성되어 있으므로 볼류메트릭 비디오는 종래의 2차원 평면 영상에 비하여 방대한 용량을 가진다. 이를 사용자에게 제공하기 위해서 볼류메트릭 비디오를 구성하는 포인트 클라우드 데이터의 압축 및 전송에 대한 연구가 활발히 진행 중이다. 하지만 볼류메트릭 비디오 압축 및 전송 기술 연구에 비하여 볼류메트릭 비디오 플레이어에 대한 논의는 상대적으로 미비한 상황이다. 기존의 포인트 클라우드의 출력을 지원하는 프로그램은 주로 정적 객체의 출력을 지원하는 뷰어의 성격이 강하기 때문에 동적 객체를 재생하는 기능이 제한적이다. 따라서 본 논문에서는 동적 객체를 재생할 수 있는 플레이어를 구현하고, 나아가 실험을 통해 각 모듈의 소비 시간을 측정하여 사용자의 안정적인 볼류메트릭 비디오 소비를 저해하는 모듈을 특정한다. 실험 결과를 바탕으로 안정적인 볼류메트릭 비디오 소비를 위한 기술의 현실적인 한계를 확인하고, 이를 개선하기 위한 향후 연구 방향에 대하여 논의한다.

키워드 : 볼류메트릭 비디오 서비스, 포인트 클라우드, Draco, V-PCC, Open3D

Key Words : Volumetric video service, point cloud, Draco, V-PCC, Open3D

ABSTRACT

Volumetric video represented as a point cloud provides a high degree of freedom and immersion since it enables the user to consume the video according to the desired point of view. The volumetric video consists of one or more dynamic objects, which can be expressed as 3D geometric data such as point clouds. Since the dynamic point cloud object is composed of several static objects with temporal correlation, the volumetric video has a large volume compared to the conventional video. Therefore, research on compression and

* 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).

• First Author : Yonsei University, Division of Software, aykim90@yonsei.ac.kr, 학생회원

o Corresponding Author : Yonsei University, Division of Software, kdseo@yonsei.ac.kr, 중신회원

* Yonsei University, Division of Software, eunbin.an@yonsei.ac.kr, 학생회원

논문번호 : 202204-059-D-RE, Received April 27, 2022; Revised July 12, 2022; Accepted August 8, 2022

transmission of point cloud data to provide the volumetric video service is being actively conducted. However, compared to the research on volumetric video compression and transmission, the discussion on the volumetric video player has been relatively considered insignificant. Conventional applications for rendering point-cloud primarily support rendering static objects. In this paper, we implemented a volumetric video player for rendering dynamic objects and measured the consumption time of each module to specify the module causing the delay. Based on the experiment results, we confirm that there is a realistic limitation to consuming volumetric video, and then discuss future research to improve the limitations.

1. 서 론

포인트 클라우드(Point Cloud)로 구성된 볼류메트릭 비디오(Volumetric Video)는 사용자가 전방위(Omnidirectional)에서 영상을 감상할 수 있으며, 이에 6-DoF(Degree of Freedom) 환경에서 소비할 수 있는 실감 미디어(Immersive Media)의 일종으로 주목받고 있다. 국내/외 기업에서는 차세대 미디어로써 볼류메트릭 비디오의 중요성을 바탕으로 이를 촬영할 수 있는 볼류메트릭 캡처 스튜디오를 활발히 설립하고 있다. 뿐만 아니라 콘서트, 스포츠 중계 서비스와 뮤지컬 공연을 볼류메트릭 비디오를 통해 제공함으로써 시장 선점에 노력하고 있다¹⁾.

포인트 클라우드 기반의 볼류메트릭 비디오는 색상 정보 뿐만 아니라 각 포인트의 위치에 대한 표현으로 인하여 방대한 데이터량을 가지는 특성이 있으며, 이를 고려한 압축 기술의 연구가 활발히 진행되고 있다. Google은 3D 기하구조(Geometry)인 메쉬(Mesh)와 포인트 클라우드를 부/복호화하기 위한 오픈 소스 라

이브리리로 Draco를 공개하였으며²⁾, ISO MPEG (Moving Picture Experts Group)은 MPEG-I Part 5의 V3C(Visual Volumetric Video-based Coding) and V-PCC(Video-based Point Cloud Compression)³⁾와 Part 6의 G-PCC(Geometry-based Point Cloud Compression)⁴⁾를 포인트 클라우드의 압축을 위한 표준으로 제정하였다.

포인트 클라우드의 압축 기술 연구와 더불어 볼류메트릭 비디오를 효율적으로 전송하기 위한 연구도 활발히 진행 중이다. DASH-PC는 MPEG-DASH (Dynamic Adaptive Streaming over HTTP) 프로토콜을 활용하여 포인트 클라우드를 전송하기 위해, MPD(Media Presentation Description) 구조와 포인트 클라우드 서브 샘플링 알고리즘을 제안하였으며⁵⁾, PCC-DASH는 복수의 볼류메트릭 객체를 전송하기 위한 MPD 구조를 제안하였다⁶⁾.

또한 클라우드 기반의 볼류메트릭 미디어 전송 기법을 통해 네트워크 지연 시간을 줄임으로써 몰입감을 증대하고, 동시에 전송 대역폭을 효율적으로 활용

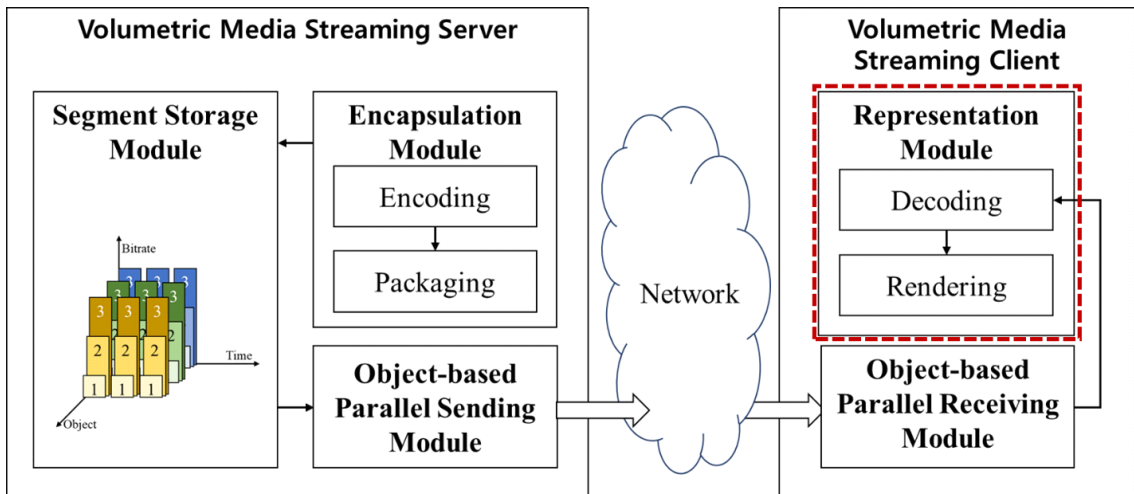


그림 1. 볼류메트릭 비디오 전송 시스템의 구조도
Fig. 1. Volumetric Video Streaming System Architecture

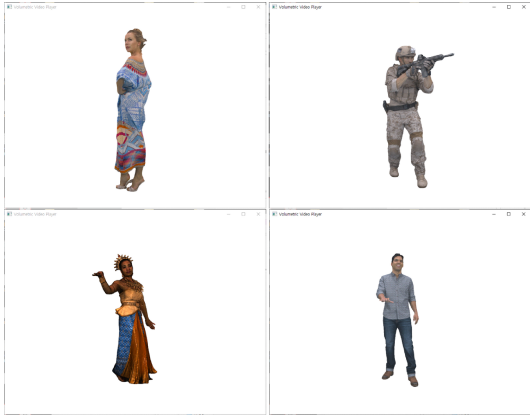


그림 2. 다양한 볼류메트릭 비디오를 소비하는 장면
Fig. 2. Playback scene of various volumetric video

하는 방안이 연구되었다^{7,8)}. 러한 방향의 연구는 사용자의 시점에 상응하는 부분을 클라우드 상에서 렌더링하여 전송하기 때문에 클라이언트는 볼류메트릭 비디오가 아니라 실질적으로 종래의 비디오를 소비하게 된다.

이처럼 볼류메트릭 비디오의 획득과 압축, 그리고 전송 기술에 대한 논의가 활발히 이루어지고 있는 것에 비하여 볼류메트릭 비디오 자체를 소비할 수 있는 플레이어에 대한 구현 및 논의는 여전히 미비하다. MeshLab과 같은 포인트 클라우드 뷰어(Viewer)의 경우, 단일 정적 객체의 소비에는 적절하나 동적 포인트 클라우드 객체로 구성된 볼류메트릭 비디오를 소비하기에는 한계가 있다. 따라서 동적 포인트 클라우드 객체를 재생하기 위한 볼류메트릭 비디오 플레이어에 대한 구현 관련 이슈를 파악하고, 이를 토대로 볼류메트릭 비디오를 전송하기 위한 플랫폼 설계가 필요하다.

본 논문에서 다루는 볼류메트릭 비디오 플레이어는 그림 1의 볼류메트릭 비디오 전송 시스템 중 재생 모듈(Representation Module)에 해당하며, 이를 사용하여 다양한 데이터 셋을 재생한 결과를 그림 2를 통해 확인할 수 있다.

II. 관련 연구

2.1 포인트 클라우드 기반의 볼류메트릭 비디오 압축 기술

포인트 클라우드는 3차원 공간에 포인트 데이터가 존재하며, 각 포인트마다 3차원 좌표정보와 색상정보를 가진다. 따라서 포인트 클라우드 기반의 볼류메트릭 영상은 용량이 매우 크다. 예를 들어, 평균 100만

개의 포인트로 구성된 볼류메트릭 객체를 한 프레임으로 영상을 구성한다면, 30fps의 10초 길이의 영상에 해당하는 데이터량은 약 4.19 GB가 된다. 따라서 포인트 클라우드 기반의 볼류메트릭 비디오를 전송하기 위하여 압축은 필수적이며, 이를 위해 압축 기술을 지원하는 다양한 코덱이 존재한다.

Google에서 제공하는 Draco는 3D 그래픽의 효율적 저장과 전송 성능 향상을 목표로 하며, 특히 정적 객체의 압축에 높은 성능을 보여준다²⁾. 또한 C++ 기반의 라이브러리를 제공할 뿐만 아니라 자바스크립트 기반의 API도 지원하여 웹상에서도 보다 빠르게 3D 객체를 출력할 수 있다. Draco는 공간 분할(Spatial Partition)을 활용하는 kd-tree를 기반으로 포인트 클라우드를 압축한다⁹⁾. Draco는 공간적 압축만 제공하고 시간적 압축은 고려하지 않기 때문에, 본 논문에서는 원시 데이터 300개를 압축하여 시간 순서로 압축 비트스트림을 구성하였다.

ISO MPEG에서는 2016년 10월 제 116차 회의를 통해 6-DoF 단계의 표준화 계획을 포함한 MPEG-I를 발족하고, 실감미디어 표준화 작업을 활발히 진행 중이다. 특히, 2017년 10월 제 120차 회의에서 포인트 클라우드 데이터를 정적 객체(Static Object), 동적 객체(Dynamic Object), 그리고 동적 획득(Dynamic Acquired)과 같이 세가지 종류로 분류하고 각 데이터의 특징에 따른 압축 기법 적용을 시도하였다. MPEG의 포인트 클라우드 압축 표준의 일종인 G-PCC는 8진(Octree) 트리 구조를 기반으로 기하 정보를 압축하고, 속성 정보는 RAHT(Region-Adaptive Hierarchical Transform)나 LOD(Level of Details) 기반의 압축 기술을 활용하여 정적 객체와 동적 획득 데이터의 압축을 지원한다. 정적 객체는 포인트 클라우드로 구성된 단일 객체이며, 동적 획득은 LiDAR(Light Detection and Ranging)와 같은 데이터를 의미한다.

또 다른 MPEG의 포인트 클라우드 압축 표준인 V-PCC는 포인트 클라우드가 시간 상에서 연속적으로 움직임을 갖는 동적 객체를 압축하기 위한 표준이다^{10,11)}. V-PCC는 기존의 비디오 압축 코덱을 내장하여 활용하며, 내장 코덱으로 AVC(Advanced Video Coding), HEVC(High Efficiency Video Coding) 및 VVC(Versatile Video Coding) 중 하나를 선택하여 활용할 수 있다. V-PCC 인코더는 크게 패치 생성(Patch Generation) 과정과 비디오 압축(Video Compression) 과정으로 구성된다. 패치 생성 과정은 3차원 포인트 클라우드 정보를 2차원 패치 영상으로 변환하는 과정이며, 패치 생성 결과로써 기하 영상

(Geometry Map), 점유 지도(Occupancy Map), 그리고 속성(Attribute) 정보가 생성된다. 이 세 가지의 패치 영상을 기존의 비디오 코덱을 활용하여 압축하는 것이 비디오 압축 과정이다. V-PCC 디코더는 SPS(Sequence Parameter Set) 정보를 사용하여 하나로 압축된 비트스트림을 기하 영상, 점유지도, 속성 정보 및 패치 정보를 복호화하고, 복호화된 정보를 기반으로 포인트 클라우드를 복원한다. 그 후 불연속적인 부분의 보완을 위한 후처리(Post-processing)로써 스무딩(Smoothing) 과정을 진행한다.

2.2 볼류메트릭 비디오 렌더링 라이브러리

일련의 동적 볼류메트릭 객체들로 구성된 볼류메트릭 비디오를 화면에 출력하기 위하여 다양한 렌더링 라이브러리가 존재한다.

PCL(Point Cloud Library)은 포인트 클라우드를 처리하기 위한 라이브러리이며, 2011년에 처음 등장하였다. 포인트 클라우드 파일 입출력과 필터링, 특징 추정, 표면 재구성 등 다양한 알고리즘을 지원하며, VTK를 기반으로 포인트 클라우드를 화면에 출력하는 Visualization 모듈을 지원한다¹²⁾.

Open3D는 3D 그래픽 데이터를 보다 간편하고 빠르게 렌더링할 수 있도록 제공하는 오픈 소스 라이브러리이다. C++과 Python을 지원하며, 다른 3D 그래픽 렌더링 라이브러리에 비하여 보다 간결하게 구현이 가능하다는 장점이 있다. Open3D는 3D 그래픽의 입/출력과 함께 이를 담는 데이터 구조와 기본 처리 알고리즘을 제공한다. 또한 사용자 시점 카메라의 이동을 지원하여 6-DoF의 환경에서 미디어 소비를 가능하게 하며, OpenGL을 기반으로 한 Visualization 모듈을 통해 GUI 제작의 편의성을 도모하였다¹³⁾. 본 논문에서는 상대적으로 써드 파티(Third party) 라이브러리의 의존성이 적은 Open3D를 활용하여 렌더러(Renderer)를 구현하였다.

2.3 뷰포트 적응적 전송

비디오나 볼류메트릭 비디오와 같은 실감미디어는 6-DoF 환경에서 주로 소비가 이루어지기 때문에, 사용자가 직접적으로 소비하는 영역은 사용자의 시선이 위치한 뷰포트(Viewport) 영역에 국한된다. 이러한 특성을 고려하여 360도 비디오를 위한 뷰포트 적응적 전송은 뷰포트 영역의 영상을 고품질로 압축 및 전송하고, 나머지 영역에 대하여 상대적으로 저품질을 적용하는 기법이다^{14,15)}. 이는 볼류메트릭 비디오 전송에서도 적용할 수 있다. 포인트 클라우드 기반의 볼류

메트릭 객체를 여러 구역으로 분할 압축하여 전송을 위한 세그먼트를 구성하고, 사용자의 시점에 상응하는 부분을 우선적으로 전송함으로써 대역폭을 절약할 수 있다. 클라이언트에서는 일부 영상만을 복호화 하기 때문에 전체 영상을 복호화 하는 것에 비해 처리 시간을 절약할 수 있으며, 불필요한 자원 소모를 피할 수 있다. 이를 위하여 향후 동적 포인트 클라우드 객체의 일부분만을 독립적으로 압축하는 기술 연구와 볼류메트릭 객체의 여러 부분을 동시에 소비할 때 객체 상호간의 시간공적 동기화를 맞추는 연구 등이 필요하다. 최종적으로 볼류메트릭 비디오를 효율적으로 전송하고 사용자의 몰입감을 저해하지 않으면서 소비하기 위한 기술의 연구가 필요하다.

III. 볼류메트릭 비디오 플레이어 설계 및 구현

포인트 클라우드 기반의 볼류메트릭 미디어를 압축 및 전송, 그리고 최종적으로 사용자가 이를 소비하기 위해서 볼류메트릭 비디오 플레이어의 구현이 필수적이다. 이에 볼류메트릭 비디오 전송 시스템의 구현에 앞서, 그림 1의 재생 모듈을 독립적으로 구현한 결과물이 볼류메트릭 비디오 플레이어이다. 따라서 플레이어는 별도의 송수신 기능 없이, 로컬(Local)에서 파일을 읽고 재생한다. 이후 읽은 파일의 종류를 판단하여 복호화가 필요한 경우, 디코더 모듈을 거쳐 원시 데이터로 복호화한다. 이때 각 압축 비트스트림에 알맞은 디코더로 복호화 과정을 거친 포인트 클라우드 데이터를 Open3D에서 지원하는 데이터 구조로 알맞게 변환시켜주는 과정이 요구된다. 복호화와 구조 변

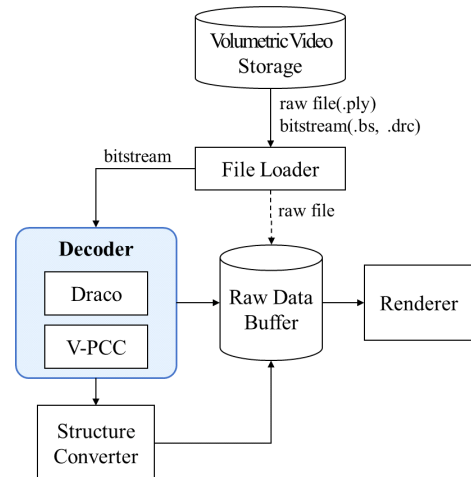


그림 3. 볼류메트릭 비디오 플레이어의 구조도
Fig. 3. Volumetric Video Player Architecture

환 과정을 거친 포인트 클라우드 데이터는 버퍼에 담겨있으며, 원하는 영상의 재생 속도에 따라 렌더러에서 동적 포인트 클라우드 객체로 구성된 볼류메트릭 비디오를 출력한다. 이러한 일련의 과정을 포함하는 볼류메트릭 비디오 플레이어의 기본 구조도는 그림 3과 같다.

3.1 볼류메트릭 비디오 저장소(Volumetric Video Storage)

로컬에 위치한 저장소(Storage)에는 압축 전 원시 데이터인 ply 파일과 Draco와 V-PCC로 압축된 비트스트림이 각각 drc 및 bs 파일로 저장되어 있다. 이때 ply 파일은 포인트 클라우드로 구성된 N개의 정적 볼류메트릭 객체가 시간 순서에 따라 저장되어 있다. 이때 N은 한 시퀀스(Sequence)를 구성하는 프레임의 수이다. 예를 들어, 300 프레임으로 구성된 redandblack 볼류메트릭 비디오는 300개의 ply 파일을 원시 데이터로 가지게 된다. drc 파일은 원시 데이터인 ply 파일들을 각각 Draco로 압축한 결과물이다. 따라서 drc 파일도 N개의 압축 비트스트림이 시간 순서로 저장되어 있으며, ply와 drc 파일은 한 개의 파일이 하나의 프레임에 해당한다. V-PCC 압축 비트스트림인 bs 파일은 N개의 ply 파일을 하나의 비트스트림으로 압축한 결과물로써 한 시퀀스당 하나의 비트스트림 파일이 존재한다.

3.2 파일 로더(File Loader) 모듈

파일 로더 모듈은 저장소로부터 원하는 파일을 읽어오고, 파일의 확장자를 토대로 디코더의 활용 여부를 판단한다. 원시 데이터는 별도의 복호화가 필요하지 않기 때문에 복호화 과정을 건너뛰고 곧바로 원시 데이터 버퍼(Raw Data Buffer)에 연결되며, 복호화가 필요한 압축 비트스트림은 디코더 모듈을 거쳐 복호화 과정을 거친 후 원시 데이터 버퍼에 포인트 클라우드 구조로 저장된다.

3.3 디코더(Decoder) 모듈

디코더 모듈은 각 압축 비트스트림에 알맞은 코덱을 활용하여 포인트 클라우드 데이터로 복호화한다. 디코더는 현재 MPEG의 V-PCC와 Google의 Draco를 지원한다. V-PCC의 테스트 모델인 TMC2 15.0 버전을 활용하였으며, 내장 비디오 코덱으로는 VVC의 테스트 모델인 VTM 13.0을 사용한다. Draco는 1.5.0 버전의 C++ 기반 디코더 API를 통해 구현하였다.

3.4 구조 변환(Structure Converter) 모듈

구조 변환 모듈은 복호화된 포인트 클라우드 데이터의 재생을 위하여 Open3D에서 지원하는 데이터 구조로 변환하는 과정을 수행한다. Open3D의 포인트 클라우드 데이터 구조는 그림 4와 같으며, 각 정보는 float64형의 3차원 벡터 형태로 저장 되어있다. 이때 포인트(Points)는 각 점이 3차원 공간에서 위치한 좌표를 나타내며, 노말(Normal)은 포인트를 정규화(Normalization)한 값이다. 그리고 컬러(Color) 정보는 각 포인트의 RGB 색상 정보를 의미한다.

반면에, Draco를 통해 부/복호화된 데이터는 Open3D와 상이한 구조를 갖는다. 그림 5에서 보여주는 Draco의 파일 포맷은 헤더와 메타데이터, Connectivity, 그리고 Attributes로 구성된다. 특히 Attributes는 특정 데이터의 속성 정보와 그 데이터를 담고 있으며, Attributes의 헤더를 통해 특정 데이터의 속성 및 구성을 확인할 수 있다. 또한 데이터의 종류에 따라 색상 정보, 텍스처의 좌표 정보, 포인트의 위치 정보 등을 속성 정보로 나타낼 수 있다. 이 중 포인트 클라우드의 필수적인 정보는 포인트의 위치 정보와 색상 정보이다. Draco는 포인트 위치 정보를 float32 형으로 표현하며, 색상정보는 uint8_t형으로 RGB 값을 저장한다. 따라서 Draco를 통해 복호화한 데이터를 그대로 활용할 수 없기 때문에, 이를 Open3D의 데이터 구조에 알맞게 변환하는 과정이 요구된다.

또한, 구조 변환 모듈에서는 데이터 구조 간의 변환을 수행하는 과정 중에, 색상 정보의 세부적인 표현이 상이한 부분을 보정하는 동작도 수행한다. 예를 들어, Draco는 포인트의 색상 정보를 0에서 255 범위의 값으로 표현하는 반면, Open3D는 이를 정규화 하여 0에

PointCloud		
Points	Normals	Colors

그림 4. Open3D의 포인트 클라우드 데이터 구조
Fig. 4. Point cloud data structure of Open3D

Draco Header	Metadata	Connectivity	Attributes
Attribute Header	Attribute Type		
	Attribute	Attribute	Attribute

그림 5. Draco의 파일 포맷
Fig. 5. File format of Draco

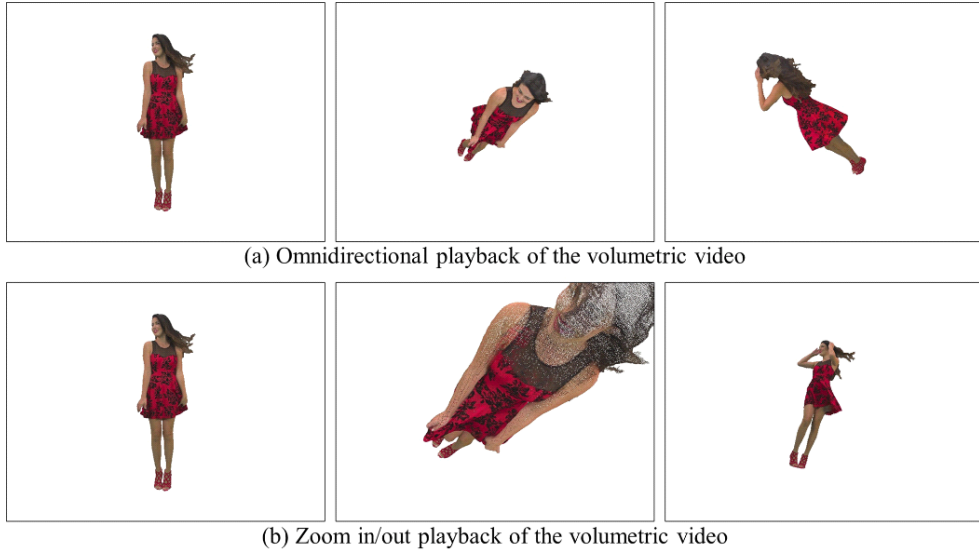


그림 6. 볼류메트릭 비디오의 재생 장면(좌측부터 각각 4번, 141번, 244번 프레임)
 Fig. 6. Playback scene of the volumetric video (From left, frame number 4, 141, and 244, respectively)

서 1 사이의 값으로 표현한다. 따라서 복호화된 색상 정보를 정규화 하는 과정이 추가적으로 필요하며, 이를 수행하는 동작이 구조 변환과정 중 함께 진행된다.

3.5 렌더러(Renderer) 모듈

렌더러 모듈은 볼류메트릭 비디오를 화면에 출력한다. 이때 볼류메트릭 비디오를 소비하는 결과 화면을 그림 6을 통해 확인할 수 있는데, 이 그림에는 redandblack 시퀀스를 재생하는 장면이 나타나 있다. 왼쪽부터 각각 4번째, 141번째, 244번째 프레임의 재생 장면을 나타내어 영상의 초반, 중반, 후반 부분의 소비 상황을 확인할 수 있다. 사용자는 마우스나 키보드를 활용하여 시점을 이동하며 그림 6(a)과 같이 전방위에서 볼류메트릭 비디오를 소비할 수 있다. 141번째 프레임의 경우 볼류메트릭 비디오의 좌측 상단부를 소비하고 있으며, 볼류메트릭 비디오의 후방부를 소비하는 모습을 244번째 프레임을 통해 확인할 수 있다. 또한 영상을 확대하거나 축소하여 소비하는 장면을 그림 6(b)의 141번째 프레임과 244번째 프레임을 통해 확인할 수 있다. 기준이 되는 4번째 프레임과 비교하였을 때 상대적으로 볼류메트릭 비디오의 크기에 변화가 있으며, 확대한 경우인 141번째 프레임의 경우 볼류메트릭 비디오를 구성하는 포인트를 확인할 수 있다.

IV. 실험 결과

볼류메트릭 비디오와 같은 몰입형 실감 미디어를 소비할 때 중요한 평가가치인 QoE(Quality of Experience)는 사용자의 몰입감을 평가하는 척도이다. 특히 지연 시간은 이러한 몰입감을 저해하는 대표적인 요소로써 영상의 재생 중 끊김을 유발한다. 이때, 지연 시간으로 인하여 비디오의 재생 중 끊김을 느끼지 않고 소비하기 위해서는 영상의 프레임 레이트(Frame rate)를 고려하여 각 프레임 사이의 거리보다 지연시간이 더 커서는 안 된다. 따라서 본 논문에서는 볼류메트릭 비디오 플레이어의 각 모듈에서 발생하는 평균 처리 시간을 측정하고 지연 시간에 가장 많은 영향을 주는 모듈을 특정하였다. 또한, 성능 평가를 위한 데이터 셋으로 8i 사에서 제공하는 redandblack 시퀀스를 활용하였으며^[16], 실험에 활용한 컴퓨터 및 구

표 1. 실험 환경 정보
 Table 1. Environmental Information for the Experiments

Hardware	CPU	Intel i7-10700K
	RAM	64GB
	GPU	NVIDIA GTX 1080
Software	GCC	C++ 17
	V-PCC	15.0 (with VTM 13.0)
	Draco	1.5.0
	Open3D	0.14.1

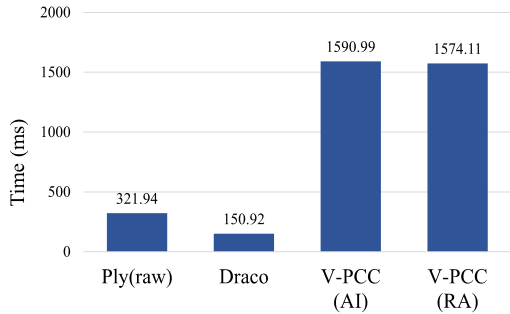


그림 7. 프레임 당 평균 처리 시간(ms)
Fig. 7. Average processing time per frame(ms)

현 환경에 대한 상세정보는 표 1에 기술되어 있다.

그림 7은 원시 데이터 및 압축 비트스트림의 한 프레임 당 처리에 걸리는 평균 시간을 ms 단위로 측정 한 결과이다. 원시 데이터는 별도의 복호화 및 구조 변환 과정을 거치지 않기 때문에 한 프레임을 처리하기 위해 소비되는 시간이 파일을 읽는 시간과 동일하며, 평균적으로 약 321 ms가 걸린다.

Draco를 이용하여 압축한 비트스트림의 경우에는 한 프레임의 평균 처리 시간이 약 150 ms로 측정되었으며, 이는 원시 데이터의 프레임 당 평균 처리시간의 약 50%에 해당한다. 반면, V-PCC를 이용하여 압축한 비트스트림의 평균 프레임 처리 시간은 AI(All Intra) 모드인 경우 약 1,590 ms, RA(Random Access) 모드인 경우 약 1,574 ms가 소요되었다. 이때, Draco 및 V-PCC AI 모드, V-PCC RA 모드의 실험 데이터 셋에 대하여 압축된 비트스트림을 읽는데 걸리는 지연 시간은 1 ms 이하로 측정되었으며, 구조를 변환하는데 평균적으로 44 ms의 지연시간이 발생했다.

실험에서 측정된 각 모듈의 처리시간을 통해 V-PCC에 비하여 Draco가 압축된 비트스트림의 평균 프레임 처리 시간이 매우 짧은 것을 확인할 수 있다. 하지만 전송 및 저장 과정에서 지연 시간이 발생할 수 있음을 고려했을 때, 복호화기의 압축 성능이 사용자의 몰입감에 영향을 줄 수 있다. 그림 8은 본 논문에서 사용한 각 복호화기의 데이터 압축비를 계산하여 비교한 그래프이다. 이때, 압축비 계산은 원시 데이터의 크기를 압축된 비트스트림의 크기로 나누어 계산한다.

4.45GB 크기의 redandblack 원시 데이터 시퀀스를 Draco와 V-PCC AI 모드, V-PCC RA 모드로 부호화한 데이터의 크기는 각각 541MB, 7,873KB, 4,242KB였으며, Draco에 비하여 V-PCC가 높은 성능을 가지는 것을 확인할 수 있다. 특히 V-PCC의 RA모드에서

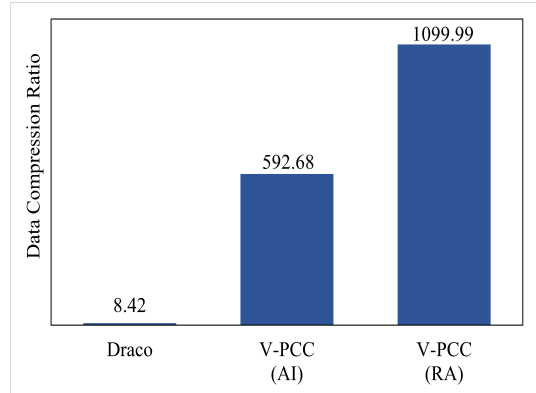


그림 8. 데이터 압축비
Fig. 8. Data compression ratio

는 AI 모드에 비해 두 배 높은 압축 성능을 보였다. 실험 결과를 통해 Draco와 달리 V-PCC에 적용된 시간적 중복성을 고려한 압축 기술이 압축 성능에 큰 영향을 미쳤다는 것을 확인할 수 있다.

결과적으로 복호화 과정과 구조 변환 과정에서 가장 많은 지연이 발생하며, 해당 지연 시간을 줄이지 않고 볼류메트릭 비디오를 정상적으로 재생하기에는 무리가 있다. 따라서 볼류메트릭 비디오를 안정적으로 소비하기 위해서는 초기에 충분한 지연 시간을 두고 복호화된 볼류메트릭 비디오를 버퍼에 확보하여 프레임 레이트를 유지할 수 있어야 한다. 또한 시간적 중복성을 제거하는 압축 기술의 추가 또는 최적화 작업을 통해 복호화로 인한 지연시간을 최소화하고 압축 성능을 극대화할 수 있는 기술 개발이 필요하다.

V. 결 론

본 논문에서는 6-DoF 환경에서 볼류메트릭 비디오를 재생할 수 있는 플레이어를 설계 및 구현하였다. 또한 볼류메트릭 비디오 플레이어를 구성하는 각 모듈의 수행 시간을 프레임 단위로 측정함으로써 볼류메트릭 비디오의 안정적 소비를 저해하는 모듈을 특정하였다. 그 결과 디코더 모듈과 구조 변환 모듈에서 지연이 발생하는 것을 확인하였고, 이를 개선하기 위하여 플레이어의 초기 지연시간을 두거나 디코더의 성능을 개선하는 방안을 논의하였다. 향후 볼류메트릭 비디오의 압축 및 전송 기술의 개선을 통해 사용자의 몰입감을 고려한 볼류메트릭 비디오 전송 시스템을 구축하고자 한다.

References

- [1] Y. H. Seo, "Volumetric photorealistic 4D video technology," *Broadcasting and Media Mag.*, vol. 26, no. 2, pp. 56-66, Apr. 2021.
- [2] Google, "Draco: 3D Data Compression," 2018. Retrieved Apr. 20, 2022, from <https://github.com/google/draco>
- [3] 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)."
- [4] ISO/IEC DIS 23090-9, "Coded representation of immersive media – Part 9: Geometry-based point cloud compression."
- [5] M. Hosseini and C. Timmerer, "Dynamic adaptive point cloud streaming," in *Proc. 23rd Packet Video Wkshp.*, pp. 25-30, 2018. (<https://doi.org/10.1145/3210424.3210429>)
- [6] J. van der Hoof, T. Wauters, F. De Turck, C. Timmerer, and H. Hellwagner, "Towards 6DoF HTTP adaptive streaming through point cloud compression," in *Proc. 27th ACM Int. Conf. Multimedia*, pp. 2405-2413, 2019. (<https://doi.org/10.1145/3343031.3350917>)
- [7] S. Gül, D. Podborski, T. Buchholz, T. Schierl, and C. Hellge, "Low-latency cloud-based volumetric video streaming using head motion prediction," in *Proc. 30th ACM Wkshp. Netw. and Oper. Syst. Support for Digital Audio and Video*, pp. 27-33, 2020. (<https://doi.org/10.1145/3386290.3396933>)
- [8] 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," in *Proc. 11th ACM Multimedia Syst. Conf.*, pp. 357-360, 2020. (<https://doi.org/10.1145/3339825.3393583>)
- [9] S. Pacheco-Gutierrez, I. Caliskanelli, and R. Skilton, "Point cloud compression and transmission for remote handling applications," *J. Software*, vol. 16, no. 1, pp. 14-23, Jan. 2021. (<https://doi.org/10.17706/jsw.16.1.14-23>)
- [10] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC)," *APSIPA Trans. Sign. and Inf. Process.*, vol. 9, no. 13, pp. 1-17, Apr. 2020. (<https://doi.org/10.1017/ATSIP.2020.12>)
- [11] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, and J. Llach, "Emerging MPEG standards for point cloud compression," *IEEE J. Emerging and Sel. Topics in Cir. and Syst.*, vol. 9, no. 1, pp. 133-148, Dec. 2018. (<https://doi.org/10.1109/JETCAS.2018.2885981>)
- [12] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," *IEEE ICRA*, Shanghai, China, May 2011. (<https://doi.org/10.1109/ICRA.2011.5980567>)
- [13] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," arXiv preprint arXiv:1801.09847, 2018. (<https://doi.org/10.48550/arXiv.1801.09847>)
- [14] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," *2016 IEEE ISM*, pp. 107-110, 2016. (<https://doi.org/10.1109/ISM.2016.0028>)
- [15] Z. Xu, X. Zhang, K. Zhang, and Z. Guo, "Probabilistic viewport adaptive streaming for 360-degree videos," in *2018 IEEE ISCAS*, pp. 1-5, 2018. (<https://doi.org/10.1109/ISCAS.2018.8351404>)
- [16] M. Krivokuća, P. A. Chou, and P. Savill, "8i voxelized surface light field (8iVSLF) dataset," ISO/IEC JTC1/SC29 WG11 (MPEG) input document m42914, Ljubljana, Jul. 2018.

김 아 영 (A-young Kim)



2016년 2월 : 연세대학교 미래
캠퍼스 컴퓨터정보통신공학
부 학사

2016년 3월~현재 : 연세대학교
전산학과 석박통합과정
<관심분야> 영상부호화, 영상통
신, 실감미디어 통신시스템

[ORCID:0000-0002-3793-1365]

안 은 빈 (Eun-bin An)



2016년 8월 : 연세대학교 미래캠
퍼스 컴퓨터정보통신공학부
학사

2017년 3월~현재 : 연세대학교
전산학과 석박통합과정
<관심분야> 영상부호화, 영상통
신, 실감미디어 통신시스템

[ORCID:0000-0001-7681-4682]

서 광 덕 (Kwang-deok Seo)



1996년 2월 : KAIST 전기 및
전자공학과 학사

1998년 2월 : KAIST 전기 및
전자공학과 석사

2002년 8월 : KAIST 전기 및
전자공학과 박사

2002년 8월~2005년 2월 : LG
전자 단말연구소 선임연구원

2012년 9월~2013년 8월 : Courtesy Professor, Univ.
of Florida, USA

2005년 3월~현재 : 연세대학교 소프트웨어학부 교수
<관심분야> 영상부호화, 영상통신, 디지털 방송, 멀
티미디어 통신시스템

[ORCID:0000-0001-5823-2857]