

MMT 기반 V3C 데이터 패킷화 방안

문형준, 김연웅, 박성환, 남귀중 *김규현

경희대학교

98015@khu.ac.kr, rladusdnd135@khu.ac.kr, gocheenee@khu.ac.kr, nkj0427@khu.ac.kr

*kyuhyeonkim@khu.ac.kr

MMT based V3C data packetizing method

Hyeongjun Moon, Yeonwoong Kim, Seonghwan Park, Kwijung Nam *Kyuhyeon Kim

Kyunghee University

요약

3D Point Cloud는 3D 콘텐츠를 더욱 실감 나게 표현하기 위한 데이터 포맷이다. Point Cloud 데이터는 3차원 공간상에 존재하는 데이터로 기존의 2D 영상에 비해 거대한 용량을 가지고 있다. 최근 대용량 Point Cloud의 3D 데이터를 압축하기 위해 V-PCC(Video-based Point Cloud Compression)와 같은 다양한 방법이 제시되고 있다. 따라서 Point Cloud 데이터의 원활한 전송 및 저장을 위해서는 V-PCC와 같은 압축 기술이 요구된다. V-PCC는 Point Cloud의 데이터들을 Patch로써 뜯어내고 2D에 Projection 시켜 3D의 영상을 2D 형식으로 변환하고 2D로 변환된 Point Cloud 영상을 기존의 2D 압축 코덱을 활용하여 압축하는 기술이다. 이 V-PCC로 변환된 2D 영상은 기존 2D 영상을 전송하는 방식을 활용하여 네트워크 기반 전송이 가능하다. 본 논문에서는 V-PCC 방식으로 압축한 V3C 데이터를 방송망으로 전송 및 소비하기 위해 MPEG Media Transport(MMT) Packet을 만드는 패킷화 방안을 제안한다. 또한 Server와 Client에서 주고받은 V3C(Visual Volumetric Video Coding) 데이터의 비트스트림을 비교하여 검증한다.

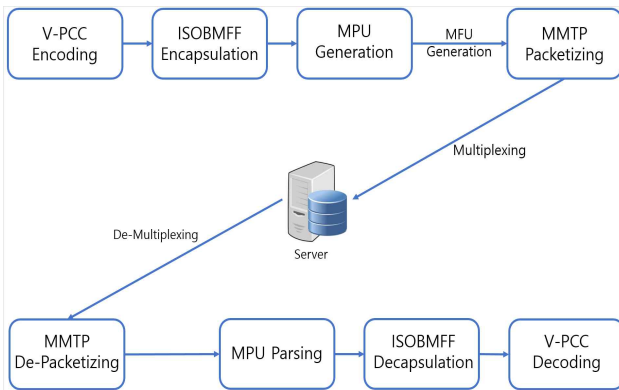
1. 서론

최근 실감 나는 3D 영상 표현을 위해 3D Point Cloud 데이터를 활용하는 방식이 활발하게 개발되고 있다. Point Cloud 데이터는 3차원의 객체를 3개의 좌표로 각 좌표에 대한 색상 정보까지 포함한 데이터 포맷이다. Point Cloud는 건축 분야, 자율주행, 메타버스 그리고 AR/VR[1][2] 등 다양한 분야에 사용되고 있다. Point Cloud 영상은 대용량이므로 이를 활용하기 위해서는 Point Cloud를 압축하는 기술이 필요하다. 이에 따라, 국제 표준화 기구인 MPEG(Moving Picture Experts Group)에서는 3D Point Cloud를 기존의 2D 코덱을 활용하여 압축하는 기술인 V-PCC[3] 표준을 제정하였다. V-PCC 기술은 3D Point Cloud 데이터를 Patch 별로 나누어 2D 영상으로 Projection 한 뒤, 기존 2D 영상 코덱을 활용하여 압축한다. 압축된 V-PCC 데이터는 V3C 비트 스트림으로 Point Cloud 데이터의 점의 유무에 대한 정보를 담고 있는 Occupancy, 3차원 점의 깊이 대한 정보를 담고 있는 Geometry 그리고 점의 색상 정보를 갖는 Attribute 유닛들로 이루어져 있다. 이 압축된 V-PCC 데이터를 효율적으로 저장하고, 전송하기 위해 ISOBMFF(ISO Base Media File Format)[4] 형식으로 캡슐화할 수 있다. 캡슐화된 3D Point Cloud 콘텐츠는 기존의 2D 영상 콘텐츠처럼 네트워크를 기반으로 전송하여 소비하기 위해 DASH[5], MMT[6]와 같은 전송 포맷을 사용해야 한다. MMT는 다양한 전송 환경에서 호환성 있게 전송이 가능하다는 특성으로 인해 다양한 환경에서 V3C 데이터를 효과적으로 서비스할 수 있을 것으로 사료된다. 따라서 본 논문

에서는 V3C 데이터를 MMT로 활용하여 전송하기 위한 MPU 생성 및 MMT 패킷화 방안을 제안하고, 제안된 방안을 검증하기 위한 검증 실험을 진행하고자 한다.

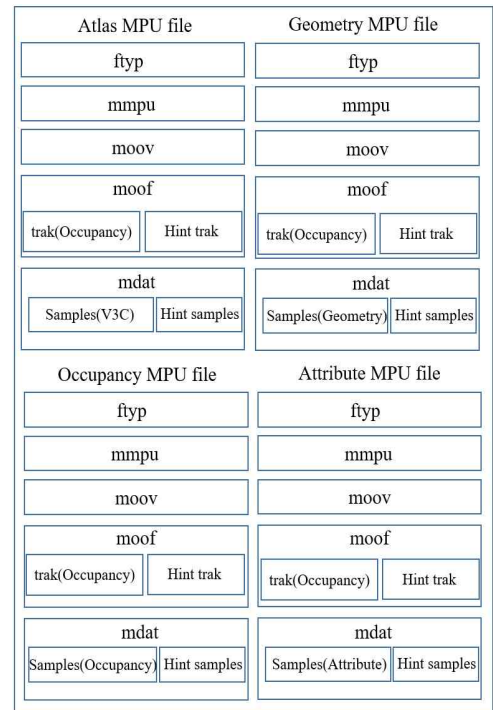
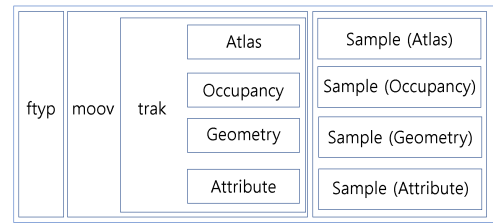
2. 본론

V-PCC를 통해 압축된 V3C 데이터는 저장된 V3C 비트스트림은 VPS, Atlas, Occupancy, Geometry 그리고 Attribute 유닛으로 구성되어 있으며 Carriage of V3C 표준에 따라 ISOBMFF 형식으로 캡슐화될 수 있다.[7] V3C 데이터의 유닛들은 ISOBMFF 캡슐화 과정에서 각각 트랙으로 구성된다. VPS 유닛과 Atlas 유닛은 V3C Atlas 트랙을 구성하며, V3C Atlas 트랙은 Sample Entry box에 파라미터 유닛을, 'mdat' box에 Atlas 패치 데이터를 포함한다. 또한 V3C Atlas 트랙에는 Track reference box가 존재하여, V3C 비디오 트랙과 동기화 정보를 제공한다. 다른 V3C 비디오 트랙인 Occupancy, Geometry 그리고 Attribute 유닛은 기존 비디오 코덱의 ISOBMFF 캡슐화 방식과 같은 구조를 가진다.



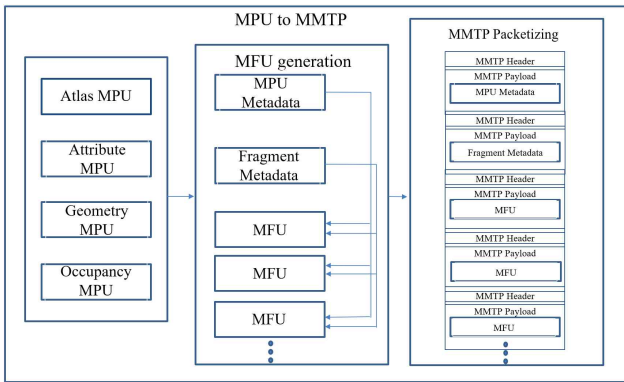
[그림 1] MMT를 이용한 V3C 데이터 송수신 방안

[그림 1]은 본 논문에서 제안하는 MMT 기반 V3C 데이터의 패킷화 방안 및 검증 방안의 구조도를 나타낸 것이다. V-PCC 방식으로 압축된 Point Cloud 데이터는 V3C 비트 스트림 형태로 저장된다. 인코딩된 V3C 비트스트림은 ISOBMFF 형식으로 캡슐화되며 MPU Generator를 통해 MPU를 생성할 수 있다. 생성된 MPU는 MMTP 패킷화를 거쳐 Multiplexing을 통해 Server에 전송된다. 전송된 MMTP 패킷은 검증 실험을 통해 데이터의 손실 없이 전송됨을 확인할 수 있다. 제안된 검증 실험은 MMTP De-packetizing과 MPU Parsing, ISOBMFF Decapsulation 모듈을 통해 V3C 비트스트림으로 복원한 뒤, V-PCC 디코딩을 통해 포인트 클라우드 데이터를 확인하는 방법으로 진행된다.



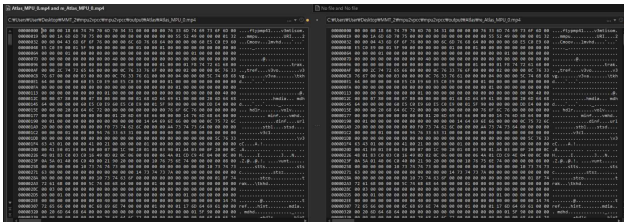
[그림 2] 캡슐화된 V3C 데이터의 MPU 생성

[그림 2]와 같이 ISOBMFF의 데이터 형식에 존재하는 Atlas 트랙과 비디오 트랙들인 Occupancy, Geometry 그리고 Attribute 트랙은 각각 MPU 파일을 생성한다. MPU 파일을 생성하기 위해선 MPU Metadata에 MPU ID정보를 포함하는 'mmpu' box의 추가적인 생성이 필요하며, Fragmented ISOBMFF 파일 구조로 변경하기 위해 'moof' box를 생성해야 한다. 이 MPU에는 MMTP 패킷화를 위해 MFU들의 경계 정보를 제공하는 hint track이 포함됐다. 이 MPU는 다음 과정을 통해서 MMTP 패킷을 생성할 수 있다.

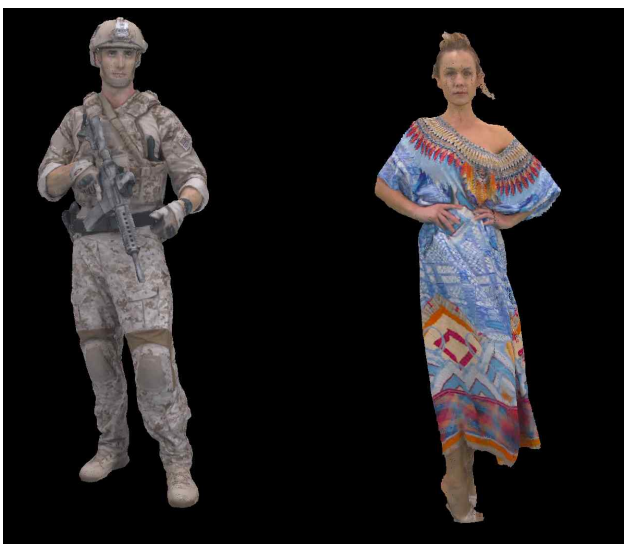


[그림 3] V3C 데이터 MMTP 패킷화 과정

MMTP 패킷을 만들기 위해선 Media Fragment Unit(MFU)를 생성해야 한다. 이 MFU는 비디오 코덱에서 생성된 비디오 Access Unit(AU)과 같거나 더 작은 단위를 의미한다. [그림 3]과 같이 MPU Metadata, Fragment Metadata 그리고 MFU를 생성한 뒤, 각각의 패킷에 header를 붙여 패킷을 생성한다. 이때 MPU Metadata는 'ftyp', 'mmpu' box와 'moov' box의 정보를 포함한다. Fragment Metadata는 'moof' box의 정보와 'mdat' box의 size에 대한 정보를 포함한다. MFU는 'mdat' box에 있는 hint sample과 미디어 sample의 1대1 대응으로 생성된다. 생성된 MMTP 패킷은 Multiplexing을 진행한 뒤 전송한다.



[그림 4] 비트스트림 전송 전후 비교



[그림 5] V3C 데이터 복원 결과

전송된 MMTP 패킷은 본 논문에서 제안된 검증 실험을 통해서 V3C 비트스트림을 MMT를 이용하여 전송할 데이터의 손실이나 누락되는 부분이 없는지 확인했다. [그림 4]처럼 재구성된 V3C 비트스트림

데이터를 Binary compare 프로그램을 이용해 확인해보았다. V3C 비트스트림을 MPU로 모두 재구성 했을 때, 기존의 MPU와 V3C 데이터가 동일한 것을 확인할 수 있다. ISOBMFF 형식의 V3C 비트스트림은 차이가 있었지만 누락된 데이터가 없는 것을 확인했다. [그림 5]의 렌더링 된 결과 통해 MMT을 이용한 서비스가 가능한 것을 확인했다. 본 실험은 V3C 비트스트림 데이터를 MPEG에서 제공된 Open V3C를 통해 렌더링을 진행했다. 본 논문을 통해 V3C 데이터가 MMT로 서비스 될 수 있는 것을 확인했다.

3. 결론

본 논문에서는 MMTP Packetizing을 이용하여 V-PCC로 압축된 V3C 데이터를 전송하는 과정을 검증했다. 그 결과 ISOBMFF 기반의 V-PCC로 압축된 데이터가 MPU와 MMTP 패킷으로 생성되어 수신 단에 MMTP 패킷을 Multiplexing을 이용하여 전송됐다. 전송된 데이터도 디코딩 과정을 통해 검증해본 결과 V3C 데이터가 문제없이 전송된 것을 확인할 수 있었다.

참고문헌

- [1] Rusinkiewicz, S. and Levoy, M. 2000. QSplat: a multiresolution point rendering system for large meshes. In Siggraph 2000. ACM, New York, NY, 343-352. DOI=<http://doi.acm.org/10.1145/344779.344940>
- [1] "Ubiquitous AR to dominate focused VR by 2022," <https://techcrunch.com/2018/01/25/ubiquitous-ar-to-dominate-focused-vr-by-2022/>, accessed: 2019.
- [2] "Creating 3D Content with Reality Composer," https://developer.apple.com/documentation/realitykit/creating_3d_content_with_reality_composer, accessed: 2019.
- [3] "Text of ISO/IEC CD 23090-5 Video-based Point Cloud Compression," ISO/IEC JTC1/SC29/WG11 MPEG2019/N18670, Gothenburg, Sweden, Oct. 2019.
- [4] ISO/IEC 14496-12-MPEG-4 Part 12, ISO bas media file format, 2014.07
- [5] "Text of ISO/IEC 23009-1 Dynamic adaptive streaming over HTTP(DASH)", ISO/IEC JTC1/SC29/WG11 MPEG2017/N17559, San Diego, USA, Oct. 2017.
- [6] "Text of ISO/IEC 23008-1 MPEG Media Transport", ISO/IEC JTC1/SC29/WG11 MPEG2019/N18427, Geneva, Sweden, March. 2019.
- [7] ISO/IEC JTC 1/SC 29/WG 07/N0004, "Text of ISO/IEC FDIS 23090-9 Geometry-based Point Cloud Compression," Oct. 2020.