

V-PCC 부호화기의 그리드 기반 세그먼트 정제 고속화

김유라, 김용환

한국전자기술연구원 지능형영상처리연구센터
keyura@keti.re.kr, yonghwan@keti.re.kr

Fast Grid-Based Refine Segmentation on V-PCC encoder

Yura Kim, Yong-Hwan Kim,

Intelligent Image Processing Research Center, Korea Electronics Technology Institute

요 약

Video-based Point Cloud Compression(V-PCC) 부호화기의 세그먼트 정제(Refining segmentation) 과정은 3D 세그먼트를 2D 패치 데이터로 효율적으로 변환하기 위한 V-PCC 부호화기의 핵심 파트이지만, 많은 연산량을 필요로 하는 모듈이다. 때문에 이미 TMC2 에 Fast Grid-based refine segmentation 과정이 구현되어 있으나, 아직도 세그먼트 정제 기술의 연산량은 매우 높은 편이다. 본 논문에서는 현재 TMC2 에 구현되어 있는 Fast Grid-based Refine Segmentation 을 살펴보고, 복셀(Voxel) 타입에 따른 특성에 맞춰 두 가지 조건을 추가하는 고속화 알고리즘을 제안한다. 실험 결과 압축성능(BD-BR)은 TMC2 와 거의 차이를 보이지 않았지만, 모듈 단위 평균 10% 연산량이 절감되는 것을 확인하였다.

1. 서론

2021 년 ISO/IEC MPEG-I Part 5 로 표준화된 V-PCC[1]는 2D 비디오 압축 방식을 활용한 3D 포인트 클라우드 데이터의 압축방식이다. 최근 들어 다양한 실감형/몰입형 디바이스들이 출시되면서 3D 데이터를 이용한 서비스 수요가 증가하고 있는데, 3D 데이터의 데이터량은 그 크기가 실시간 전송이 불가능 할 정도로 커서 V-PCC 와 같은 3D 데이터 압축 기술이 필수적이다. 특히 V-PCC 는 이미 시장에 보급되어 있는 H.264 나 HEVC 와 같은 비디오 코덱을 활용하여 3D 포인트 클라우드 데이터를 압축 하는 구조[2]이기 때문에 이를 활용하여 빠르게 시장에서 자리 잡을 수 있는 3D 데이터 압축 방식으로 평가 되고 있다.

하지만 3D 포인트 클라우드 데이터를 2D 형태의 패치 데이터로 변환하기 위해 여러 단계와 많은 연산이 필요하다. 그림 1. 에서는 V-PCC 부호화기에서 3D 포인트들을 2D 패치로 변환을 하는 과정은 개념적으로 확인할 수 있다. 우리는 V-PCC 의 2D 패치 생성 과정 중 세그먼트 정제(Refining Segmentation)에

대해 살펴보고, 세그먼트 정제 과정의 연산량 절감을 위한 방법을 제시하고 이에 대한 실험 결과를 보여주려 한다.

본 논문의 구성은 다음과 같다. 2 장에서는 세그먼트 정제 과정과 V-PCC 레퍼런스 소프트웨어인 TMC2[3]에 구현된 Fast Grid-Based Refine Segmentation(FGBRS) 알고리즘에 대해 설명한다. 3 장에서 FGBRS 알고리즘에서 사용된 복셀(Voxel) 타입에 대한 특징을 살펴 보고 이 특징을 이용한 고속화 방법을 제안한다. 4 장에서는 제안된 방법으로 V-PCC 부호화 실험 결과를 확인하고, 5 장에서 결론을 제시한다.

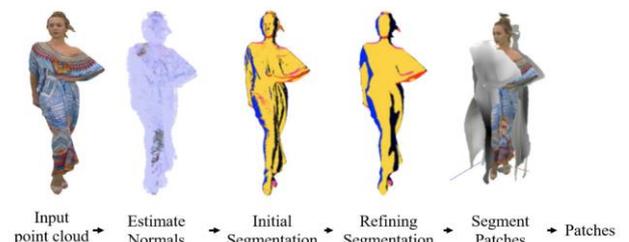


그림 1. V-PCC 의 2D 패치 생성 과정

2. V-PCC 의 세그먼트 정제

그림 1 에서 확인 수 있는 3D 포인트클라우드에서 2D 패치를 만드는 주요 과정을 다음과 같다.

- 1) 세그먼트 초기화(Initial Segmentation): 모든 포인트에 대해 각 포인트가 어느 투영면에 투영 될 지를 결정한다.
- 2) 세그먼트 정제(Refine segmentation): 각 포인트의 투영면 정보를 주변 포인트의 투영면 정보를 바탕으로 업데이트 한다.
- 3) 세그먼트 패치(Segment Patches): 동일한 투영면 정보를 가진 포인트들(Connected Component)을 모아 3D 패치를 구성한다.

세그먼트 초기화에서 포인트의 투영면을 결정할 때에는, 각 포인트의 법선을 미리 구한 후 이 값을 이용해 해당 포인트가 어느 투영면에 투영되는 것이 좋을 지를 결정한다. 이렇게 결정된 초기 투영면 인덱스(Projection Plane Index, PPI)는 오직 해당 포인트의 법선 값에 의해 결정되므로, 주변 포인트들의 PPI 와 연속성이 떨어져 세그먼트 패치 단계에서 너무 많은 3D 패치들을 생성하게 된다.

세그먼트 정제 과정은 이러한 단점을 극복하기 위해 인접한 주변 포인트들의 PPI 정보들을 이용하여 각 포인트의 PPI 를 평활화(smoothing)하는 과정이다. 이 과정에서는 법선 값뿐만 아니라 주변 포인트들의 PPI 정보를 이용해 가중된 새로운 값을 이용하여 PPI 를 업데이트 한다. 이러한 PPI 업데이트 과정을 반복적으로 수행하면, 3D 패치 생성시 더 많은 포인트들이 하나의 패치가 되고, 결과적으로 전체 패치의 수를 줄일 수 있게 된다.

V-PCC 초기 anchor 소프트웨어에서는 모든 3D 포인트에 대해 정제를 수행하고, 이를 외부에서 설정한 수(maxIterCount)만큼 반복 수행하도록 설계되었다. MPEG 에서 사용하고 있는 테스트 시퀀스들이 각 frame 마다 70 만개~200 만개의 포인트로 구성되어 있는데, 매 frame 의 각 포인트를 대상으로 정제를 반복 수행하는 매우 복잡한 과정이었다.

128 차 MPEG 미팅에서 3D 포인트 클라우드를 그리드 단위(Voxel)로 묶어 정제를 진행하는 Grid-Based Refine Segmentation (GBRS)[5]이 제안되었다. GBRS 는 기존 포인트 단위 세그먼트 정제 대비 최고 80% 정도의 정제 시간 절감을 이루었고, 표준에 채택되어 TMC2 에 적용되었다. 다시 134 차 MPEG 미팅에서 복셀의 특성을 이용한 고속화 알고리즘(Fast Grid-Based Refine Segmentation)[6]이 제안되었다. FGBRS 는 GBRS 에 비해서도 평균 60% 연산량 절감을 이루었다. FGBRS 는 TMC2 부호화기에 채택되었고, TMC2 v14 버전 이후 기본 옵션으로 사용되고 있다.

그림 2 는 TMC2 에 구현되어 있는 FGBRS 의 순서도이다.

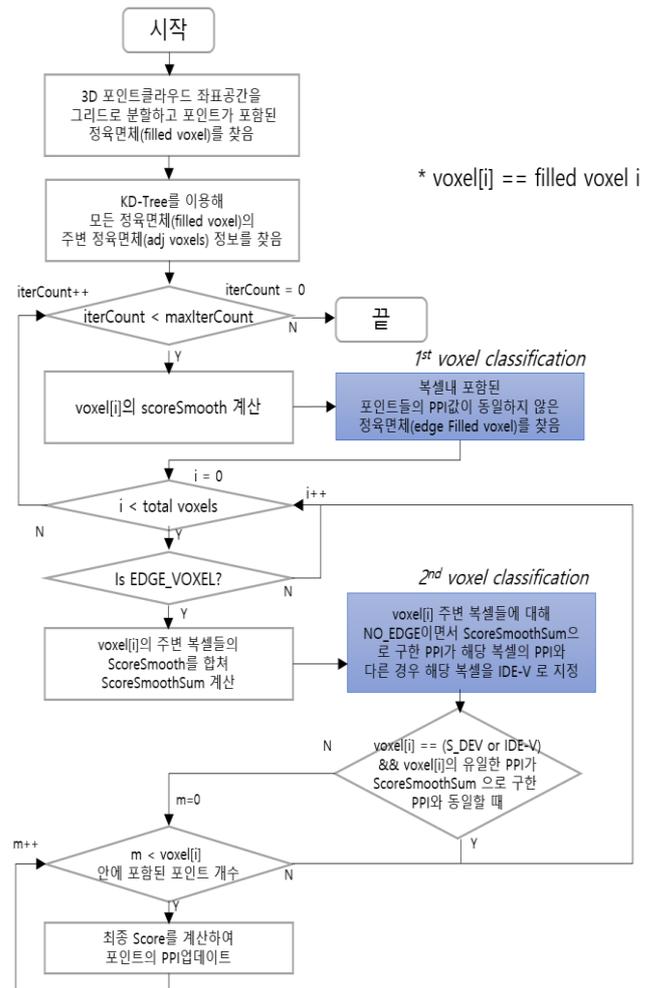


그림 2. Fast Grid-Based Refine Segmentation 순서도

FGBRS 의 핵심은 복셀 내의 포인트들이 이미 모두 같은 PPI 를 가지고 있어 평활하다고 판단되는 복셀(no edge-voxel, NE-V)에 대해서는 정제를 진행하지 않는 것이다. 실제로 복셀 내의 모든 포인트들의 PPI 가 동일하고, 일정거리 안의 주변 포인트들의 최다 발생 PPI 값과 복셀 내 포인트의 PPI 가 동일할 경우 정제를 수행해도 PPI 가 변경되는 경우는 거의 없다

FGBRS 알고리즘에서 복셀 분류방식은 그림 3 에서 확인할 수 있다. 크게 복셀 내의 포인트들의 PPI 가 모두 같은 경우를 NE-V, 복셀 내 포인트들의 PPI 값이 서로 다른 경우를 Direct Edge Voxel(DE-V)으로 분류한다. 또, DE-V 근접 복셀들 중에 NE-V 로 분류되었지만, 일정거리 안의 주변 복셀 내 포인트들의 최다 PPI 값과 현재 DE-V 복셀 내 포인트들의 최다 PPI 값이 다른 경우 해당 주변 복셀을 Indirect edge voxel(IDE-V)로 분류한다. 또 포인트를 1 개만 가지는 복셀(S-DEV)들이 존재하는데, 이 복셀 역시 예외로 분류한다.

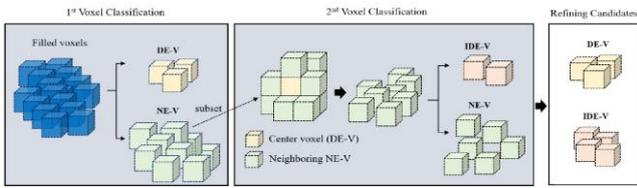


그림 3. FGBRS 알고리즘의 복셀 분류 방식 [6]

이와 같이 복셀의 특성에 따라 에지로 판정된 일부 복셀에 포함되는 포인트들의 PPI 만 업데이트를 진행하는데, 평균적으로 첫 iteration 에서는 약 30%~50%의 복셀만이 정제되고, 마지막 iteration 에서는 12%~27%의 복셀만이 정제된다.

3. 제안하는 고속 FGBRS 방법

세그먼트 정제과정은 2 장에서 설명한 것과 같이 외부에서 지정한 반복횟수(maxIterCount)만큼 반복적으로 수행 되는데, S-DEV 는 한번 지정되면 마지막 반복작업을 할때까지 항상 에지로 분류되는 복셀이다. 때문에 S-DEV 는 매번 PPI 업데이트 여부를 판단하게 되는데, 전체 복셀 대비 5%~13%로 적지 않은 수의 복셀이 이에 해당된다. 다음 표 1 은 테스트 시퀀스들의 처음 32 프레임에 대한 평균 filled Voxel 및 S-DEV 개수를 보여준다. 이때 voxelDimension 값은 Loot, Red and black, Soldier, Queen 은 2 이고, 나머지 시퀀스는 4 이다.

표 1. 복셀 타입별 개수 및 비율

시퀀스	filled Voxel 개수	S-DEV 개수	S-DEV 비율(%)	마지막 정제수행시 NE-V 개수	NE-V 비율(%)
Loot	214995	25583	11.9	157142	73.1
Red and black	196311	26429	13.5	135033	68.8
Soldier	295128	41053	13.9	204447	69.3
Queen	249351	28506	11.4	171659	68.8
Longdress	57440	3521	6.2	45792	79.7
Basketball_player	201232	11330	5.6	172279	85.6
Dancer	182020	11028	6.1	151156	83.0



(a) 흰점: S-DEV 에 해당하는 포인트들 (b) 흰점: 마지막 iteration 에서 PPI 가 업데이트 되는 포인트들

그림 4. MPEG-I 테스트 시퀀스 Loot

그림 4 의 (a) 모델을 보면 S-DEV 에 해당하는 포인트들을 나타내는 흰점이 모델 전체적으로 분포하는데, 이 S-DEV 들은 매 iteration 에서 정제를 수행하게 되지만, 정작 그림 4 의 (b)와 같이 정제 후 PPI 가 변하는 포인트들을 살펴보면 S-DEV 와 연관성이 적음을 알 수 있다. 또한 실험을 통해 S-DEV 로 지정된 복셀의 경우 일정거리 안의 주변 복셀 내 포인트들의 최다 발생 PPI 값과 S-DEV 의 PPI 값이 같다면 PPI 값이 거의 변하지 않는다는 것을 확인 할 수 있었다.

다음으로 그림 2 의 순서도에서 확인할 수 있듯이 각 iteration 단계 초반에서는 정제과정을 통해 변경된 PPI 값을 적용하기 위해 모든 복셀에 대해 scoreSmooth 값을 계산한다. 그러나 이전 iteration 단계에서 NE-V 로 지정되어 정제를 수행하지 않아 PPI 변경이 없던 복셀들은 변경된 PPI 값이 없으므로 굳이 scoreSmooth 값을 다시 계산할 이유가 없다. 그럼에도 현재 구현된 방식에서는 이를 다시 계산하고 있다. 표 1 의 마지막 정제 수행시 NE-V 비율을 확인 해보면 68~85% 정도의 복셀이 불필요한 연산을 하고 있다는 것을 알 수 있다. 단순 반복문과 증가 연산이므로 연산량 자체는 크지 않으나, 모든 포인트에 대한 임의 메모리 접근이 필요한 부하가 큰 과정이므로, 이를 개선하기 위해 복셀 내 포인트의 PPI 업데이트 여부를 확인하는 것이 필요하다.

위에 언급한 문제점들을 해결하기 위해, 본 논문에서는 2 가지 조건을 추가하였다.

- 1) S-DEV 의 PPI 와 주변 복셀 내 포인트들의 최다 발생 PPI 가 동일하다면, NE-V 로 판단한다.
- 2) 이전 iteration 에서 업데이트되지 않은 복셀이라면, 현재 iteration 에서 scoreSmooth 값을 다시 계산하지 않는다.

2 가지 조건을 추가하여도 V-PCC 의 압축성능에는 큰 차이가 없었으며, 평균 10% 정도의 정제 연산 복잡도를 줄일 수 있었다.

4. 실험결과

실험 환경은 다음과 같다.

- HW: Intel® Xeon® Gold6242R CPU @3.10GHz, Memory 256GB
 - OS: Windows 10 x64
 - TMC2 Version : Tag. v17-release(2022 년 3 월 7 일 commit)
- 실험에는 V-PCC 성능평가에 사용하는 MPEG 의 공식적인 7 개 시퀀스와 Common-Test-Condition 조건 및 C2AI(All-Intra) 와 C2RA(Random-Access)조건을 사용하였다. 표 2~4 의 실험은 32 프레임 기준이다.

표 2. 세그먼트 정제 과정 평균 수행 시간

시퀀스 / maxIterationCount	original (ms)	Proposed (ms)	시간감소율 (%)
Loot / 10	8143.34	7727.50	5.11
Redandblack / 10	8328.31	7725.06	7.24
Soldier / 10	12347.47	11525.75	6.65
Queen / 10	12385.31	11700.16	5.53
Longdress / 50	2204.53	1789.34	18.83
basketball_player/20	5039.50	4373.50	13.22
Dancer / 20	4635.44	3998.91	13.73
평균			10.05

표 2 은 각 시퀀스 별로 세그먼트 정제 과정의 평균 수행시간을 보여준다. 세그먼트 정제과정은 AI 나 RA 조건, QP 조건등에 영향을 받지 않는 과정이므로, 시퀀스만 구분하여 측정하였다. 측정 결과 최소 5% 에서 최대 18% 수행시간이 줄어드는 것을 확인 할 수 있었다. 특히 maxIterationCount 값이 50 인 longdress 시퀀스의 경우 2 번째 제안된 조건으로 인해 18%의 큰 폭으로 수행시간이 절감되는 효과가 있었다.

표 3 All-Intra 실험결과

시퀀스	Geom. BD-TotalRate [%]		End-to-End BD-TotalRate [%]		
	D1	D2	Luma	Chroma Cb	Chroma Cr
loot	-0.03	0.01	0.04	0.12	0.30
redandblack	0.02	-0.13	-0.05	-0.18	0.11
soldier	-0.14	0.09	-0.07	-0.26	-0.12
queen	0.15	0.07	0.04	-0.26	-0.05
longdress	-0.14	-0.04	-0.02	-0.04	0.08
basketball	0.04	0.04	0.05	-0.62	0.14
dancer	-0.06	0.00	0.05	0.52	0.74
평균	-0.02	0.01	0.01	-0.10	0.17

표 4. Random-Access 실험결과

시퀀스	Geom. BD-TotalRate [%]		End-to-End BD-TotalRate [%]		
	D1	D2	Luma	Chroma Cb	Chroma Cr
loot	-0.05	-0.34	-0.04	-0.02	-0.93
redandblack	-0.63	-0.78	0.00	0.24	-0.25
soldier	0.36	0.48	-0.08	-0.07	0.70
queen	0.45	0.32	0.78	-1.70	-0.89
longdress	0.20	-0.23	-0.06	0.27	0.32
basketball	-0.02	0.01	0.12	0.39	0.74
dancer	-0.02	-0.05	-0.09	-0.19	0.35
평균	-0.02	-0.09	0.09	-0.24	0.01

압축성능의 차이를 확인하기 위한 BD-BR 차이값은 아래 표 3 All-Intra, 표 4 Random-Access 조건 실험 결과와 같이 미미한 것으로 확인 되었다.

5. 결론

V-PCC 의 세그먼트 정제과정은 2D 패치 생성에 큰 영향을 주는 과정으로, V-PCC 압축에 핵심적인 과정임에는 틀림없다. 본 논문에서는 세그먼트 정제과정의 높은 연산 복잡도를 낮추기 위한 몇 가지 방법을 제시했다. 제안하는 방법을 적용하여 모듈 단위 10%정도의 연산량을 줄일 수 있었다. 그러나 V-PCC 부호화기의 실시간 부호화를 위해서는 더 많은 최적화 및 고속화 필요하다. 다음 연구에서는 세그먼트 정제의 적응적인 조기 종료 기법에 대해 고민해보려 한다.

ACKNOWLEDGEMENT

이 논문은 2020 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2020-0-00231, 5G 엣지클라우드 기반 VR.AR 저지연 스트리밍 기술 개발)

[참고 문헌]

- [1] Coded Representation of Immersive Media Part 5: Visual Volumetric Video-Based Coding (V3C) and Video-Based Point Cloud Compression (V-PCC), Standard ISO/IEC FDIS 23090-5, 2020.
- [2] T. Dong, K. Kim and E. S. Jang, "Performance Evaluation of the Codec Agnostic Approach in MPEG-I Video-Based Point Cloud Compression," *IEEE Access*, vol. 9, pp. 167990-168003, 2021.
- [3] TMC2 reference software <https://github.com/MPEGGroup/mpeg-pcc-tmc2>
- [4] E. Faramarzi, M. Budagavi, and R. Joshi, "Gridbased Partitioning", ISO/IEC JTC1/SC29/WG11 M47600, Mar. 2019
- [5] J. Kim, Y.-H. Kim, E.-S. Ryu, "Fast Grid-based Refining Segmentation Method in V-PCC", ISO/IEC JTC1/SC29/WG11 M56635, April 2021
- [6] J. Kim and Y. -H. Kim, "Fast Grid-Based Refining Segmentation Method in Video-Based Point Cloud Compression," *IEEE Access*, vol. 9, pp. 80088-80099, 2021,