Check for updates
일반논문 (Regular Paper)
방송공학회논문지 제31권 제5호, 2026년 9월 (JBE Vol.31, No.5, September 2026)
https://doi.org/10.5909/JBE.2026.31.5.1119
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

3D Gaussian Splatting의 G-PCC 압축 효율 향상을 위한 기하 옥트리 손실 기반 가우시안 최적화 방법

황 정 원[a)], 김 민 태[a)], 심 동 규[a)‡]

Gaussian Optimization based on Geometry Octree Loss for Efficient G-PCC Compression of 3D Gaussian Splatting

Jeongwon Hwang[a)], Mintae Kim[a)], and Donggyu Sim[a)‡]

요 약

본 논문에서는 G-PCC Amd1의 3D Gaussian Splatting 압축 과정에서 가우시안의 위치를 조정하여 공간적으로 중복된 가우시안을 병합함으로써 부호화 효율을 향상시키는 기하 옥트리 손실 기반의 가우시안 최적화 방법을 제안한다. 속성만을 조정하는 기존 가우시안 최적화 기법과 달리, 제안하는 방법은 옥트리 부호화의 복셀 격자 좌표계에서 인접 가우시안 간 거리를 반영한 기하 옥트리 손실을 추가적으로 고려하여 가우시안의 위치까지 조정 대상에 포함한다. 이를 통해 화질 저하를 최소화하는 선에서 가우시안 쌍을 동일 복셀로 수렴시키고 하나의 점으로 병합하여 부호화 대상 점의 수를 줄인 뒤, 속성을 다시 학습하여 렌더링 왜곡을 보정한다. 참조 소프트웨어 TMC13-GS v2.0-rc2에서 실험한 결과, 제안하는 방법은 앵커 대비 RGB-PSNR과 YUV-SSIM 기준 BD-rate가 평균 -29.75%, -30.82%, 기존 가우시안 최적화 기법 대비 -4.83%, -5.28% 개선되어 렌더링 품질을 유지하면서 부호화 효율을 향상시킴을 확인하였다.

Abstract

In this paper, we propose a Gaussian optimization method based on a geometry octree loss that improves the coding efficiency of 3D Gaussian Splatting compression in G-PCC Amd1 by adjusting the positions of Gaussians and merging spatially redundant ones. Unlike the existing Gaussian optimization tools, which adjust only the attributes of the Gaussians, the proposed method additionally takes into account a geometry octree loss reflecting the distances between adjacent Gaussians on the voxel grid coordinate system of the octree coding, and thereby includes the positions of the Gaussians in the adjustment. Through this, pairs of Gaussians converge into the same voxel while minimizing quality degradation, and are merged into a single point to reduce the number of points to be coded, after which the attributes are re-optimized to compensate for the rendering distortion. Experimental results on the reference software TMC13-GS v2.0-rc2 show that the proposed method improves the BD-rate by an average of -29.75% and -30.82% in terms of RGB-PSNR and YUV-SSIM over the anchor, and by -4.83% and -5.28% over the existing Gaussian optimization tools, confirming that the coding efficiency is improved while the rendering quality is maintained.

Keyword : 3D Gaussian Splatting, Gaussian Splat Coding, G-PCC, Octree geometry coding, Gaussian optimization

Copyright Ⓒ 2026 Korean Institute of Broadcast and Media Engineers. All rights reserved.
“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (http://creativecommons.org/licenses/by-nc-nd/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

Ⅰ. 서 론

최근 메타버스, 가상·증강현실(Virtual/Augmented Reality, VR/AR), 자유시점 비디오(Free-viewpoint Video, FVV) 등 몰입형 미디어 응용이 확산되면서, 실제 장면을 사실적으로 재현하고 임의의 시점에서 렌더링하는 3차원 장면 표현 기술의 중요성이 커지고 있다. 이러한 요구에 대응하여 제안된 3D Gaussian Splatting(3DGS)[1]은 3차원 공간을 방향과 크기가 서로 다른 타원체 형태의 비등방성(anisotropic) 가우시안 프리미티브(Gaussian primitive)들의 집합으로 표현하며, 실시간 렌더링과 사실적인 새로운 시점 합성(Novel-View Synthesis, NVS)을 동시에 달성하여 몰입형 미디어의 유력한 표현 방식으로 주목받고 있다. 장면을 구성하는 각 가우시안은 중심이 되는 3차원 위치(position)와 함께 형태를 결정하는 크기(scale)와 회전(rotation), 빛의 투과율을 나타내는 불투명도(opacity), 시점이나 조명 각도에 따른 색상 변화를 모사하는 구면 조화(Spherical Harmonics, SH) 계수를 속성으로 갖는다. 이때 하나의 가우시안은 위치 3개, 크기 3개, 회전 사원수 4개, 불투명도 1개, 그리고 3차 SH 기준 DC 3개와 AC 45개로 구성된 48개의 색상 계수를 포함하여 총 59개의 부동소수점 파라미터로 표현된다. 하나의 복잡한 실제 장면을 사실적으로 표현하기 위해서는 수십만에서 수백만 개의 가우시안이 필요하므로, 학습된 3DGS 데이터 하나의 크기는 수백 MB에서 수 GB에 이른다[2]. 이러한 방대한 데이터 크기는 저장과 전송에 상당한 부담으로 작용하며, 특히 대역폭이 한정된 스트리밍 환경에서 3DGS를 실용화하기 위해서는 시각적 품질을 유지하면서 전송 비트레이트를 낮추는 효율적인 압축 기술이 필수적으로 요구된다.

이러한 배경에서, 국제 표준화 기구인 ISO/IEC JTC1 SC29 WG7 MPEG의 3DGH(3D Graphics and Haptics Coding) 서브 그룹에서는 Gaussian splat(GS) 기반 장면 데이터의 압축 및 전송을 위한 Gaussian Splat Coding(GSC) 표준화 작업을 활발히 추진하고 있다. GSC는 크게 비디오 코덱 프레임워크를 재활용하는 비디오 기반 압축 방법[3]과 포인트 클라우드(Point cloud) 코덱을 확장하는 기하 기반 압축 방법[4]으로 나뉘어 탐색 실험이 진행 중이며, 이 중 Geometry-based Point Cloud Compression(G-PCC) 기반 GSC 앵커 모델[5] 개발은 2024년 11월 회의를 기점으로 본격화되었다. 기존의 G-PCC 참조 소프트웨어[6]는 위치 및 단순 색상 정보 중심의 일반적인 포인트 클라우드 데이터 부호화를 목적으로 설계되어 GS 데이터를 구성하는 고차원 속성 정보를 지원하는 데 한계가 있었으며, 이를 극복하기 위해 GS 전용으로 확장된 참조 소프트웨어인 TMC13-GS[7]가 개발되었다.

나아가 2025년 4월 회의에서는 HMD 및 모바일 환경에서의 저전력·실시간 디코딩 요구사항을 충족하기 위해 경량화 GSC 표준을 Fast-track으로 추진하기로 합의하였으며, G-PCC 프로파일 확장을 통한 단일 프레임 기하(1F-geo) 압축 기술 개발이 핵심 과제로 부상하였다[8]. 이러한 흐름 속에서 현재 진행 중인 G-PCC Amd1(Amendment1) 표준화는 기존 포인트 클라우드 압축 표준[9]에 Gaussian splat 전용 프로파일 및 확장 기능을 추가하는 개정 절차로 공식화되었으며, 해당 회의에서 Working Draft (WD)[10] 단계에 진입한 이후 현재 Committee Draft Amendment(CDAM)[11] 단계에 이르렀다. 현재 G-PCC Amd1의 공통 실험 조건(Common Test Conditions, CTC)[12]은 기하 정보를 옥트리(Octree) 기반으로 준무손실(near-lossless) 부호화하고, 속성 정보는 파라미터의 특성에 따라 RAHT(Region Adaptive Hierarchical Transform)[13], 예측 변환(Predicting Transform), 리프팅 변환(Lifting Transform)[14] 기법을 혼합 적용하여 손실(Lossy) 부호화하는 옥트리-하이브리드(Octree-hybrid) 방식을 주요 앵커 기준으로 확립하고 있다.

이처럼 옥트리-하이브리드 기반의 표준 프레임워크가 확립됨에 따라, 최근 표준화 회의에서는 코덱의 내부 구조를

a) 광운대학교 컴퓨터공학과(Kwangwoon University)
‡ Corresponding Author : 심동규(Donggyu Sim)
E-mail: dgsim@kw.ac.kr
Tel: +82-2-940-5470
ORCID: https://orcid.org/0000-0002-2794-9932
※ 이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학ICT연구센터(ITRC)의 지원(IITP-2026-RS-2022-00156225) 및 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-학·석사연계 ICT 핵심인재양성의 지원(IITP-2026-RS-2022-00156215)을 받아 수행된 연구임.
· Manuscript received August 23, 2026; Revised September 14, 2026; Accepted September 15, 2026.

직접 변경하는 대신 부호화 이전 단계에서 압축 효율을 극대화하는 가우시안 최적화 기술이 앵커의 핵심 성능 향상 도구로 연이어 채택되고 있다. 대표적으로 원본 영상(ground truth) 없이 렌더링 왜곡과 RAHT의 추정 엔트로피 손실을 최소화하도록 가우시안의 속성 파라미터만을 추가로 조정하는 Gaussian Self-finetuning[15]이 제안되어 인코더 선택 도구로 채택되었다. 이 기법은 이웃한 가우시안 간 속성 값을 서로 닮게 만드는 손실 항을 함께 사용하여 공간적 속성 상관성을 높임으로써, RAHT를 포함한 속성 부호화 도구 전반의 압축 효율을 개선한다. 이와 더불어 2026년 4월 회의에서는 앞서 채택된 Self-finetuning 전 단계에 Gaussian Pruning[16]을 결합하는 확장 기법이 추가적인 선택 도구로 채택되었으며, 이 기법은 렌더링에 기여하지 않는 가우시안을 화소 및 시점 가시성(visibility) 기반의 중요도로 평가하여 물리적으로 제거한 뒤 화질을 복원함으로써 GSC CTC 시퀀스 전반에서 압축 성능을 한층 더 끌어올렸다.

그러나 이와 같은 Pruning 및 Self-finetuning은 코덱 부호화 과정에서 발생하는 공간적 중복 문제를 직접적으로 해결하지 못한다는 한계가 존재한다. 이는 현재 G-PCC Amd1 앵커가 좌표 양자화 단계에서만 손실을 허용하고 이후의 좌표는 무손실로 부호화하여, 동일 복셀(Voxel)로 양자화된 점들을 중복점으로 온전히 보존하기 때문이다. 그 결과 복호화 시 동일 위치에 다수의 점이 복원되고 각각의 속성이 개별적으로 부호화되면서 기하 및 속성 비트의 낭비를 초래한다. Gaussian Pruning은 오직 렌더링 가시성만을 제거 기준으로 삼으므로 시각적 기여도는 높으나 실제로는 동일하거나 인접한 복셀 내에 촘촘히 겹쳐 있는 중복점을 걸러내지 못하며, Gaussian Self-finetuning은 가우시안의 위치를 고정한 상태에서 파라미터를 갱신하므로 속성 정보의 부호화 효율은 향상시킬 수 있으나 기하 정보나 옥트리 점유(occupancy) 구조 자체는 개선의 대상으로 다루지 않는다. 결국 기존의 가우시안 최적화 기법들은 우수한 렌더링 품질 보존과 속성 압축 성능에도 불구하고 위치가 고정된다는 전제에서 비롯된 공간적 중복을 남기며, 이는 옥트리 부호화 과정의 불필요한 비트 소모로 이어진다는 한계가 있다.

이에 본 논문은 옥트리 압축 특성을 적극적으로 반영한 가우시안 위치 재배치와 병합을 결합하여 추가적인 비트율 절감을 달성하는 세 단계의 가우시안 최적화 방법을 제안한다. 이하 기존 채택 도구는 Self-finetuning으로, 제안하는 방법의 각 학습 단계는 self-finetuning으로 표기하여 구분한다. 먼저 기하 인지 self-finetuning 단계에서는 코덱이 실제로 부호화하는 복셀 격자 좌표계 상에서 인접 가우시안 간 거리를 반영하는 미분 가능한 기하 옥트리 손실을 정의하고 이를 기존 Self-finetuning의 손실과 함께 학습에 반영한다. 이를 통해 화질 저하를 최소화하는 범위에서 인접한 가우시안 쌍을 선택적으로 동일 복셀로 이동시킴으로써 옥트리 점유 노드(occupied node) 수를 감소시킨다. 이어지는 복셀 단위 가우시안 병합 단계에서는 학습을 통해 동일 복셀 좌표를 갖게 된 가우시안들을 하나의 점으로 병합하여 부호화 대상이 되는 점의 개수를 직접 감소시키며, 마지막 속성 보정 self-finetuning 단계에서는 병합된 모델의 위치를 고정한 채 속성을 다시 학습함으로써 병합 과정에서 발생한 속성 왜곡을 보정한다. 제안하는 방법은 기존의 Gaussian Pruning 및 Self-finetuning과 유기적으로 결합되어, 가시성 기반 중요도만으로는 제거되지 않는 잉여 가우시안을 부호화 관점의 공간적 중복성 차원에서 추가적으로 해소한다. MPEG GSC CTC 시퀀스에 대한 실험을 통해 제안하는 방법이 G-PCC Amd1 앵커와 기존 가우시안 최적화 기법을 적용한 경우 모두에 대해 추가적인 BD-rate 절감을 달성함을 확인하였다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 G-PCC Amd1 표준의 구조와 선행 연구인 Gaussian Pruning 및 Self-finetuning에 대해 설명한다. Ⅲ장에서는 제안하는 손실 함수의 정의와 이를 포함한 전체 파이프라인을 기술한다. Ⅳ장에서는 제안하는 방법의 성능을 기존 방법과 비교하여 평가하며, Ⅴ장에서 결론을 맺는다.

Ⅱ. 배경 기술

본 논문에서 제안하는 방법은 G-PCC Amd1의 부호화 이전 단계에서 동작한다. 이에 본 장에서는 제안 방법의 상세

한 설명에 앞서 G-PCC Amd1의 전체 구조를 소개하고, 선행 가우시안 최적화 기법인 Gaussian Pruning과 Self-fine-tuning에 대해 설명한다.

1. G-PCC Amd1 프레임워크

G-PCC Amd1은 기존 포인트 클라우드 압축 표준인 G-PCC의 부호화 구조를 GS 데이터에 적용하기 위해 확장한 표준으로, 가우시안 프리미티브의 위치를 희소 포인트 클라우드의 좌표로, 나머지 파라미터를 해당 점의 속성으로 대응시켜 부호화한다. 이때 옥트리와 RAHT 등 핵심 부호화 엔진은 기존 G-PCC의 것을 재사용하며, GS 고유의 고차원 파라미터를 이 엔진에 정합시키기 위한 전처리 및 후처리가 새롭게 확장되었다. 그림 1은 G-PCC Amd1 참조 소프트웨어의 전체 구조를 나타낸다. 부호화기는 전처리, 기하 부호화, 속성 부호화로 구성되며, 입력 GS 데이터는 전처리를 거쳐 정수 격자로 변환된 뒤 기하와 속성으로 나뉘어 부호화되어 하나의 G-PCC GSC 비트스트림을 구성한다. 복호화기는 이 비트스트림으로부터 기하와 속성을 복원한 뒤, 후처리에서 전처리의 역과정을 거쳐 최종 GS 데이터를 재구성한다. 이때 속성 부호화는 복원된 기하 정보를 참조하여 수행되므로, 부호화와 복호화 모두 기하가 먼저 처리된 뒤 속성이 처리되는 종속 관계를 갖는다.

전처리 단계는 부호화에 앞서 입력 가우시안을 코덱이 처리할 수 있는 형태로 변환하며, 색공간 변환, 크기·회전 정준화(canonicalization), 양자화의 세 과정으로 구성된다. 먼저 색공간 변환은 시점 의존적(view-dependent) 색상을 표현하는 SH 계수에 RGB-YUV 변환을 직접 적용하여 채널 간 상관성을 제거한다. 정준화는 가우시안의 대칭성으로 인해 크기와 회전의 조합이 달라도 동일한 형태가 표현되는 중복을 제거하여 가우시안 간 통계적 일관성을 높인다. 이어 양자화는 각 파라미터를 파라미터별 최소·최대 구간에서 지정된 비트 심도로 정수화하며, CTC는 위치에 18-bit, 나머지 속성에 각각 12-bit를 할당한다. 이때 정수화된 위치 좌표가 옥트리 부호화의 입력이 되므로, 위치 양자화의 비트 심도에 따라 점들이 배치되는 복셀 격자의 해상도가 결정된다.

기하 부호화는 양자화된 위치 정보를 압축하며, 옥트리 부호화와 예측 기하(Predictive geometry) 부호화의 두 가지 방법을 지원한다. 옥트리 부호화는 점군 전체를 포함하는 바운딩 박스를 루트 노드로 하여 재귀적으로 8개의 서브 노드로 분할함으로써 옥트리 구조를 형성하고, 각 노드의 점 존재 여부를 1비트의 점유 정보로 표현하여 문맥 기반 엔트로피 부호화로 전송한다. 그림 2 (a)는 옥트리 부호화의 공간 분할 과정을, 그림 2 (b)는 그로부터 생성되는 옥트리와 점유 정보를 나타낸다. 점유된 노드는 8개 자식 노드의 점유 여부를 나타내는 8비트의 정보를 발생시키며, 이들은 순차적으로 부호화되어 기하 비트스트림을 구성한다. 분할은 노드가 단위 복셀에 도달할 때까지, 즉 위치 양자화 비트 심도와 동일한 깊이 d에 이를 때까지 반복되므로, 부호화되는 기하 비트량은 옥트리에 점유된 노드의 수에 직접적으로 좌우된다. 반면 예측 기하 부호화는 점들을 트리 형태로 연결하고 부모 노드로부터 자식 노드의 위치를 예측하여 그 잔차만을 부호화하는 방식으로, 인접한 점 사이의 위치 상관이 큰 데이터에서 높은 효율을 보인다.

속성 부호화는 이웃 점 간의 속성 유사성을 활용하여 압축을 수행하며, RAHT와 예측 변환, 리프팅 변환의 세 가지 방법을 지원한다. RAHT는 Haar 웨이블릿 변환을 3차원으로 확장한 기법이다. Haar 웨이블릿 변환은 인접한 두 값을 그 합과 차에 해당하는 저주파 성분과 고주파 성분으로 분

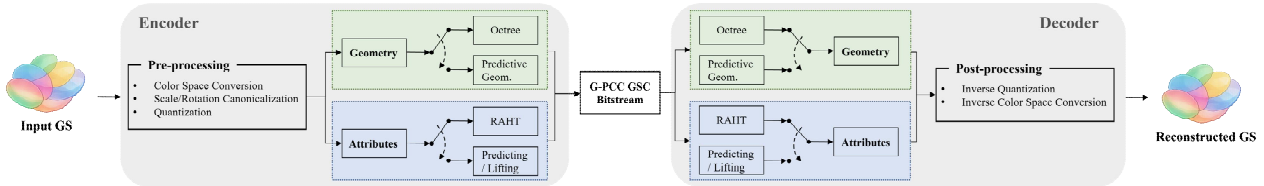

그림 1. G-PCC Amd1의 부호화 및 복호화 구조
Fig. 1. Encoding and decoding structure of G-PCC Amd1

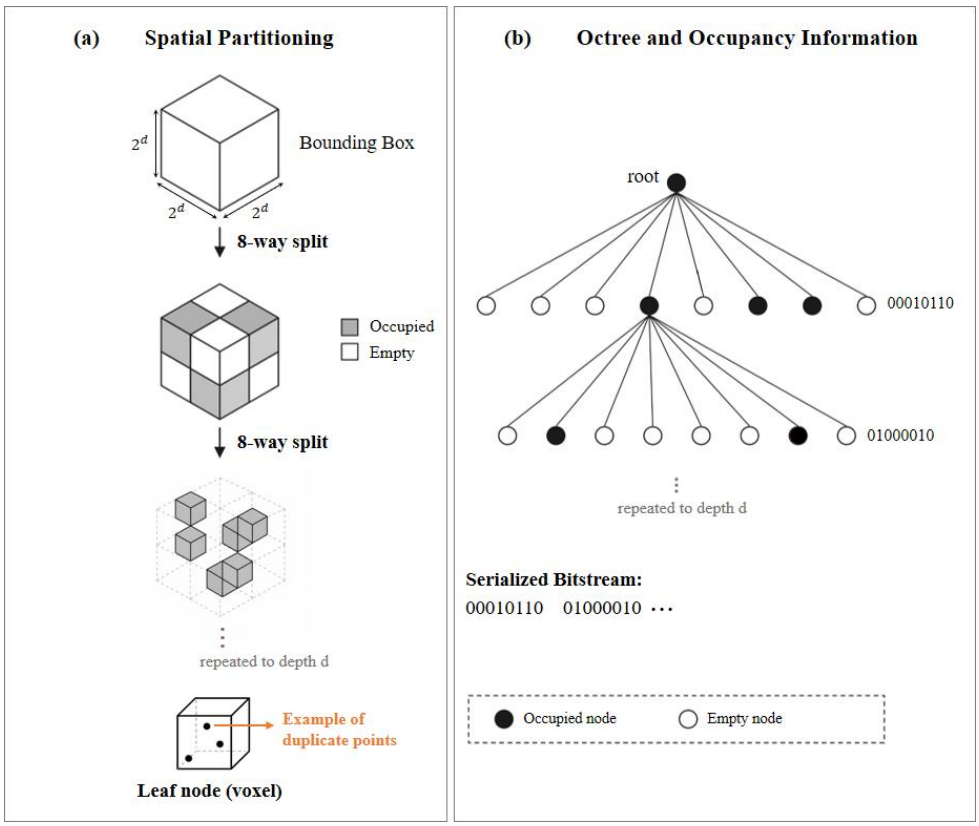

그림 2. 옥트리 부호화의 (a) 공간 분할과 (b) 점유 정보
Fig. 2. (a) Spatial partitioning and (b) occupancy information in octree coding

해하며, 두 값이 유사할수록 고주파 성분이 0에 수렴한다. RAHT는 이를 3차원 공간에 적용하여, 공간을 계층적으로 분할한 뒤 최하위 계층부터 각 축 방향으로 인접한 두 점의 속성을 저주파 성분(DC)과 고주파 성분(AC)으로 분해한다. 이때 두 성분은 각 점이 포함하는 하위 점의 수를 가중치로 반영하여 계산되므로, 점이 조밀한 영역일수록 큰 가중치를 갖는다. 분해된 저주파 성분만을 상위 계층으로 전달하여 동일한 변환을 재귀적으로 반복하며, 이 과정에서 발생한 고주파 성분들을 엔트로피 부호화한다. 결과적으로 장면 전체가 하나의 DC 성분으로 수렴하고, 나머지는 계층별 AC 성분으로 표현된다. 한편, 예측 변환은 LoD(Level of Detail) 구조상의 이웃 점들로부터 속성 값을 예측하여 그 잔차를 부호화하며, 리프팅 변환은 예측 변환에 갱신 단계를 추가하여 detail 계수를 부호화한다. 세 방법은 구현 방식은 상이하나 모두 공간적으로 인접한 점들의 속성이 유사하다는 전제에 기반하므로, 이웃 간 속성 차이가 작을수록 부호화되는 계수 또는 잔차의 크기가 감소한다.

이처럼 G-PCC Amd1은 기하와 속성 각각에 대해 복수의 부호화 도구를 제공하며, 대상 데이터의 특성에 따라 이들을 선택적으로 조합하여 사용한다. 본 논문에서는 GSC CTC가 규정하는 앵커 구성에 따라, 기하 정보는 옥트리 부호화로 압축하고 속성 정보는 파라미터의 특성에 맞추어 세 가지 변환을 조합하는 옥트리-하이브리드 구성을 기준으로 실험을 수행하였다. 이때 하이브리드 방식은 불투명도와 크기·회전에 예측 변환을, SH의 DC 및 1·2차 AC 성분에 RAHT를, 3차 AC 성분에 리프팅 변환을 적용한다. 한편, G-PCC Amd1 앵커 구성의 옥트리 기하 부호화는 양자화 이후의 좌표를 무손실로 부호화하므로, 동일한 정수 좌표를 갖게 된 복수의 점들이 그대로 보존된다. 이처럼 그림 2 (a)와 같이 동일한 최하위 노드에 속하여 같은 복셀

좌표를 갖게 된 점들을 중복점이라고 하며, 이들은 복호화 시 동일 위치에 해당 개수만큼 복제된다. 이때 중복점은 하나의 점유 노드를 공유함에도 각각의 속성이 개별적으로 부호화되므로, 점의 개수에 비례하여 속성 비트가 소모된다. 제안하는 방법은 이러한 옥트리 기하 부호화 과정의 동작 특성에 주목하여, 점유된 노드의 수와 각 노드에 포함된 중복점을 함께 줄임으로써 기하 및 속성 비트스트림을 감소시키는 방향으로 설계된다.

2. Gaussian Pruning 및 Self-finetuning

1절에서 기술한 G-PCC Amd1의 기본 부호화 구조에서 전처리는 입력 가우시안을 코덱이 부호화할 수 있는 형식으로 변환하는 과정이다. 이에 더해, 코덱을 변경하지 않고도 부호화 효율을 높이기 위해 전처리 단계를 확장하여 입력 가우시안을 코덱에 유리한 형태로 가공하는 가우시안 최적화 기법들이 제안되었다. 이러한 가우시안 최적화 기법은 표준 호환성을 유지하면서 부호화 효율을 개선한다는 공통점을 가지며, 현재 G-PCC Amd1에는 Gaussian Pruning과 Self-finetuning이 선택적 인코더 도구로 채택되어 참조 소프트웨어에 통합되어 있다. 그림 3은 두 기법이 가우시안 최적화 단계를 구성하는 부호화 파이프라인의 전체 구조를 나타낸다.

Gaussian Pruning은 렌더링 결과에 거의 기여하지 않는 가우시안을 부호화 이전에 제거하여 부호화 대상 점의 수를 직접 감소시키는 기법이다. 먼저 각 가우시안에 대해 중요도 점수를 산출하는데, 중요도 점수 I_i는 식 (1)과 같이 여러 시점 c와 화소 p에 걸친 해당 가우시안의 가시성 $w_i(c,p)$의 제곱합으로 정의된다.

$$I_i = \sum_c \sum_p w_i(c,p)^2 \tag{1}$$

여기서 $w_i(c,p)$는 렌더링 시 가우시안 i가 해당 화소 색상에 기여하는 정도를 나타내는 알파 블렌딩(alpha-blending) 가중치이다. 이에 따라 크고 불투명한 가우시안은 높은 점수를, 작거나 투명하거나 다른 가우시안에 가려진 가우시안은 낮은 점수를 갖는다. 이후 중요도가 낮은 하위 일정 비율의 가우시안을 제거하며, Amd1 앵커에서는 모든 시퀀스에 대해 30%의 제거 비율을 일괄 적용한다.

Pruning은 점을 물리적으로 제거하므로 필연적으로 화질 저하를 수반한다. Gaussian Self-finetuning은 Pruning 이후 남은 가우시안의 속성 파라미터를 미세하게 조정하여 이러한 화질 저하를 회복하는 동시에, 속성 정보의 부호화 효율을 함께 개선하는 기법이다. 이를 위해 Self-finetuning은 렌더링 왜곡과 부호화율을 함께 고려하는 율-왜곡(rate-distortion) 손실을 최소화하도록 속성을 갱신한다. 이 손실 L은 식 (2)와 같이 렌더링 손실 D와 속성 근사 엔트로피 손실 R의 가중합으로 정의되며, 가중치 λ가 두 손실 항 사이의 균형을 조절한다.

$$L = D + \lambda R \tag{2}$$

렌더링 손실 D는 입력 가우시안을 목표 시점에서 렌더링하여 원본 영상을 대신하도록 생성한 유사 원본 영상

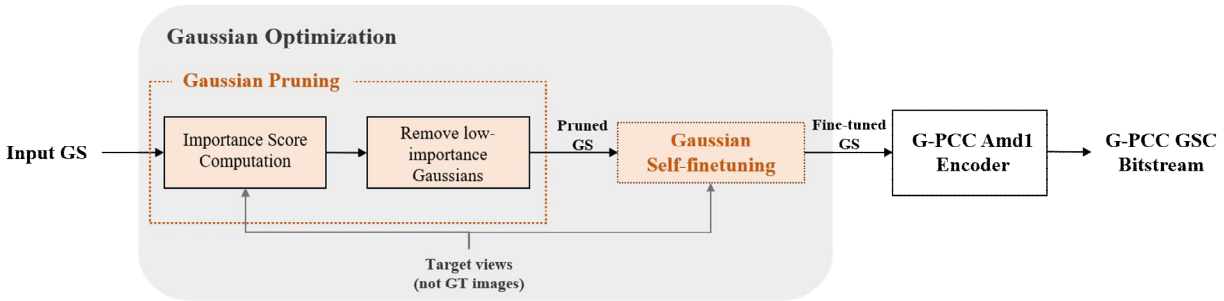

그림 3. 가우시안 최적화가 적용된 G-PCC Amd1 부호화 파이프라인
Fig. 3. G-PCC Amd1 encoding pipeline with Gaussian optimization

(pseudo-ground truth)과 최적화 중인 가우시안의 렌더링 결과 간 차이를 측정하여 화질 저하를 억제한다. 이는 3DGS의 학습에서와 동일하게 식 (3)과 같이 화소 단위의 L1 손실과 구조적 유사도(Structural Similarity, SSIM) 손실의 가중합으로 정의된다.

$$D = (1-\lambda_{SSIM})L_1 + \lambda_{SSIM}L_{D-SSIM} \tag{3}$$

여기서 L_1은 두 영상의 화소 색상 간 절댓값 차이이며, SSIM은 밝기·대비·구조를 함께 반영하는 유사도 지표로서 $1-SSIM$이 구조적 왜곡으로 작용한다. 한편 R은 부호화율을 직접 모델링하기 어려운 RAHT를 대신하여, 이웃한 가우시안 간 속성 값의 차이를 통해 부호화율을 근사하는 항이다. 이는 식 (4)와 같이 각 가우시안과 그 이웃 간 속성 값의 차이를 합하여 정의된다.

$$R = \sum_i \sum_{j \in N_k(i)} \| a_i - a_j \|_1 \tag{4}$$

여기서 a_i와 a_j는 각각 가우시안 i와 이웃 가우시안 j의 속성이고, $N_k(i)$는 가우시안 i의 k-최근접 이웃(k-nearest neighbors, KNN) 집합을 나타낸다. 이 항을 최소화하면 속성의 공간적 평활도가 증가하여 이웃 가우시안 간 속성 차이가 줄어들고, RAHT를 비롯한 속성 부호화 도구가 내부적으로 이 차이를 부호화하므로 결과적으로 비트스트림의 크기가 감소한다.

이와 같이 G-PCC Amd1의 가우시안 최적화 파이프라인은 Pruning으로 점의 수를 줄이고 Self-finetuning으로 남은 가우시안의 속성을 조정하는 2단계로 구성된다. 이러한 2단계 파이프라인에 대해 GSC CTC에서 앵커 대비 정면 촬영(Forward facing) 시퀀스에서 -32.8%, 객체 중심(Object centric) 시퀀스에서 -26.0%의 BD-rate가 보고되었으며, 이는 Self-finetuning 단독의 -15.0% 및 -4.3% 대비 크게 향상된 수치이다[16]. 이처럼 두 기법은 렌더링 품질을 보존하면서 가우시안 점의 수와 속성 부호화 효율을 개선하지만, 그 최적화는 속성 축에 국한된다. Pruning은 가시성이 낮은 점을 제거하고 Self-finetuning은 속성 값을 조정할 뿐, 두 기법 모두 가우시안의 위치 자체는 최적화 대상으로 삼지 않는다. 이로 인해 동일 복셀에 다수의 가우시안이 잔존하여 옥트리 점유 노드를 불필요하게 증가시키는 공간적 중복 문제는 여전히 해소되지 않으며, 이러한 한계는 본 논문의 출발점이 된다.

Ⅲ. 제안하는 방법

본 장에서는 가우시안의 위치를 학습하여 공간적 중복을 해소하는 기하 옥트리 손실 기반의 가우시안 최적화 방법을 제안한다. 그림 4는 제안하는 방법이 적용된 전체 부호화 파이프라인을 나타내며, 크게 기하 옥트리 손실을 적용한 기하 인지 self-finetuning, 복셀 단위 가우시안 병합, 그리고 속성 보정 self-finetuning의 세 과정으로 구성된다.

먼저, 기하 인지 self-finetuning은 복셀 격자 좌표계 상에서 정의된 기하 옥트리 손실을 기존 손실에 추가하고, 가우시안의 위치를 속성과 함께 최적화하는 과정이다. 이 손실항은 인접한 가우시안 쌍을 동일 복셀로 끌어당기는 힘으로 작용하며, 화질을 보존하려는 렌더링 손실과 경쟁함으로써 시각적 왜곡을 적게 유발하는 점들을 선택적으로 동

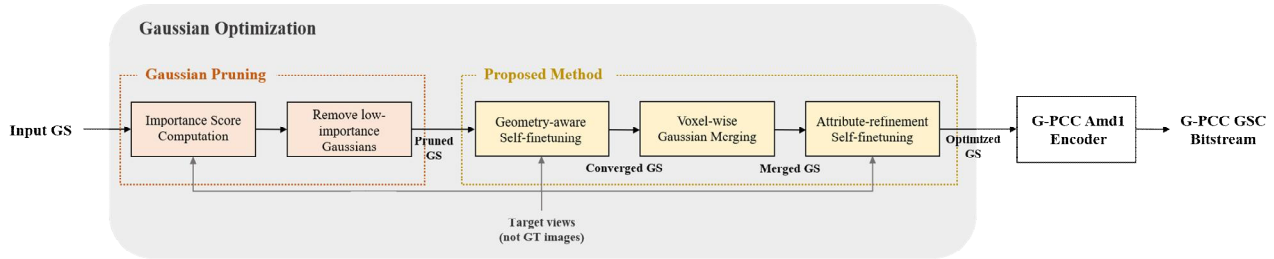

그림 4. 제안하는 방법이 적용된 G-PCC Amd1 부호화 파이프라인
Fig. 4. G-PCC Amd1 encoding pipeline with the proposed method

일 복셀에 수렴시킨다. 다음으로 복셀 단위 가우시안 병합 단계에서는 학습을 통해 동일 복셀 좌표를 공유하게 된 가우시안들을 하나의 점으로 병합하여 전체 가우시안 점의 수를 감소시킨다. 마지막으로, 속성 보정 self-finetuning 단계에서는 가우시안의 위치를 고정한 상태에서 속성 파라미터만을 다시 학습하여 병합 과정에서 발생할 수 있는 미세한 화질 저하를 회복한다. 이때, 제안하는 방법은 G-PCC Amd1의 선택적 인코더 도구로 채택된 Gaussian Pruning을 먼저 수행한 뒤 그 결과 모델에 적용되며, 최종적으로 최적화된 GS 데이터는 앵커와 동일한 부호화기로 전달된다. 아래 절에서 각 과정을 보다 상세히 기술한다.

1. 기하 옥트리 손실

제안하는 방법의 핵심은 가우시안의 위치를 옥트리 부호화에 유리한 방향으로 이동시키는 미분 가능한 손실 항이다. Ⅱ장 1절에서 기술하였듯이 옥트리 부호화는 노드가 단위 복셀에 도달할 때까지 분할을 반복하므로 옥트리의 최대 깊이는 위치 양자화의 비트 심도에 의해 고정되는 반면, 각 레벨에서 실제로 점유된 노드의 수는 점들이 공간상에 분포한 양상에 따라 달라진다. 옥트리의 한 점유 노드는 자식 노드들의 점유 여부를 나타내기 위해 8비트의 점유 정보를 발생시키므로, 결과적으로 부호화되는 기하 비트량은 옥트리 전체의 점유 노드 수에 비례한다. 따라서 인접한 가우시안들을 동일 복셀로 수렴시키면 점유 노드의 수가 감소하여 기하 비트량을 절감할 수 있으며, 이에 본 논문에서는 학습 과정에서 가우시안의 위치를 이러한 방향으로 이동시키는 기하 옥트리 손실을 제안한다.

이러한 손실 함수를 정의하기 위해서는 먼저 점 간 거리를 측정하는 기준 좌표계가 옥트리 부호화가 수행되는 복셀 격자와 일치해야 한다. 옥트리 부호화는 양자화된 정수 좌표를 입력으로 처리하며, 임의의 두 점이 동일한 복셀을 공유하는지 여부는 이 격자 공간 위에서 결정되기 때문이다. 반면 가우시안의 위치는 월드 좌표계의 실수값으로 표현되고, 복셀 하나가 대응하는 실제 크기는 장면의 바운딩 박스 크기와 위치 양자화의 비트 심도에 따라 결정된다. 따라서 월드 좌표계에서 측정한 거리는 옥트리 부호화가 인식하는 거리와 일치하지 않으며, 이에 가우시안 i의 위치 x_i를 식 (5)와 같이 복셀 격자 좌표 $q(x_i)$로 변환한 뒤 거리를 측정한다.

$$q(x_i) = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}(2^b - 1) \tag{5}$$

식 (5)에서 b는 위치 양자화의 비트 심도를 나타내며, $x_{\min}$과 $x_{\max}$는 양자화 구간의 경계로서 동일한 격자를 구성하기 위해 부호화기의 전처리 과정에서 산출되는 값을 그대로 사용한다. 이때 $q(x_i)$는 부호화기의 위치 양자화와 동일한 비트 심도의 스케일을 사용하되, 학습 과정에서는 정수화하지 않은 실수값으로 유지한다. 정수화된 좌표는 가우시안이 복셀 내부에서 이동하는 동안 값이 변하지 않아 손실에 변화가 나타나지 않으며, 결과적으로 위치에 대한 기울기가 0이 되어 학습이 불가능해지기 때문이다. 이러한 복셀 격자 좌표계의 실수값으로부터 정의되는 두 가우시안 i, j 사이의 거리 $d_{i,j}$는 식 (6)과 같다. 이때 $d_{i,j}$가 작아질수록 두 가우시안이 같은 복셀에 속할 가능성은 높아지므로, 본 논문에서 제안하는 기하 옥트리 손실 L_{geo}는 $d_{i,j}$를 감소시키도록 식 (7)과 같이 정의된다.

$$d_{i,j} = \| q(x_i) - q(x_j) \|_2 \tag{6}$$

$$\begin{aligned} L_{geo} &= \sum_i \sum_{j \in N_k(i)} l_{geo}(d_{i,j}), \\ l_{geo}(d_{i,j}) &= 1 - \exp\left(-\left(\frac{d_{i,j}}{v}\right)^2\right) \end{aligned} \tag{7}$$

식 (7)에서 $l_{geo}(d_{i,j})$는 개별 이웃 쌍 (i,j)에 대한 손실이며, $N_k(i)$는 속성 근사 엔트로피 손실과 동일한 이웃 집합으로 학습 시작 시점의 위치를 기준으로 한 차례 구성한 뒤 학습 전 구간에 걸쳐 고정하고 두 손실이 이를 공유한다. v는 손실이 유효하게 작용하는 거리 범위를 결정하는 기준 반경이며, 이때 손실은 $d_{i,j}$가 아니라 이를 v로 나눈 값에 의해 결정된다. $d_{i,j}/v$는 해당 쌍의 거리가 기준 반경의 몇 배인지를 나타내는 값으로, 손실은 이 상대적인 척도에 따라 각 쌍에 서로 다른 세기의 힘을 가한다.

기하 옥트리 손실을 이와 같은 형태로 설계한 이유는 크

게 두 가지이다. 첫째, 서로 가까운 가우시안 쌍에 대해서만 선택적으로 작용하여야 한다. 부호화 관점에서 제거할 수 있는 중복은 이미 거의 같은 위치에 놓인 가우시안들이며, 이러한 쌍은 조금만 이동시켜도 동일 복셀로 모을 수 있다. 반면 멀리 떨어진 쌍까지 끌어당기면 장면 전체가 수축하여 기하 구조가 훼손되고, 이는 곧 렌더링 품질 저하로 이어진다. 따라서 손실은 일정 거리 이상 떨어진 쌍에 대해서는 작용하지 않아야 하며, 그 기준이 되는 거리를 결정하는 값이 v이다. 식 (7)의 지수 항은 $d_{i,j}$가 v에 비해 커질수록 손실을 1로 포화시키므로, v를 크게 넘어서는 쌍은 더 이상 손실을 증가시키지 않아 학습에서 자연히 배제된다. 둘째, 가우시안의 위치에 대해 미분 가능하여야 한다. 손실이 궁극적으로 감소시키고자 하는 값은 옥트리의 점유 노드 수이나, 이 값은 앞서 기술한 반올림된 좌표와 마찬가지로 가우시안이 복셀 경계를 넘어서는 순간에만 변하므로 직접 손실로 사용할 수 없다. 이에 점유 노드의 수를 직접 다루는 대신 연속적으로 정의되는 $d_{i,j}$를 손실의 기준으로 삼으며, 따라서 손실은 $d_{i,j}$에 대해 미분 가능한 형태여야 한다. 제안하는 손실 함수는 지수 함수를 통해 $d_{i,j}$의 변화가 손실에 연속적으로 반영되도록 하며, 이에 따라 위치에 대한 기울기가 전달되어 학습을 통해 가우시안의 위치를 이동시킬 수 있다. 이때 위치에 전달되는 기울기는 두 가우시안을 끌어당기는 힘으로 작용하며, 그 세기는 식 (8)과 같이 거리에 대한 기울기로 결정된다.

$$\frac{\partial l_{geo}}{\partial d_{i,j}} = \frac{2d_{i,j}}{v^2}\exp\left(-\left(\frac{d_{i,j}}{v}\right)^2\right) \tag{8}$$

식 (8)의 기울기는 모든 거리에서 연속적으로 정의되므로 둘째 조건을 만족하며, $d_{i,j}$가 v에 비해 충분히 커지면 지수 항에 의해 0으로 수렴하므로 첫째 조건 또한 만족한다. 그림 5는 개별 이웃 쌍의 손실 l_{geo}와 그 기울기가 거리에 따라 변화하는 양상을 나타낸다. 기울기는 v를 곱하여 $d_{i,j}/v$만의 함수가 되도록 하였으며, 이에 따라 두 곡선은 v의 값이 달라지더라도 각각 동일한 형태를 유지한다. 그림 5의 기울기 곡선은 세 구간으로 구분된다. 먼저 $d_{i,j}$가 v에 비해 충분히 작은 구간을 살펴보면, 이는 이미 서로 충분히 가까워져 동일 복셀에 놓일 가능성이 높은 쌍에 해당한다. 이러한 쌍을 계속 끌어당기는 것은 추가적인 비트 절감 없이 렌더링 화질만 저하시키므로, 손실은 이 구간에서 약한 기울기만을 갖는다. 다음으로 $d_{i,j} = v/\sqrt{2}$에서는 기울기가 최댓값을 갖는다. 이 부근은 아직 서로 다른 복셀에 놓여 있으나 비교적 적은 이동만으로 동일 복셀에 모일 수 있는 쌍에 해당하며, 이동에 필요한 거리가 짧은 동시에 가장 강한 힘을 가하여 병합을 유도한다. 마지막으로 $d_{i,j}$가 v를 크게 넘어서는 구간에서는 기울기가 0으로 수렴한다. 이 구간은 동일 복셀로 모으기 위해 큰 이동이 필요한 쌍에 해당하

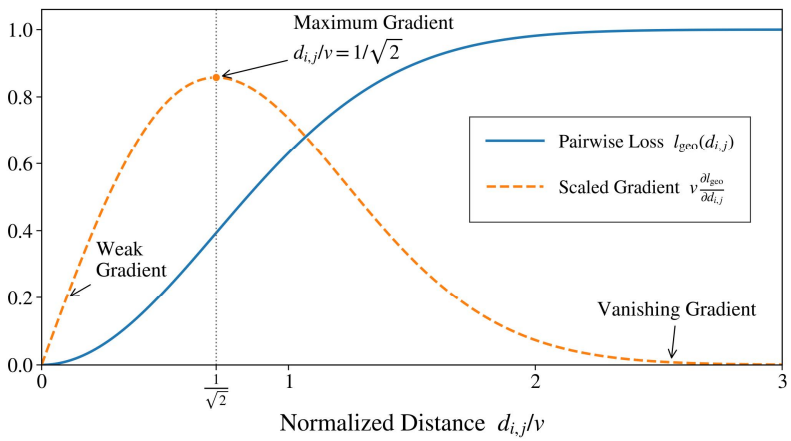

그림 5. 기하 옥트리 손실 함수와 거리에 대한 기울기
Fig. 5. Geometry octree loss and its gradient

므로, 이를 이동시키면 점유 노드를 줄이더라도 렌더링 품질이 크게 떨어진다. 따라서 손실이 위치에 가하는 힘은 거리가 멀수록 약해진다.

이와 같이 손실의 작용 방식은 $d_{i,j}$와 v의 비율에 따라 결정되므로, 기하 옥트리 손실을 실제 학습에 적용하기 위해서는 v를 설정하여야 한다. v는 손실의 작용 범위를 결정하는 값이므로, v를 정하는 것은 곧 전체 가우시안 이웃 쌍 중 어느 범위까지를 손실의 대상으로 삼을지를 정하는 것이다. 이때 $d_{i,j}$는 복셀 단위로 측정한 거리이며, 복셀 하나의 크기는 장면의 바운딩 박스를 위치 양자화의 비트 심도로 나누어 결정된다. 따라서 복셀 하나가 실제 공간에서 차지하는 크기는 장면마다 다르며, 가우시안이 같은 밀도로 분포하더라도 복셀 단위로 측정한 이웃 간 거리의 분포는 장면에 따라 달라진다. 따라서 v를 모든 시퀀스에서 고정된 값으로 지정하면, 가우시안이 조밀하게 분포한 장면에서는 거의 모든 쌍이 작용 범위에 포함되어 장면 전체가 수축하고, 희소하게 분포한 장면에서는 어떤 쌍도 힘을 받지 못하여 병합이 일어나지 않는다. 이에 v를 직접 지정하는 대신, 손실의 작용 범위에 포함되는 이웃 쌍의 비율 τ를 지정한다. 이와 같이 정의한 작용 범위에 포함되는 이웃 쌍의 비율 $\eta(v)$는 식 (9)와 같으며, 각 시퀀스에서 $\eta(v)=\tau$를 만족하는 v를 산출한다.

$$\eta(v)=\frac{1}{n}\sum_{i}\sum_{j\in N_k(i)}\mathbf{1}\left[\exp\left(-\left(\frac{d_{i,j}}{v}\right)^2\right)>\varepsilon\right] \tag{9}$$

식 (9)에서 n은 전체 이웃 쌍의 개수, ε은 기울기가 유효하다고 간주하는 최소 임계값이며, $\mathbf{1}[\cdot]$은 조건을 만족할 때 1, 그렇지 않을 때 0을 갖는 지시 함수이다. v가 커질수록 작용 범위에 포함되는 쌍이 늘어나므로 $\eta(v)$는 v에 대해 단조 증가하며, 따라서 목표 비율 τ에 대하여 $\eta(v)=\tau$를 만족하는 v가 유일하게 결정된다. 이렇게 산출된 v는 해당 장면에서 가장 가까운 τ만큼의 이웃 쌍이 포함되는 거리 범위를 반영한 값이 된다. 본 논문에서는 τ를 모든 시퀀스에 공통으로 적용함으로써, 각 시퀀스의 분포에 맞는 v가 산출되면서도 손실이 유효하게 작용하는 이웃 쌍의 비율은 장면과 무관하게 일정하게 유지되도록 하였다.

2. 최종 손실 함수와 위치 학습

1절에서 정의한 기하 옥트리 손실은 기존 Self-finetuning의 손실에 가중합되어 최종 손실을 구성하며, 이는 식 (10)과 같이 정의된다.

$$L=D+\lambda R+wL_{geo} \tag{10}$$

식 (10)에서 D와 R은 각각 식 (3)과 식 (4)의 렌더링 손실과 속성 근사 엔트로피 손실이며, w는 기하 옥트리 손실의 가중치이다. 세 항은 각각 화질, 속성 비트량, 기하 비트량을 담당한다. 이 중 R과 L_{geo}는 모두 미분 불가능한 부호화 비트량을 미분 가능한 형태로 근사한다는 점에서 동일한 구조를 가지나, R이 속성 값의 공간적 유사성을 높여 부호화되는 계수의 크기를 감소시키는 반면 L_{geo}는 부호화 대상이 되는 점유 노드의 수 자체를 감소시킨다는 점에서 구별된다. 또한 D와 L_{geo}는 가우시안의 위치에 대해 상반된 방향의 힘을 가한다. L_{geo}가 인접한 가우시안을 동일 복셀로 끌어당기는 반면, D는 위치 이동이 렌더링 결과를 훼손하는 경우 이를 원래 위치로 되돌린다. 따라서 두 힘이 경쟁하는 과정에서 위치 이동에 따른 화질 저하가 작은 쌍만이 선택적으로 동일 복셀에 수렴하며, 화질 기여가 큰 가우시안은 이동하지 않고 위치를 유지한다. 이와 같이 제안하는 방법은 개별 점에 대해 명시적인 율-왜곡 판정 절차 없이도, 기존 Self-finetuning과 동일하게 학습 과정에서 손실 간 경쟁을 통해 율과 왜곡의 균형을 달성한다. 이때 w는 두 손실이 위치에 가하는 힘의 균형을 조절한다. v가 손실이 작용하는 쌍의 범위를 정하는 반면, w는 그 범위 안에 놓인 쌍이 실제로 이동하는지를 결정한다. 즉 동일 복셀에 수렴하는 쌍은 손실의 작용 범위에 포함된 쌍 가운데 L_{geo}가 가하는 힘이 D가 가하는 힘보다 큰 쌍에 한정되며, 그 수는 두 기울기의 상대적인 크기에 의해 정해진다. D의 기울기는 위치 변화에 따른 렌더링 결과의 변화가 큰 장면일수록 크게 나타나므로 장면의 시각적 특성에 따라 그 크기가 달라진다. 따라서 w를 고정된 값으로 지정하면 두 기울기의 비율이 시퀀스마다 달라져, 어떤 시퀀스에서는 L_{geo}가 D에 눌려 수렴이 거의 일어나지 않고 다른 시퀀스에서

는 반대로 화질 저하가 크게 발생한다. 이에 w를 직접 지정하는 대신, 두 손실이 위치에 가하는 기울기 크기의 비율 ρ를 지정하고 이로부터 각 시퀀스의 w를 산출한다. 구체적으로 학습 시작 시점에 D와 L_{geo}가 위치에 가하는 기울기의 크기를 한 차례 측정하여, L_{geo}의 기울기가 D의 기울기의 ρ배가 되도록 w를 계산한다. ρ 역시 모든 시퀀스에 공통으로 적용하여, 두 손실이 위치에 가하는 힘의 균형이 장면과 무관하게 일정하게 유지되도록 하였다.

한편, 기존 Self-finetuning은 양자화 오차를 반영하기 위해 양자화 인지 학습(Quantization-Aware Training, QAT) 방식으로 수행되며, 이에 따라 순전파 과정에서 파라미터들은 양자화 및 역양자화를 거쳐 렌더링에 적용된다. 다만 1절에서 기술한 바와 같이 양자화 과정의 반올림 연산은 기울기가 0이 되어 렌더링 손실의 역전파를 차단하므로, 이를 해결하기 위해 역전파에서는 직통 추정기(Straight-Through Estimator, STE)가 적용된다. 제안하는 방법에서는 위치 파라미터 역시 새로운 학습 대상으로 포함되므로, 기존 파라미터에 쓰이던 QAT와 STE 기법을 위치 최적화 과정에도 동일하게 적용하여 위치에 대한 기울기가 전달되도록 한다.

3. 가우시안 병합 및 속성 보정

기하 인지 self-finetuning의 학습이 완료된 시점에서 다수의 가우시안이 동일 복셀 좌표를 공유하게 된다. 학습 과정에서 $d_{i,j}$는 정수화 이전의 실수 좌표에서 계산되므로 두 가우시안이 실제로 동일한 복셀에 속하는지를 그대로 나타내지는 않는다. $d_{i,j}$가 작더라도 복셀 경계를 사이에 둔 쌍은 서로 다른 복셀에 속할 수 있고, 반대로 $d_{i,j}$가 다소 크더라도 하나의 복셀 안에 함께 놓일 수 있다. 어느 쌍이 동일 복셀에 속하게 되었는지는 학습이 완료된 뒤 $q(x_i)$를 반올림하여 정수 좌표로 변환하는 시점에 확정되며, 이 정수 좌표가 각 가우시안이 속한 복셀을 지정한다. 이때 Ⅱ장 1절에서 기술하였듯이 앵커 구성에서 기하 부호화는 양자화 이후의 좌표를 무손실로 부호화하여 동일 좌표의 점들을 중복점으로 보존하며, 복호화 시에도 동일 위치에 해당 개수만큼 복제된다. 이에 따라 각 점의 속성 또한 여전히 개별적으로 부호화된다. 따라서 학습으로 형성된 중복을 실제 비트 절감으로 전환하기 위해서는 중복점을 하나로 병합하는 과정이 필요하다. 본 논문에서는 동일한 정수 좌표를 갖는 가우시안들의 집합 C를 식 (11)과 같이 단순 평균을 통해 하나의 가우시안으로 대체한다.

$$\widehat{x_C} = \frac{1}{|C|}\sum_{i\in C}\widetilde{x_i}\ , \qquad \widehat{a_C} = \frac{1}{|C|}\sum_{i\in C}a_i \tag{11}$$

식 (11)에서 $\widetilde{x_i}$와 a_i는 각각 가우시안 i의 양자화 및 역양자화를 거친 위치와 속성을 나타낸다. 집합 C에 속한 가우시안들은 모두 동일한 위치로 역양자화되므로 $\widehat{x_C}$는 해당 복셀의 중심과 일치하며, 병합에 따른 위치 오차는 발생하지 않는다. 반면 속성은 색상, 불투명도, 크기, 회전 등 종류에 따라 서로 다른 성격을 갖는다. 그러나 식 (4)의 속성 근사 엔트로피 손실은 속성의 종류를 구분하지 않고 최근접 이웃 간 차이를 최소화하며, 기하 옥트리 손실 또한 동일한 최근접 이웃 그래프 상에서 정의된다. 따라서 동일 복셀로 병합되는 가우시안들은 학습 과정에서 모든 속성의 차이가 이미 축소된 이웃 쌍에 해당한다. 이에 모든 속성에 대해 동일하게 단순 평균을 적용한다. 이 과정을 통해 부호화 대상이 되는 전체 점의 수가 감소한다.

병합 이후의 속성 평균값은 해당 복셀을 대표하는 값이기는 하나 렌더링 품질 관점에서 최적인 값은 아니므로, 병합은 일정한 화질 저하를 수반한다. 이를 회복하기 위해 병합된 모델에 대해 속성 보정 self-finetuning을 수행한다. 이 단계에서는 위치를 고정한 채 식 (2)의 기존 손실만을 적용한다. 이는 앞선 단계에서 확보한 복셀 점유 구조를 유지하기 위함으로, 위치를 고정한 상태에서 속성만을 다시 학습함으로써 감소한 점의 수를 유지하는 동시에 렌더링 화질을 회복한다.

Ⅳ. 실험 결과

1. 실험 환경

본 논문에서 제안하는 방법의 성능을 평가하기 위해

G-PCC Amd1 참조 소프트웨어인 TMC13-GS v2.0-rc2[17]의 가우시안 최적화 단계에 제안하는 방법을 구현하고, 앵커 및 앵커에 기존 가우시안 최적화 기법을 적용한 경우와의 성능 비교를 수행한다. 여기서 앵커는 가우시안 최적화를 적용하지 않은 G-PCC Amd1을 의미하며, 기존 가우시안 최적화 기법은 선택 도구로 채택된 Gaussian Pruning과 Gaussian Self-finetuning을 적용하는 경우를 의미한다. 제안하는 방법은 Gaussian Pruning 이후 기하 인지 self-finetuning, 복셀 단위 가우시안 병합, 속성 보정 self-finetuning을 순차적으로 수행한다. 세 조건은 모두 동일한 부호화 설정을 사용하며, 부호화기와 복호화기의 어떠한 모듈도 수정하지 않는다. 부호화 조건은 CTC[12]에 규정된 옥트리-하이브리드 구성을 따라 기하는 준무손실로, 속성은 손실 부호화하였으며, 위치 비트 심도는 18-bit, 나머지 속성의 비트 심도는 각각 12-bit로 설정하였다. 압축 성능은 네 개의 rate point에 대해 평가하였으며, 각 rate point의 속성 양자화 파라미터는 표 1과 같다. 실험에 사용한 데이터셋은 GSC CTC의 단일 프레임 테스트 시퀀스 19종이며, 이 중 7종은 정면 촬영 시퀀스이고 나머지 12종은 객체 중심 시퀀스이다.

표 1. Rate point별 양자화 파라미터
Table 1. Quantization parameters for each rate point

Parameter	Rate 1	Rate 2	Rate 3	Rate 4
Scale	38	30	22	14
Rotation	38	30	22	14
Opacity	46	38	30	22
SH-DC	26	18	10	6
SH-AC 1st (1~3)	38	30	22	14
SH-AC 2nd (4~8)	46	38	30	22
SH-AC 3rd (9~15)	49	41	33	25

가우시안 최적화 단계의 학습 설정은 다음과 같다. Gaussian Pruning은 GSCM 기반 중요도 산출 방식[18]을 적용하여 전체 가우시안의 30%를 제거하였으며, 이어지는 기하 인지 self-finetuning은 10,000회, 가우시안 병합 이후의 속성 보정 self-finetuning은 5,000회 반복하였다. 이때 기하 인지 self-finetuning의 초기 2,000회 구간에서는 기하 옥트리 손실을 적용하지 않고 속성이 먼저 안정화되도록 하였다. 각 가우시안에 대해 탐색하는 최근접 이웃의 수 k는 16으로 설정하였고, 유효 이웃 쌍 비율 τ와 기울기 비율 ρ는 대표 시퀀스에 대한 파라미터 탐색 실험을 통해 렌더링 PSNR이 유지되는 범위에서 BD-rate 성능이 가장 우수한 값으로 결정하였으며, 각각 0.333과 0.018을 모든 시퀀스에 공통으로 적용하였다. 가우시안의 위치는 월드 좌표계의 실수값으로 표현되어 장면의 크기에 따라 동일한 학습률이 서로 다른 이동량에 대응하므로, 위치의 학습률은 장면별 복셀 크기에 비례하도록 설정하여 모든 시퀀스에서 한 번의 갱신에 따른 이동량이 복셀 단위로 동일하도록 하였다. 그 외 속성의 학습률과 속성 근사 엔트로피 손실의 가중치는 기존 Self-finetuning의 설정에 따라 각각 기본 학습률의 0.1배와 5×10^{-10}을 사용하였으며, 이는 기존 가우시안 최적화 기법과 제안하는 방법에 동일하게 적용하였다.

화질 평가는 GSC CTC의 방식에 따라 각 시점에서 원본 GS 모델을 렌더링한 결과를 기준으로 복원된 모델의 렌더링 결과를 비교하여 수행하였다. 화질 지표로는 RGB-PSNR과 YUV-SSIM을 사용하였으며, 압축 성능은 전체 비트량에 대한 두 지표의 BD-rate를 평가하였다. 제안하는 방법은 Python 3.10 기반의 PyTorch 2.4.1 및 CUDA 12.4 환경에서 구현하였으며, 실험은 AMD Ryzen Threadripper PRO 3975WX 프로세서, NVIDIA RTX A5000 GPU, 256GB RAM을 갖춘 Ubuntu 20.04 환경에서 수행하였다.

2. 압축 성능 평가

제안하는 방법의 압축 성능을 앵커, 그리고 앵커에 Gaussian Pruning과 Self-finetuning을 적용한 기존 가우시안 최적화 기법과 비교한 결과를 표 2에 나타내었다. 실험 결과, 제안하는 방법은 앵커 대비 평균 BD-rate 성능이 RGB-PSNR 기준 -29.75%, YUV-SSIM 기준 -30.82%로 나타나 우수한 압축 성능을 달성하였다. 특히, 기존 가우시안 최적화 기법과 비교하더라도 각각 -4.83%와 -5.28%의 추가적인 부호화 효율 개선을 이루어냈다. 시퀀스 계열별로 살펴보면 기존 가우시안 최적화 기법 대비 RGB-PSNR 기준 정면 촬영 계열에서 -6.95%, 객체 중심 계열에서 -3.59%의 BD-rate를 기록하였다.

개별 시퀀스 분석 결과, 전체 19종 중 18종에서 기존 가우시안 최적화 기법 대비 BD-rate 절감이 확인되었다. library에서만 RGB-PSNR 기준 +0.23%의 미세한 손실이 나

표 2. 시퀀스별 BD-rate 성능
Table 2. BD-rate performance for each sequence

Class	Sequence	BD-rate [%]			
		Anchor vs. Proposed		Anchor w/ Pruning and Self-finetuning vs. Proposed	
		RGB-PSNR	YUV-SSIM	RGB-PSNR	YUV-SSIM
Forward facing	bartender_tracked	-43.38	-44.33	-12.50	-13.27
	bartender_semitracked	-37.31	-36.92	-1.98	-1.94
	cinema_tracked	-41.76	-41.91	-14.89	-14.27
	cinema_semitracked	-39.67	-39.82	-5.06	-5.06
	breakfast_tracked	-33.85	-36.69	-4.67	-5.82
	breakfast_semitracked	-33.69	-34.62	-4.38	-5.34
	breakfast_untracked	-33.51	-21.95	-5.19	-2.90
Forward facing Avg.		-37.60	-36.61	-6.95	-6.94
Object centric	manwithfruit_tracked	-29.33	-23.58	-0.34	+1.38
	lego_ferrari	-25.50	-39.37	-1.74	-3.26
	lego_bugatti	-9.55	-23.75	-5.96	-3.08
	cricket_player	-33.04	-33.17	-1.84	-6.73
	plant	-31.14	-34.03	-2.58	-2.89
	solo_tango_female	-28.94	-35.06	-3.12	-6.18
	solo_tango_male	-31.01	-30.91	-6.96	-5.89
	tango_duo	-31.41	-36.19	-3.28	-4.58
	tennis_player	-25.21	-35.14	-2.32	-11.18
	library	-32.75	-34.12	+0.23	-2.69
	flowerdance	-17.72	+22.23	-6.50	-1.31
	gymnast	-6.57	-26.24	-8.70	-5.31
Object centric Avg.		-25.18	-27.44	-3.59	-4.31
Overall Avg.		-29.75	-30.82	-4.83	-5.28

타났으나, YUV-SSIM 기준으로는 -2.69%의 성능 향상을 보였다. 가장 큰 폭의 부호화 효율 개선을 보인 시퀀스는 cinema_tracked와 bartender_tracked로 RGB-PSNR 기준 각각 -14.89%와 -12.50%를 기록하였다. 반면 동일한 장면을 대상으로 하는 bartender_semitracked는 -1.98%에 그쳤으며, 이러한 시퀀스별 성능 편차의 원인은 이어지는 3절에서 가우시안 수의 변화와 함께 분석한다. 한편 flowerdance는 앵커 대비 YUV-SSIM 기준 +22.23%로 유일하게 큰 폭의 손실이 나타났는데, 이는 기존 가우시안 최적화 기법에서도 +22.78%로 동일하게 관측되는 값이다. 즉 해당 손실은 Pruning 과정에서 제거된 가우시안이 구조적 유사도에 미친 영향으로, 제안하는 방법은 이 시퀀스에서도 기존 가우시안 최적화 기법 대비 RGB-PSNR 기준 -6.50%, YUV-SSIM 기준 -1.31%의 성능 향상을 달성하였다.

그림 6은 대표 시퀀스에 대하여 네 개의 rate point에서 측정한 전체 비트량과 렌더링 화질의 관계를 나타낸 것으로, 앵커와 기존 가우시안 최적화 기법, 제안하는 방법의 곡선을 함께 도시하였다. 세 방법이 도달하는 최고 화질의 상한선은 유사한 반면, 동일한 화질을 달성하는 데 요구되는 비트량에서는 뚜렷한 차이가 나타난다. 제안하는 방법은 모든 rate point에서 기존 가우시안 최적화 기법보다 낮은 비트량으로 동등한 렌더링 화질을 달성하였으며, 두 곡선 사이의 간격은 비트율 구간에 관계없이 일정하게 유지된다. 이는 제안하는 방법에 의한 비트량 절감이 특정 비트율 구간에서만 발생하는 것이 아니라, 부호화 대상 가우시안의 수가 감소함에 따라 전 구간에서 안정적으로 확보됨을 의미한다. 다만, 일부 객체 중심 시퀀스의 고비트율 구간에서 앵커가 더 높은 화질에 도달하는 경향이 관찰되었다. 이는 Pruning 단계에서 제거된 가우시안으로 인해 복원 가능한 화질의 상한이 제한되기 때문으로 기존 가우시안 최적화 기법에서도 동일하게 관측되며, 제안하는 방법은 해당 시퀀스에서도 기존 가우시안 최적화 기법 대비 일관된 성능 향상을 보였다.

표 3은 기존 가우시안 최적화 기법과 제안하는 방법의 가우시안 최적화 시간, 그리고 두 방법의 앵커 대비 G-PCC Amd1 부호화 및 복호화 시간을 측정한 결과이다. 이때 가

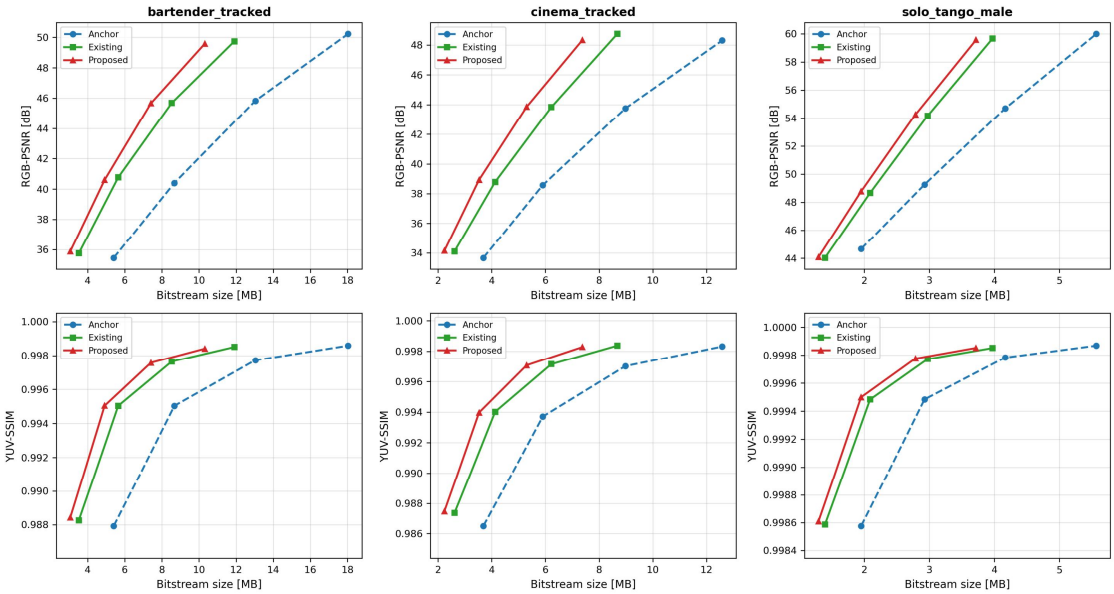

그림 6. 대표 시퀀스에 대한 RD 곡선
Fig. 6. RD-curve comparison for representative sequences

표 3. 시퀀스별 실행 시간 비교
Table 3. Comparison of running time for each sequence

Class	Sequence	Gaussian optimization [min]		Encoding time [%]		Decoding time [%]	
		Anchor w/ Pruning and Self-finetuning	Proposed	Anchor w/ Pruning and Self-finetuning	Proposed	Anchor w/ Pruning and Self-finetuning	Proposed
Forward facing	bartender_tracked	12.41	18.45	70.64	56.53	50.58	44.30
	bartender_semitracked	15.47	23.16	60.53	57.64	43.03	42.29
	cinema_tracked	9.23	13.90	76.54	58.46	61.83	47.04
	cinema_semitracked	12.18	19.03	61.94	58.36	45.68	42.60
	breakfast_tracked	11.17	16.95	66.45	65.59	51.38	43.78
	breakfast_semitracked	11.05	16.99	63.85	67.14	47.72	48.08
	breakfast_untracked	10.90	16.53	66.60	73.39	60.01	53.50
Forward facing Avg.		11.77	17.86	66.65	62.45	51.46	45.94
Object centric	manwithfruit_tracked	9.15	13.43	62.84	61.85	69.30	70.38
	lego_ferrari	22.63	33.59	67.93	71.50	72.01	63.58
	lego_bugatti	12.70	18.62	49.27	47.73	46.15	43.48
	cricket_player	13.06	18.74	44.60	44.48	45.45	43.42
	plant	10.47	15.52	72.35	67.93	77.64	72.14
	solo_tango_female	10.31	15.51	65.99	64.60	75.36	74.78
	solo_tango_male	12.04	16.23	49.27	44.73	45.51	41.25
	tango_duo	11.18	16.45	45.42	44.22	47.47	44.50
	tennis_player	13.19	18.84	45.51	45.35	44.56	44.34
	library	52.58	77.88	43.39	41.25	47.24	45.79
	flowerdance	23.89	35.13	58.09	54.19	45.89	45.59
	gymnast	6.36	9.31	46.68	45.89	46.50	44.60
Object centric Avg.		16.46	24.10	54.28	52.81	55.26	52.82
Overall Avg.		14.73	21.80	58.84	56.36	53.86	50.29

우시안 최적화는 rate point에 독립적으로 수행되므로 시퀀스당 한 차례 측정하였으며, 부호화 및 복호화 시간은 네 개의 rate point에서 측정한 앵커 대비 상대값이다. 제안하는 방법은 앵커 대비 부호화 시간이 평균 56.36%, 복호화 시간이 평균 50.29%로 나타나 기존 가우시안 최적화 기법의 58.84%와 53.86% 대비 각각 2.48%p와 3.57%p 단축되었다. 이는 Pruning에 더하여 가우시안 병합이 부호화 대상 점의 수를 추가로 감소시킨 결과이다. 반면 가우시안 최적화 시간은 기존 가우시안 최적화 기법이 평균 14.73분, 제안하는 방법이 평균 21.80분으로 1.48배 증가하였다. 이는 학습 반복 횟수가 10,000회에서 15,000회로 증가한 비율인 1.5배와 거의 일치하며, 최적화 시간의 증가가 제안하는 방법의 추가적인 연산에 따른 것이 아니라 속성 보정 self-finetuning 단계의 추가에서 비롯되었음을 보여준다.

3. 가우시안 수 감소에 따른 비트량 및 화질 분석

본 절에서는 2절에서 확인된 부호화 효율 개선이 가우시안 수의 감소에서 비롯되었음을 비트량과 렌더링 화질의 관점에서 분석한다. 표 4는 제안하는 방법의 가우시안 병합 단계 전후 가우시안 수와 그에 따른 비트량 변화를 나타낸다. 병합률은 Pruning 후 가우시안 수 대비 병합으로 제거된 가우시안의 비율을, 비트량 감소율은 기존 가우시안 최적화 기법 대비 동일 rate point에서 측정한 감소 비율을 의미한다.

정면 촬영 시퀀스에서는 cinema_tracked와 bartender_tracked가 각각 17.40%와 16.17%의 병합률을 기록하였으며, 이에 따라 기하 비트량이 14.43%와 13.49% 감소하였다. 동시에 부호화 대상 가우시안 자체가 감소함에 따라 속성 비트량 또한 14.69%와 13.07% 감소하였다. 이는 표 2에서 두 시퀀스가 RGB-PSNR 기준 기존 가우시안 최적화 기법 대비 각각 -14.89%와 -12.50%로 가장 큰 성능 향상을 보인 결과와 일치한다. 반면, 객체 중심 시퀀스에서는 병합률이 대체로 낮게 나타났으며, lego_ferrari, lego_bugatti, cricket_player, tango_duo, gymnast에서는 병합이 발생하지 않았다. 이러한 편차는 시퀀스 계열이 아니라 각 장면을 구성하는 가우시안의 공간적 밀도 차이에서 기인한다. $\eta(v)=\tau$를 만족하는 기준 반경 v는 시퀀스에 따라 4.58복셀에서 149.60복셀까지 약 33배의 차이를 보였으며, v가 큰 시퀀스일수록 병합률이 낮게 나타났다. v가 크다는 것

표 4. 제안하는 방법에 의한 가우시안 수 및 비트량 변화
Table 4. Number of Gaussians and bitrate reduction by the proposed method

Class	Sequence	Number of Gaussians			Bitrate reduction [%]	
		After pruning	After merging	Merge ratio [%]	Geometry	Attribute
Forward facing	cinema_tracked	296,275	244,729	17.40	14.43	14.69
	bartender_tracked	397,407	333,148	16.17	13.49	13.07
	breakfast_tracked	369,708	356,455	3.58	4.57	4.95
	breakfast_semitracked	371,885	360,540	3.05	4.21	3.43
	breakfast_untracked	350,458	340,846	2.74	3.54	3.02
	cinema_semitracked	477,899	471,950	1.24	3.13	5.30
	bartender_semitracked	549,119	544,822	0.78	1.96	4.28
Object centric	solo_tango_male	128,429	119,948	6.60	10.99	5.60
	library	2,137,233	2,079,522	2.70	4.13	3.48
	plant	132,816	130,864	1.47	3.16	3.20
	solo_tango_female	131,528	130,418	0.84	1.87	2.58
	manwithfruit_tracked	32,208	32,167	0.13	0.45	0.32
	tennis_player	134,476	134,350	0.09	0.35	1.58
	flowerdance	700,000	699,755	0.03	0.14	2.04
	lego_ferrari	139,135	139,135	0.00	0.00	0.13
	lego_bugatti	139,583	139,580	0.00	0.06	0.12
	cricket_player	137,401	137,397	0.00	0.01	0.59
	tango_duo	139,338	139,338	0.00	0.01	0.48
	gymnast	70,000	70,000	0.00	0.01	1.27

은 인접한 가우시안 쌍이 복셀 단위로 멀리 떨어져 있음을 의미하므로, 희소한 공간 분포를 갖는 시퀀스에서는 동일한 복셀을 공유할 수 있는 가우시안 쌍 자체가 존재하지 않는다. 동일한 장면을 대상으로 하는 bartender_tracked와 bartender_semitracked의 v가 각각 4.58과 57.80으로 12배 이상 차이를 보이는 것 또한 이러한 특성을 뒷받침한다. 이러한 병합률의 차이는 표 3의 실행 시간에도 반영되어 정면 촬영 계열이 객체 중심 계열보다 부호화 및 복호화 시간이 더 크게 감소하였다.

한편 병합이 발생하지 않은 시퀀스에서도 표 2와 같이 기존 가우시안 최적화 기법 대비 성능 향상이 관측되었다. 제안하는 방법은 기하 인지 self-finetuning과 속성 보정 self-finetuning을 합하여 기존 가우시안 최적화 기법보다 많은 학습을 수행하므로, 가우시안 병합 이외의 요인을 확인하기 위한 추가 실험을 수행하였다. 기존 가우시안 최적화 기법에 속성 보정 self-finetuning과 동일한 횟수의 학습만을 추가한 경우, RGB-PSNR 기준 기존 가우시안 최적화 기법 대비 cinema_tracked와 bartender_tracked에서 각각 -2.42%와 -1.27%의 성능에 그쳤다. 이는 제안하는 방법이 달성한 -14.89%와 -12.50%의 각각 16%와 10% 수준으로, 병합이 발생하는 시퀀스에서 관측된 성능 향상의 대부분이 가우시안 병합에 기인함을 보여준다. 반면 병합이 발생하지 않은 gymnast와 lego_bugatti에서는 표 4와 같이 기하 및 속성 비트량의 감소가 거의 나타나지 않으므로, 이들 시퀀스의 성능 향상은 비트량의 절감이 아니라 동일 비트량에서의 렌더링 화질 향상에서 비롯된다. 두 시퀀스에서 속성 학습만을 추가한 경우 각각 -4.98%와 -2.88%를, 또한 기하 옥트리 손실 없이 위치만을 학습 대상에 포함한 경우 각각 -3.86%와 -3.04%를 기록하였다. 기존 가우시안 최적화 기법이 가우시안의 위치를 고정한 채 속성만을 조정하는 반면, 제안하는 방법은 위치 또한 학습 대상에 포함하므로 Pruning으로 제거된 가우시안이 담당하던 영역을 인접 가우시안이 이동하여 보완할 수 있다. 따라서 병합이 발생하지 않은 시퀀스의 성능 향상은 추가적인 속성 학습과 위치 자유도의 확대가 화질 향상으로 이어져 성능이 개선된 것으로 확인하였다.

그림 7은 압축 및 복원 과정을 거친 모델이 원본 3DGS 모델과 비교하여 어느 정도의 시각적 품질을 유지하는지 확인하기 위한 것으로, 병합률이 높게 나타난 bartender_tracked, cinema_tracked, solo_tango_male의 Rate 4에서의 렌더링 결과를 나타낸다. 좌측은 확대 영역의 위치를 표시한 전체 화면이며, 우측은 해당 영역을 확대하여 원본 모

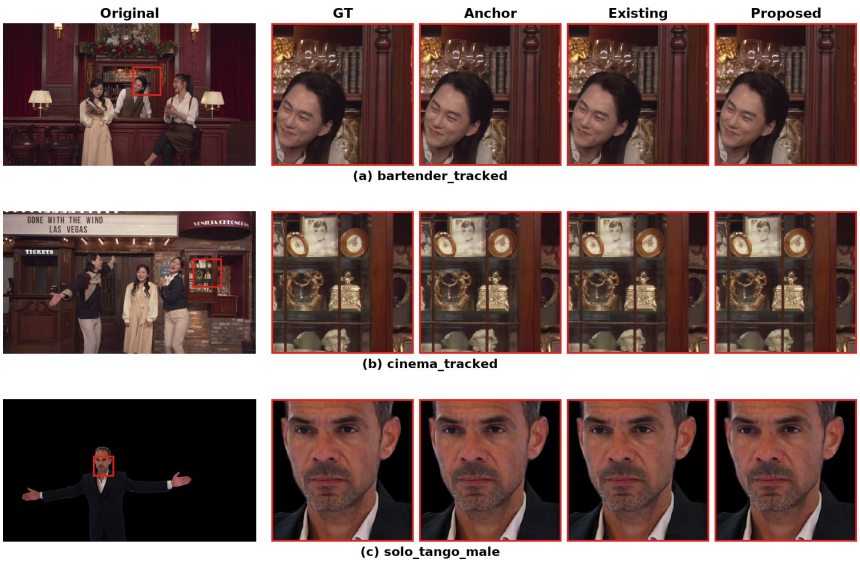

그림 7. 원본 3DGS 모델과 각 방법으로 압축·복원한 모델의 렌더링 결과 비교
Fig. 7. Rendering comparison between the original 3DGS model and the models compressed by each method

델과 세 방법의 복원 결과를 순서대로 배열한 것이다. bartender_tracked의 경우 전체 가우시안의 16.17%가 병합되었음에도, 머리카락 경계와 배경의 유리잔과 같은 고주파 영역에서 기존 가우시안 최적화 기법과 구별되는 화질 저하가 관측되지 않았다. cinema_tracked의 진열장 내부 장식과 solo_tango_male의 얼굴 세부 표현에서도 동일한 경향이 확인된다. 이때 제안하는 방법은 세 시퀀스에서 기존 가우시안 최적화 기법 대비 6.4%에서 15.0%까지 더 적은 비트량을 사용하였다. 이는 가우시안 병합 과정에서 발생한 속성의 변화가 이어지는 속성 보정 self-finetuning을 통해 보완됨에 따라, 제안하는 방법이 시각적 품질의 손실 없이 가우시안 수의 감소에 따른 비트량 절감을 달성함을 입증한다.

V. 결 론

본 논문은 G-PCC Amd1 기반 3DGS 압축에서 가우시안의 공간적 중복으로 인한 기하 및 속성 비트량의 낭비를 해소하기 위해, 가우시안의 위치를 학습 대상으로 포함하는 기하 옥트리 손실 기반 가우시안 최적화 방법을 제안하였다. 제안하는 방법은 옥트리 부호화가 수행되는 복셀 격자 좌표계 상에서 인접 가우시안 간 거리를 미분 가능한 형태로 정의한 기하 옥트리 손실을 렌더링 손실과 속성 근사 엔트로피 손실로 구성된 기존 Self-finetuning의 손실에 결합하여 가우시안의 위치를 학습함으로써, 위치 이동에 따른 화질 저하가 작은 가우시안 쌍을 선택적으로 동일 복셀로 수렴시킨다. 또한 동일 복셀로 수렴한 가우시안들을 하나의 점으로 병합하여 부호화 대상이 되는 점의 수를 직접 감소시키고, 위치를 고정한 상태에서 속성을 다시 학습하여 병합 과정에서 발생한 왜곡을 회복한다. 실험 결과, 제안하는 방법은 가우시안 최적화를 적용하지 않은 G-PCC Amd1 앵커 대비 RGB-PSNR과 YUV-SSIM 기준 BD-rate에서 각각 평균 -29.75%와 -30.82%의 성능 향상을 보였다. 또한 Gaussian Pruning 및 Self-finetuning을 앵커에 적용한 경우 대비 각각 평균 -4.83%와 -5.28%의 추가적인 성능 향상을 보였으며, 이를 통해 렌더링 품질을 유지하면서도 압축 효율을 효과적으로 개선함을 확인하였다. 향후 연구에서는 유효 이웃 쌍 비율과 기울기 비율을 각 시퀀스의 율-왜곡 특성에 따라 결정하고, 단일 프레임뿐만 아니라 다중 프레임 시퀀스에서도 동작할 수 있도록 확장하는 연구를 수행할 예정이다.

참 고 문 헌 (References)

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3D Gaussian Splatting for Real-Time Radiance Field Rendering,” ACM Transactions on Graphics, Vol. 42, No. 4, Article 139, pp. 1-14, July 2023.
doi: https://doi.org/10.1145/3592433

[2] J. Hwang, M. Kim, J. Byeon, and D. Sim, “Analysis of Representation and Compression Performance of Point Cloud and 3D Gaussian Splatting for Efficient 3D Data Representation,” The Korean Institute of Broadcast and Media Engineers Fall Conference, Seoul, Korea, pp. 283-286, 2025.

[3] S. Li, Y. Liao, and L. Yu, “[GSC][JEE2] A Potential Video-based Anchor for Gaussian Splats Coding,” ISO/IEC JTC 1/SC 29/WG 4/m72063, The 150th MPEG meeting, Online, March 2025.

[4] K. Unno, D. Fujii, K. Nonaka, and K. Kawamura, “[JEE6] Preliminary software implementation based on G-PCC 1st edition for anchor generation of 3D Gaussian coding,” ISO/IEC JTC 1/SC 29/WG 7/m70095, The 148th MPEG meeting, Kemer, Turkey, November 2024.

[5] M. Lee, J. Lee, H. Oh, and H. Hur, “[G-PCC][JEE6] Anchor for 3D gaussian coding with TMC13-v27,” ISO/IEC JTC 1/SC 29/WG 7/m69983, The 148th MPEG meeting, Kemer, Turkey, November 2024.

[6] G-PCC Test Model v27, ISO/IEC JTC 1/SC 29/WG 7/N00940, The 147th MPEG meeting, Sapporo, Japan, July 2024.

[7] Description of JEE 6.2 on anchor generation, ISO/IEC JTC 1/SC 29/WG 4/N0585, The 148th MPEG meeting, Kemer, Turkey, November 2024.

[8] K. Lee, J. Jeong, and H. Choo, “Standardization status of Gaussian Splat Coding (GSC) in MPEG,” Broadcast and Media Magazine, Vol. 30, No. 3, pp. 41-50, July 2025.

[9] ISO/IEC 23090-9:2023, Information technology – Coded representation of immersive media – Part 9: Geometry-based point cloud compression, March 2023, https://www.iso.org/standard/78990.html

[10] WD of ISO/IEC 23090-9 Amd1 - Geometry-based Gaussian Splat Coding, ISO/IEC JTC 1/SC 29/WG 7/N01361, The 153rd MPEG meeting, Online, January 2026.

[11] Text of ISO/IEC 23090-9 CDAM - Geometry-based gaussian splat coding, ISO/IEC JTC 1/SC 29/WG 7/N01454, The 154th MPEG meeting, Santa Eularia, Spain, April 2026.

[12] Common test conditions for Gaussian splat coding, ISO/IEC JTC 1/SC 29/WG 7/N01414, The 153rd MPEG meeting, Online, January 2026.

[13] R. L. de Queiroz and P. A. Chou, “Compression of 3D Point Clouds

Using a Region-Adaptive Hierarchical Transform," IEEE Transactions on Image Processing, Vol. 25, No. 8, pp. 3947-3956, August 2016. doi: https://doi.org/10.1109/TIP.2016.2575005

[14] G-PCC 2nd Edition codec description, ISO/IEC JTC 1/SC 29/WG 7/N00942, The 147th MPEG meeting, Sapporo, Japan, July 2024.

[15] W. Kawai and O. Nakagami, "[GSC][JEE 6.2-related] Improving G-PCC Anchor with Gaussian Self-Finetuning," ISO/IEC JTC 1/SC 29/WG 7/m73559, The 151st MPEG meeting, Daejeon, Korea, June 2025.

[16] W. Kawai and O. Nakagami, "[G-PCC][Amd1] Gaussian Pruning for G-PCC Amd1 Anchor," ISO/IEC JTC 1/SC 29/WG 7/m76518, The 154th MPEG meeting, Santa Eularia, Spain, April 2026.

[17] G-PCC Amd1 test model TM13-GS v2, ISO/IEC JTC 1/SC 29/WG 7/N01494, The 154th MPEG meeting, Santa Eularia, Spain, April 2026.

[18] B. Brand, J. Keinert, L. Müller, and C. Schinabeck, "[V-PCC][Amd1] Compact representation of Gaussian splats by adaptive quantization for V-PCC," ISO/IEC JTC 1/SC 29/WG 7/m75039, The 153rd MPEG meeting, Online, January 2026.

저 자 소 개

황 정 원

- 2025년 2월 : 광운대학교 컴퓨터정보공학부 학사
- 2025년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 석사과정
- ORCID : https://orcid.org/0009-0008-8301-8762
- 주관심분야 : 3D데이터압축, 영상압축, 컴퓨터비전

김 민 태

- 2023년 2월 : 광운대학교 컴퓨터공학과 학사
- 2025년 2월 : 광운대학교 컴퓨터공학과 석사
- 2025년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 박사과정
- ORCID : https://orcid.org/0009-0007-9100-8567
- 주관심분야 : 3D데이터압축, 영상압축, 컴퓨터비전

심 동 규

- 1993년 2월 : 서강대학교 전자공학과 공학사
- 1995년 2월 : 서강대학교 전자공학과 공학석사
- 1999년 2월 : 서강대학교 전자공학과 공학박사
- 1999년 3월 ~ 2000년 8월 : 현대전자 선임연구원
- 2000년 9월 ~ 2002년 3월 : 바로비젼 선임연구원
- 2002년 4월 ~ 2005년 2월 : University of Washington Senior research engineer
- 2005년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 교수
- ORCID : https://orcid.org/0000-0002-2794-9932
- 주관심분야 : 영상신호처리, 영상압축, 컴퓨터비전