

특집논문 (Special Paper)

방송공학회논문지 제30권 제6호, 2025년 11월 (JBE Vol.30, No.6, November 2025)

<https://doi.org/10.5909/JBE.2025.30.6.1027>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

저전력 임베디드 시스템을 위한 GPU 가속 Visual SLAM 및 GPCC 압축 기법

안 승 혁^{a)}, 양 이 석^{a)}, 김 영 규^{a)}, 류 은 석^{a)‡}

GPU-Accelerated Visual SLAM and GPCC Compression for Low-Power Embedded Systems

Seung Huk Ahn^{a)}, Isaac Yang^{a)}, Yeong-Gyu Kim^{a)}, and Eun-Seok Ryu^{a)‡}

요 약

본 논문은 저전력 임베디드 플랫폼에서 GPU를 활용한 시각 동시적 위치 추정 및 지도작성 (vSLAM)과 멀티 스레드를 활용한 기하학 기반 포인트 클라우드 압축 기법 (GPCC)을 제안한다. 제안한 시스템은 이동형 로봇에 장착된 RGB-D 센서로 데이터를 취득하고, NVIDIA Jetson Nano에서 실시간으로 처리한다. VSLAM 단계에서는 GPU 병렬 가속을 통해 오도메트리 연산 시간을 Baseline 대비 평균 47% 단축하였으며, ICP 정량 평가 (RMSE 0.0237m)를 통해 매핑 화질 저하가 미미함을 입증하였다. GPCC 단계에서는 CPU 멀티 스레딩을 적용하여 표준 데이터셋에서 평균 43%의 인코딩 시간 절감을 달성하였다. 또한, 전력 분석 결과 작업 시간 단축으로 총 전력 효율성이 향상됨을 확인하였다. 본 연구는 제한된 자원 환경에서도 실시간 3D 매핑과 압축이 가능함을 보여주며, 향후 모바일 로봇, 디지털 트윈 등 다양한 응용 분야에서의 활용 가능성을 제시한다.

Abstract

This paper presents an integrated framework that utilizes GPU for visual Simultaneous Localization and Mapping (vSLAM) with Geometry-based Point Cloud Compression (GPCC) using multi-threading for efficient 3D mapping and compression on low-power embedded platforms. The system is implemented on an NVIDIA Jetson Nano, processing data from an RGB-D camera. The vSLAM pipeline leverages GPU acceleration to reduce odometry estimation latency by 45% and up to 49% compared to a CPU-only method. Crucially, this speedup is achieved with little to no loss of mapping quality, as quantitatively verified by ICP analysis (0.0237m RMSE). For compression, CPU multi-threading is applied to the GPCC encoder, achieving an average encoding time reduction of 43% on standard datasets. Furthermore, a power analysis confirms the system's suitability for embedded use. While the instantaneous power (W) increases, the significant reduction in processing time leads to greater overall power (Joule) efficiency. This work demonstrates that real-time, high-quality 3D mapping and standard-compliant compression are simultaneously achievable on resource-constrained mobile devices.

Keyword : Point cloud compression, SLAM, Low-power platform, ROS, 3D reconstruction

I. 서론

최근 가상 현실 (VR), 증강 현실 (AR), 디지털 트윈과 같은 기술의 발전으로, 실제 환경의 고정밀 3차원 공간 데이터를 실시간으로 취득하고 재구성하려는 수요가 급격히 증가하고 있다^[1]. 이러한 몰입형 콘텐츠 환경에서는 사용자의 시점 변화에 따라 장면이 즉시 반응해야 하며, 이를 위해서는 정확한 위치 추정과 동시에 3차원 지도를 생성하는 동시적 위치 추정 및 지도작성 (SLAM) 기술이 핵심으로 주목받고 있다^[2].

SLAM은 카메라나 LiDAR와 같은 센서를 통해 주변 공간을 인식하고 이를 2D 또는 3D 지도 형태로 변환하는 기술이다^[3]. 이러한 센서는 자동차, 로봇, 드론 등 다양한 이동체에 장착되어 실내외의 공간 매핑에 활용된다^[4]. 특히 3차원 매핑 시에는 LiDAR 스캔이나 영상 기반 깊이 정보를 통해 포인트 클라우드 데이터 (PCD) 혹은 다각형 3D 파일 형식 (PLY)과 같은 대용량 데이터 파일이 생성된다. 문제는 이러한 포인트 클라우드 데이터가 크기가 방대하여, 현장 데이터 취득 시 저장 및 전송 과정에서 심각한 부담이 발생한다는 점이다^[5]. 기존에는 데이터를 모두 저장한 뒤 무선 네트워크를 통해 서버로 전송하거나, 직접 저장 장치를 서버로 이동시켜 옮기는 방식이 주로 사용되었다^[6]. 그러나 네트워크 환경이 불안정하면 데이터 손실 위험이 있고, 날씨나 환경 변

화로 동일 데이터를 재취득하기 어려운 경우도 많다^[7,8]. 이 때문에 불필요한 시간 낭비와 비효율성이 빈번히 발생한다.

이러한 한계를 극복하기 위해서는 데이터의 정확성을 유지하면서도 현장에서 곧바로 압축할 수 있는 경량화 기술이 필요하다. 특히 차량이나 이동형 로봇처럼 고성능 컴퓨터를 탑재하기 어려운 환경에서는 저전력 시스템에서도 실행이 가능한 SLAM 및 압축 기술이 요구된다. 최근 국제 표준화단체 중 동영상 전문가 그룹 (MPEG)에서 제안한 기하학 기반 포인트 클라우드 압축 기법 (GPCC)은 대규모 포인트 클라우드 데이터를 효율적으로 줄일 수 있으나, 연산량이 커서 일반적인 임베디드 보드에서는 실행이 제한적이다^[9].

본 논문은 그림 1과 같이 저전력 그래픽 처리 장치 (GPU) 기반 이동형 로봇에 vSLAM을 적용하여 실시간 포인트 클라우드를 생성하고, 이를 중앙 처리 장치 (CPU) 멀티 스레드를 활용한 GPCC 기법으로 곧바로 압축 및 저장하는 통합 파이프라인을 제안한다. 본 시스템은 이동형 로봇 기반 플랫폼에서 구현되었으며, 차량 주행을 통한 도로 매핑이나 실내외 로봇 주행 등 고성능 장비를 탑재하기 어려운 다양한 상황에서 활용 가능하다. 또한 다양한 GPCC 부호화 파라미터 실험을 통해 압축 효율성과 임베디드 환경에서의 실행 가능성을 검증한다.

II. 관련 연구

VSLAM은 연속적인 카메라 프레임을 통해 설정된 지점을 추적하여 3차원 위치를 삼각측량한다. 그와 동시에 이 정보를 활용하여 카메라의 위치를 추정한다. 이러한 시스템의 목표는 탐색을 위해 주변 환경을 자체 위치와 연관지어 매핑하는 것이다. 주로 RGB-D 카메라 또는 LiDAR 센서를 활용하여 매핑을 진행하는데, 이는 많은 컴퓨팅 리소스가 필요하므로 저전력 시스템 환경에서는 사용이 어렵다. 또한 이러한 매핑된 데이터는 용량이 크기 때문에 흔히 고사양 컴퓨터를 사용하여 매핑을 한다. 현재 포인트 클라우드 데이터를 효율적으로 저장하기 위한 연구가 진행되고 있고, vSLAM도 저전력 시스템에서 사용할 수 있도록 연구 중이다. 본 장은 경량화 vSLAM과 포인트 클라우드 압축

a) 성균관대학교 실감미디어공학과(Department of Immersive Media Engineering, Sungkyunkwan University)

‡ Corresponding Author : 류은석(Eun-Seok Ryu)
E-mail: esryu@skku.edu
Tel: +82-2-760-0677

ORCID: <https://orcid.org/0000-0003-4894-6105>

※ 이 논문의 결과 중 일부는 한국방송·미디어공학회 2025년 하계학술대회에서 발표한 바 있음

※ This research was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No.RS-2024-00457605, Development of Movable Robot Cluster-based Volumetric Video Acquisition and Computing Technology for Mega-scale Theaters and Stadiums)

※ This research was also supported by Global Standardization and Commercialization of Copyright Technology Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism (No.RS-2024-00439789, Development of International Standards for CT XR Content Copyright Protection Technologies)

· Manuscript September 25, 2025; Revised November 10, 2025; Accepted November 11, 2025.

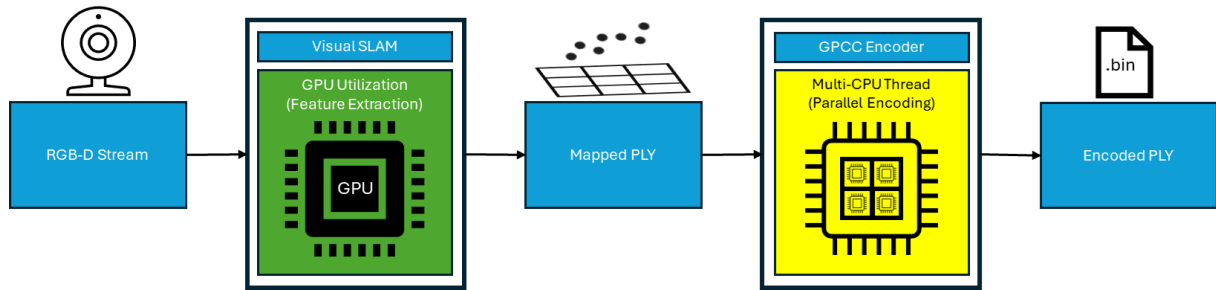


그림 1. 전체 워크플로우. RGB-D 스트림을 GPU 가속 vSLAM으로 처리해 PLY 맵을 생성하고, 이를 GPCC 멀티 스레드로 압축한다.

Fig. 1. Overall workflow. RGB-D stream is processed by GPU-accelerated vSLAM to create a PLY map, which is then compressed via multi-threaded GPCC.

기법에 대해 하나씩 설명한다.

1. 경량화 vSLAM

기존 기계식 LiDAR 센서 기반 방식과 달리 고정형 LiDAR SLAM (SSL SLAM) 기법은 인텔 사의 RealSense L515 카메라와 같이 고정형 LiDAR를 대상으로 제한된 경량화 SLAM 프레임워크이다^[10]. 고정형 LiDAR는 좁은 시야각과 높은 주사 주파수를 가지므로, 기존의 LOAM, LeGO-LOAM과 같은 알고리즘을 그대로 적용할 경우, 연산량이 증가하여 추적이 불안정해지는 문제가 발생한다^[11,12]. 이를 해결하기 위해 SSL SLAM 기법은 회전 불변 특징 추출을 통해 edge와 plane 특징을 안정적으로 검출하고, 로컬 맵 기반 매칭으로 오도메트리를 추정하며, 옥트리 (Octree) 기반 맵 구성으로 효율적인 전역 맵을 구축한다. SSL SLAM은 RealSense L515 카메라를 탑재한 무인 운반차 (AGV)에서 구현되었으며, 기존 LOAM 방식 대비 회전 상황에서의 강인성과 실시간 처리를 인텔 사의 NUC PC에서 가능하다는 것을 입증하였다. 따라서 SSL SLAM은 저전력 시스템에서도 실시간 위치 추정과 고정밀 매핑이 가능함을 보여주었다.

NUC PC의 경우 인텔 사의 i5-10520u CPU가 장착되어 있으며, 저전력이지만 최대 25W까지 전력을 사용하여 본 논문에서 사용할 이동형 로봇에 장착하기에는 어려움이 있다. 따라서 SSL SLAM 기법을 이동형 로봇에 장착할 수 있는 저전력 임베디드 시스템에 최적화하기 위해 CPU 및 GPU를 모두 사용하는 기법을 적용하였다.

2. 포인트 클라우드 압축 기법

포인트 클라우드 압축 기술 중 MPEG의 GPCC는 주로 LiDAR 센서나 RGB-D 카메라로 취득한 데이터를 대상으로 하며, 기하학적 정보는 Octree, 예측 트리 (Predictive Geometry Tree), 트라이슈프 (Trisoup) 방식으로 압축할 수 있고 속성 데이터 정보는 RAHT 또는 Predlift를 통해 부호화한다^[13,14,15,16,17]. 또한 2018년도에 MPEG에서는 포인트 클라우드를 2차원 평면에 투영 후 비디오 코덱을 이용하여 압축하는 기법 (VPCC)을 제안하였으나, VPCC는 연속된 시퀀스 형태의 밀집 포인트 클라우드를 영상으로 변환하여 처리하는 방식이므로, SLAM으로부터 생성되는 단일 PLY 파일과 같은 데이터에는 직접 적용하기 어렵다^[18,19]. 여러 기하학 압축 방법 중 Octree는 GPCC의 기본 방식으로, 다양한 환경에서 안정적으로 동작하며 포인트 클라우드가 불균일하게 분포되어 있어도 압축 효율과 복잡도의 균형이 우수하다. Predtree 방식은 특정 상황에서 압축률이 높을 수 있으나, 손실률이 높아지고 구현 복잡도가 증가하는 단점이 있다. Trisoup 방식 또한 표면 기반 데이터에 적합하나, SLAM을 기반으로 한 불균일한 포인트 클라우드에는 효율적이지 않다.

따라서 본 논문은 저전력 임베디드 환경에서도 신뢰성 있게 활용할 수 있는 Octree 기반의 기하학 압축을 선택하였다. 속성 데이터 부호화의 경우, RAHT는 포인트 클라우드의 색상 및 반사 강도 정보를 효율적으로 압축하면서도 구현이 간단하고 연산량이 적어, Predlift보다 경량 환경에

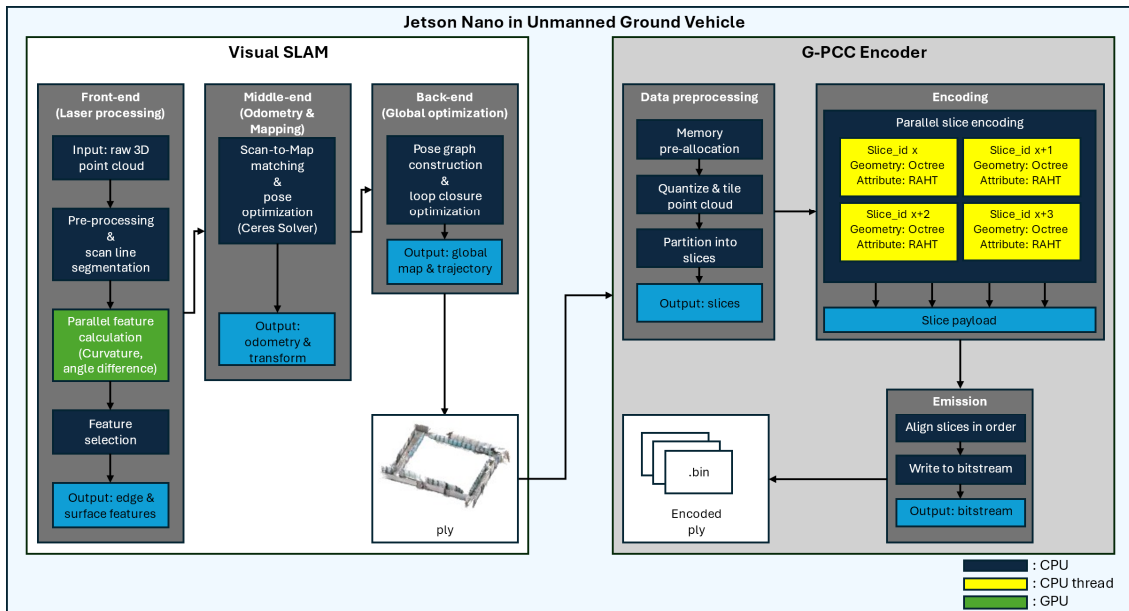


그림 2. 제안하는 방법론의 전체 파이프라인. vSLAM의 프론트엔드는 GPU가, 미들엔드와 백엔드는 CPU가 처리한다. GPCC 인코더는 4개의 CPU 스레드 풀을 활용하여 3D 맵을 병렬 압축한다.

Fig. 2. Proposed methodology's full pipeline. The vSLAM Front-end is GPU-accelerated, while the Middle/Back-end runs on the CPU. The GPCC encoder uses a 4-worker CPU thread pool for parallel compression of the 3D map.

적합하다. 이에 따라 기하학 정보는 Octree 방식, 속성 정보는 RAHT 방식을 적용하여 SLAM 기반 포인트 클라우드 데이터의 효율적이고 실시간 가능한 압축을 수행한다.

GPCC는 현재 지속해서 연구가 진행 중인 표준이다. 다음 버전에서는 전체 파일을 복호화하는 것이 아니라 압축된 일부 포인트 클라우드만 압축을 풀 수 있는 기술이 추가될 예정이고, 기존에 있는 기술의 성능 개선을 계획하고 있다^[20]. 이것과 같이 포인트 클라우드의 압축 기술이 활발하게 연구가 되고 있고, 앞으로 더 많은 환경에서 이 기술이 사용될 것으로 보인다.

III. GPU 가속 vSLAM 및 GPCC 기법

본 논문은 저전력 시스템에서 효율적으로 vSLAM을 진행하기 위해 CPU와 GPU를 사용한 SSL SLAM 기법과 멀티 CPU 스레드를 활용한 GPCC 통합 프레임워크를 제안한다. 이 방법은 저전력 시스템에서 CPU의 한계를 넘어서

GPU까지 사용하여 원활한 실시간 vSLAM을 가능하게 하고, GPCC를 사용하여 데이터를 저장하여 저장 공간을 확보함으로써, 효율적인 3D reconstruction을 가능하게 한다. 그림 2는 제안하는 방법론에 대한 전체 파이프라인을 보여준다. 본 장은 제안한 방법의 핵심 구성 요소인 (1) 적응형 GPU 기반 vSLAM 기법, (2) 멀티 CPU 스레드 기반 GPCC 기법으로 나누어 설명한다.

1. 적응형 GPU 기반 vSLAM 기법

본 논문은 RGB-D 카메라를 이용한 vSLAM을 GPU 기반으로 최적화하여 실시간 매핑 성능을 확보하였다. 임베디드 환경에서는 연산 자원이 제한적이므로, 높은 연산 복잡도를 가지는 특징 추출과 최적화 단계가 병목 현상을 유발한다. 이를 해결하기 위해 GPU 기반 병렬 연산과 시간 제약 최적화를 적용하여 연산 속도를 향상시켰다. 이를 위해 먼저 전처리 및 특징 추출을 진행한다. 센서로부터 받은 포인트 클라우드 $P = \{p_i\}$ 는 좌표 변환을 통해 센서 좌표

계에서 로봇 좌표계로 정규화된다. 이후 스캔 각도 변화량이 일정 임계값을 초과할 때 새로운 세그먼트로 구분한다. 각 세그먼트 내에서 포인트의 곡률은 인접한 10개의 점을 사용하여 다음 식 (1)에 따라 계산한다.

$$c_j = \left(\sum_{k=-5, k \neq 0}^5 p_{j+k} - 10p_j \right)^2 \quad (1)$$

이 식을 이용하여 포인트 클라우드의 각 점 p_j 에 대해 주변 점들과의 위치 차이를 기반으로 곡률을 계산한 값이 크면 해당 점이 주변과 급격한 변화를 가지는 **edge** 특성을 갖는 점으로 분류된다. 또한 인접 포인트들의 방위각 차이를 이용하여 각도 변화량을 식 (2)와 같이 정의한다.

$$\Delta\theta_j = \left| \arctan\left(\frac{y_{j-5}}{z_{j-5}}\right) - \arctan\left(\frac{y_{j+5}}{z_{j+5}}\right) \right| \quad (2)$$

여기서 y_{j-5}, z_{j-5} 는 현재 포인트보다 앞선 위치의 좌표이고, y_{j+5}, z_{j+5} 는 뒤쪽 포인트의 좌표이다. 이 값이 크면 곡률과 마찬가지로 급격한 방향 변화를 가지는 포인트로 간주하여, **edge** 후보가 된다. 곡률 c_j 값이 큰 경우 해당 점은 **edge** 특징점으로 분류되고, 곡률이 작은 경우 **plane** 특징점으로 분류된다. 이 과정은 CUDA 커널을 통해 GPU 상에서 병렬적으로 수행되며, GPU 메모리는 사전 할당 방식으로 관리하여 반복적인 메모리 복사와 할당 오버헤드를 줄였다. 추출된 **edge** 및 **plane** 특징점들은 오도메트리를 추정 및 연산하여 로컬 지도의 특징점들과 매칭한다. 모서리 특징점은 직선과의 최소 거리 기반 잔차 값으로 정의되며, 최근접 두 점 a, b 를 통해 계산한다. **Plane** 특징점은 이웃 점들로 추정한 법선 벡터 n 과 **plane** 파라미터 d 를 이용하여 식 (3)과 같이 잔차 값을 정의한다.

$$r_{edge}(p) = \frac{|(Rp + t - a) \times (Rp + t - b)|}{|a - b| \sin \theta}, \quad (3)$$

$$r_{surf}(p) = n^T(Rp + t) + d$$

이때 R 과 t 는 추정된 회전 행렬과 변환 벡터, a 와 b 는

주변 **edge**의 특징점, n 과 d 는 각각 평면의 법선 벡터와 상수항이다. r_{edge} 는 점이 두 특징점을 잇는 직선에서 떨어진 거리를 나타내며, r_{surf} 는 점이 특정 **plane**에서 벗어난 정도를 의미한다. 이 두 잔차 값은 비선형 최적화 과정에서 동시에 최소화되어 로봇의 위치 추정 및 맵핑의 정확도를 높인다. 이러한 잔차 값을 최소화하는 기법은 **Ceres Solver**를 이용해 수행되며, 저사양 환경을 고려하여 반복 횟수를 2회로 제한하고 연산 시간을 0.01초로 상한을 두는 시간 제약 최적화를 적용하였다. 추정된 프레임별 포즈는 전역 최적화를 위해 **GTSAM** 기반의 포즈 그래프에 추가된다. 새로운 노드는 일정 이동 거리 $\Delta d > \tau_d$, 회전 변화량 $\Delta\theta > \tau_\theta$, 혹은 누적 프레임 수가 일정 값 이상일 때만 새로운 노드가 추가된다. 루프 클로저가 감지되면 기하적 일관성 검증을 거쳐 루프 팩터가 그래프에 추가되고, **Levenberg - Marquardt** 최적화로 전역 지도를 정합하여 **PLY** 파일 형태로 맵핑된 지도를 저장한다.

2. 멀티 CPU 스레드 기반 GPCC 기법

GPCC는 입력된 포인트 클라우드를 양자화하고 이를 기반으로 공간적 구조와 속성을 부호화하여 압축 효율을 극대화한다. 본 논문은 기하학 정보 부호화 방식으로는 **Octree**를, 속성 부호화 방식은 **RAHT** 기법을 선택하였다. **Octree**는 **LiDAR**와 같이 희소하고 불균일한 점 분포에서도 공간을 계층적으로 분할하여 빈 영역을 효율적으로 처리할 수 있으며, **RAHT**는 점들의 속성을 가중합과 차 연산을 통해 변환하여, 불균일한 샘플링 환경에서도 안정적인 압축 성능을 보장한다. 이러한 특성은 다른 방법들에 비해 임베디드 환경에 유리하므로, 저전력 시스템에서의 압축을 위해 멀티 스레드 기반 GPCC 알고리즘을 적용하였다. 먼저 전처리 및 양자화를 위해 입력된 포인트 클라우드 $p_i = (x_i, y_i, z_i)$ 는 정수 격자에 맞게 양자화된다. 이 과정은 스케일 s 를 적용하여 좌표를 정수 단위로 변환하고 오프셋 o 를 감하여 원점을 교정하며, 지정된 클램핑 범위인 c_{min}, c_{max} 를 벗어나면 잘라내는 방식으로 식 (4)를 통해 수행한다.

$$\tilde{p}_i = clip(|sp_i| - o, c_{min}, c_{max}) \quad (4)$$

양자화된 점들은 이후 Octree 구조에 삽입되어 기하학 정보가 부호화된다. Octree는 공간을 8분할하여 계층적으로 표현하는 자료 구조이며, 각 노드의 리프에는 포함된 점의 개수가 기록된다. 리프 노드의 점 개수는 식 (5)와 같이 정의된다.

$$n_{leaf} = \sum_{i=1}^N 1\{\tilde{p}_i \in node\} \quad (5)$$

여기서 N 은 전체 포인트의 개수, $1\{\bullet\}$ 은 조건이 참이면 1을 반환하는 지기 함수이다. 이 식을 활용하여 Octree 기반 기하학 정보 부호화에서 점유 여부를 효율적으로 표현한다. 속성 정보는 RAHT를 통해 부호화된다. 자식 노드의 속성값이 x_a, x_b , 각 노드의 가중치가 w_a, w_b 일 때 평균과 차이값은 식 (6)과 같이 계산된다.

$$y_{sum} = \frac{w_a x_a + w_b x_b}{w_a + w_b}, \quad y_{diff} = x_a - x_b \quad (6)$$

이 식의 y_{sum} 은 저주파 성분을 나타내어 전체적인 경향을 보존하며, y_{diff} 는 고주파 성분을 나타내어 지역적인 변화를 반영한다. 이 성분들은 이후 양자화와 산술 부호화를 통해 압축된다. 이러한 기법은 포인트 클라우드를 분할하여 연산하지 않으면 저전력 시스템에서는 사용하기 어렵다. 그래서 대규모 포인트 클라우드를 효율적으로 부호화하기 위해, 본 논문은 슬라이스 단위 분할 및 비트스트림 생성 과정을 도입하였다. 이 과정은 포인트 클라우드의 크기를 제어하면서도 임베디드 환경에서 안정적인 압축을 가능하게 한다. 먼저 전체 포인트 집합 P 를 슬라이스 개수 M 에 따라 분할 한다. 각 슬라이스 S_m 의 최대 크기는 N_{max} 로 제한되며 식 (7)과 같이 나타낼 수 있다.

$$P = \bigcup_{m=1}^M S_m, \quad |S_m| \leq N_{max} \quad (7)$$

이때 슬라이스 분할은 포인트의 공간적 분포에 따라 등점수 또는 등구간 방식으로 수행할 수 있으며, 한 슬라이스가 차지하는 포인트 수를 제어함으로써 메모리 사용량과 지연 시간을 관리할 수 있다. 각 슬라이스는 독립적으로

Octree 기반 기하학 정보 부호화와 RAHT 기반 속성 정보 부호화를 수행한 뒤, 헤더 정보를 포함한 페이로드로 식 (8)과 같이 구성된다.

$$SlicePayload_m = [Header_m \parallel G_m \parallel A_m] \quad (8)$$

여기서 $Header_m$ 은 슬라이스의 메타데이터, G_m 은 기하학 정보, A_m 은 속성 정보를 의미한다. 최종적으로 모든 슬라이스 페이로드는 순차적으로 결합하고, 식 (9)를 사용하여 하나의 비트스트림 B 로 형성된다.

$$B = \parallel_{m=1}^M SlicePayload_m \quad (9)$$

이 접근법은 한 번에 대규모 데이터를 처리하는 기존 방식보다 메모리 효율성과 실시간성을 크게 향상시킨다. 특히 임베디드 장치에서는 작은 단위의 슬라이스만 순차적으로 메모리에 로드하면 되므로, 과부하 없이 안정적으로 부호화할 수 있다. 또한 슬라이스 크기 N_{max} 와 개수 M 은 플랫폼 성능에 따라 조정 가능한 파라미터로, 압축 효율과 처리 지연 간의 균형을 맞추는 핵심 요소로 작용한다.

IV. 실험 환경 및 실험 결과

1. 실험 환경 및 구현 세부 사항

본 장은 최적화된 vSLAM 방법을 그림 3과 같은 환경 세팅을 사용하여 건물의 복도를 매핑하고, 이를 활용하여 제안한 GPCC를 사용하여 PLY 인코딩을 진행한다. 실험에서는 Nvidia 사의 Jetson Nano 4gb를 사용했으며, 18650 배터리가 장착된 무인지상차량 (UGV)을 사용하였다. 카메라는 Intel 사의 RealSense L515 카메라를 사용하였다. 그림 3은 이 두 장치의 기능에 대한 설명을 적었다^[21,22]. Jetson Nano의 메모리 용량의 경우 4gb가 장착되어 있지만, swap 기법을 활용하여 총 메모리를 8gb까지 사용하도록 설정하였다. 본 실험은 저전력 임베디드 시스템인 Jetson Nano의 GPU와 CPU를 모두 사용하여 실시간

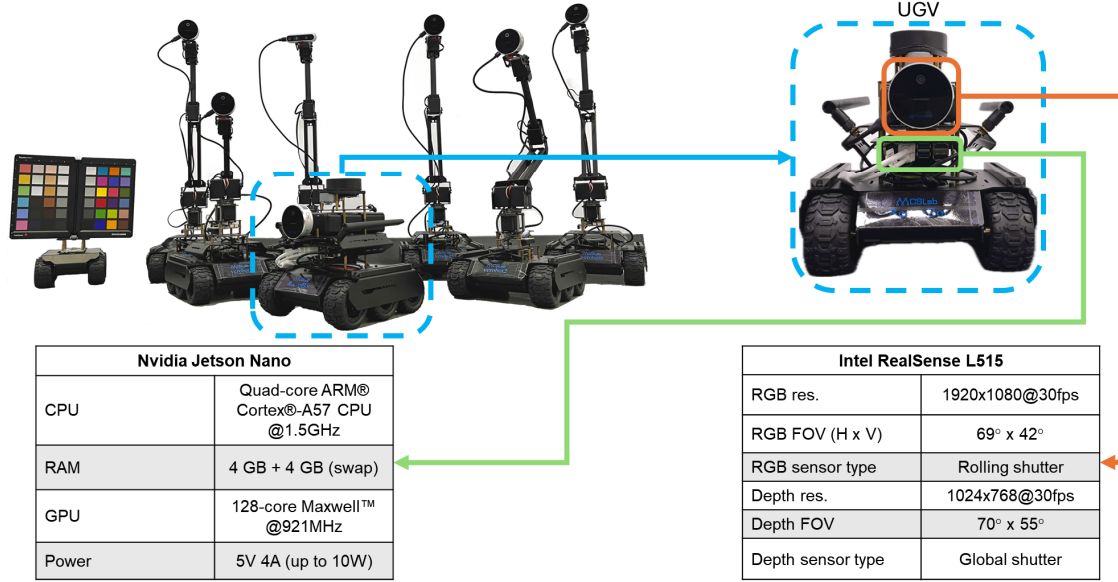


그림 3. 다양한 로봇들 및 본 실험에 활용한 로봇. 실험 환경은 Intel RealSense L515 센서와 NVIDIA Jetson Nano로 구성되며, 실시간 매핑 성능 평가에 사용되었다.

Fig. 3. Various robots including the robot used int this experiment. The setup consists of an Intel RealSense L515 sensor and an NVIDIA Jetson Nano, used to evaluate real-time mapping performance.

vSLAM으로 매핑을 진행하고, 멀티 CPU 스레드 GPCC 기법을 활용하여 효율적으로 용량을 확보하였다. 매핑 환경은 한 건물의 복도를 목표로 하고 진행하였다. RealSense L515 카메라의 깊이 정보 스트림은 320*240, RGB 스트림은 640*480 화질에 모두 30fps로 진행하였다. 또한 CPU 스레드는 Jetson Nano의 모든 스레드인 4개를 사용하도록 설정하였다.

2. 평가 방법

제안한 최적화된 vSLAM과 GPCC 성능은 다음과 같이 평가하였다. 먼저, vSLAM 성능은 실시간성을 확인하기 위해 Baseline인 SSL SLAM 기법과 제안한 방식의 오도메트리 추정 속도를 측정하였다. 오도메트리 계산은 현재 프레임과 다음 프레임 간의 특징점 추출 결과를 기반으로 하며, 평균 처리 시간이 곧 실시간 가능 여부를 결정한다. 이를 위해 직선 매핑과 회전 매핑을 각각 100프레임 단위별 평균 오도메트리 처리 시간을 측정하고, 전체 1,000프레임의 결

과를 통해 시스템의 처리율인 초당 프레임 수 (fps)를 산출하였다. 또한 edge 및 surface 맵의 포인트 수 증가를 기록하여 최적화 과정이 매핑 화질 저하를 유발하지 않는지를 검증하였다.

GPCC 성능 측정은 압축 효율성과 복원 화질을 함께 고려하여 평가를 진행하였다. VSLAM으로 생성된 PLY 파일을 입력으로 사용하여 QP 값의 변화에 따른 파일 크기와 비트레이트를 측정하고, 식 (10)을 이용해 R, G, B 채널별 최대 신호 대 잡음비 (PSNR)를 계산하였다.

$$PSNR = 10 \cdot \log_{10} \left(\frac{L^2}{MSE} \right) \quad (10)$$

여기서 L 은 포인트 클라우드 좌표계의 최대 범위를 의미하며, MSE 는 원본 포인트 p_i 와 복원된 포인트 \hat{p}_i 간의 평균 제곱 오차이다. 이를 바탕으로 부호율-변형 (RD) 곡선을 작성하여 압축 강도에 따른 화질 변화를 분석하였다. 또한, 압축 강도가 인코딩 복잡도(시간)에 미치는 영향을 분

석하기 위해 두 가지 시나리오로 나누어 실험을 진행하였다. 첫째는 기하학 정보는 무손실 (lossless)로, 속성 정보만 손실 (lossy) 압축하는 경우이며, 둘째는 속성 정보는 무손실로, 기하학 정보만 손실 압축하는 경우이다. 각 시나리오 별로 QP 값을 변경하며 인코딩 시간을 측정하여 복잡도를 비교 분석하였다.

VSLAM의 평가는 제안한 GPU 가속 방식이 최종 매핑 화질에 미치는 영향을 정량적으로 평가하기 위해, Baseline 방식으로 생성된 맵과 제안한 방식으로 생성된 맵 간의 Iterative Closest Point (ICP) 정렬을 수행하였다. 맵의 정합 화질은 Inlier Root Mean Square Error (Inlier RMSE)와 Fitness 총 두 가지 지표로 평가하였다. Inlier RMSE는 식 (11)과 같이 두 맵 간의 inlier 포인트들의 평균 제곱근 오차를 나타내며, 0의 값에 가까울수록 정밀하게 정합되었음을 의미한다.

$$RMSE_{inlier} = \sqrt{\frac{1}{N_{inlier}} \sum_{i=1}^{N_{inlier}} \|p_i^{(r)} - (Rp_i^{(s)} + t)\|^2} \quad (11)$$

여기서 $p_i^{(r)}$ 은 기준 포인트 클라우드의 i 번째 점, $p_i^{(s)}$ 는 제안한 방식으로 생성된 포인트 클라우드의 대응점, R 과 t 는 ICP에서 추정된 회전 행렬과 변환 벡터, N_{inlier} 는 대응 임계값 안에 포함된 점의 개수다. Fitness는 식 (12)와

같이 baseline 맵의 전체 포인트 중 제안한 기법으로 형성된 맵과 일정 대응 거리인 τ 이내로 정합된 포인트의 비율을 나타낸다. 이때 대응 거리의 값이 1의 값에 가까울수록 두 맵의 일치도가 높음을 의미한다.

$$Fitness = \frac{N_{inlier}}{N_{ref}} = \frac{|\{i \mid \|p_i^{(r)} - (Rp_i^{(s)} + t)\| < \tau\}|}{N_{ref}} \quad (12)$$

멀티 스레드를 이용한 GPCC 기법의 효율성은 JPEG Pleno 데이터베이스 중 8iVFB v2 데이터셋인 “soldier_vox10”, “longdress_vox10”, “loot_vox10”, “redandblack_vox10”을 추가로 활용하여 Baseline 대비 제안 기법의 인코딩 시간을 비교하였다^[23]. 각 데이터셋의 단일 PLY 파일을 사용하여 실험을 수행하였으며 압축률 및 인코딩 시간을 분석하였다.

3. 실험 결과

직선 주행 시퀀스인 그림 4 (a)의 Baseline은 100프레임 지점 135.2ms에서 1,000프레임 지점 187.9ms까지 맵 누적에 따라 처리 시간이 점진적으로 증가하는 경향을 보였다. 반면, 제안한 방식은 동일 구간에서 87.2ms에서 96.0ms 사이의 일정한 처리 시간을 유지하였다. 연산 부하가 상대적으로 높은 그림 4 (b)의 회전 주행 시퀀스는 Baseline의 처

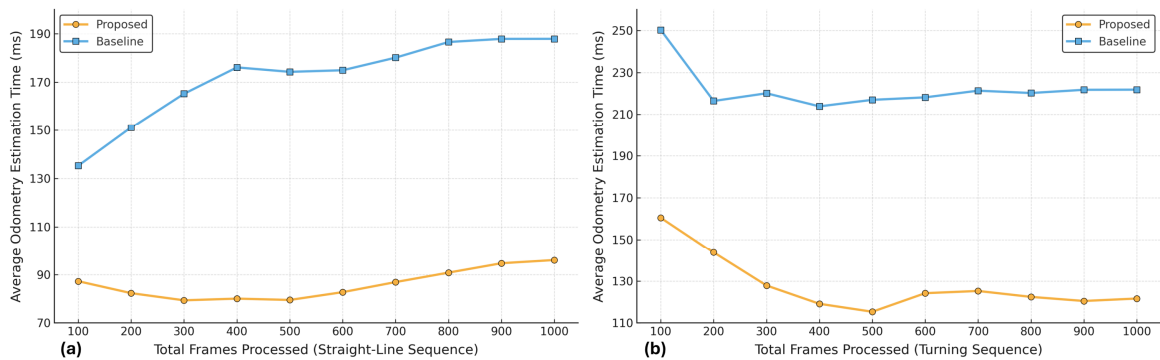


그림 4. 매핑 시퀀스별 평균 오도메트리 추정 시간 비교. (a) 직선 주행 시퀀스. Baseline은 187ms까지 증가하나, 제안 방식은 96ms 이하로 유지한다. (b) 회전 주행 시퀀스. Baseline은 평균 222.14ms로 측정되었으나, 제안한 방식은 평균 128.05ms 실시간 처리를 보인다.

Fig. 4. Comparison of average odometry estimation time by mapping Sequence. (a) Straight-line sequence. The Baseline's performance increases up to 187ms, while the Proposed method remains under 96ms. (b) Turning sequence. The Baseline averaged 222.14ms, while the Proposed method averaged 128.05ms.

리 시간이 초기 100프레임에서 250.3ms를 기록한 후 1,000 프레임에서 221.9ms로 수렴하였다. 제안한 방식은 초기 160.5ms에서 1,000프레임 121.6ms로 전 구간 Baseline 대비 낮은 처리 시간을 유지하였다. 이는 1,000프레임 처리 시점을 기준으로 Baseline 대비 직선 주행에서 48.9%, 회전 주행에서 45.2%의 시간을 단축한 결과이다. 이로써 제안 방식이 맵 크기나 주행 상황 변화와 관계없이 안정적인 실시간 오도메트리 연산이 가능함을 확인하였다.

제안된 GPU를 사용한 SLAM 기법이 매핑 화질에 미치는 영향을 평가하기 위해 정량적 및 시각적 비교를 수행하였다. 먼저, 정량적 평가를 위해 Baseline 방식으로 생성된 맵과 제안 방식으로 생성된 맵 간의 ICP 정렬을 수행하였다. 측정 결과, Inlier RMSE는 0.0237m, Fitness (Inlier

Ratio)는 0.8938로, 두 맵이 기하학적으로 차이점이 적다는 것을 확인하였다. 그림 5는 이러한 정량적 결과를 시각적으로 보여준다. 원본 장면 (Original scene), Baseline 맵, 제안한 방식의 맵을 비교하였을 때, 초록색 화살표와 같은 세부 텍스처가 제안한 방식에서도 Baseline과 동등한 수준으로 보존되었음을 알 수 있다. 따라서, 제안한 방식은 매핑 화질의 저하를 최소화하고, 오도메트리 연산 속도를 단축시켰음을 입증하였다.

앞선 결과는 제안 방식의 속도 향상과 화질 보존을 확인하였으며, 이에 더해 저전력 시스템 적합성을 검증하기 위해 vSLAM과 GPCC 인코딩 수행 중 Jetson Nano의 평균 전력 소모량 (W)을 측정하였으며, 그 결과는 그림 6과 같다. 그림 6 (a)는 vSLAM 매핑 시의 전력 소모를 보여준다.

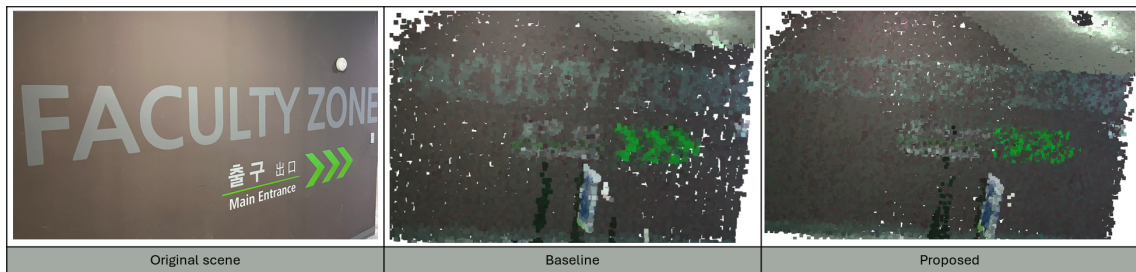


그림 5. 매핑 결과의 시각적 화질 비교. Baseline (SSL SLAM) 맵과 제안 방식 맵의 텍스처 디테일을 비교하여, 화질 저하가 미미함을 보여준다.
Fig. 5. Visual quality comparison of mapped results. Compares texture details of the Baseline (SSL SLAM) and Proposed maps, demonstrating minimal quality degradation.

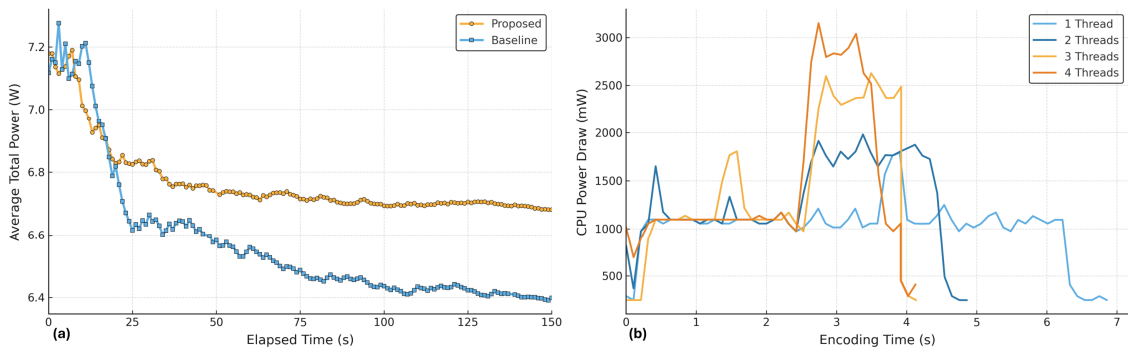


그림 6. VSLAM 및 GPCC 인코딩 평균 전력 소모량 비교. (a) vSLAM 매핑 시 전력 소모. 제안 방식은 GPU 가속으로 Baseline 대비 순간 전력이 0.3W 상승한다. (b) vSLAM으로 생성된 맵의 GPCC 인코딩 시 전력 소모. 제안 방식은 Baseline 대비 순간 전력은 높지만, 인코딩 시간을 39.7% 단축시켜 총 전력 소모 (Joule) 관점에서 더 효율적이다.

Fig. 6. Comparison of average power consumption during vSLAM and GPCC Encoding. (a) Power consumption during vSLAM mapping. The Proposed method's GPU acceleration results in a 0.3W higher instantaneous power draw over Baseline. (b) Power consumption during GPCC encoding of the vSLAM-generated map. The Proposed method has higher instantaneous power than Baseline but is more power-efficient by reducing encoding time by 39.7%.

Baseline은 평균 6.4W 수준을 유지한 반면, 제안한 방식은 특징 추출 단계의 GPU 가속을 위해 평균 6.7W 수준으로 Baseline 대비 약 0.3W 높은 전력을 소모한다.

그림 6 (b)는 vSLAM으로 생성한 실제 맵 데이터를 GPCC로 인코딩할 때 스레드 개수에 따른 전력 소모를 비교한다. Baseline인 1 스레드를 사용하는 기존 GPCC는 인코딩(6.17초) 동안 평균 6.4W 수준을 유지하나, 제안한 방식은 인코딩(3.72초) 동안 병렬 처리를 위해 Jetson Nano에 있는 모든 CPU 코어를 활용하여 순간 전력 소모 (W)가 평균 7.1W 수준으로 0.7W 더 높게 측정되었다.

그러나 제안한 방식은 표 1에서 GPCC 인코딩 시간을 평균 43% 단축시켰을 뿐만 아니라, 그림 6 (b)의 vSLAM을 사용하여 생성한 맵 인코딩 시간 또한 6.17초 (Baseline)에

서 3.72초 (Proposed)로 약 39.7% 단축시켰다. 따라서 순간 전력 소모는 높지만, 작업 완료에 필요한 총 전력 소모 (Joule = watt * second) 관점에서는 제안 방식이 더 효율적이며, 이는 배터리로 구동되는 저전력 임베디드 환경에 적합함을 시사한다.

표 1은 JPEG Pleno 8iVFB v2 데이터셋을 대상으로 Baseline 대비 제안 기법의 인코딩 시간을 비교한 것이다. 기하학 정보 및 속성 정보는 모두 QP 값을 0으로 두고 실험을 진행하였다. 실험 결과, 모든 시퀀스에서 인코딩 시간이 39.24%에서 47.65%까지 단축되었으며, 평균적으로 약 43%의 성능 향상이 달성되었다. 이러한 결과는 제안된 병렬화 및 최적화 기법이 압축 효율을 유지하면서도 인코딩 지연을 크게 줄일 수 있음을 시사한다.

표 1. JPEG Pleno 8iVFB v2 데이터셋^[23]의 인코딩 시간 비교. 제안한 방식은 Baseline과 동일한 결과를 보여주며 인코딩 시간을 평균 약 43% 단축했다.

Table 1. Encoding time comparison on the JPEG Pleno 8iVFB v2 dataset^[23]. The Proposed method reduces encoding time by an average of 43% compared to Baseline while maintaining the same result.

Dataset	Encoding time (Baseline)(s)	Encoding time (Proposed)(s)(↓)
soldier_vox10_0767.ply (1,103,974 points)	6.402	3.352 (47.65%)
longdress_vox10_1137.ply (916,250 points)	5.182	3.149 (39.24%)
loot_vox10_1289.ply (835,458 points)	4.596	2.641 (42.55%)
redandblack_vox10_1609.ply (816,879 points)	4.639	2.658 (42.7%)

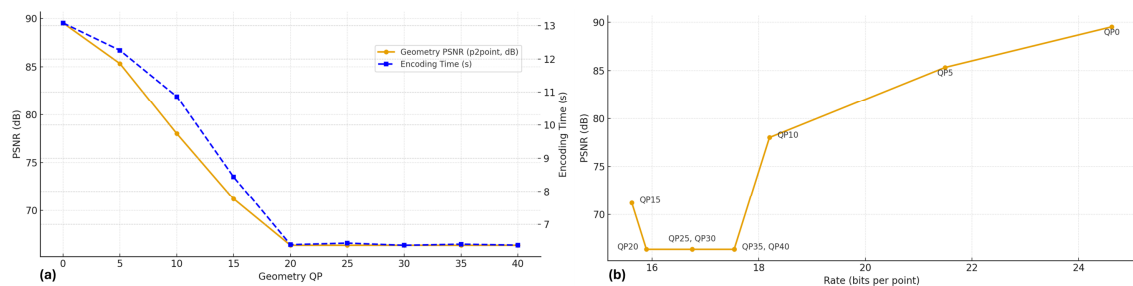


그림 7. 기하학 정보 압축 성능 및 복잡도. (a) QP값 별 PSNR 및 인코딩 시간. QP 값 0은 QP 값 20 대비 인코딩 시간이 13.08초로 2배 이상 증가한다. (b) RD 성능. QP 15 이후는 Octree 오버헤드 비트 증가로 비단조적 (non-monotonic) 형태를 보인다.

Fig. 7. Geometry compression performance and complexity. (a) PSNR and Encoding time by QP. Encoding time more than doubles from 6.4s (QP 20) to 13.08s (QP 0). (b) RD performance. The graph shows a non-monotonic trend after QP 15 due to increased Octree overhead bits.

그림 7은 vSLAM과 GPCC의 전력 소모 특성에 이어, 압축 효율과 연산 복잡도에 영향을 미치는 QP 값에 따른 성능을 분석한 결과이며, 특히 그림 7 (a)는 기하학 정보의 QP 값에 따른 $PSNR$ 값과 인코딩 시간 (복잡도)을 보여준다. QP 값은 낮아질수록 고화질 결과를 생성하여 인코딩 시간이 급격히 증가하는 경향을 보인다. 구체적으로, 인코딩 시간은 QP 값 20에서 6.40초였으나, QP 0 (무손실)에서는 13.08초로 2배 이상 증가하였다. 반면, QP 값 20 지점부터는 QP 값을 더 높여도 $PSNR$ 값이 66.42dB로, 인코딩 시간이 약 6.4초로 더 이상 변하지 않는 포화 (saturation) 구간이 관찰되었다. 이 현상은 기하학적 정보의 양자화가 최대치에 도달하여, 세부적인 디테일이 모두 손실되었기 때문으로 분석된다.

그림 7 (b)는 기하학 정보의 RD 성능을 나타낸다. 특히 포화 구간인 QP 값 15 이상에서는 일반적인 RD 곡선과 다른 현상이 관찰된다. $PSNR$ 값이 66.42dB로 정체됨에도 불구하고, 비트레이트 (bpp)는 15.89bpp (QP 값 20)에서 17.54bpp (QP 값 35)까지 오히려 소폭 증가하는 비단조적 (non-monotonic) 형태를 보였다. 이는 과도한 양자화가 점들의 공간적 분포를 비효율적으로 변경하여, Octree 트리 구조 자체를 부호화하는 데 드는 오버헤드 비트가 증가하기 때문에 발생하는 GPCC의 특성이다. 반면, QP 값 10과 QP 0 (무손실) 구간은 78.01dB에서 89.53dB로

비트레이트와 $PSNR$ 값이 함께 향상되는 RD 성능을 보였다.

그림 8은 속성 정보의 압축 성능과 복잡도를 분석한 결과로, 그림 8 (a)의 경우 QP 값에 따른 파일 크기, 평균 $PSNR$, 인코딩 시간의 관계를 보여준다. 파일 크기는 QP 값이 0 (무손실)일 때 3.20MB에서 QP 값이 40일 때 2.01MB까지 감소하였으며, 평균 $PSNR$ 값도 56.47dB에서 20.37dB로 함께 감소하였다. 특히 인코딩 시간 (복잡도)의 경우, 속성 압축의 인코딩 시간은 QP 값과 관계없이 14.3초에서 14.8초 사이로 차이가 미미한 일정한 수준을 유지하였다. 이는 그림 7 (a)에서 기하학 QP 값에 따라 인코딩 시간이 2배 이상 증가했던 것과는 대비되는 결과로, RAHT 기반의 속성 압축이 Octree 기반의 기하학 압축보다 QP 변화에 따른 연산 복잡도 영향이 적다는 것을 의미한다.

그림 8 (b)는 R, G, B 채널별 RD 성능을 보여준다. 세 채널의 $PSNR$ 값은 동일한 경향을 보이며, QP 값이 낮아질수록 $PSNR$ 값과 비트레이트가 함께 증가하는 전형적인 RD 커브를 나타낸다. 평균 $PSNR$ 값은 QP 값 0 (무손실)에서 56.47dB, QP 값 10에서 38.40dB, QP 20에서 33.11dB로 점차 감소한다. 특히 QP 20을 초과하는 높은 QP 값은 비트레이트 절감 효과가 미미한 반면, $PSNR$ 값이 20.37dB (QP 40)까지 저하되는 비효율적인 압축을 보였다. 이 수치는 이어지는 그림 9에서 관찰되는 시각적 화질

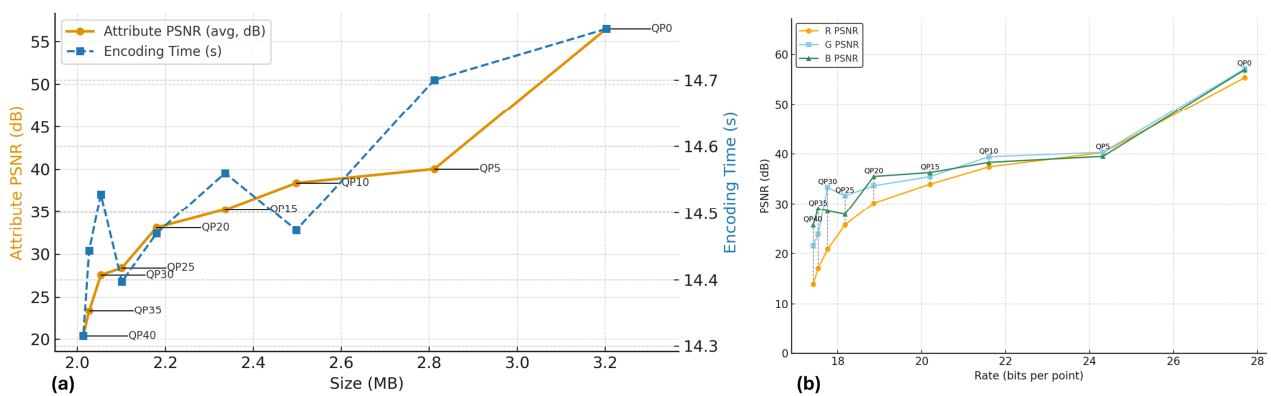


그림 8. 속성 정보 압축 성능 및 복잡도. (a) QP 값별 성능. QP 0에서 QP 40으로 화질 및 크기가 감소하고, 인코딩 시간은 약 14.55초로 일정하였다. (b) R,G,B 채널별 RD 성능. 전형적인 RD 경향을 보이는 것을 알 수 있다.

Fig. 8. Attribute compression performance and complexity. (a) Performance by QP. Quality and size decrease from QP 0 to QP 40 and encoding time is constant at about 14.55s. (b) R,G,B Channel RD Performance. Shows a typical RD curve trend.

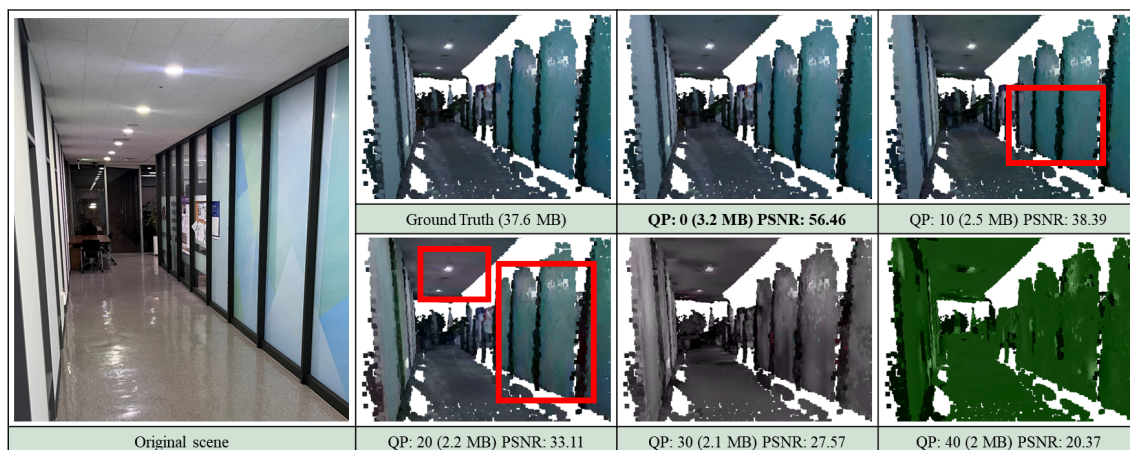


그림 9. 속성 인코딩 QP 값에 따른 압축 화질 비교. QP 값이 증가할수록 용량은 작아지지만, 세밀한 색감 및 구조에서 화질 저하가 두드러진다.
Fig. 9. Visual comparison of reconstructed bitstream under different QP values for attribute encoding. As QP increases, file size decreases, but quality degradation appears in fine colors and structures.

저하와 일치하는 결과이다.

그림 9는 서로 다른 QP 값에서 복원된 장면의 시각적 화질을 비교한 것이다. Ground Truth (37.6 MB) 대비 QP 값을 0, 10, 20, 30, 40에서 압축을 수행하였으며, 각 부호화된 파일 크기와 PSNR 값을 함께 표기하였다. 낮은 QP 값에서는 원본과 유사한 시각적 화질을 유지하였으나, QP 값이 증가할수록 디테일이 손실되며 화질 저하가 발생하였다. 특히 빨간색 사각형으로 표시된 영역에서 세부 구조의 왜곡이 뚜렷하게 관찰되었으며, 이는 압축 강도가 높을수록 시각적 인지 화질에 큰 영향을 준다는 점을 보여준다.

V. 결 론

본 논문은 저전력 임베디드 환경에서 GPU를 활용한 vSLAM과 멀티 CPU 스레드를 적용한 MPEG GPCC 기반 포인트 클라우드 압축을 통합한 프레임워크를 제안한다.

실험 결과, vSLAM의 오도메트리 추정 시간은 GPU 가속을 통해 Baseline 대비 약 45%에서 49% 단축되었으며, 특히 부하가 큰 회전 구간에서도 160ms 이하의 안정적인 실시간 성능을 보장하였다. 이러한 속도 향상은 매핑 화질 저하 없이 달성되었음을 ICP 정량 평가(RMSE 0.0237m,

Fitness 0.8938)를 통해 입증하였다. 또한, GPCC 인코딩 과정에 CPU 병렬화를 적용하여 표준 데이터셋에서 평균 43%의 인코딩 시간 절감을 달성하였다. QP의 복잡도 분석 결과, 기하학 정보 압축은 고화질일수록 연산 복잡도가 급증하는 반면, 속성 정보 압축은 복잡도 변화가 미미함을 확인하여 효율적인 파라미터 설정의 가이드를 제시하였다. 전력 분석에서는 제안한 방식이 순간 전력 소모는 다소 높으나, 작업 시간 단축을 통해 총 전력 효율성 측면에서 저전력 시스템에 적합함을 확인하였다.

본 연구는 제한된 자원의 임베디드 시스템에서도 고화질의 실시간 3D 매핑과 효율적인 표준 압축이 동시에 가능함을 입증하였다. 이는 모바일 로봇, 디지털 트윈, XR 등 대규모 3D 데이터의 실시간 처리가 요구되는 다양한 응용 분야에서 핵심적인 기반 기술로 활용될 것으로 기대된다. 향후 연구로는 제안한 프레임워크를 다중 로봇 협업 환경과 엣지-클라우드 하이브리드 구조로 확장하여, 실시간 데이터 전송 및 분산 압축을 통한 대규모 공간 매핑의 효율성을 높이는 방향으로 발전시킬 예정이다. 또한 GPCC가 아닌 다양한 포인트 클라우드 압축 기법을 탐구하여 압축률과 복원 화질 간의 균형을 최적화하고, 포인트 클라우드 기반 3D 가우시안 스플래팅과의 연계를 통해 몰입형 미디어 및 실감형 디지털 트윈 응용으로의 확장 가능성을 검토할 계획이다.

참 고 문 헌 (References)

- [1] R. Minerva, G. M. Lee, and N. Crespi, "Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models", *Proceedings of the IEEE*, Vol.108, No.10, pp.1785-1824, Oct. 2020.
doi: <https://doi.org/10.1109/JPROC.2020.2998530>
- [2] X. Sheng, S. Mao, Y. Yan, and X. Yang, "Review on SLAM Algorithms for Augmented Reality", *Displays*, Vol.84, 102806, 2024.
doi: <https://doi.org/10.1016/j.displa.2024.102806>
- [3] H. Taheri and Z. C. Xia, "SLAM: Definition and Evolution", *Engineering Applications of Artificial Intelligence*, Vol.97, 104032, 2021.
doi: <https://doi.org/10.1016/j.engappai.2020.104032>
- [4] I. Yang, Y.-G. Kim, M. Park, J.-B. Jeong, and E.-S. Ryu, "3D Gaussian Splatting-based Static Scene Volumetric Video Capturing System Using Remote Controlled Movable Robots", *Proceedings of the 37th Workshop on Image Processing and Image Understanding (IPIU)*, Feb. 2025.
- [5] M. Huang, Q. Xu, and W. Xu, "Point Cloud Geometry Compression Based on the Combination of Interlayer Residual and IRN Concatenated Residual", *Graphical Models*, Vol.140, 101279, 2025.
doi: <https://doi.org/10.1016/j.gmod.2025.101279>
- [6] L. A. Curiel-Ramirez, R. A. Ramirez-Mendoza, and J. Izquierdo-Reyes, "Hardware in the Loop Framework Proposal for a Semi-Autonomous Car Architecture in a Closed Route Environment", *International Journal of Interactive Design and Manufacturing*, Vol.13, pp.1647 - 1658, 2019.
doi: <https://doi.org/10.1007/s12008-019-00619-x>
- [7] Y. Ryu, I. Yang, S.-B. Jo, M. Park, and E.-S. Ryu, "Challenges and Solutions for MPEG Immersive Video Acquisition in Dynamic Unstructured Outdoor Environment", *Proceedings of the Korean Institute of Broadcast and Media Engineers (KIBME) Summer Conference*, pp.369 - 372, Jun. 2024. <https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11849106>
- [8] L. Wijayathunga, A. Rassau, and D. Chai, "Challenges and Solutions for Autonomous Ground Robot Scene Understanding and Navigation in Unstructured Outdoor Environments: A Review", *Applied Sciences*, Vol.13, No.9877, 2023.
doi: <https://doi.org/10.3390/app13179877>
- [9] X. Li, E.-Y. Chang, J. Cha, and E. S. Jang, "Efficient Inter-Prediction Method Using Reference Frame Accumulation for MPEG GPCC", *IEEE Access*, Vol.13, pp.96135-96146, 2025.
doi: <https://doi.org/10.1109/ACCESS.2025.3575099>
- [10] H. Wang, C. Wang, and L. Xie, "Lightweight 3-D Localization and Mapping for Solid-State LiDAR", *IEEE Robotics and Automation Letters*, Vol.6, No.2, pp.1801 - 1807, Apr. 2021.
doi: <https://doi.org/10.1109/LRA.2021.3060392>
- [11] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time", *Proceedings of Robotics: Science and Systems Conference (RSS)*, pp.109 - 111, 2014.
doi: <https://doi.org/10.15607/rss.2014.x.007>
- [12] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, pp.4758 - 4765, 2018.
doi: <https://doi.org/10.1109/IROS.2018.8594299>
- [13] ISO/IEC JTC 1/SC 29/WG 7, "GPCC Codec Description", *ISO/IEC Standard Document N0057*, Jan. 2021.
- [14] R. Schnabel and R. Klein, "Octree-based Point-cloud Compression", *Proceedings of the 3rd Eurographics/IEEE VGTC Conference on Point-Based Graphics (SPBG'06)*, Goslar, Germany, pp.111 - 121, 2006.
doi: <https://dl.acm.org/doi/10.5555/2386388.2386404>
- [15] D. Flynn et al., "Predictive Geometry Coding", *ISO/IEC JTC1/SC29 WG7 3DG Document m51012*, Oct. 2019.
- [16] S. Lasserre, "Part 1 Improving Trisoup Summary, Results and Perspective", *ISO/IEC JTC1/SC29 WG7 3DG Document m59288*, Mar. 2022.
- [17] K. Mammou et al., "Proposal for Improved Lossy Compression in TMC1", *ISO/IEC JTC1/SC29 WG7 3DG Document m42640*, Apr. 2018.
- [18] Y. Choi, J.-B. Jeong, and E.-S. Ryu, "Efficient Conversion Method of Dynamic Point Cloud for Generating Dynamic Mesh", *Journal of Broadcast Engineering*, Vol.28, No.5, pp.633 - 645, Sep. 2023.
doi: <https://doi.org/10.5909/JBE.2023.28.5.633>
- [19] Y. Hwang, J. Kim, and K. Kim, "2D Interpolation of 3D Points using Video-based Point Cloud Compression", *Journal of Broadcast Engineering*, Vol.26, No.6, pp.692 - 703, 2021.
doi: <https://doi.org/10.5909/JBE.2021.26.6.692>
- [20] H. J. Heo, "MPEG Geometry-based Point Cloud Compression (GPCC) Standard Technology", *Broadcasting and Media Magazine*, Vol.28, No.3, pp.12-22, 2023. <https://www.kibme.org/resources/journal/2023/0807171041050.pdf>
- [21] NVIDIA Jetson Nano Product Development, <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/> (accessed Sep. 10, 2025)
- [22] Intel RealSense LiDAR Camera L515 Specifications, <https://www.intel.com/content/www/us/en/products/sku/201775/intel-realsense-lid-ar-camera-l515/specifications.html> (accessed Sep. 10, 2025)
- [23] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i Voxelized Full Bodies - A Voxelized Point Cloud Dataset", *ISO/IEC JTC 1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, Geneva, Switzerland, Jan. 2017. <https://plenodb.jpeg.org/pc/8ilabs>

저 자 소 개



안 승 혁

- 2018년 3월 ~ 2024년 2월 : 동국대학교 산업시스템공학과 공학사
- 2024년 9월 ~ 현재 : 성균관대학교 실감미디어공학과 석사과정
- ORCID : <https://orcid.org/0009-0002-4475-6742>
- 주관심분야 : 로봇틱스, 실감미디어, 포인트 클라우드 압축, SLAM



양 이 삭

- 2016년 3월 ~ 2023년 8월 : 성균관대학교 컴퓨터공학과 학사
- 2023년 9월 ~ 현재 : 성균관대학교 실감미디어공학과 석박통합과정
- ORCID : <https://orcid.org/0009-0007-3176-6972>
- 주관심분야 : 멀티미디어 통신 시스템, 실감미디어, 볼류메트릭 비디오, 로봇틱스



김 영 규

- 2024년 3월 ~ 현재 : 성균관대학교 실감미디어공학과 석사과정
- ORCID : <https://orcid.org/0009-0008-5470-3103>
- 주관심분야 : 볼류메트릭 비디오, 3D 가우시안 스피래팅, 압축, 방사형 필드, 동적 재구성



류 은 석

- 1999년 8월 : 고려대학교 컴퓨터학과 학사
- 2001년 8월 : 고려대학교 컴퓨터학과 석사
- 2008년 2월 : 고려대학교 컴퓨터학과 박사
- 2008년 3월 ~ 2008년 8월 : 고려대학교 연구교수
- 2008년 9월 ~ 2010년 12월 : 조지아공대 박사후과정
- 2011년 1월 ~ 2014년 2월 : InterDigital Labs Staff Engineer
- 2014년 3월 ~ 2015년 2월 : 삼성전자 수석연구원/파트장
- 2015년 3월 ~ 2019년 8월 : 가천대학교 컴퓨터공학과 조교수
- 2019년 9월 ~ 현재 : 성균관대학교 실감미디어공학과 학과장, 지능형 멀티미디어 연구센터 센터장
- ORCID : <https://orcid.org/0000-0003-4894-6105>
- 주관심분야 : 멀티미디어 시스템, 비디오 코딩 및 국제 표준, HMD/VR 응용분야