JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 25, No. 10, pp. 3031-3040, Oct. 2024

Check for updates

Real-3DGS: 자율주행 시뮬레이션을 위한 센서 퓨전 기반 현실 규모의 환경 모델링

서 유 리[1] · 허 채 연[2] · 김 찬 수[3*]
[1]전남대학교 인공지능학부 학사과정 [2]전남대학교 인공지능융합학과 석사과정 [3]전남대학교 지능형모빌리티융합학과 교수

Real-3DGS: Sensor Fusion-based Realistic Scale Environmental Modeling for Simulation of Autonomous Driving

Yu-Ri Seo[1] · Chae-Yeon Heo[2] · Chan-Soo Kim[3*]

[1]Bachelor's Course, Department of Artifical Intelligence, Chonnam National University, Gwangju 61186, Korea
[2]Master's course, Department of Artificial Intelligence Convergence, Chonnam National University, Gwangju 61186, Korea
[3]Professor, Department of Intelligent Mobility, Chonnam National University, Gwangju 61186, Korea

[요 약]

최근 급격한 발전을 이루고 있는 자율주행 기술은 현실에서 테스트할 수 없는 위험한 상황을 안전하게 평가하는 데 있어 어려움을 겪고 있다. 위험한 시나리오를 평가하기 위해 가상 시뮬레이션 기술이 도입되고 있지만, 현재 시뮬레이션 주행 환경은 CAD 프로그램을 통해 수작업으로 재구성하기 때문에 재현성 및 현실성이 떨어진다. 이를 해결하기 위해, 본 연구에서는 센서 퓨전을 활용한 3D Gaussian Splatting 기반 현실 스케일의 환경 모델링 기법인 Real-3DGS을 제안한다. 현실감 높지만 카메라 이미지만을 사용하여 거리 정확도가 부정확한 기본 3DGS의 단점을 해결하기 위해, 제안된 Real-3DGS는 카메라 센서의 색상 정보가 추가된 LiDAR 점군 데이터로부터 3DGS에서 가우시안의 초기치를 생성하였다. 또한, LiDAR 기반의 SLAM 에서 추정된 위치 정보를 기반으로 실제 위치에 맞게 카메라 위치 초기치를 재정렬 하였고, LiDAR에서 취득된 거리 정보를 손실함수에 반영하여 거리 정보의 정확도를 개선하였다. 실험 결과, 제안된 모델은 이미지 품질 및 깊이 성능 측면에서 기존 모델보다 우수한 성능을 보였다.

[Abstract]

The recent rapid advancements in autonomous driving technology face significant challenges in safely evaluating dangerous situations that cannot be evaluated in real-world environments. Although virtual simulation technology has been introduced to evaluate dangerous scenarios, current simulated driving environments are manually reconstructed using CAD programs, resulting in limitations in reproducibility and realism. To solve these problems, this paper proposes Real-3DGS, a realistic scale environmental modeling method based on 3D Gaussian splatting using sensor fusion. To address the limitations of the naive 3DGS, which has high realism but lacks accuracy in distance measurements because it employs images, the proposed Real-3DGS generates initial Gaussians from LiDAR point clouds augmented with color information from images. Additionally, initial camera poses are aligned based on pose information estimated by LiDAR-based SLAM, and the distance information obtained from LiDAR is incorporated into the loss function to enhance the accuracy of regenerated distances. Experimental results show that the proposed method outperforms the naive 3DGS in image quality and depth performance.

색인어 : 환경 모델링, 3차원 가우시안 스플래팅, 현실적인 스케일, 센서 퓨전, 자율주행
Keyword : Environmental Modeling, 3D Gaussian Splatting, Realistic Scale, Sensor Fusion, Autonomous Driving

http://dx.doi.org/10.9728/dcs.2024.25.10.3031

Received 10 August 2024; **Revised** 26 September 2024
Accepted 26 September 2024

***Corresponding Author; Chan-Soo Kim**

Tel: +82-62-530-4227
E-mail: chansoo.kim@jnu.ac.kr

Ⅰ. 서 론

자율주행 기술은 최근 급격한 발전을 이루며, 양산 자동차에 탑재되기 위한 다양한 기능 개발 및 검증 과정이 활발히 진행되고 있다. 자율주행 자동차의 안전성과 신뢰성을 확보하기 위해 일반 도로 주행 환경에서의 검증이 중요한 과제로 부각되고 있으며, 이를 위해 임시적으로 설정된 자율주행 시범 운영지구 구간에서는 다양한 주행 상황에 대한 기능적 검증과 성능 평가가 이루어지고 있다[1]. 이는 실제 도로 환경에서 발생할 수 있는 다양한 주행 상황을 재현하고 평가하는 데 중요한 역할을 하고 있다.

그러나 자율주행 기술의 특성상 작은 실수로도 인명적·재산적으로 치명적인 피해를 초래할 수 있기 때문에, 현실 도로에서는 테스트할 수 없는 위험한 상황이나 다양한 시나리오를 안전하게 평가하는 데 한계가 있다. 이러한 한계를 극복하기 위해 가상 환경 시뮬레이션을 통한 평가와 검증 방법이 중요하게 대두되고 있다. 가상 환경 시뮬레이션은 자율주행 기술의 안전성과 신뢰성을 높이기 위해, 현실에서 구현하기 어려운 극한의 상황을 평가하고 검증하는 데 유용한 도구로 활용되고 있다[2].

자율주행의 기능 검증을 위한 가상 환경 시뮬레이션은 세 가지 필수 요건을 충족해야 한다. 첫째, 가상 환경은 실제 환경과 유사한 현실적인 이미지를 재현할 수 있어야 한다. 둘째, 자율주행에서는 주변 객체와의 거리 정보가 매우 중요하기 때문에, 현실 환경의 거리 정보를 정확하게 묘사하고 표현할 수 있어야 한다. 셋째, 자율주행 가상 환경 시뮬레이션은 실제 환경을 묘사하기 위한 실시간성을 보장할 수 있어야 한다. 이러한 요구사항을 충족시키기 위해서는 높은 거리 정확도를 보장하는 현실적인 스케일의 고품질 3D 환경 모델링과 실시간성 및 재현성을 보장하는 이미지 렌더링이 매우 중요하다.

고품질 3D 환경 모델을 생성하는 방법으로는 여러 가지가 있다. 대부분의 가상 시뮬레이션 프로그램에서는 도시 설계도를 기반으로 3차원 게임 환경처럼 모델을 만드는 방법을 활용한다[3]. 이 방법은 기구성된 에셋을 배치하여 도로 주행 환경을 구축하기 때문에 만들기 용이하고 수정 및 보완이 간편하다는 장점을 가진다. 하지만, 이러한 방법은 실제 환경의 복잡성을 표현할 수 없고, 현실과 다른 이미지를 렌더링한다는 단점이 있다.

또 다른 방법으로는 고정밀 LiDAR (Light Detection and Ranging)를 탑재한 MMS (Mobile Mapping System) 장비를 사용하여 실제 환경을 스캔하고 이를 환경 모델로 변환하는 방법이 있다[4]. 이 방법은 실제 환경에서 취득한 데이터를 활용하기 때문에 복잡성을 잘 표현할 수 있지만, MMS 장비의 가격이 비싸고 색상 정보가 포함된 점군 데이터로 환경을 모델링하기 때문에 렌더링된 이미지가 희소한 정보를 제공한다는 단점이 있다.

최근에는 여러 시점의 카메라 이미지 정보를 취합하여 3D 환경을 구축하는 NeRF (Neural Radiance Fields)[5] 및 3DGS (3D Gaussian Splatting)[6] 방법이 발전하고 있다. 이 방법은 카메라 이미지 정보만을 활용하기 때문에 데이터 취득 비용이 저렴하며, 새로운 시점에서 렌더링된 이미지의 현실성과 실시간성이 뛰어나다는 장점이 있다. 하지만, 깊이 정보를 포함하지 않는 이미지 정보만을 활용하기 때문에 주변 객체와의 거리 정보를 정확히 묘사하기 어렵다는 단점이 있다.

자율주행을 위한 가상 환경 시뮬레이션 요건에 맞는 고품질 3D 환경 모델을 생성하기 위해, 본 논문에서는 카메라와 LiDAR의 센서 퓨전을 통한 3DGS 기반 새로운 환경 모델링 기법인 Real-3DGS 기법을 제안한다. 기존 3DGS는 여러 시점에서 촬영한 이미지를 활용하여 새로운 시점에서도 현실성 있는 이미지를 재생성하는 장점을 가지지만, 이미지 정보만을 이용하기 때문에 거리 정확도가 떨어지는 단점이 있다. 제안된 방법은 저가의 LiDAR 센서 정보를 융합하여 3DGS의 실시간성과 재현성을 유지하면서도 정확한 거리 정보를 제공하는 것을 목표로 한다. 주요 기여점으로는 먼저 센서 융합을 활용한 3DGS 기반 주행 환경 모델링을 위한 새로운 파이프라인을 제안하였다. 그리고 정성적·정량적 결과를 통해 제안된 방법이 단일 센서 사용 대비 향상된 3D 재구성 정확도와 렌더링 품질을 달성한 것을 증명하였다.

이를 수행하기 위해, 본 논문에서는 LiDAR 기반의 SLAM (Simultaneous Localization And Mapping)을 사용하여 위치 전처리를 수행하고, 카메라와 LiDAR 정보를 조합하여 포인트 전처리를 수행하며, LiDAR에서 취득한 거리 정보를 손실함수에 반영함으로써 기존 3DGS 기법에 비해 정확한 거리 정보를 제공한다. 해당 연구는 자율주행 개발에 있어 효율적이고 정확한 3D 환경 모델링에 대한 증가하는 수요를 해결하며, 가상 현실, 증강 현실 및 기타 3D 비전 작업에도 적용 가능성을 갖는다.

Ⅱ. 배경 연구

2-1 NeRF (Neural Radiance Field)

NeRF[5]은 여러 시점에서 취득된 이미지 정보를 통합하여 3D 장면을 표현하고 새로운 시각에서 이미지를 합성하는 분야에서 혁신적인 변화를 가져왔다. 이 기술은 미분 가능한 볼륨 렌더링 기법을 사용하여 고품질의 이미지를 생성할 수 있으며, 복잡한 기하학적 구조와 미세한 텍스처를 정확하게 재현할 수 있는 장점을 지닌다. NeRF는 신경망을 이용하여 3D 공간 내의 각 위치와 방향에서의 색상과 밀도를 예측함으로써 새로운 시점에서의 이미지를 생성한다.

NeRF의 작동 원리는 다음과 같다. 먼저 3D 공간에서 샘플링된 지점에 대해 신경망이 해당 지점의 색상과 밀도를 예측한다. 이후, 볼륨 렌더링을 통해 이 예측값들을 통합하여 최종

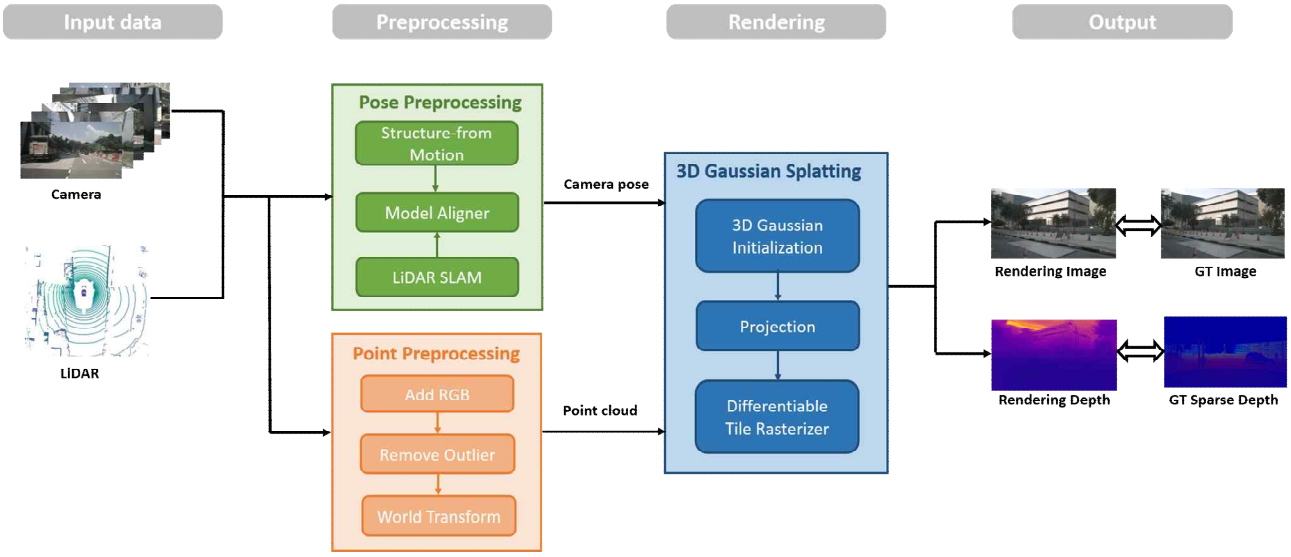


그림 1. 모델 구조
Fig. 1. Model architecture

이미지를 생성한다. 이 방식은 특히 정적 장면에서 뛰어난 성능을 발휘하며, 기존의 3D 모델링 기술에 비해 높은 품질을 제공한다.

그러나 NeRF는 고품질의 결과를 얻기 위해 많은 양의 훈련 데이터가 필요하며, 이로 인해 데이터 수집과 처리에 많은 시간과 자원이 소요된다. 또한, NVIDIA V100을 사용할 때 한 장의 이미지를 렌더링하는 데 30초 이상의 시간이 소요된다. 이러한 단점들로 인해 NeRF는 실시간 렌더링을 요구하는 자율주행을 위한 가상 환경 시뮬레이션에는 적합하지 않다.

2-2 3DGS (3D Gaussian Splatting)

3DGS[6]은 NeRF의 한계를 극복하기 위한 대안으로 등장하였다. 3DGS는 3D 공간상의 가우시안 분포를 사용하여 장면을 표현하고, 이를 2D 이미지 평면으로 투영(splatting)한다. 3DGS는 각 가우시안의 위치, 크기, 색상, 불투명도 등의 속성을 기반으로 래스터화하여 초당 150여 개의 이미지를 출력하는 빠른 렌더링 속도를 달성한다.

3DGS는 NeRF에 비해 훨씬 빠른 렌더링 속도를 제공하며, 실시간 응용이 가능하고, 메모리 효율성이 높아 대규모 환경 모델링에 적합하다. 또한, 3DGS는 복잡한 신경망 학습 과정이 필요 없기 때문에 상대적으로 적은 계산 자원으로도 높은 성능을 발휘할 수 있다. 이는 자율주행, 가상현실(VR), 증강현실(AR) 등의 실시간 응용 분야에서 매우 유리한 특성이다. 3DGS는 다양한 환경 조건에서도 안정적인 성능을 보이며, 복잡한 장면의 빠르고 효율적인 렌더링을 가능하게 한다.

그러나 3DGS는 주로 카메라 데이터만을 사용하여 새로운 환경을 모델링하기 때문에 실제 환경의 스케일을 정확히 반영하기 어렵다. 이로 인해 생성된 3D 모델이 실제 환경의 물리적인 규모와 일치하지 않을 수 있으며, 이는 자율주행 시뮬레이션과 같이 거리 정확도가 중요한 어플리케이션에서는 치명적인 문제가 된다. 본 연구에서는 이 문제를 해결하기 위해 LiDAR를 추가하여 정확한 3차원 정보를 획득하고, 이를 기반으로 실제 환경과 유사한 스케일의 3DGS 지도를 생성하는 방법을 제안한다.

Ⅲ. 제안 방법

본 논문에서는 카메라와 LiDAR의 센서 퓨전을 통한 3DGS 기반 새로운 환경 모델링 기법인 Real-3DGS 기법을 제안한다. 제안된 기법은 카메라의 고해상도 이미지와 LiDAR의 정확한 깊이 정보를 통합하여 고품질의 3D 환경 모델을 생성하고, 이를 효율적으로 렌더링한다.

3-1 개요

그림 1은 본 논문에서 제안된 Real-3DGS 모델 구조를 나타낸다. 본 논문에서는 실시간 렌더링의 장점을 활용하기 위해 3DGS 기반의 렌더링 방법을 사용한다. 기본 3DGS는 여러 단계를 거쳐 이미지를 렌더링한다. 먼저, SfM (Structure from Motion) 알고리즘[7]을 통해 카메라 이미지에서 추출된 SIFT 특징 정보[8]를 정합하고 최적화를 수행함으로써 카메라의 최적 위치를 계산한다. 그 후, 카메라의 최적 위치로부터 정합된 SIFT 특징의 3차원 위치 정보를 삼각화 기법을 통해 복원하여 3D 점군 데이터를 생성한다. 이렇게 생성된 최적 카메라 위치와 3D 점군 데이터는 3DGS의 입력 정보로 사용된다.

3DGS의 초기화 단계에서는 3D 점군 데이터를 기반으로 가우시안을 위한 초기값을 생성하며, 초기 속성 값인 공분산(Covariances), 색상(Colors), 불투명도(Opacities)은 임의로 설정된다. Projection 단계에서는 3D 가우시안을 이미지 평면으로 투영하여 2D 가우시안 형태로 변환한다. 이어서 Differentiable Tile Rasterizer 단계에서는 미분 가능한 형태의 Tile Rasterization을 통해 2D 가우시안들을 하나의 이미지로 생성한다. 마지막으로, Adaptive Density Control 단계에서는 생성된 이미지와 GT 이미지의 손실을 계산한 후, 그 손실을 기반으로 가우시안의 형태를 조정한다. 이 과정에서 투명도가 낮은 가우시안은 제거되고, 높은 가우시안은 가우시안의 크기에 따라 나누어서 분리되거나 복제된다. 이러한 과정을 통해 3차원의 환경 모델을 생성하고, 새로운 시점에서의 이미지를 실시간으로 현실감 있게 렌더링할 수 있다.

하지만, 기본 3DGS는 가상 환경을 구축하는 데 몇 가지 한계를 가지고 있다. 첫째, 거리 정보가 주어지지 않는 카메라 정보만을 활용하여 카메라 위치 정보를 추정하기 때문에, 실제 스케일의 정보대로 위치 정보를 추출하지 못하는 문제가 있다. 둘째, 기본 3DGS는 이미지를 출력할 뿐, 시점에서부터 주변 환경까지의 거리 정보를 제공하지 않는다.

이러한 문제를 해결하기 위해 Real-3DGS에서는 두 가지 개선점을 도입하였다. 첫째, LiDAR를 추가하여 정확한 3차원 정보를 획득하고, 이를 통해 실제 환경과 유사한 스케일의 3DGS 환경 모델을 생성하였다. 이를 위해 LiDAR 점군 데이터를 전처리하는 점군 데이터 전처리 단계와 LiDAR SLAM을 기반으로 한 위치 추정을 통해 정확한 위치 정보를 획득하는 위치 전처리 단계를 추가하였다. 둘째, 각 픽셀이 카메라와 물체 사이의 거리를 나타내는 깊이 이미지를 출력하도록 깊이 렌더링 과정을 추가하였다. 깊이 렌더링 과정의 결과와 LiDAR 데이터 사이에서의 손실 함수를 반영함으로써, 거리 추정 성능을 높였다.

3-2 위치 데이터 전처리

3-1에서 기본 3DGS는 SfM 알고리즘을 통해 카메라의 최적 위치를 계산한다고 언급하였다. 하지만 깊이 정보가 없는 카메라 센서만 사용하기 때문에 정확한 위치를 취득할 수 없다. 이를 해결하기 위해, 우리는 LiDAR SLAM으로 추정한 위치를 통해서 실제 환경과 알맞은 위치 정보를 획득하여 이를 반영하였다.

그림 2에 있는 Model Aligner 알고리즘은 SfM으로 추정한 카메라 위치가 LiDAR SLAM으로 얻은 위치 정보를 참조하여 실제 환경의 스케일에 맞게 재정렬될 수 있도록 한다. Model Aligner의 목표는 SfM으로 추정한 위치가 LiDAR SLAM으로 추정한 위치와 정합될 수 있게 만드는 이동, 회전, 축척 정보를 가진 최적의 변환 행렬 모델을 구하는 것이다. 이를 위해, Model Aligner 알고리즘은 임의의 입력 모델을 반복적으로 적용하여 여러 개의 입력 모델 중 입력 정보들 사이의 유사도가 최선이 되는 입력 모델을 찾는 RANSAC 알고리즘[9]을 응용하였다.

Algorithm 1: Model Aligner

Input : SfM Pose Data: D_{src}
LiDAR SLAM Pose Data: D_{dst}
Number of Iterations: *iterations*
Output: Best Model: M_{best}

$M_{best} \leftarrow \emptyset$
best_inlier_count $\leftarrow 0$

for $i \leftarrow 1$ **to** *iterations* **do**
  $x \leftarrow$ random_sample(D_{src}, n)
  $y \leftarrow$ random_sample(D_{dst}, n)
  $M \leftarrow$ umeyama_model(x, y)
  inliers $\leftarrow$ cal_inlier(M, x, y)
  inlier_count $\leftarrow$ len(*inliers*)

  if *inlier_count* > *best_inlier_count* **then**
    best_inlier_count $\leftarrow$ *inlier_count*
    $M_{best} \leftarrow M$

return *best_model*

그림 2. Model Aligner 슈도코드
Fig. 2. Model Aligner pseudocode

먼저, SfM으로 추정한 위치와 LiDAR SLAM으로 추정한 위치를 각각 랜덤하게 n개를 샘플링하고 이를 x, y로 부르겠다. RANSAC의 입력 모델을 선정하기 위해, x와 y 위치 정보를 사용하여 변환 행렬 모델을 계산하는 Umeyama 알고리즘[10]이 사용되었다. x를 y에 맞추는 변환 행렬을 계산하기 위해, 수식(1)과 같이 x와 y 간의 공분산을 구하고 수식 (2)와 같이 특이값 분해를 수행한다.

$$H = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_y)^T(x_i - \mu_x), \tag{1}$$

$$H = UDV^T \tag{2}$$

특이값 분해로 얻은 U, V를 통해 수식 (3), (4), (5)를 수행함으로써, 회전 행렬 R, 축척 c, 이동 행렬 t를 구할 수 있고, 이를 통해 수식(6)과 같이 x를 y에 재정렬하는 변환 행렬 모델 M을 계산할 수 있다.

$$S = \begin{cases} I & \text{if } \det(H) \geq 0 \\ diag(1,1,-1) & \text{if } \det(H) < 0 \end{cases}, \tag{3}$$

$$R = USV^T,\ c = \frac{\sigma_y^2}{tr(DS)}, \tag{4}$$

$$t = \mu_y - cR\mu_x, \tag{5}$$

$$M = \begin{bmatrix} cR & t \\ 0 & 1 \end{bmatrix} \tag{6}$$

계산된 변환 행렬 모델 M이 유사도가 최선이 되는 최적 입력 모델인지 확인하기 위해, 변환 행렬 모델 M에 의해 변환된 x'과 y 사이의 거리가 특정 threshold보다 작으면

inlier라고 판단하고, inlier의 개수를 센다. 만약 계산된 inlier의 개수가 현재까지의 최적 변환 행렬 모델 M_{best}의 inlier 개수보다 많으면 현재 모델을 최적 변환 행렬 모델 M_{best}로 선정한다. 여러 반복을 통해 선정된 최적 변환 행렬 모델 M_{best}은 최종적으로 SfM으로 추정한 카메라 위치와 LiDAR SLAM으로 얻은 위치 정보 사이의 변환 행렬모델을 잘 추정할 수 있다. 우리는 이 변환 행렬 모델로 SfM으로 추정한 카메라 위치를 LiDAR SLAM으로 추정한 카메라 위치에 맞게 재정렬할 수 있다.

3-3 점군 데이터 전처리

카메라로 취득한 모든 장면의 이미지를 사용하여 Structure-from-Motion(SfM)[7] 알고리즘을 통해 최적 카메라 위치 및 3D 점군 데이터를 추출한다. 이후, 3-2에서 설명한 Locally Optimized RANSAC[9] 및 Umeyama[10] 기반의 위치 데이터 전처리 모델을 활용하여 카메라 위치를 실제 스케일에 맞게 재정렬한다.

이후 입력 데이터 전처리 과정에서는 Camera-LiDAR 간의 캘리브레이션 정보를 이용하여 LiDAR 점군 데이터에 RGB 값을 추가한다. 카메라의 시야 범위(view frustum) 밖의 점을 제거한 후, 모든 점들을 실세계 좌표계로 변환하여 누적한다.

LiDAR 점군 데이터 매핑은 LiDAR 센서로부터 얻은 3D 점군 데이터를 시각적으로 풍부한 3D 모델로 만드는 과정이다. 카메라 이미지와 매핑하여 각 점에 RGB 값을 추가한다. 카메라와 LiDAR가 취득한 데이터는 정확한 시간 동기화가 필요하며, 동기화된 시점의 카메라 이미지와 LiDAR 점군 데이터들은 캘리브레이션 정보를 통해 LiDAR 점군 데이터를 카메라 이미지 평면으로 투영할 수 있다. 투영된 점이 위치하는 이미지의 픽셀에서 RGB 값을 가져오고 이를 LiDAR 점의 새로운 특징 정보로 할당한다. 이를 통해 점군 데이터에 색상 정보를 부여하여 시각적으로 더 풍부한 3D 모델을 생성할 수 있다.

카메라의 시야 범위(view frustum)는 카메라의 위치와 방향, 그리고 렌즈의 초점 거리와 같은 파라미터에 의해 정의된다. LiDAR 점군 데이터 중 카메라의 시야 범위 내에 있는 점들만 선택하고, 나머지 점들은 제거한다. 이 과정은 카메라의 투영행렬을 사용하여 각 점들이 시야 범위 내에 있는지 확인하는 방식으로 이루어진다. 시야 범위 밖의 점들을 제거함으로써 렌더링 성능을 최적화하고 불필요한 데이터 처리를 줄이는 데 도움을 준다. 이는 메모리 사용량을 줄이고, 처리 시간을 단축시키는 효과가 있다.

카메라와 LiDAR 각각의 점군 데이터는 각기 다른 좌표계를 갖고 있다. 이를 통합하기 위해 모든 점들을 공통된 실세계 좌표계로 변환한다. 이 과정에서는 카메라와 LiDAR의 상대적 위치 및 방향 정보(외부 파라미터)를 사용하여 점들을 실세계 좌표계로 변환한다. 이를 통해 여러 시점에서 수집된 점군 데이터를 하나의 일관된 실세계 좌표계로 통합할 수 있으며, 전체 환경을 하나의 3D 모델로 표현할 수 있다. 이렇게 하면 다양한 시점에서 수집된 데이터들이 정확히 정렬되고, 누적된 3D 환경 모델을 생성할 수 있다.

3-4 깊이 렌더링

기존 3DGS는 깊이를 렌더링하지 않는다. 따라서 본 논문에서는 깊이를 렌더링할 수 있도록 기존 3DGS가 사용하는 Differentiable Tile Rasterizer에 깊이를 추출하는 메서드를 추가하였다. 깊이를 렌더링하기 위해 색상을 rasterizing 하는 수식을 참고하였다.

$$C = \sum_{i \in N} c_i \alpha_i T_i, \quad where \;\; T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{7}$$

여기서 C는 픽셀 색상, c는 splat 색상, α는 학습된 불투명도에 2D 가우시안을 곱한 값이다. 이 공식은 카메라에 더 가깝게 위치한 불투명한 가우시안의 색상 c를 우선시하기 때문에 최종 결과 C에 큰 영향을 미친다. 해당 파이프라인을 활용하여 가우시안의 깊이도 아래 수식(8)을 통해 렌더링한다.

$$D = \sum_{i \in N} d_i \alpha_i T_i \tag{8}$$

여기서 D는 렌더링 된 깊이, d_i는 카메라에서 바라본 각 가우시안의 깊이이다. 위 식도 카메라에 더 가깝게 위치한 가우시안의 깊이를 우선시하고, α_i와 T_i의 직접적인 활용을 가능하게 하여 최소한의 계산 부하로 빠른 깊이 렌더링을 촉진한다.

이렇게 추출된 깊이 값은 각 픽셀의 깊이 이미지를 형성하며, 3D 환경에서의 물체와 카메라 사이의 거리를 정확하게 반영할 수 있다.

3-5 손실 함수

본 연구에서는 색상 이미지와 깊이 정보를 활용하여 3D 환경 모델의 정확성을 평가하고 최적화하기 위해 다양한 손실 함수를 사용하였다. 각 손실 함수는 예측된 값과 실제 값 간의 차이를 최소화하는 것을 목표로 한다.

1) 색상 손실 함수(L_{rgb})

색상 손실 함수는 예측된 색상 이미지와 실제 색상 이미지 간의 손실을 계산하는 방법을 나타낸다. 이 함수는 두 부분으로 구성된다. 첫째, L1 손실은 예측된 색상 이미지와 실제 색

상 이미지 간의 절대 차이의 평균을 계산하여 이미지의 색상 정보를 정확히 재현하기 위해 사용된다. 둘째, 구조적 유사성 지수(SSIM) 손실은 예측된 이미지와 실제 이미지 간의 구조적 유사성을 평가하는 손실 함수로, 이미지의 밝기, 대비, 구조 정보를 종합적으로 고려하여 손실을 계산한다. λ는 L1 손실과 SSIM 손실 간의 가중치를 조절하는 파라미터이다. RGB 손실 함수는 다음과 같이 표현된다.

$$L_{rgb} = (1-\lambda)L1 + \lambda L_{S\sim} \tag{9}$$

여기서 L_{rgb}는 색상 손실, $L1$은 L1 손실, $L_{S\sim}$은 SSIM 손실, 그리고 λ는 두 손실간의 가중치를 조절하는 파라미터이다.

2) 깊이 손실 함수(L_{depth})

깊이 손실 함수는 예측된 깊이 값과 실제 깊이 값 간의 차이를 계산하는 손실 함수이다. 이 함수는 평균 절대 오차(Mean Absolute Error)를 기반으로 하며, 예측된 깊이 값과 실제 깊이 값 간의 절대 차이의 평균을 계산하여 깊이 정보의 정확성을 향상시킨다. 깊이 손실 함수는 다음과 같이 표현된다.

$$L_{depth} = \frac{1}{N}\sum_{i=1}^{N}|D_{GT} - D_{pred}| \tag{10}$$

여기서 N은 데이터 포인트의 수, D_{GT}는 실제 깊이 값, D_{pred}는 예측된 깊이 값을 나타낸다. 이때 D_{GT}는 LiDAR로 추출한 point를 2차원 이미지 평면에 투영한 희소한 깊이 지도이고 이는 그림 1에서 볼 수 있다.

3) Total 손실 함수(L_{total})

총 손실 함수는 색상 손실과 깊이 손실을 결합하여 최종 손실 값을 계산한다. 이 함수는 두 손실 간의 가중치를 조절하는 하이퍼 파라미터 α를 포함하며, 최적의 성능을 얻기 위해 실험적으로 설정된다. 본 연구에서는 α를 0.6으로 설정하였다. 총 손실 함수는 다음과 같이 표현된다.

$$L_{total} = \alpha L_{rgb} + (1-\alpha)L_{depth} \tag{11}$$

여기서 L_{total}은 최종 손실 값을 의미하며, α는 색상 손실과 깊이 손실 간의 가중치를 조절하는 하이퍼파라미터이다. 이 수식은 두 손실 간의 균형을 맞추기 위해 사용된다.

이와 같은 손실 함수를 통해 전처리된 LiDAR 포인트와 Model Aligner로 재정렬된 카메라 위치는 같은 실세계 좌표계로 통합되어 3DGS를 학습할 수 있다. 제안된 방법은 적은 양의 입력 데이터로도 실시간 렌더링 성능을 달성하는 것을 목표로 하며, 자율주행 시뮬레이션에서의 정확성과 효율성을 높이는 데 기여한다.

Ⅳ. 실험 및 결과 분석

4-1 실험 설정

1) 데이터셋

nuScenes[11] 데이터셋은 자율주행 연구 및 개발을 위한 고품질의 다중 센서 데이터셋으로 객체 검출 및 추적, 맵핑 및 로컬라이제이션, 행동 예측 및 경로 계획 알고리즘을 개발하고 평가하는데 사용된다. 해당 데이터셋에서 사용하는 32채널 LiDAR는 초당 20회 스캔된 360° 시점을 제공해주고 카메라도 6대를 사용하여 360° 정보를 제공하고 센서들이 동일한 시간 기준으로 데이터를 수집하였기 때문에 센서 퓨전을 하는데 적합하다.

2) 실험 환경 설정

우리는 nuScenes 데이터셋의 싱가포르 도시 환경 중 일부를 선택하여 알고리즘을 테스트하였다. epoch는 70000으로 설정하고 learning rate나 density control threshold와 같은 하이퍼 파라미터는 기존 3DGS[6]와 동일하게 설정하여 모델 학습을 진행하였다. 실험을 위해 NVIDIA A5000 GPU 장비를 사용하였고 OS 환경은 Ubuntu 20.04, CUDA는 11.7 버전을 사용하였다.

4-2 실험 결과 분석

1) RGB 렌더링 성능 결과

우리는 총 네 가지 모델로 실험을 진행하였는데, (A)는 기존 3DGS 모델이고, (B)와 (C)는 Model Aligner로 재정렬된 카메라 포즈를 사용하지만 초기 가우시안의 시드 값이 각각 SFM 점군 데이터, LiDAR 점군 데이터로 다른 모델이다. 마지막으로 (D)는 LiDAR 점군 데이터에서 깊이 최적화까지 반영한 우리가 최종적으로 제안한 모델이다. 이미지 렌더링 성능을 정량적으로 평가하기 위해 이미지 품질 측정에 사용되는 PSNR, SSIM, LPIPS 평가 지표를 사용하였다. 수식 (12)에서 표현된 PSNR은 원본 이미지와 복원된 이미지의 픽셀값 차이로 두 이미지 간의 유사성을 평가한다. 수식 (13)의 SSIM은 밝기, 대비, 구조 정보를 모두 고려하여 이미지 구조적 유사성을 측정하고, 수식 (14)의 LPIPS는 딥러닝 모델을 사용하여 인간의 시각적 인식과 유사한 방식으로 이미지 유사성을 평가한다. PSNR과 SSIM은 높을수록 LPIPS는 낮을수록 원본 이미지를 잘 복원했다는 의미이고 수식은 아래와 같다.

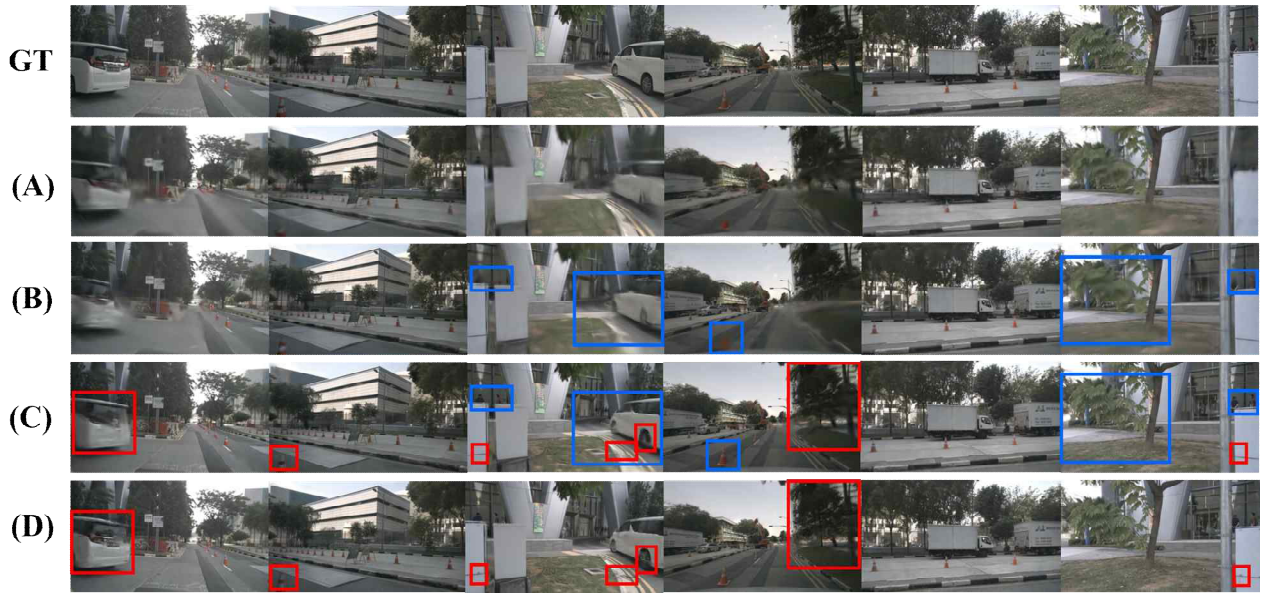


그림 3. 색상 이미지 렌더링 결과
Fig. 3. Color image rendering results

표 1. 이미지 복원 성능 정량적 평가
Table 1. Quantitative evaluation of image restoration

	Pose	Point	Depth	PSNR↑	SSIM↑	LPIPS↓
(A)	X	X	X	26.29	0.815	0.393
(B)	O	X	X	27.68	0.849	0.334
(C)	O	O	X	28.85	0.873	0.290
(D)	O	O	O	29.03	0.883	0.268

표 2. 깊이 성능 정량적 평가
Table 2. Quantitative evaluation of depth rendering

	Pose	Point	Depth	Abs Rel↓	Sq Rel↓	RMSE ↓	RMSE log↓	$d_{1.25}$↑
(A)	X	X	X	0.81	11.56	18.06	2.16	0.02
(B)	O	X	X	0.36	4.08	9.15	0.72	0.45
(C)	O	O	X	0.30	3.32	8.57	0.63	0.54
(D)	O	O	O	0.11	0.94	4.36	0.26	0.87

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(i,j)-\hat{I}(i,j))^2}\right), \quad (12)$$

$$S\sim = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}, \quad (13)$$

$$LPIPS = \sum_l \frac{1}{H_l W_l}\sum_{h=1}^{H_l}\sum_{w=1}^{W_l} \| w_l(f_l(x-y)_{hw}) \|_2^2 \quad (14)$$

Model Aligner가 포함된 우리의 제안 방법인 (B), (C), (D)가 기존 방법 (A)보다 좋은 성능을 보여주고, SfM 점군 데이터를 사용한 (B)보다 LiDAR 점군 데이터로 초기 가우시안을 만들어 주는 모델들인 (C), (D)가 더 우수한 성능을 보여주는 것을 표 1을 보면 확인할 수 있다. 그림 3은 원본 이미지에 대해 각 모델이 색상 이미지를 렌더링한 결과이다. 파란색 박스는 SfM 점군 데이터와 LiDAR 점군 데이터에 따른 결과이고 빨간색 박스는 깊이 최적화 반영 여부에 따른 결과이다. 정성적인 결과에서는 기존 방법인 (A)보다 나머지 모델들이 좋은 렌더링 결과를 보여주고 이는 아래로 내려갈수록 점차 향상된 것을 볼 수 있다. 예를 들어 여섯 번째 장면에서 LiDAR 점군 데이터를 사용한 모델들이 사람과 나무의 잎을 흐릿하지 않게 잘 묘사하였고 이는 카메라에 맞게 전처리 된 LiDAR 점군 데이터의 풍부한 3차원 정보에 기인한다고 볼 수 있다. 여기에서 깊이 최적화까지 추가한 모델인 (D)가 자동차와 같은 동적 객체, 콘, 나무 등 전반적으로 흐릿한 부분 없이 본 이미지를 잘 복원하였다.

2) 깊이 성능 결과

이 성능도 위의 색상 렌더링 성능을 비교한 것처럼 4가지 모델로 평가하였고 정량적 결과는 표 2에 나와있다. 이때, 실제 깊이 이미지는 LiDAR 기반 희소 깊이 이미지를 사용하였다. 표 2에 나와 있는 정량적 지표들은 깊이 추정 성능을 평가할 때 주로 사용되고 앞의 4개 열은 오차, 뒤에 1개 열은 정확도를 나타낸다. Abs와 Sq는 실제 깊이 값과 모델이 예측한 깊이 값의 차이를 상대적으로 계산한 지표이다. Abs는 이 차

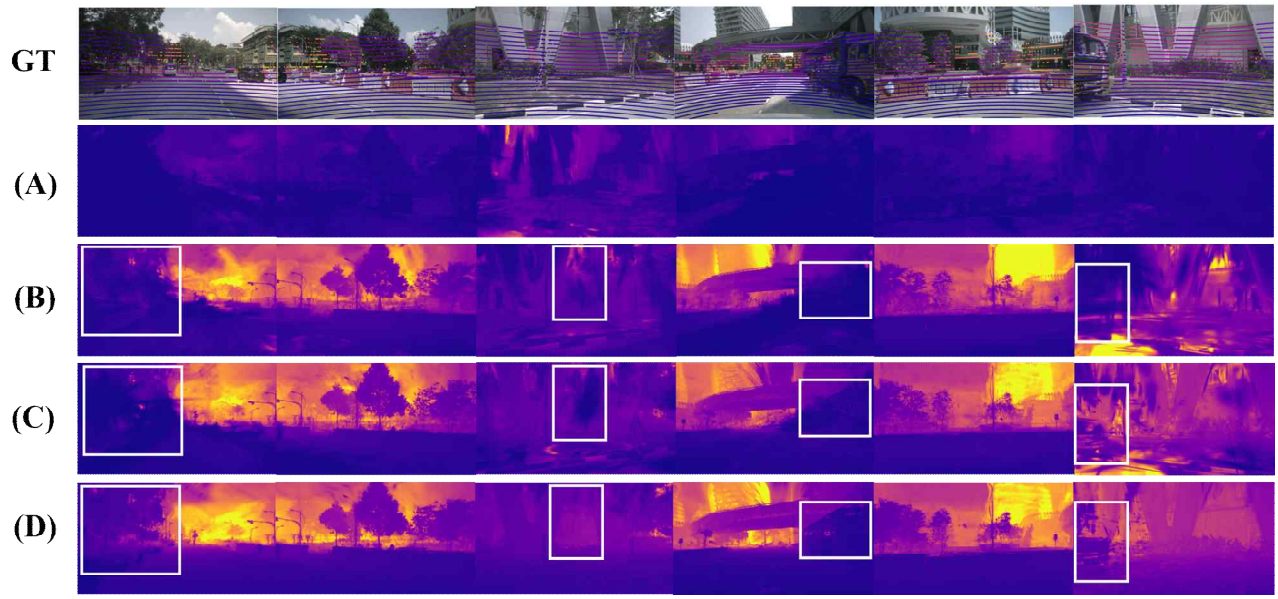


그림 4. 깊이 이미지 렌더링 결과
Fig. 4. Depth image rendering results

이를 절댓값으로 구하고, Sq는 차이를 제곱하여 계산한다. 실제 깊이 차이를 미터 단위로 측정하기 위해서는 RMSE를 사용하며, 이를 로그 스케일로 변환한 것이 RMSE log이다. 정확도는 예측 깊이와 실제 깊이 중 큰 값을 분자, 작은 값을 분모로 두고 특정 threshold 값 보다 작으면 True Positive로 간주한다. 본 연구에서는 threshold 값을 1.25로 사용하였다. 해당 지표들에 대한 수식은 아래와 같다.

$$Abs = \sum_{p} \frac{|d_p - \hat{d}_p|}{d_p}, \tag{15}$$

$$Sq = \frac{1}{T}\sum_{p} \frac{(d_p - \hat{d}_p)^2}{d_p}, \tag{16}$$

$$R = \sqrt{\frac{1}{T}\sum_{p} \frac{(d_p - \hat{d}_p)^2}{d_p}}, \tag{17}$$

$$R_{\log} = \sqrt{\frac{1}{T}\sum_{p} (\log(\hat{d}_p) - \log(d_p))^2}, \tag{18}$$

$$Accuracy = \max\left(\frac{\hat{d}_p}{d_p}, \frac{d_p}{\hat{d}_p}\right) = \sigma < threshold \tag{19}$$

위 지표들로 깊이를 정량적으로 평가하였을 때 본 논문에서 제안했던 위치 전처리, 점군 데이터 전처리,깊이 최적화를 하나씩 추가할수록 점차 좋은 결과가 나타났고 모두가 반영된 최종 제안 모델인 (D)가 가장 좋은 성능을 보여준다.

그림 4는 각 모델들이 렌더링한 깊이 이미지인데 이때, 가까운 거리일수록 어둡고 먼 거리일수록 밝은 색상을 띤다. 그림 4에서 첫 번째 장면과 세 번째 장면을 보면 (D)가 거리의 차이를 잘 구분하여 렌더링한 것을 알 수 있다.

첫 번째 장면에서 왼쪽 도로를 보면 (D)가 나머지 모델들보다 깊이감을 두드러지게 잘 표현하였다. 세 번째 장면에서는 (C)는 건물의 벽에서 움푹 들어간 부분을 가까운 거리로, 튀어 나온 부분을 먼 거리로 판단하였지만 (D)는 그렇지 않고 튀어나온 부분을 어둡게 들어간 부분을 밝게 렌더링하여 잘 구별하였다. 그리고 네 번째와 여섯 번째 장면을 보면 트럭과 같은 객체들도 (D)가 가장 좋은 렌더링 성능을 보여준다.

그림 5는 10 m 간격으로 깊이 성능을 비교한 그래프이다. (a)의 RMSE 오차 그래프를 보면, 거리가 멀어질수록 모델별로 차이가 뚜렷하게 나타났다. (A)는 거리가 증가할수록 오차가 선형적으로 증가했으며, (B)와 (C)도 90 m 이상의 거리에서 40 m 이상의 오차를 보였지만 (D)는 그보다 절반 정도인 약 20 m의 오차만을 나타냈다. (b)의 threshold가 1.25 미만인 정확도 그래프를 보면, (A)는 거의 0에 가까운 정확도를 보였고, (B)와 (C)는 10-20 m의 가까운 거리에서도 정확도가 60%를 넘지 않아 성능이 좋지 않았다. 이에 반해, (D)는 90-100 m 구간에서도 70%를 넘는 정확도를 달성했다. 이

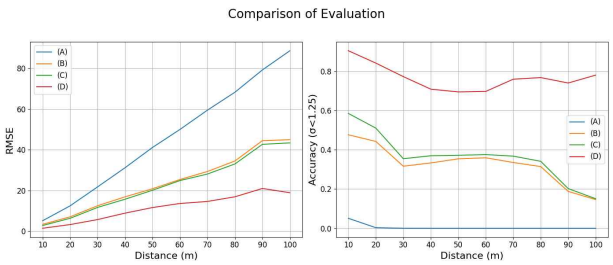


그림 5. 구간별 깊이 성능 분석
Fig. 5. Depth performance analysis

를 통해 최종 제안 모델인 (D)가 모든 거리에서 깊이 추정 성능이 우수함을 알 수 있고 이는 특히 먼 거리에서 두드러진다.

V. 결 론

본 논문에서는 3D Gaussian Splatting(3DGS)을 활용하여 현실적인 스케일로 렌더링함으로써 정확한 가상환경을 구축하는 방법을 제안한다. 제안된 모델의 구조는 3DGS의 렌더링 방식을 기반으로 하여 Model Aligner와 전처리 기법을 통해 센서 퓨전된 데이터를 사용하며, 색상 이미지뿐만 아니라 깊이 정보를 렌더링하고 최적화하는 과정을 추가하였다. 이를 통해 실제 환경의 스케일과 위치를 반영한 가상 환경을 만들어낼 수 있어 자율주행과 같은 대규모 환경 렌더링에 기여 할 수 있다. 성능 평가 결과 색상 및 깊이 측면에서 기존 방법보다 성능이 향상됨을 확인하였고 특히 먼 거리일수록 정확한 깊이 추정 성능을 보여주었다. 본 연구의 한계점으로는 주변 환경이 정적이다는 가정하에 프레임워크가 설계되어 동적 객체에 의한 부정확성이 나타난다. 그리고 자율주행 시뮬레이션을 통해 안전한 자율주행 시스템을 구축하기 위해서는 표 2의 결과보다 깊이 성능이 향상되어야 한다. 따라서 향후 연구로는 깊이 정보를 더욱 정확히 추정할 수 있도록 3차원 정보를 최적화할 수 있는 Geometric Loss을 반영하며, 환경 모델링을 위해 동적 객체 부분의 배제하는 방안이 필요함을 제안한다.

감사의 글

본 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2022-00166693). 이 논문은 2024년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임(P0020536, 2024년 산업혁신인재성장지원사업). 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업 연구 결과로 수행되었음(IITP-2023-RS-2023-00256629).

참고문헌

[1] Hankyoreh. 10 Autonomous Vehicle Pilot Operation Zones Have Been Added, Bringing the Total to 34 Zones Across 17 Provinces and Cities [Internet]. Available: https://www.hani.co.kr/arti/economy/economy_general/1118176.html.

[2] CARLA Simulator. Open-Source Simulator for Autonomous Driving Research [Internet]. Available: https://carla.org/.

[3] MathWorks. RoadRunner: Design 3D Scenes for Automated Driving Simulation [Internet]. Available: https://www.mathworks.com/products/roadrunner.html.

[4] Leica Geosystems. Leica Pegasus: Two Ultimate Mobile Sensor Platform [Internet]. Available: https://leica-geosystems.com/products/mobile-mapping-systems/capture-platforms/leica-pegasus_two-ultimate.

[5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis,” in *Proceedings of the 16th European Conference on Computer Vision (ECCV 2020)*, Glasgow, UK, pp. 405-421, August 2020. https://doi.org/10.1007/978-3-030-58452-8_24

[6] 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, 139, August 2023. https://doi.org/10.1145/3592433

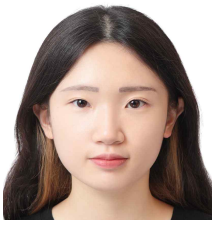
[7] J. L. Schönberger and J.-M. Frahm, “Structure-from-Motion Revisited,” in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Las Vegas: NV, pp. 4104-4113, June 2016. https://doi.org/10.1109/CVPR.2016.445

[8] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, November 2004. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[9] O. Chum, J. Matas, and J. Kittler, “Locally Optimized RANSAC,” in *Proceedings of the 25th DAGM Symposium*, Magdeburg, Germany, pp. 236-243, September 2003. https://doi.org/10.1007/978-3-540-45243-0_31

[10] S. Umeyama, “Least-Squares Estimation of Transformation Parameters between Two Point Patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 4, pp. 376-380, April 1991. https://doi.org/10.1109/34.88573

[11] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, ... and O. Beijbom, “nuScenes: A Multimodal Dataset for Autonomous Driving,” in *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle: WA, pp. 11618-11628, June 2020. https://doi.org/10.1109/CVPR42600.2020.01164



서유리(Yu-Ri Seo)

2021년～현　재: 전남대학교 인공지능학부 학사과정
※관심분야 : 자율주행 (Autonomous Driving), 3D 컴퓨터 비전 (3D Computer Vision)

허채연(Chae-Yeon Heo)

2021년～2024년 9월: 전남대학교 인공지능학부 (공학사)
2024년 9월～현　재: 전남대학교 인공지능융합학과 석사과정
※관심분야 : 3D 컴퓨터 비전 (3D Computer Vision)

김찬수(Chan-Soo Kim)

2013년 : 한양대학교 기계공학부 (공학사)
2020년 : 한양대학교 대학원 (공학박사-미래자동차공학과)

2020년 3월～2021년 2월: 한양대학교 미래자동차연구소 박사후연구원
2021년 3월～2021년 8월: 현대자동차 로보틱스랩 로봇지능팀 책임연구원
2021년 9월～현　재: 전남대학교 AI융합대학 지능형모빌리티융합학과 조교수
※관심분야 : 자율주행, SLAM, 센서 퓨전, 3차원 렌더링 등