

# GG-SLAM: Gaussian Grouping과 Splatting을 이용한 객체 수준 SLAM 시스템

## GG-SLAM: An Object-Level SLAM System Using Gaussian Grouping and Splatting

정 현 도<sup>1</sup> · 김 곤 우<sup>†</sup>  
Hyundo Jung<sup>1</sup>, Gon-woo Kim<sup>†</sup>

**Abstract:** Simultaneous Localization and Mapping (SLAM) is crucial for applications such as autonomous driving, robot navigation, and 3D reconstruction. With advancements in Virtual Reality (VR) and Augmented Reality (AR), the demand for immersive and realistic experiences is increasing. This paper introduces a real-time SLAM system enhanced with object-level information using Gaussian Grouping, based on the Gaussian Splatting technique. Our system utilizes predefined masks generated by Tracking Anything to segment objects effectively, enabling the real-time construction of high-quality 3D maps. By applying Gaussian Grouping, the system efficiently identifies static objects while integrating object-level data, maintaining the accuracy of traditional Gaussian Splatting-based SLAM. Experimental results demonstrate the system's high accuracy and efficiency across various environments, enhancing object recognition and interaction for robots, as well as enabling natural interactions with virtual objects in VR/AR applications. This study advances SLAM technologies by providing a high-quality 3D reconstruction method that incorporates object-level information, expanding its applicability to diverse domains.

**Keywords:** SLAM, Gaussian Splatting, Gaussian Grouping, Object-Level SLAM

### 1. 서 론

최근 컴퓨터 비전과 로봇 공학 분야에서 동시 위치추정 및 지도작성(SLAM)은 자율 주행, 로봇 내비게이션, 3D 재구성 등 다양한 응용 분야에서 핵심적인 역할을 하고 있다. 특히, 가상 현실(VR)과 증강 현실(AR) 산업의 발전으로 인해, 사용자에게 더욱 몰입감 있고 현실적인 경험을 제공하기 위한 기술의 필요

성이 증가하고 있다. 예를 들어, Meta Quest 3S와 같은 최신 VR 기기는 고해상도 디스플레이와 향상된 센서 기술을 통해 사용자 경험을 향상시키고 있으며, 이러한 기기들은 SLAM 기술을 활용하여 주변 환경을 인식하고 상호작용할 수 있다.

Gaussian Splatting<sup>[1]</sup>은 최근 제안된 기법으로, Neural Radiance Field (NeRF)<sup>[2]</sup>의 효율적인 대안으로 주목받고 있다. 이 기법은 3D 공간에서 가우시안 분포를 활용하여 현실감 있는 시각적 표현을 실시간으로 구현할 수 있으며, 복잡한 장면에서도 효율적인 처리가 가능하다. 이러한 특징은 VR/AR 애플리케이션에서 고품질의 그래픽을 제공하고, 사용자와의 상호작용을 향상시키는 데 큰 도움이 될 수 있다.

본 논문에서는 Gaussian Splatting을 활용한 Gaussian Grouping<sup>[3]</sup> 기반의 SLAM 시스템을 제안한다. 이 시스템은 Segment Anything<sup>[4]</sup>을 기반으로 한 Tracking Anything<sup>[5]</sup>에서 사전에 생성된 마스크를 활용하여 객체 수준의 정보를 지도 작성 과정에 통합한다. Gaussian Grouping은 3D 장면 내에서 동일한 객체에 속하는

Received : Oct. 24, 2024; Revised : Nov. 13, 2024; Accepted : Nov. 14, 2024

※ This research was supported in part by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0020536, HRD Program for Industrial Innovation) and in part by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2025-RS-2020-II201462, 50).

1. Graduate Student, IRL, Department of Intelligent Systems and Robotics, Chungbuk National University, Cheong-Ju, Korea (dsowrd@naver.com)

† Professor, Corresponding author: Department of Intelligent Systems and Robotics, Chungbuk National University, Cheong-Ju, Korea (gwkim@cbnu.ac.kr)

Gaussian들을 그룹화할 수 있게 하며, 이를 통해 객체 수준의 정보를 다양한 목적으로 활용할 수 있다. 예를 들면, 실내 환경에서 잠재적으로 움직일 수 있는 물체나 불필요한 물체들을 제거하거나 이동할 수 있고, 물체를 정확하게 구분함으로써 VR/AR 애플리케이션에서 가상 객체와의 자연스러운 상호작용을 가능하게 한다. 이를 통해 현실적이고 의미 있는 지도 작성 및 VR/AR 산업에서 상호작용이 풍부한 가상 환경 구축에 중요한 역할을 한다.

본 연구의 주요 기여:

1. Tracking Anything<sup>10</sup>로부터 생성된 사전 정의된 마스크를 이용한 실시간 Gaussian Grouping SLAM 시스템 구현.
2. Gaussian Grouping을 통한 객체 구분.
3. 기존 Gaussian Splatting 기반 SLAM의 정확도를 유지하면서 객체 수준 정보 통합.

이를 통해 로봇의 주변 환경 이해, 물체 인식 및 상호작용, 3D 스캔을 통한 환경 정보 보존 등 다양한 분야에서 현실적이고 의미 있는 3D 지도 작성이 가능해질 것으로 기대할 수 있다. 또한, VR/AR 산업에서 더욱 현실적이고 상호작용이 풍부한 가상 환경을 구축하는 데 기여할 수 있을 것이다.

## 2. Gaussian Splatting 기반 기존 연구

### 2.1 SLAM 기술 개요

동시 위치추정 및 지도작성(SLAM)은 로봇 공학과 컴퓨터 비전 분야에서 중요한 역할을 하며, 자율 주행, 로봇 내비게이션, 증강 현실(AR), 가상 현실(VR) 등 다양한 응용 분야에서 활용되고 있다. SLAM은 로봇이나 카메라가 알 수 없는 환경에서 자신의 위치를 추정하면서 동시에 그 환경의 지도를 작성하는 문제를 해결한다.

초기의 SLAM 연구는 확장 칼만 필터(EKF)나 입자 필터와 같은 확률론적 방법을 사용하여 로봇의 위치와 환경의 특징점을 동시에 추정하였다. 예를 들어, MonoSLAM<sup>6</sup>은 단일 카메라를 사용하여 실시간으로 위치 추정과 지도 작성을 수행하였고, ORB-SLAM<sup>7,8</sup>은 ORB 특징점을 활용하여 높은 정확도와 실시간 성능을 달성하였다.

이후 딥러닝의 발전으로 신경망을 활용한 SLAM 연구가 활발히 진행되고 있다. DeepVO<sup>9</sup>는 심층 신경망을 사용하여 시각적 주행 거리계(Visual Odometry)를 학습하였으며, DS-SLAM<sup>10</sup>은 딥러닝 기반의 객체 분할을 통해 동적인 객체를 제거하여 SLAM 성능을 향상시켰다.

또한 환경에 대한 고차원적인 이해를 위해 의미론적 SLAM에 대한 연구도 이루어지고 있다. SemanticFusion<sup>11</sup>은 CNN을

활용하여 실시간으로 3D 지도와 의미론적 분할을 결합하였고, MaskFusion<sup>12</sup>은 객체 마스크를 사용하여 동적인 객체를 인식하고 재구성하였다. 최근에는 Segment Anything<sup>14</sup>과 같은 범용적인 객체 분할 기법이 등장하여, 객체 수준의 정보를 SLAM에 통합하는 데 활용되고 있다.

### 2.2 Gaussian Splatting 기반 SLAM

기존의 NeRF (Neural Radiance Fields)<sup>12</sup> 표현은 고품질의 시각적 재구성을 가능하게 하지만, 복잡한 신경망 학습과 높은 계산량으로 인해 실시간 처리에 한계가 있다. 이러한 한계를 극복하기 위해 Gaussian Splatting<sup>11</sup>이 제안되었으며, 이는 가우시안 분포를 활용하여 효율적인 실시간 재구성을 가능하게 한다.

이러한 Gaussian Splatting<sup>11</sup>을 SLAM에 적용하여 고품질의 시각적 재구성을 실현하는 연구가 주목받고 있다<sup>13-16</sup>. Gaussian Splatting은 3D 공간에서 가우시안 분포를 활용하여 포인트 클라우드를 표현함으로써, 복잡한 장면에서도 고품질의 시각적 재구성을 실시간으로 수행하는 기법이다.

GS-ICP-SLAM<sup>17</sup>은 Gaussian Splatting과 Generalized Iterative Closest Point (G-ICP)<sup>18</sup>를 결합한 새로운 밀집 표현 SLAM 접근법으로, 실시간으로 고품질의 3D 지도를 생성한다<sup>17</sup>. 이 연구에서는 트래킹과 맵핑 모두에 단일한 가우시안 맵을 사용하여 상호적인 이익을 얻을 수 있다. 이를 통해 G-ICP와 3DGS의 공분산을 공유하고 스케일 정렬 기술을 사용하여 계산 비용을 줄이고 3DGS의 빠른 수렴을 촉진하였다. 실험 결과, 시스템 전체 속도가 최대 107 FPS에 달하며, 재구성된 지도의 품질이 우수함을 보여주었다.

그러나 Gaussian Splatting은 주로 외형과 기하학적 모델링에 집중되어 있어, 객체 수준의 세부적인 장면 이해나 객체 수준의 정보의 통합에는 한계가 있다. 이러한 한계를 극복하기 위해 Gaussian Grouping 기법이 제안되었다<sup>3</sup>.

### 2.3 Gaussian Grouping

Gaussian Grouping<sup>3</sup> 기법은 Gaussian Splatting을 확장하여 개방형 3D 장면에서 객체의 재구성과 분할을 동시에 수행한다.

Gaussian Grouping에서는 각 가우시안에 간결한 식별 인코딩(Identity Encoding)을 추가하여, 3D 장면 내에서 동일한 객체에 속하는 가우시안들을 그룹화할 수 있게 한다. 비용이 많이 드는 3D 라벨링 작업을 대신하여, Segment Anything Model (SAM)<sup>14</sup>이 예측한 2D 마스크를 활용하고, 3D 공간의 일관성 정규화(Spatial Consistency Regularization)를 도입하여 렌더링 과정에서 식별 인코딩을 학습한다.

본 연구에서는 이러한 Gaussian Grouping 기법을 활용하여, segmentation 정보를 통합한 고품질의 3D 지도를 생성하고, 객체 수준의 장면 이해를 향상시킬 수 있음을 확인하였다.

### 3. Gaussian Grouping SLAM

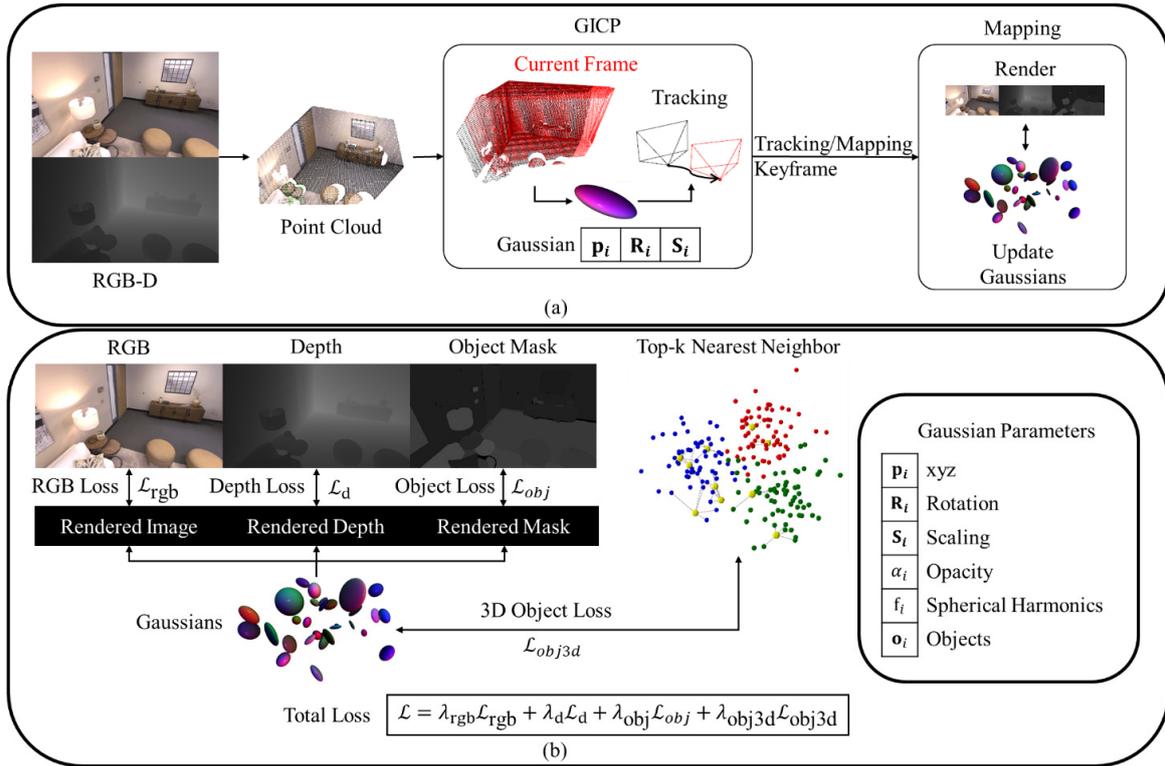
#### 3.1 시스템 아키텍처 개요

본 연구에서는 GS-ICP-SLAM<sup>[17]</sup>을 기반으로 시스템을 설계하였다.

또 기존 GS-ICP-SLAM에 객체의 Gaussian Grouping을 위한 파라미터를 추가하여, 객체 단위의 정보도 함께 학습할 수 있도록 시스템을 확장하였다. 이를 통해 각 가우시안은 위치, 회전, 스케일 등의 기하학적 정보뿐만 아니라, 객체 식별 정보도 포함한다. 이러한 추가 정보를 통해 객체 수준의 정보를 통합한 고품질의 3D 지도를 생성할 수 있다.

전체적인 시스템의 흐름은 [Fig. 1]에서 확인할 수 있다. 시스템은 크게 두 가지 주요 단계로 구성된다:

1. **트래킹 및 가우시안 초기화 단계:** 각 프레임마다 RGB 이미지와 깊이 이미지를 사용하여 포인트 클라우드를 생성하고, G-ICP를 통해 회전 및 스케일을 계산. 트래킹 및 맵핑을 위한 키프레임으로 선정되면, 포인트 클라우드의 위치, 색상, 회전, 스케일 등의 정보를 사용하여 가우시안들을 초기화.
2. **매핑 및 가우시안 최적화 단계:** 가우시안들을 렌더링하여 생성된 이미지와 실제의 RGB, 깊이, 객체 마스크와 비교하여 가우시안 내의 파라미터들을 최적화. 최적화되는 파라미터들은 위치  $\mathbf{p}_i$ , 회전  $\mathbf{R}_i$ , 스케일  $\mathbf{S}_i$ , 불투명도  $\alpha_i$ , 색상 표현  $f_i$  (구면 조화 함수 사용), 객체 식별 정보  $\mathbf{o}_i$  등이 포함. 이러한 아키텍처를 통해, 시스템은 실시간으로 트래킹과 맵핑을 수행하면서도 객체 수준의 정보를 통합하여 고품질의 3D 지도를 생성할 수 있다.



[Fig. 1] Overview of the GG-SLAM System. (a) Tracking and Gaussian Initialization Process: For each frame, a point cloud is generated using RGB and depth images. Rotation and scale are calculated using Generalized Iterative Closest Point (GICP). Gaussian parameters ( $\mathbf{p}_i$ : XYZ location,  $\mathbf{R}_i$ : Rotation,  $\mathbf{S}_i$ : Scaling) are utilized in the GICP process. When a frame is identified as a tracking or mapping keyframe, Gaussians are initialized using the position, color, rotation, and scale of the point cloud. (b) Mapping and Gaussian Optimization Process: The parameters of the Gaussians are optimized by comparing the rendered images of the Gaussians with the ground truth RGB, depth, and object masks. In addition, a Top-k Nearest Neighbor approach is used to optimize the Gaussians in 3D by checking neighborhood consistency. The parameters being optimized include: [ $\mathbf{p}_i$ : XYZ location,  $\mathbf{R}_i$ : Rotation,  $\mathbf{S}_i$ : Scaling,  $\alpha_i$ : Opacity,  $f_i$ : Color (represented using Spherical Harmonics),  $\mathbf{o}_i$ : Objects]. This optimization ensures that the Gaussians accurately capture both the geometric and object-level properties of the scene, enabling high-fidelity reconstruction and object-level understanding within the SLAM system

### 3.2 Gaussian Splatting 기반 SLAM 모듈

#### 3.2.1 카메라 포즈 추정

GS-ICP-SLAM<sup>[17]</sup>에서 제안된 G-ICP 기반 트래킹 방식을 채택하여 카메라의 위치와 자세를 추정하였다.

구체적으로, 각 가우시안의 공분산 행렬  $\mathbf{C}$ 를 특이값 분해하여 스케일과 회전 정보를 추출하고, 가우시안의 기하학적 특성을 보존하면서 스케일을 정규화하는 타원 정규화(ellipse regularization) 방법을 사용하였다. 이는 다음과 같이 표현된다:

$$\mathbf{S}' = \frac{1}{\text{median}(s_0, s_1, s_2)} \cdot \mathbf{S} \quad (1)$$

여기서:

- $\mathbf{S}'$ : 정규화된 스케일 행렬
- $\text{median}(s_0, s_1, s_2)$ : 스케일 요소들의 중앙값

이러한 정규화를 통해 가우시안의 상대적인 형태를 유지하면서도, 스케일을 일관성 있게 맞출 수 있다. 이는 다양한 형태의 가우시안들이 존재하는 복잡한 장면에서도 효과적인 트래킹을 가능하게 한다.

또한, 동적 키프레임 선택 방식을 사용하여, G-ICP 과정에서 계산된 현재 프레임과 맵 사이의 기하학적 대응성을 활용하였다<sup>[17]</sup>.

#### 3.2.2 Gaussian Splatting을 이용한 3D 맵 생성

본 연구에서는 맵핑 전용 키프레임 선택 방식<sup>[17]</sup>을 채택하여, 트래킹 성능의 저하 없이 맵핑 품질을 향상시켰다. 트래킹은 기존의 키프레임을 사용하여 안정적으로 수행하고, 맵핑은 추가로 선택된 맵핑 전용 키프레임을 활용한다. 이때, 맵핑 전용 키프레임은 트래킹 과정에 영향을 주지 않으면서도 맵의 정밀도를 높인다.

실시간으로 동작하는 시스템에서 제한된 관측 데이터로 인해 맵핑 시 국소 최적해(local minima)에 빠질 수 있는 문제가 발생할 수 있다. 이를 해결하기 위해 다음과 같은 방법을 사용한다:

1. 무작위 키프레임 선택: 학습 단계마다 지금까지 선택된 키프레임 중 하나를 무작위로 선택하여 학습에 사용한다.
2. 가우시안 프루닝(pruning): 학습 과정에서 국소 최적해에 빠진 가우시안들을 제거하여 맵핑 품질을 향상시킨다.

이러한 접근법을 통해 맵핑 품질과 시스템의 안정성을 향상시킬 수 있다.

### 3.3 사전 정의된 마스크를 활용한 객체 수준 정보 통합

객체 마스크 생성을 위해 Tracking Anything<sup>[5]</sup>모델의 출력물을 활용하였다. 이 모델은 다양한 비디오 세그멘테이션 작업에



[Fig. 2] Object Masks Generated by Tracking Anything. (Left) Original RGB image. (Right) Generated masks (colored with RGB values for visualization)

서 범용적으로 사용될 수 있는 접근 방식을 제공하며, 이미지 수준의 세그멘테이션 결과를 시간적으로 전파하여 일관된 객체 마스크를 생성한다. [Fig. 2]에서 생성된 마스크를 확인할 수 있으며, 실제 구현에서는 객체마다 ID가 부여된 그레이스케일 이미지를 사용한다.

### 3.4 Gaussian Grouping을 통한 객체 구분

#### 3.4.1 Gaussian Grouping

Gaussian Splatting은 3D 씬의 고품질 및 실시간 새로운 시점 합성을 가능하게 하지만, 세밀한 객체 수준의 장면 이해가 부족하다는 한계가 있다. 이를 해결하기 위해 본 논문에서는 Gaussian Grouping 기법을 도입하였다. 주요 특징은 다음과 같다:

1. Identity Encoding을 통한 객체 그룹핑
2. 2D 마스크 예측과 3D 공간 일관성 정규화
3. 고해상도 및 효율적인 3D 세그멘테이션
4. Local Gaussian Editing

Gaussian Grouping은 이러한 특징들을 통해 높은 시각적 품질과 세밀한 장면 이해를 제공하며, 다양한 3D 비디오 세그멘테이션 작업에 유연하게 적용할 수 있다.

#### 3.4.2 객체 구분을 위한 데이터 처리 및 클러스터링

Gaussian Grouping은 3D 씬에서 객체를 재구성하고 세그멘테이션할 수 있도록 설계된 방법으로, 객체 인스턴스 소속에 따라 Gaussian을 그룹화하여 세밀한 객체 수준의 장면 이해를 가능하게 한다.

초기 Gaussian 맵 생성을 위해, 포인트 클라우드(point cloud) 데이터, 색상 정보, 회전 및 스케일링 파라미터, 깊이 값(z-values), 그리고 추적 가능한 인덱스(trackable\_idx)를 입력 데이터로 사용한다. 이 과정은 여러 세부 단계로 나뉘며, 각 Gaussian의 위치와 색상을 정확하게 정의하고 객체 식별을 위한 초기 설정을 포함한다.

먼저, 포인트 클라우드 데이터와 색상 정보를 통합하여 각 Gaussian의 위치와 색상을 정의한다.

각 Gaussian  $i$ 에 대해 RGB 색상 벡터  $\mathbf{c}_i = [R_i, G_i, B_i]^T$ 가 주어진다. 이 RGB 색상은 Spherical Harmonics (SH) 계수로 변환되어 Gaussian의 색상 특성을 나타내는 벡터로 변환된다.

다음으로, 각 Gaussian에 고유한 Identity Encoding을 부여한다. Identity Encoding은 객체 인스턴스 소속을 나타내는 16차원의 학습 가능한 벡터로 정의된다. 이를 통해 동일한 객체에 속한 Gaussian들은 유사한  $\mathbf{e}_i$  값을 가지게 되며, 이는 객체의 개별 식별과 관리에 용이하게 한다.

객체 ID 초기화 단계에서는 각 Gaussian에 대한 객체 ID가 무작위로 생성된 RGB 값을 기반으로 설정된다. 이 RGB 값은 Spherical Harmonics 계수로 변환되어 초기 Identity Encoding 벡터  $\mathbf{o}_i$ 를 형성한다. 초기화 과정은 다음과 같이 수식으로 표현된다:

$$\mathbf{o}_i = \frac{\mathbf{c}_{\text{rand},i} - 0.5}{c_0} \quad (2)$$

$$\mathbf{o}_i \in \mathbb{R}^{16} \quad (3)$$

여기서  $\mathbf{c}_{\text{rand},i} \in \mathbb{R}^3$ 는 Gaussian  $i$ 에 대한 무작위 RGB 값을 생성하는 벡터이다.  $\mathbf{o}_i$ 는 Gaussian  $i$ 에 할당된 초기 Identity Encoding 벡터로, 학습을 통해 최적화된다.

또한, 위치, 스케일, 회전, 불투명도(opacity) 등의 파라미터를 초기화한다.

Identity Encoding 최적화 및 객체 분류 단계에서는 Gaussian Grouping의 클러스터링을 위해 객체 구분을 위한 분류기(classifier)와 손실 함수(loss functions)를 설정한다. 이 과정은 다음과 같은 단계로 이루어진다.

먼저, 객체 구분을 위해 100개의 클래스를 분류할 수 있는 2D 컨볼루션 층을 정의하고, 크로스 엔트로피 손실 함수를 사용하여 분류 성능을 최적화한다. 분류기는 각 Gaussian이 속한 클래스(객체)를 예측하는 역할을 한다.

$$\mathbf{y}_i = \text{Classifier}(\mathbf{e}_i) \quad (4)$$

여기서  $\mathbf{y}_i \in \mathbb{R}^{100}$ 은 Gaussian  $i$ 에 대한 클래스 확률 분포이다.

모델의 학습을 최적화하기 위해 복합 손실 함수(composite loss function)를 사용하였으며, 이는 객체 분류, RGB 이미지 재구성, 깊이 이미지 재구성, 그리고 3D 객체 일관성을 포함한다.

모델의 전체 손실 함수  $\mathcal{L}$ 는 다음과 같이 정의된다:

$$\mathcal{L} = \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \lambda_d\mathcal{L}_d + \lambda_{\text{obj}}\mathcal{L}_{\text{obj}} + \lambda_{\text{obj3d}}\mathcal{L}_{\text{obj3d}} \quad (5)$$

여기서,  $\mathcal{L}_{\text{rgb}}$ 는 RGB 이미지 재구성 손실,  $\mathcal{L}_d$ 는 깊이 이미지 재구성 손실,  $\mathcal{L}_{\text{obj}}$ 는 객체 분류 손실,  $\mathcal{L}_{\text{obj3d}}$ 는 3D 객체 일관성 손실 그리고  $\lambda_{\text{rgb}}$ ,  $\lambda_d$ ,  $\lambda_{\text{obj}}$ ,  $\lambda_{\text{obj3d}}$ 들은 각 손실 구성 요소의 가중치 계수이다.

RGB 손실은 L1 손실과 SSIM 손실의 가중 합으로 정의된다. 이는 이미지의 픽셀 단위 차이와 구조적 유사성을 동시에 고려하기 위함이다.

$$\mathcal{L}_{\text{rgb}} = (1 - \lambda_{\text{dssim}})\mathcal{L}_{\text{L1}} + \lambda_{\text{dssim}}(1 - \mathcal{L}_{\text{SSIM}}) \quad (6)$$

여기서  $\lambda_{\text{dssim}}$ 는 SSIM 손실의 가중치 계수,  $\mathcal{L}_{\text{L1}}$ 는 L1 손실 그리고  $\mathcal{L}_{\text{SSIM}}$ 는 SSIM 손실을 의미한다.

깊이 손실은 예측된 깊이 이미지  $\hat{D}$ 와 실제 깊이 이미지  $D$  간의 L1 손실로 정의된다. 깊이 값의 범위 조절을 위해 스케일링 상수  $d_{\text{max}}$ 를 사용한다.

$$\mathcal{L}_d = \frac{1}{N} \sum_{i=1}^N \left| \frac{D_i}{d_{\text{max}}} - \frac{\hat{D}_i}{d_{\text{max}}} \right| \quad (7)$$

여기서  $N$ 은 깊이 이미지의 전체 픽셀 수이다.

객체 분류 손실은 각 Gaussian의 Identity Encoding  $\mathbf{e}_i$ 를 분류기(classifier)에 입력하여 얻은 로짓(logits)과 실제 객체 레이블  $\mathbf{y}_i$  간의 크로스 엔트로피 손실로 정의된다.

$$\mathcal{L}_{\text{obj}} = \frac{1}{N} \sum_{i=1}^N \text{CrossEntropy}(\mathbf{z}_i, \mathbf{y}_i) \quad (8)$$

여기서  $\mathbf{z}_i$ 는  $\mathbf{z}_i = \mathbf{W} \cdot \mathbf{e}_i + \mathbf{b}$ 이며, CrossEntropy는 크로스 엔트로피 손실 함수 그리고  $N$ 은 클래스 수이다. 분류기는 1x1 Conv2D 층으로 구현되며, 각 Gaussian의 Identity Encoding을 클래스 로짓으로 변환한다.

3D 객체 일관성 손실은 Gaussian의 위치 정보와 클래스 확률 분포 간의 공간적 일관성을 유지하기 위해 사용된다. 이는 주기적으로 계산되며, Kullback-Leibler (KL) 다이버전스를 기반으로 한다.

$$\mathcal{L}_{\text{obj3d}} = \frac{\lambda_{\text{val}}}{C} \sum_{j=1}^C \text{KL}(P_j \parallel Q_j) \quad (9)$$

여기서  $P_j$ 는 샘플링된 Gaussian  $j$ 의 클래스 확률 분포,  $Q_j = \{\mathbf{e}'_{j1}, \mathbf{e}'_{j2}, \dots, \mathbf{e}'_{jk}\}$ 는 Gaussian  $j$ 의  $k$ 개의 최근접 이웃의 클래스 확률 분포 그리고  $\text{KL}(P_j \parallel Q_j)$ 는 KL 다이버전스를 의미한다.

$$\mathcal{L}_{\text{obj3d}} = \frac{\lambda_{\text{val}}}{C} \sum_{j=1}^C \sum_{k=1}^K P_{jk} \log \left( \frac{P_{jk}}{Q_{jk}} \right) \quad (10)$$

$P_{jk}$ 는 샘플링된 Gaussian  $j$ 의 클래스  $k$ 에 대한 확률,  $Q_{jk}$ 는 Gaussian  $j$ 의 클래스  $k$ 번째 이웃 Gaussian의 클래스  $k$ 에 대한 확률,  $\lambda_{val}$ 은 손실 가중치 계수 그리고  $C$ 는 클래스 수를 의미한다. 모델의 전체 손실 함수는 식 (5)와 같다.

## 4. 실험 및 결과

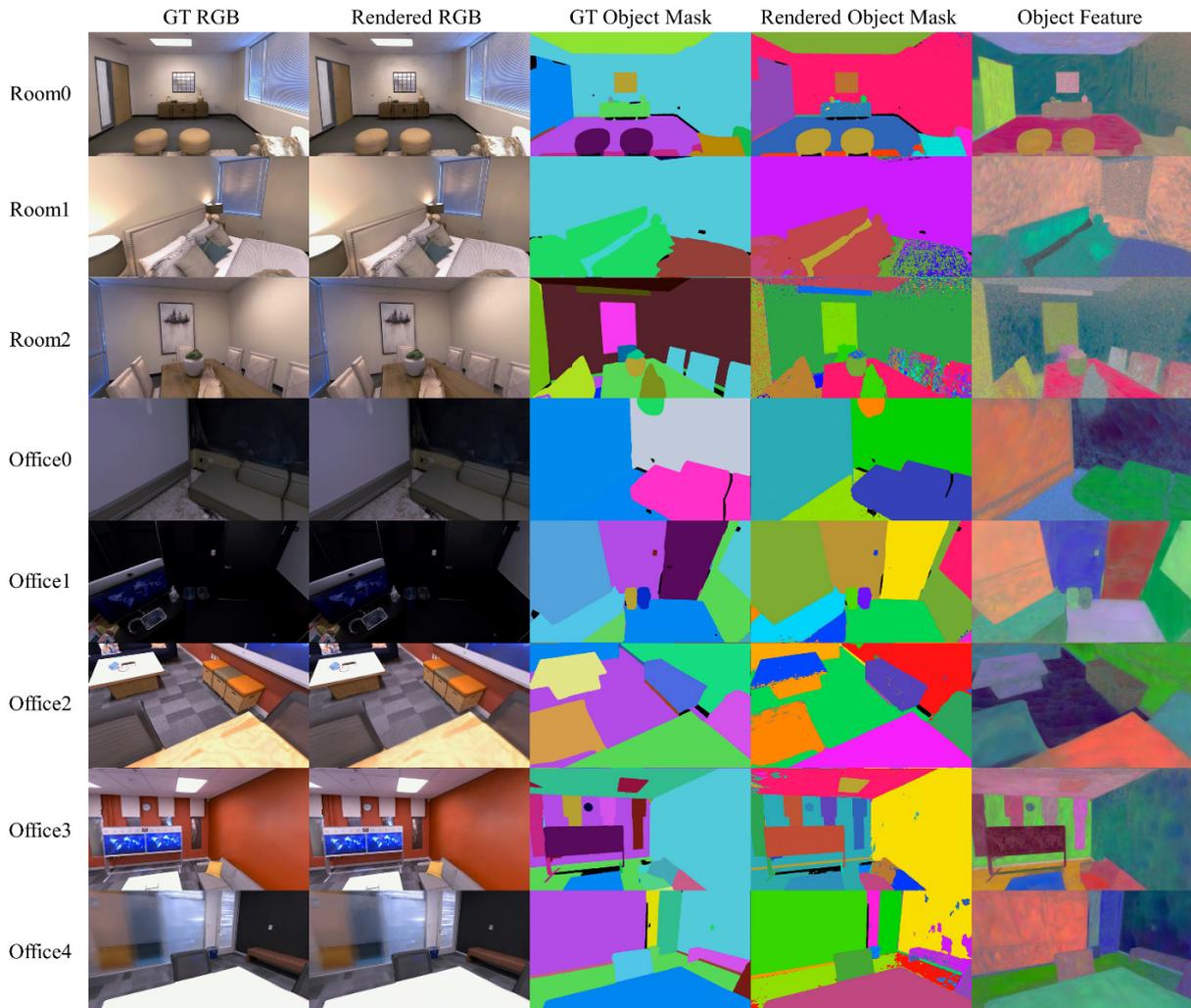
### 4.1 실험 설정

본 연구에서는 Gaussian Grouping 기반의 SLAM 시스템을 평가하기 위해 Replica 데이터셋<sup>[14]</sup>을 사용하였다. 본 연구에서는 [5]에서 제안된 Tracking Anything Model (TAM)을 활용하여 RGB 이미지를 기반으로 사전에 객체 마스크를 생성하였다.

### 4.2 평가 지표

실험은 NVIDIA RTX A6000 GPU를 기반으로 수행되었다. 분류기는 클래스 수를 100으로 설정하고, Gaussian 수에 맞춰 입력 채널 수를 조정한 1x1 Conv2D 층으로 구현하였다. 분류기의 손실 함수로는 크로스 엔트로피 손실(Cross Entropy Loss)을 사용하였다. 분류기 학습을 위해 Adam 옵티마이저를 사용하였고, 학습률은  $5 \times 10^{-4}$ 로 설정하였다. 손실 가중치는 RGB 손실에 대해  $\lambda_{rgb} = 1.0$ , 깊이 손실에 대해  $\lambda_d = 0.1$ , 객체 분류 손실에 대해  $\lambda_{obj} = 0.1$ , 그리고 3D 객체 일관성 손실에 대해  $\lambda_{obj3d} = 1.0$ 으로 조정하였다.

평가 지표로는 이미지 재구성 품질을 평가하기 위한 Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure



[Fig. 3] Ground Truth and Results on the Replica Dataset. Rendered RGB images using the original Gaussian Splatting method. Rendered Object Masks with Gaussian Grouping parameters. Object Feature images obtained by applying PCA to 16-dimensional vectors and mapping them to 3-channel RGB images. Note that mask quality is influenced by the ground truth and the Tracking Anything model, which may not perform well when frames undergo significant changes, resulting in some masks containing noise or dots

(SSIM), Learned Perceptual Image Patch Similarity (LPIPS)를 사용하였다.

#### 4.3 비교 실험

본 비교 실험 결과는 Gaussian Grouping 기법이 SLAM 시스템에 객체 수준 정보를 효과적으로 통합하면서도, 기존의 성능을 유지하거나 향상시킬 수 있음을 입증하였다. 정성적 결과는 [Fig. 3]에서 확인할 수 있다.

맵핑 정확도 실험 결과는 [Table 1]과 같다.

제안한 Gaussian Grouping 기반 SLAM 시스템의 성능을 기존의 Orbeez-SLAM<sup>[20]</sup>, Point-SLAM<sup>[21]</sup>, GS-SLAM<sup>[22]</sup>, SplaTAM<sup>[13]</sup>, GS-ICP-SLAM<sup>[17]</sup>과 비교하였다. 비교 대상인 이들 알고리즘은 Gaussian Splatting을 활용한 다양한 SLAM 시스템으로, 본

연구에서는 GS-ICP-SLAM의 성능 지표를 그대로 사용하였다. 본 연구에서 사용한 Tracking 결과는 GS-ICP-SLAM과 유사한 Root Mean Square Error (RMSE)를 보이므로 실험을 진행하지 않았다. 따라서, 기존 방법과 유사한 성능을 유지하면서도, 추가적으로 객체 수준 정보를 통합한 Gaussian Grouping 기법의 효과를 중점적으로 평가하였다.

[Table 2]에서는 기존 Semantic Segmentation을 기반으로 객체 정보가 결합된 SLAM 방법들과의 비교 실험을 수행하였다. 비교 결과, 제안한 Gaussian Grouping 기반 SLAM 시스템은 Semantic Segmentation 기반 SLAM 방법들과 유사하거나 더 우수한 정확도를 보였다.

추가적으로, [Table 3]에서는 초당 프레임 수(FPS)를 10~30으로 제한하여 정확도에 대한 실험을 수행하였다. 특히, 10FPS 일 때 제안한 Gaussian Grouping 기반 SLAM 시스템은 원본

[Table 1] Mapping Accuracy and Processing Speed on the Replica Dataset. All performance metrics for Orbeez-SLAM, Point-SLAM, SplaTAM, and GS-ICP-SLAM are extracted from the GS-ICP-SLAM paper<sup>[17]</sup>. The proposed Gaussian Grouping-based SLAM system effectively integrates additional object parameters without significantly compromising rendering quality, maintaining processing speed comparable to GS-ICP-SLAM while achieving mapping accuracy that is on par with or exceeds other state-of-the-art SLAM methods. **Bold** values indicate the best performance, and underlined values indicate the second best

| Methods  | Metrics | Room0        | Room1        | Room2        | Office0      | Office1      | Office2      | Office3      | Office4      |
|--|---------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Orbeez-SLAM <sup>[20]</sup>                        | PSNR ↑  | 12.13        | 15.28        | 15.87        | 17.59        | 19.26        | 10.30        | 11.55        | 12.65        |
|  | SSIM ↑  | 0.699        | 0.787        | 0.808        | 0.795        | 0.672        | 0.783        | 0.760        | 0.838        |
|  | LPIPS ↓ | 0.612        | 0.477        | 0.509        | 0.515        | 0.413        | 0.554        | 0.540        | 0.528        |
|  | FPS ↑   | 24.05        | 24.78        | 24.00        | 19.22        | 26.74        | 23.88        | 24.45        | 26.11        |
| Point-SLAM <sup>[21]</sup>                         | PSNR ↑  | <u>33.38</u> | 34.10        | <u>36.32</u> | 38.72        | 39.31        | <u>34.22</u> | 34.10        | 34.82        |
|  | SSIM ↑  | <b>0.979</b> | <b>0.977</b> | <b>0.985</b> | 0.985        | <u>0.987</u> | 0.962        | 0.963        | <b>0.981</b> |
|  | LPIPS ↓ | 0.097        | 0.115        | 0.101        | 0.089        | 0.110        | 0.152        | 0.119        | 0.131        |
|  | FPS ↑   | 0.26         | 0.30         | 0.31         | 0.33         | 0.34         | 0.30         | 0.28         | 0.30         |
| GS-SLAM <sup>[22]</sup>                            | PSNR ↑  | 31.56        | 32.86        | 32.59        | 38.70        | <u>41.17</u> | 32.36        | 32.03        | 32.92        |
|  | SSIM ↑  | 0.968        | 0.973        | 0.971        | <b>0.986</b> | <b>0.993</b> | <b>0.978</b> | <b>0.970</b> | 0.968        |
|  | LPIPS ↓ | 0.094        | 0.075        | 0.093        | 0.050        | <u>0.033</u> | 0.094        | 0.110        | 0.112        |
|  | FPS ↑   | 8.34         | -            | -            | -            | -            | -            | -            | -            |
| SplaTAM <sup>[13]</sup>                            | PSNR ↑  | 32.60        | 33.55        | 34.83        | 38.09        | 39.02        | 31.95        | 29.53        | 31.55        |
|  | SSIM ↑  | <u>0.975</u> | 0.969        | <u>0.982</u> | <u>0.982</u> | 0.982        | 0.966        | 0.949        | 0.951        |
|  | LPIPS ↓ | <u>0.070</u> | 0.097        | <u>0.074</u> | 0.088        | 0.093        | 0.098        | 0.119        | 0.150        |
|  | FPS ↑   | 0.24         | 0.19         | 0.19         | 0.20         | 0.22         | 0.27         | 0.26         | 0.24         |
| GS-ICP-SLAM <sup>[17]</sup><br>(limited to 30 FPS) | PSNR ↑  | <b>35.37</b> | <b>37.80</b> | <b>38.50</b> | <b>43.13</b> | <b>43.26</b> | <b>36.93</b> | <b>36.90</b> | <b>38.75</b> |
|  | SSIM ↑  | 0.963        | <u>0.971</u> | 0.975        | <b>0.986</b> | 0.985        | <u>0.974</u> | <u>0.969</u> | <u>0.973</u> |
|  | LPIPS ↓ | <b>0.048</b> | <b>0.045</b> | <b>0.048</b> | <b>0.026</b> | <b>0.029</b> | <b>0.043</b> | <b>0.042</b> | <b>0.045</b> |
|  | FPS ↑   | 29.97        | 29.98        | 29.98        | 29.98        | 29.99        | 29.97        | 29.97        | 29.97        |
| Ours<br>(limited to 30 FPS)                        | PSNR ↑  | 32.77        | <u>35.24</u> | 34.89        | <u>40.52</u> | 41.00        | 32.36        | <u>34.42</u> | <u>36.35</u> |
|  | SSIM ↑  | 0.941        | 0.957        | 0.958        | 0.978        | 0.976        | 0.948        | 0.954        | 0.961        |
|  | LPIPS ↓ | 0.081        | <u>0.074</u> | 0.083        | <u>0.047</u> | 0.060        | <u>0.091</u> | <u>0.067</u> | <u>0.070</u> |
|  | FPS ↑   | 29.96        | 29.97        | 29.98        | 29.97        | 29.97        | 29.98        | 29.95        | 29.97        |

GS-ICP-SLAM과 유사한 정확도를 유지하는 모습을 보였다.

#### 4.4 결과 분석

본 연구에서 제안한 Gaussian Grouping 기반 SLAM 시스템은 Replica 데이터셋의 시나리오를 통해 기존의 주요 SLAM 알고리즘들과 비교 실험을 수행하였다. [Table 1]에서는 제안 시스템이 타 시스템과 비교하여 매핑 정확도와 처리 속도 측면에서 우수하거나 유사한 성능을 보였음을 확인할 수 있었다. 특히, Gaussian Grouping 기법을 도입함으로써 객체 파라미터를 효과적으로 추가하면서도 GS-ICP-SLAM과 유사한 처리 속도를 유지하였으며, 매핑 정확도는 다른 SLAM 시스템들과 유사하거나 더 높은 수준을 달성하였다. 이는 제안된 기법이 SLAM

[Table 2] Mapping Accuracy and Processing Speed on the Replica Dataset. Performance metrics for SGS-SLAM and SemGauss-SLAM are extracted from their respective studies [23,24]. Although frame rate (FPS) comparisons are limited due to the use of predefined object masks, the FPS is capped at 30, allowing for the integration of additional semantic or object-level information

| Methods                       | Metrics | Average |
|-------------------------------|---------|---------|
| SGS-SLAM <sup>[23]</sup>      | PSNR ↑  | 34.66   |
|                               | SSIM ↑  | 0.973   |
|                               | LPIPS ↓ | 0.096   |
|                               | FPS ↑   | 2.11    |
| SemGauss-SLAM <sup>[24]</sup> | PSNR ↑  | 35.03   |
|                               | SSIM ↑  | 0.982   |
|                               | LPIPS ↓ | 0.062   |
|                               | FPS ↑   | -       |
| Ours<br>(limited to 30 FPS)   | PSNR ↑  | 35.94   |
|                               | SSIM ↑  | 0.959   |
|                               | LPIPS ↓ | 0.072   |
|                               | FPS ↑   | 29.97   |

[Table 3] Mapping Accuracy with Changing FPS on the Replica Dataset

| Methods           | Metrics | Room0 | Room1 | Room2 | Office0 | Office1 | Office2 | Office3 | Office4 |
|-------------------|---------|-------|-------|-------|---------|---------|---------|---------|---------|
| limited to 30 FPS | PSNR ↑  | 32.77 | 35.24 | 34.89 | 40.52   | 41.00   | 32.36   | 34.42   | 36.35   |
|                   | SSIM ↑  | 0.941 | 0.957 | 0.958 | 0.978   | 0.976   | 0.948   | 0.954   | 0.961   |
|                   | LPIPS ↓ | 0.081 | 0.074 | 0.083 | 0.047   | 0.060   | 0.091   | 0.067   | 0.070   |
| limited to 20 FPS | PSNR ↑  | 33.74 | 36.21 | 36.26 | 41.49   | 41.55   | 33.67   | 35.37   | 37.31   |
|                   | SSIM ↑  | 0.951 | 0.963 | 0.964 | 0.981   | 0.979   | 0.953   | 0.960   | 0.966   |
|                   | LPIPS ↓ | 0.067 | 0.064 | 0.071 | 0.039   | 0.051   | 0.077   | 0.059   | 0.063   |
| limited to 10 FPS | PSNR ↑  | 35.04 | 37.37 | 37.84 | 42.24   | 42.63   | 34.43   | 36.60   | 38.21   |
|                   | SSIM ↑  | 0.959 | 0.969 | 0.972 | 0.983   | 0.982   | 0.958   | 0.967   | 0.970   |
|                   | LPIPS ↓ | 0.055 | 0.052 | 0.058 | 0.032   | 0.039   | 0.068   | 0.047   | 0.053   |

시스템의 전반적인 성능을 유지시키며 객체 수준 정보 통합을 할 수 있음을 시사한다.

[Table 2]에서는 Semantic Segmentation을 SLAM 방법들과의 비교 실험 결과를 보여준다. 제안한 Gaussian Grouping 기반 SLAM 시스템은 Semantic Segmentation 기반 방법들과 비교하여 정확도가 유사하거나 더 우수하였다. 본 연구에서 마스크들이 사전 정의되었다는 점이 속도 면에서 정확한 비교가 힘들게 하지만, 처리 속도가 매우 빠르고 기존 방법에서 사용되는 Semantic Segmentation이 훈련 데이터에 크게 의존한다는 한계를 나타내므로, 객체 수준 정보를 통합할 경우 처리 속도와 일반화 성능에서 좋은 결과를 기대할 수 있는 가능성을 보여주었다.

[Table 3]에서는 다양한 FPS 제한 하에서의 매핑 정확도를 평가하였다. 10FPS에서는 제안한 시스템이 원본 GS-ICP-SLAM과 유사한 정확도를 유지함을 확인할 수 있었으며, 이는 Gaussian Grouping 기법이 SLAM 시스템의 처리 속도에 크게 영향을 미치지 않으면서도 객체 파라미터를 효과적으로 통합할 수 있음을 보여준다. 이러한 결과는 다양한 실내 환경에서의 SLAM 응용 가능성을 더욱 높이는 데 기여하였다.

종합적으로, 제안한 Gaussian Grouping 기반 SLAM 시스템은 객체 수준 정보의 통합을 통해 기존 SLAM 시스템의 성능을 유지하거나 향상시킬 수 있음을 입증하였다. 특히, 높은 매핑 정확도와 효율적인 처리 속도를 동시에 달성함으로써, 본 연구의 주요 기여인 Gaussian Grouping 기법의 유효성을 명확히 확인할 수 있었다. 향후 연구에서는 객체 정보 처리 속도 향상과 일반화 성능 강화를 통해 더욱 향상된 SLAM 시스템을 구현할 수 있을 것으로 기대된다.

## 5. 결론 및 향후 연구

본 연구에서는 Gaussian Grouping 기반의 SLAM 시스템을 제안하여, 기존의 주요 SLAM 알고리즘들과 비교 실험을 수행하였다. 제안된 시스템은 객체 수준 정보의 통합을 통해 기존

SLAM 시스템의 매핑 정확도를 유지하거나 향상시키면서도, 처리 속도를 크게 저하시키지 않는 성능을 보였다. 특히, Gaussian Grouping 기법을 도입함으로써 객체 파라미터를 효과적으로 추가할 수 있었다.

그러나 몇 가지 한계점이 존재한다. 객체 마스크가 사전에 정의되어 있다. 이는 객체 마스크의 품질이 Tracking Anything 모델의 성능에 크게 의존함을 의미하며, 모델이 정확하게 추적하지 못해 일부 마스크에 노이즈나 점(dot)이 포함되는 문제가 발생하였다.

향후 연구에서는 이러한 한계점을 극복하기 위한 다양한 방안을 모색할 예정이다. 첫째, 객체 추적을 효과적으로 처리할 수 있는 SLAM 시스템을 개발하여, 실내 환경에서의 객체 변화에 유연하게 대응할 수 있도록 할 것이다. 둘째, 실시간 추적 성능을 향상시키기 위해 최적화된 환경과 알고리즘을 도입하여, 보다 빠르고 정확한 SLAM 처리가 가능하도록 할 것이다. 셋째, 더 나은 객체 분할 방법을 연구하여, 마스크의 정확도와 안정성을 높이고, 다양한 환경에서의 일반화 성능을 강화할 것이다.

또한, 객체 제거(Object Removal) 및 인페인팅(Inpainting) 기법을 SLAM 시스템에 통합함으로써, 불필요한 객체나 노이즈를 효과적으로 제거하고, 보다 깨끗하고 정확한 맵을 생성할 수 있는 방안을 탐구할 것이다. 이를 통해, 사전에 생성된 마스크의 한계를 극복할 수 있을 것으로 기대된다. 이러한 연구는 SLAM 시스템의 전반적인 성능을 향상시키고, 다양한 실내 환경에서의 응용 가능성을 더욱 넓히는 데 기여할 것이다.

종합적으로, 본 연구는 Gaussian Grouping 기법을 활용한 SLAM 시스템의 가능성을 입증하였으며, 향후 연구를 통해 더욱 발전된 SLAM 기술을 구현할 수 있는 기반을 마련하였다. 이러한 연구는 실내 환경에서의 정밀한 매핑과 추적, 객체 수준 이해를 요구하는 다양한 응용 분야에서 큰 잠재력을 가지고 있다.

## References

- [1] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1-4, Jul., 2023, DOI: 10.1145/3592433.
- [2] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99-106, Dec., 2021, DOI: 10.1145/3503250.
- [3] M. Ye, M. Danelljan, F. Yu, and L. Ke, "Gaussian grouping: Segment and edit anything in 3d scenes," *arXiv:2312.0073*, 2023, DOI: 10.48550/arXiv.2312.00732.
- [4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, pp. 3992-4003, 2023, DOI: 10.1109/ICCV51070.2023.00371.
- [5] H. K. Cheng, S. W. Oh, B. Price, A. Schwing, and J.-Y. Lee, "Tracking anything with decoupled video segmentation," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, pp. 1316-1326, 2023, DOI: 10.1109/ICCV51070.2023.00127.
- [6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Mono-SLAM: Real-time single camera SLAM," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052-1067, Jun., 2007, DOI: 10.1109/TPAMI.2007.1049.
- [7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147-1163, Oct., 2015, DOI: 10.1109/TRO.2015.2463671.
- [8] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255-1262, Oct., 2017, DOI: 10.1109/TRO.2017.2705103.
- [9] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," *2017 IEEE international conference on robotics and automation (ICRA)*, Singapore, pp. 2043-2050, 2017, DOI: 10.1109/ICRA.2017.7989236.
- [10] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Madrid, Spain, pp. 1168-1174, 2018, DOI: 10.1109/IROS.2018.8593691.
- [11] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," *2017 IEEE International Conference on Robotics and automation (ICRA)*, Singapore, pp. 4628-4635, 2017, DOI: 10.1109/ICRA.2017.7989538.
- [12] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Munich, Germany, pp. 10-20, 2018, DOI: 10.1109/ISMAR.2018.00024.
- [13] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "SplaTAM: Splat Track & Map 3D Gaussians for Dense RGB-D SLAM," *arXiv:2312.02126*, 2024, DOI: 10.48550/arXiv.2312.02126.
- [14] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian splatting slam," *arXiv:2312.06741*, 2024, DOI: 10.48550/arXiv.2312.06741.
- [15] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-slam: Photo-realistic dense slam with gaussian splatting," *arXiv:2312.10070*, 2023, DOI: 10.48550/arXiv.2312.10070.
- [16] H. Huang, L. Li, H. Cheng, and S.-K. Yeung, "Photo-SLAM: Real-time Simultaneous Localization and Photorealistic Mapping for Monocular Stereo and RGB-D Cameras," *arXiv:2311.16728*, 2024, DOI: 10.48550/arXiv.2311.16728.

- [17] S. Ha, J. Yeon, and H. Yu, "Rgbd gs-icp slam," *arXiv:2403.12550*, 2024, DOI: 10.48550/arXiv.2403.12550.
- [18] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," *Robotics: science and systems 2009*, 2009, DOI: 10.15607/RSS.2009.V.021.
- [19] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. De Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, "The Replica dataset: A digital replica of indoor spaces," *arXiv:1906.05797*, 2019, DOI: 10.48550/arXiv.1906.05797.
- [20] C.-M. Chung, Y.-C. Tseng, Y.-C. Hsu, X.-Q. Shi, Y.-H. Hua, J.-F. Yeh, W.-C. Chen, Y.-T. Chen, and W. H. Hsu, "Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, pp. 9400-9406, 2023, DOI: 10.1109/ICRA48891.2023.10160950.
- [21] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-slam: Dense neural point cloud-based slam," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, pp. 18387-18398, 2023, DOI: 10.1109/ICCV51070.2023.01690.
- [22] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," *arXiv:2311.11700*, 2024, DOI: 10.48550/arXiv.2311.11700.
- [23] M. Li, S. Liu, H. Zhou, G. Zhu, N. Cheng, T. Deng, and H. Wang, "Sgs-slam: Semantic gaussian splatting for neural dense slam," *arXiv:2402.03246*, 2024, DOI: 10.48550/arXiv.2402.03246.
- [24] S. Zhu, R. Qin, G. Wang, J. Liu, and H. Wang, "Semgauss-slam: Dense semantic gaussian splatting slam," *arXiv:2403.07494*, 2024, DOI: 10.48550/arXiv.2403.07494.



### 정 현 도

2023 충북대학교 전자공학과(학사)  
2023~현재 충북대학교 지능로봇공학과  
(석사)

관심분야: SLAM, Deep Learning, Computer Vision



### 김 곤 우

2006 서울대학교 전기컴퓨터공학(박사)  
2006~2008 한국생산기술연구원 로봇기술  
본부 선임연구원  
2008~2012 원광대학교 조교수  
2012~현재 충북대학교 교수

관심분야: Artificial Intelligence, SLAM, Mobile Robot Navigation