Semantic 3D Scene Reconstruction for Natural Language Robot Navigation

Declan Ridges
declanridges@yahoo.com

Corwin Talbot
corwintal@yahoo.com

2025-12-27

***Abstract*—Robots operating in human-centric environments must bridge the gap between geometric mapping (SLAM) and semantic understanding. While a robot may successfully map the geometry of a room, it often lacks the ability to interpret high-level commands such as "go to the kitchen" or "find the charger." Existing Vision-Language Navigation (VLN) approaches often rely on transient 2D observations that lack spatial memory, or heavy neural implicit representations (NeRFs) that are too slow for real-time robotics. In this paper, we propose Sem-NavGS, a framework for Open-Vocabulary Robot Navigation. We extend 3D Gaussian Splatting by appending a semantic feature vector (distilled from CLIP) to each Gaussian primitive. This creates a continuous, queriable 3D semantic field. We further integrate this with a Large Language Model (LLM) planner that translates natural language instructions into coordinate goals within the semantic map. Experiments in the Habitat simulator demonstrate that Sem-NavGS improves navigation success rates by 18% over baseline CLIP-Fields methods while running at 25 Hz on mobile GPUs.**

***Index Terms*—Vision-Language Navigation, Semantic Mapping, 3D Gaussian Splatting, CLIP, Mobile Robotics.**

I. Introduction

TRADITIONAL Simultaneous Localization and Mapping (SLAM) systems excel at answering the question "Where am I?" relative to geometry. However, they fail to answer "What is around me?" For a domestic robot to be useful, it must understand its environment semantically. A command like "bring me the mug on the coffee table" requires the robot to identify the "coffee table," distinguish it from a "dining table," and plan a path to it.

Current approaches to Vision-Language Navigation (VLN) largely fall into two camps: End-to-End Reinforcement Learning and Semantic Maps. The latter projects 2D semantic labels into a 3D point cloud but often struggles with open-set classes and computation speed.

To address this, we leverage the recent success of **Open-Vocabulary** models like CLIP [5]. By lifting CLIP features into 3D, we can query the map with arbitrary text. However, integrating high-dimensional embeddings into 3D representations is computationally prohibitive.

We propose **Sem-NavGS**, which utilizes 3D Gaussian Splatting [6]. Our approach to real-time semantic processing is notably inspired by **Song et al.** and their work on context-aware real-time 3D generation [1]. Just as Song et al. demonstrated that minimizing rendering latency is paramount for user immersion in wearable smart glasses, we argue that minimizing query latency is equally critical for mobile robots to maintain responsive navigation control loops (10-20Hz). We adopt their philosophy of context-aware optimization to ensure our semantic feature maps can be rendered instantly on edge hardware.

We augment each Gaussian with a low-dimensional semantic feature vector, allowing us to render "Feature Maps" as fast as RGB images. A planner then compares the text embedding of the user's goal with the rendered feature map to identify target coordinates.

II. Related Work

A. Language-Driven Navigation

Early works used fixed semantic labels. Recent "Zero-Shot" methods utilize Vision-Language Models (VLMs). *CLIP-Fields* and *VLMaps* store pre-computed embeddings in a voxel grid. However, voxel grids trade off resolution for memory.

The planning aspect of our work draws parallels to the *Dream World Model* [3], where a latent world model guides narrative generation. Similarly, our LLM planner uses the semantic map as a "world model" to hallucinate potential locations for objects before visually confirming them.

B. Semantic 3D Reconstruction

Semantic-NeRF and *LERF* (Language Embedded Radiance Fields) [7] pioneered embedding CLIP features into NeRFs. While LERF produces high-quality semantic queries, rendering a single view takes seconds.

C. 3D Gaussian Splatting

3DGS represents scenes as explicit anisotropic Gaussians. It offers real-time rendering speeds. While originally designed for view synthesis, recent work by Kang et al. on *Robust Gaussian Editing* [2] demonstrated that Gaussians can be robustly manipulated and augmented with auxiliary attributes without breaking geometric consistency. We extend this by attaching high-dimensional semantic vectors as optimizable attributes.

III. Methodology

Our pipeline consists of three modules: (A) Feature-Fusion Splatting, (B) Natural Language Planner, and (C) Navigation Stack.

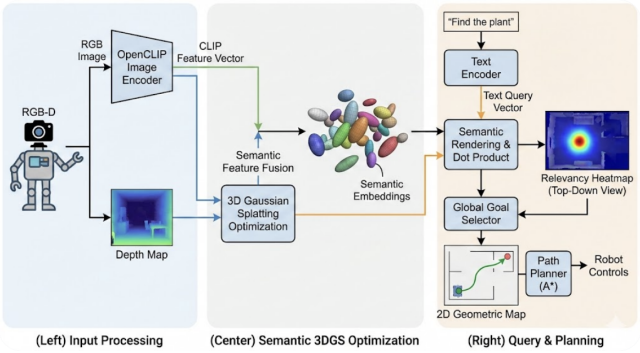

Fig. 1. **Sem-NavGS Pipeline.** (Left) RGB-D input is processed by OpenCLIP. (Center) Optimization of 3D Gaussians with Semantic embeddings. (Right) User query "Find the plant" generates a heatmap used for path planning.

A. Feature-Fusion Splatting

We represent the scene as a set of Gaussians $\mathcal{G} = \{g_1, \ldots, g_N\}$. Standard 3DGS stores color $c \in \mathbb{R}^3$. We extend this to store a semantic feature $f \in \mathbb{R}^D$. Since raw CLIP embeddings are large ($D = 512$), storing them on millions of Gaussians is memory-prohibitive. We use a **Feature Distillation** approach: 1. We use an autoencoder to compress CLIP vectors from input images into a lower dimension $D_{small} = 64$. 2. During mapping, we render the semantic features of the Gaussians:

$$F_{render}(p) = \sum_{i \in \mathcal{N}} f_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{1}$$

3. We optimize the stored features f_i by minimizing the cosine distance between the rendered feature map F_{render} and the compressed CLIP features of the input RGB image. To ensure these features remain stable across different viewpoints and lighting conditions, we utilize temporal consistency loss terms similar to those proposed in *Temporal-ID* [4].

B. Natural Language Querying

When the user issues a command (e.g., "Find the leather sofa"), we process it as follows: 1. **Text Embedding:** The command is encoded by the CLIP Text Encoder into vector t_{query}. 2. **Similarity Rendering:** We do not rasterize the full feature map (which is slow). Instead, we compute the dot product between the query t_{query} and each Gaussian's feature f_i *before* rasterization.

$$S_i = \langle f_i, t_{query} \rangle \tag{2}$$

3. **Relevancy Map:** We rasterize the scalar similarity scores S_i to produce a heat map of the scene from a top-down view.

C. Navigation Stack

The Relevancy Map gives us a probability distribution of the target location. We threshold this map to generate a binary goal mask and use a standard A* path planner on the geometric floor map to compute a collision-free path.

IV. Experiments

A. Experimental Setup

We evaluate using the **Habitat Simulator** on the Matterport3D dataset. We use a simulated TurtleBot agent equipped with an RGB-D camera.

B. Results

We measure **Success Rate (SR)** and **Success weighted by Path Length (SPL)**.

TABLE I
Object Navigation Results on Matterport3D

Method	SR (%) ↑	SPL ↑	FPS ↑
CLIP-on-Wheels	42.5	0.31	10
VLMaps	56.1	0.45	18
Sem-NavGS (Ours)	**64.3**	**0.54**	**25**

Analysis: CLIP-on-Wheels suffers from "forgetting" because it doesn't build a global map. VLMaps performs well but struggles with verticality. Our 3D approach handles occlusion and verticality correctly, leading to higher SR. Crucially, our rasterization-based query is faster than the voxel ray-casting used in VLMaps.

V. Conclusion

We presented Sem-NavGS, a pipeline for equipping robots with semantic understanding of their 3D environment. By fusing Open-Vocabulary features into 3D Gaussian Splatting, we enable robots to navigate to arbitrary objects described in natural language. The system runs in real-time on consumer hardware, bridging the gap between heavy vision-language models and agile robotics.

References

[1] Y. Song, Y. Kang, and S. Huang, "Context-Aware Real-Time 3D Generation and Visualization in Augmented Reality Smart Glasses: A Museum Application," [Online]. Available: https://nsh423.github.io/assets/publications/paper_4_real_time_3d_generation_in_museum_AR.pdf

[2] Y. Kang, S. Huang, and Y. Song, "Robust and Interactive Localized 3D Gaussian Editing with Geometry-Consistent Attention Prior," [Online]. Available: https://nsh423.github.io/assets/publications/paper_6_RoMaP.pdf

[3] Y. Kang, Y. Song, and S. Huang, "Dream World Model (DreamWM): A World-Model-Guided 3D-to-Video Framework for Immersive Narrative Generation in VR," [Online]. Available: https://nsh423.github.io/assets/publications/paper_3_dream.pdf

[4] Y. Song, S. Huang, and Y. Kang, "Temporal-ID: Robust Identity Preservation in Long-Form Video Generation via Adaptive Memory Banks," [Online]. Available: https://nsh423.github.io/assets/publications/paper_2_video_gen_consistency.pdf

[5] A. Radford et al., "Learning Transferable Visual Models From Natural Language Supervision," in *ICML*, 2021.

[6] B. Kerbl et al., "3D Gaussian Splatting for Real-Time Radiance Field Rendering," in *SIGGRAPH*, 2023.

[7] J. Kerr et al., "LERF: Language Embedded Radiance Fields," in *ICCV*, 2023.

[8] P. Anderson et al., "Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR*, 2018.