End-to-End Pipeline for Open-Vocabulary Semantic Novel View Synthesis with 3D Gaussians

Gábor Markó, Daniel Stjepanovic, Dávid Rozenberszki
Technical University of Munich, Visual Computing Group

Abstract

The task of semantic scene understanding and novel view synthesis for 3D scenes is challenging due to the limited availability of labeled 3D datasets. We propose an end-to-end pipeline that is based on two baseline methods. Firstly, we build on the Unified-Lift method, including 3D Gaussian Splatting for geometry reconstruction and 2D-to-3D lifting of semantic segmentation results. Secondly, we use the efficient voxel grid representation to store high-dimensional semantic feature vectors, as proposed in the Plenoxels method for storing the RGB and spherical harmonics values used for 3D scene reconstruction.

By combining the advantages of these baseline methods and representation forms, we can realize open-vocabulary semantic novel view synthesis. It is achieved by rasterizing Gaussians containing logits queried from high-dimensional feature embedding vectors, which are obtained by efficiently lifting 2D segmentation results first into the sparse voxel space, and then to the Gaussian representation.

We test the performance of our method on the real-world ScanNet++ dataset to examine its robustness and evaluate the functionality of our proposed pipeline.

1. Introduction

The task of 3D scene reconstruction from multi-view 2D images has been revolutionized by methods based on Neural Radience Field (NeRF) [5] and 3D Gaussian Splatting (3D-GS) [3]. Despite their exceptional fidelity for novel view synthesis, leveraging these models efficiently for 3D scene understanding tasks remains a challenge.

The efficiency of 3D-GS enables photo-realistic real-time rendering, but it is not optimal to store semantic features on the dense Gaussian primitives in a high-resolution space. To overcome this limitation, we propose a framework that unifies the 3D-GS method with the sparse, semantics-aware representation of Plenoxels [7] for semantic novel view synthesis. Based on the Unified-Lift method [8] that demonstrates 2D-to-3D lifting for scene segmentation, we propose a hybrid method that stores semantic embeddings in sparse voxels and interpolates them over dense Gaussian fields during rasterization, allowing semantic segmentation for any novel views over the full 3D scene.

The main contributions of our pipeline are the following.

1. Lifting the high-dimensional per-pixel feature embeddings resulting from 2D segmentation to 3D sparse voxels using an efficient ray tracing based projection method.
2. The projected high-dimensional per-voxel feature embeddings enable open vocabulary queries in 3D space.
3. The dual photometric and semantic representation enables efficient rasterization and segmentation due to the computational and memory efficiency of the proposed hybrid voxel- and Gaussian-based method.

2. Related Work

Lifting 2D Semantic Segmentation to 3D Scene Representations: Building semantically meaningful 3D environments from 2D inputs is a common challenge in scene understanding. Unified-Lift [8], proposed in *"Rethinking End-to-End 2D to 3D Scene Segmentation in Gaussian Splatting"*, addresses this by embedding learned semantic features into 3D Gaussians. While effective, this approach tightly couples semantics and geometry, limiting stability and feature dimensionality. Our method instead lifts 2D features into a separate sparse voxel grid, yielding more consistent and memory-efficient semantics.

3D Gaussian Splatting: We use 3D Gaussian Splatting [3] as a geometric backbone, following *"3D Gaussian Splatting for Real-Time Radiance Field Rendering"*. It optimizes anisotropic Gaussian primitives for real-time, high-fidelity rendering, avoiding volumetric grids or neural networks.

Language-Driven Segmentation with LSeg and CLIP: For per-pixel semantics, we use LSeg [4] from *"Language-driven Semantic Segmentation"*, which leverages CLIP embeddings to enable open-vocabulary segmentation. These 2D features are projected into 3D to construct our semantic volume.

ScanNet++ Dataset: Our work is evaluated on ScanNet++ [2], introduced in *"ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes"*. It provides multi-view

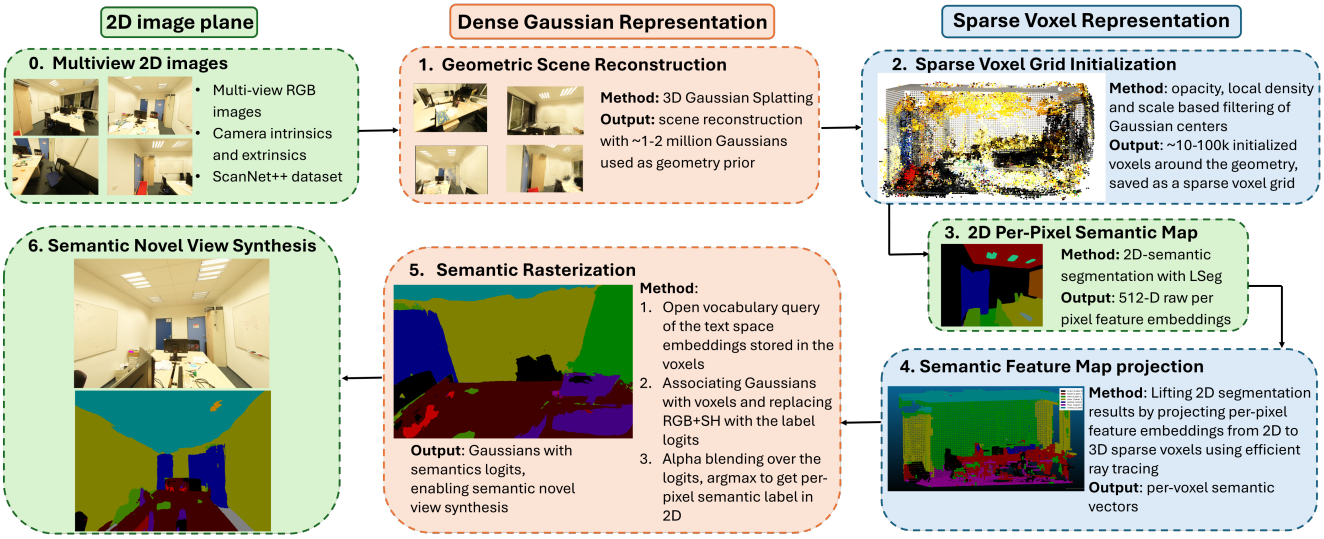

Figure 1. **Overview of our end-to-end hybrid Gaussian- and voxel-based pipeline with 2D to 3D segmentation lifting.** Data storage and processing takes place at three different levels: in 2D image space, in 3D dense space represented by Gaussian Splats, and in 3D sparse voxel space, indicated by the background color of each step. Starting from multiview 2D input images with known camera intrinsic and extrinsic parameters, the pipeline outputs 2D semantic segmentation masks for novel views based on an open-vocabulary query.

RGB-D images, accurate camera poses, and per-vertex semantics—key components for training and evaluating semantic 3D pipelines like ours. Its rich annotations and consistent camera calibration make it particularly well-suited for training multi-view 3D learning pipelines.

3. Method

Our proposed end-to-end pipeline contains the following five major steps that accomplish the task of open-vocabulary semantic novel view synthesis, as shown in Figure 1. The inputs of our pipeline are multi-view RGB images with known camera intrinsic and extrinsic parameters. In this project, we used the ScanNet++ dataset, which is a large-scale dataset with 1000+ 3D indoor scenes.

3.1. Geometric scene reconstruction

For the geometric scene reconstruction, we use the baseline 3D Gaussian Splatting method. Starting from the multi-view 2D images, an initial 3D point cloud is reconstructed using Structure-from-Motion, which provides positions and visibility for initializing the 3D Gaussians. The Gaussians are then optimized end-to-end using gradient-based optimization to refine their positions, orientations, scales, opacities, and colors by minimizing the photometric error between rasterized and ground truth images. At the end of this process, we get around 1-2 million Gaussian Splats as a dense geometric representation of our scene.

3.2. Sparse voxel grid initialization

To make the downstream tasks more efficient, we switch from the dense Gaussian representation to a sparse voxel grid representation. In this step, based on opacity, scale, and local density of the Gaussians, we filter out Gaussian centers with the goal of representing the scene with fewer instances. We use MinkowskiEngine [1] to initialize a sparse voxel grid based on the filtered Gaussian centers, which enables efficient initialization by voxelizing only the occupied regions, drastically reducing memory and computation compared to dense grids. At the end of this step, we expect to have 50-100k occupied voxels, which are sufficient to represent the geometry.

3.3. 2D semantic feature extraction

From the input images, we create 2D semantic maps using the Language-driven Semantic Segmentation (LSeg) model [4], and save out 512D raw per-pixel feature embeddings. LSeg uses a transformer-based image encoder that computes dense per-pixel embeddings of the input image. In our project, we used the ViT-L/16 model as a backbone, as suggested by the LSeg paper. To make the downstream projections and calculations more memory efficient, we save out the raw feature embeddings at the lowest dimensional latent space, which is an intermediate step in the original pipeline.

3.4. Semantic feature map projection

To lift the 2D feature maps storing semantic information into the sparse voxels, we used a simple and fast ray tracing

based method. Based on the known camera parameters and poses, we shoot rays through every pixel of the input images, and associate their feature embeddings with the first hit voxel along the ray. We do this projection for every pixel of all images, and average the feature vectors based on the number of times a voxel was hit by a ray, resulting in 512D per-voxel semantic feature embeddings.

3.5. Semantic rasterization

3.5.1. Open-vocabulary query

The query with open-vocabulary labels is performed at this step in 3D space, on the feature maps stored in the sparse voxels. As we used the raw embeddings at the projection step, we need to perform the last steps of the forward method of the ViT image encoder to transform it into text space, making it comparable with the CLIP text encoder, which computes embeddings of descriptive input labels. In our project, based on LSeg, we used the CLIP ViT-B/32 model as a text encoder. To create the per-voxel logits, we calculated the cosine similarity of the text embedding of each label and the embedded text features extracted by the image encoder.

3.5.2. Associating the Gaussians with the voxels

To associate the logits with the Gaussians, we find the nearest voxel for each Gaussian, and use its logits. We store the logit values in the Gaussians by overwriting their RGB and spherical harmonics values.

3.5.3. Alpha blending of the logit vectors of the Gaussians

As the original rasterization method used in the baseline 3DGS codebase is hardcoded for rasterizing the three RGB channels, we used GSplat [6] for the alpha blending step. This extension supports the rasterization of higher-dimensional feature vectors as well, by using the same alpha blending logic.

3.6. Semantic Novel View Synthesis

Storing the high-dimensional semantic feature embeddings in the 3D sparse voxel grid enables fast open-vocabulary semantic novel view synthesis. The query happens in the 3D space, making it possible to associate the resulting logits to the Gaussians, which can be efficiently rasterized from any novel view. With that last step, our pipeline finally produces the resulting 2D segmentation maps for any user-defined camera poses and label sets.

4. Results

4.1. Qualitative Evaluation

We visualize the semantic novel view synthesis results obtained from our pipeline on ScanNet++ scenes. These renderings on Figure 2 show that our approach produces semantically meaningful and visually consistent outputs from

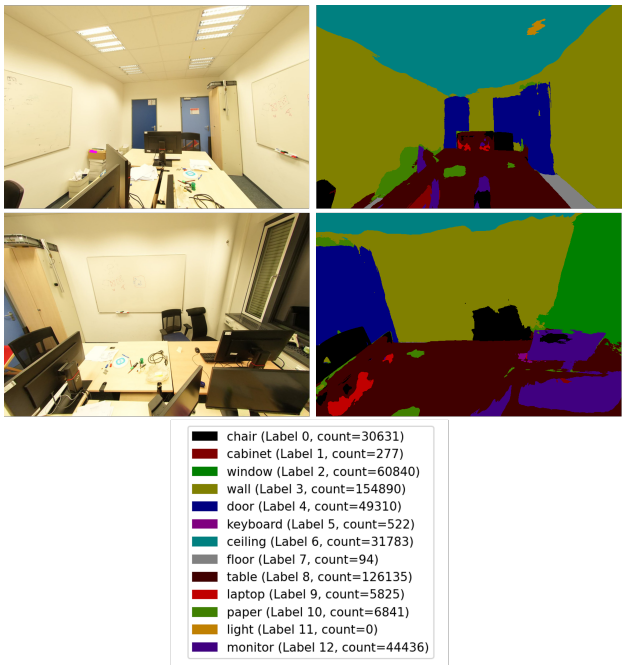

Figure 2. Open-vocabulary novel view segmentation results on the office scene of ScanNet++ dataset.

novel viewpoints. Coarse and structurally dominant objects such as walls, floors, and ceilings are segmented with high consistency, while fine-grained or small-scale objects are more challenging and often underrepresented in the final renderings. Despite not relying on per-object optimization or end-to-end semantic tuning, the system produces coherent results across unseen views.

4.2. Quantitative Evaluation

To gain insight into the quantitative performance of our method, we evaluated it on a single representative ScanNet++ scene. This scene was chosen due to its moderate size and complexity—it contains small objects and cluttered geometry but relatively few images, making it feasible to process under memory constraints.

Evaluation Method	Score
mIoU	0.391
fwIoU	0.494

Table 1. For queries ”chair”, ”cabinet”, ”window”, ”wall”, ”door”, ”keyboard”, ”ceiling”, ”floor”, ”table”, ”laptop”, ”paper”, ”light”, ”monitor” tested against the ScanNet++ annotations of the office scene.

The evaluation was performed by rendering 2D semantic predictions and comparing them to the projected ScanNet++ ground truth. While absolute mIoU values remain

modest, the fwIoU scores are noticeably higher, confirming that large-scale geometry is segmented more consistently than smaller objects. A technical limitation of our setup also restricts us to 32 output classes, whereas ScanNet++ defines 48—this mismatch further complicates a direct comparison.

4.3. Evaluation Setup and Limitations

Importantly, we did not benchmark our approach against other semantic segmentation baselines. Since we focused on a single scene and tailored our training to it, any direct comparison would risk being biased or misleading. Instead, the quantitative results serve as a proof of concept, offering a first estimate of the strengths and limitations of our hybrid pipeline. Our evaluation approach, which relies on rendering projected views from the ScanNet++ semantic mesh, is itself non-standard due to the absence of established benchmarks for semantic novel view synthesis.

Despite these constraints, the results demonstrate that our method can produce spatially and semantically coherent outputs in practice. The framework lays a foundation for future extensions to multi-scene evaluation or end-to-end optimization.

4.4. Discussion and Implementation Considerations

The primary contribution of our work lies in successfully building a complex hybrid pipeline that integrates multiple stages—geometry reconstruction, semantic feature lifting, sparse voxel reasoning, and semantic re-rendering—into a single, functioning system. Each of these components required careful balancing and integration to ensure semantic coherence and geometric consistency across views. The fact that this pipeline yields reliable results across novel viewpoints demonstrates both the conceptual viability and implementation soundness of our approach.

There are, of course, several areas where the current implementation can be improved. In the projection step from 2D to 3D, we currently use a simplified ray tracing scheme that considers only the first intersected voxel along a ray. A more advanced sampling or multi-hit strategy could potentially capture more context and improve semantic accuracy. Additionally, the use of a sparse voxel grid, while highly memory-efficient, may impose subtle limits on semantic sharpness in areas with fine detail or sparse sampling. These effects are further influenced by intermediate interpolation steps such as nearest-neighbor matching from voxels to Gaussians.

Taken together, the results demonstrate a strong foundation: a multi-stage, modular pipeline capable of producing coherent semantic novel views. While there is room for optimization in individual steps, the current version proves the feasibility of this hybrid design and lays the groundwork for future end-to-end improvements.

5. Conclusion

We presented a hybrid pipeline for semantic novel view synthesis that combines dense 3D Gaussian representations with a sparse voxel-based semantic field. Our system successfully integrates multiple complex stages—geometry reconstruction, 2D-to-3D feature lifting, and rasterizing semantics—into a functional end-to-end workflow. The results demonstrate that our approach is feasible and can produce semantically coherent outputs with limited supervision.

While the current implementation prioritizes clarity and modularity over deep optimization, the framework already supports meaningful semantic inference from novel viewpoints. By leveraging the sparse voxel representation, we enable the storage of high-dimensional features in a memory-efficient manner, laying the groundwork for lightweight semantic synthesis in complex scenes.

Overall, our work serves as a proof-of-concept showing that hybrid representations can be made to work together effectively. With further optimization and scaling, this approach has the potential to serve as a flexible and efficient backbone for real-time semantic 3D applications

The code implementation of our project is available at: github.com/gabormarko/3D-semantic-segmentation.

References

[1] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes, 2017. 1

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. 1

[4] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation, 2022. 1, 2

[5] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 1

[6] Gaussian Splatting. Rasterization api documentation, 2025. Accessed: July 23, 2025. 3

[7] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks, 2021. 1

[8] Runsong Zhu, Shi Qiu, Zhengzhe Liu, Ka-Hei Hui, Qianyi Wu, Pheng-Ann Heng, and Chi-Wing Fu. Rethinking end-to-end 2d to 3d scene segmentation in gaussian splatting, 2025. 1