Pipeline for 3D Scene Editing via 3D Gaussian Splatting

Jinyoon Kim, Manvitha Reddy , Sansshita Baskaran

Problem Statement – The "Manual Editing" Bottleneck

- **The Problem:** Editing 3D scenes currently requires expert knowledge of complex software like Blender. Users must manually fix geometry and textures.
- **The Gap:** While AI tools now allow us to edit 2D images easily with text (like Generative Fill), this capability does not exist natively for 3D scenes.
- **Our Objective:** We built an automated pipeline to bridge this gap. We chain multiple SOTA frameworks together to allow users to edit 3D scenes using only text prompts, without touching the geometry manually.

Evolution of Our Approach

- **Initial Idea:** We originally planned to edit the scene by modifying the 2D training images (e.g., inpainting the object out of every frame) and re-training the model.
- **The Failure:** We found this impractical. 2D diffusion models are stochastic; they generate inconsistent textures across different views (e.g., wood floor in View 1, tile in View 2), which breaks 3D reconstruction.
- **The Pivot:** We decided to **directly manipulate the 3D framework**.
    - We modify the 3D Gaussian structure itself (removing/adding points) to ensure geometric consistency.
    - We use 2D images only for texture guidance (inpainting holes).

The Technical Challenge - Volumetric Data

- **Representation:** We use **3D Gaussian Splatting (3DGS)**. Unlike meshes, this is a "volumetric cloud" of data without clear surfaces.
- **The Challenge:** Objects in 3DGS have internal density. If you naively delete the visible surface, you reveal the opaque inner layers (the "guts") instead of a clean background.
- **Goal:** We need a method that identifies and removes the *entire* volume of an object based only on 2D text prompts.

Dataset (Mip-NeRF 360)

- **The Benchmark:** We utilized the **Mip-NeRF 360 v2** dataset (Barron et al., CVPR 2022), which is the industry standard for evaluating modern 3D reconstruction methods.
- **Why this dataset?** Unlike synthetic datasets, these are **unbounded real-world captures**. They feature complex central objects surrounded by detailed backgrounds, making them significantly harder to edit than simple object scans.
- **Our Testbed:** We specifically selected two challenging scenes to validate our pipeline:
    - **Garden:** Outdoor scene with complex thin structures (plants) and lighting.
    - **Kitchen:** Indoor scene with high clutter and occlusions.

Segmentation - GroundingDINO + SAM2

- **The Goal:** Translate a text prompt into a precise 2D pixel mask.
- **GroundingDINO:** Provides semantic understanding. It takes a text prompt (e.g., "brown plant") and finds the bounding box .
- **SAM2 (Segment Anything):** Provides geometric precision. It refines the box into a tight pixel mask.
- **Why these integration?** Manual prompting for 100+ views is impossible. This combination automates the detection process across the entire dataset.

Inpainting - Why LaMa over Stable Diffusion?

- **Initial Plan:** We originally used **Stable Diffusion (SDXL)** to fill holes left by removed objects.
- **The Limitation:** SDXL is "too creative." Instead of emptying the space, it often hallucinates new objects (replacing a plant with a vase or pot).
- **Our Choice (LaMa):** We replaced it with **LaMa (Large Mask Inpainting)**.
- LaMa specializes in "texture continuation" using Fast Fourier Convolutions. It extends the existing floor/wall pattern into the hole without inventing new objects.

3D Generation - GaussianDreamerPro

- **The Goal:** For "Object Addition," we need to generate new 3D assets from scratch.
- **The Framework:** We used **GaussianDreamerPro**, which bridges text-to-image diffusion with 3D Gaussian Splatting.
- **Integration:** We use this as an external asset generator. It produces a `.ply` point cloud (e.g., "coffee cup") which our pipeline then imports and merges into the scene.

Pipeline Overview

System Design: We engineered a multi-stage framework that transforms 2D semantic priors into consistent 3D geometric edits.

Phase I: Volumetric Selection (Lifting)

- We aggregate 2D masks from SAM2 into a unified 3D Region of Interest (ROI), utilizing depth-based voting to strictly resolve occlusions and filter background noise.

Phase II: Scene Restoration (Inpainting)

- We eliminate the target volume and reconstruct the background texture. Our multi-pass depth comparison ensures only true "hole" pixels are inpainted by LaMa.

Phase III: Asset Integration (Synthesis)

- We synthesize new 3D assets via GaussianDreamer and mathematically merge them into the scene's coordinate system using a custom spatial placement module.

Phase I - Volumetric Selection (Occlusion-Aware Lifting)

Objective: Transform 2D semantic masks into a precise 3D Region of Interest (ROI), handling both object ambiguity and background occlusion.

1. Detection Strategy (2D):

- **Spatial Disambiguation (Garden):** Utilized a Reference Box to filter false positives in cluttered environments, strictly isolating the target plant from surrounding foliage.
- **Semantic Detection (Kitchen):** Relied on Open-Vocabulary Search (Text-Only) for unique objects, proving the system can locate targets without manual bounding boxes.

2. The Lifting Algorithm (3D):

- **Problem:** Naive projection of 2D masks selects the object and the wall behind it.
- **Solution:** Depth-Based Voting. We project every Gaussian into the view and check its depth against the rendered surface.
- **Mechanism:** A Gaussian only receives a "vote" if |Gaussian_Depth - Surface_Depth| < 5%.
- **Result:** This effectively filters out background Gaussians, creating a tight volumetric hull around the visible surface.

Phase II - Scene Restoration (Geometry-Aware Inpainting)

Objective: Eliminate the target 3D volume entirely and reconstruct the background texture without introducing artifacts.

1. Volumetric Removal (The "Artichoke" Solution):

- **Problem:** 3DGS objects have internal density. Naive deletion or alpha-thresholding often leaves an opaque inner core.
- **Algorithm: Multi-Pass Depth Comparison.** We render the scene twice: once fully (`Depth_Full`) and once with only the ROI (`Depth_ROI`).
- **Logic:** A pixel is marked as a "hole" only if `|Depth_Full - Depth_ROI| < 0.01`. This mathematically identifies pixels where the ROI was the **front-most visible object**.

2. Texture Completion (LaMa):

- **Method:** We feed the computed "Hole Masks" into **LaMa (Large Mask Inpainting)**.
- **Rationale:** Unlike diffusion models (SDXL) which attempt to generate new objects, LaMa utilizes **Fast Fourier Convolutions** to propagate existing background patterns (e.g., floor tiles) into the void, ensuring geometric continuity.

Phase III - Asset Integration (Generative 3D Composition)

Objective: Synthesize new 3D assets and seamlessly integrate them into the scene's coordinate system.

1. Asset Generation (External):

- **Tool:** We utilized **GaussianDreamerPro** to generate standalone 3D Gaussian point clouds (PLY files) from text prompts.
- **Output:** High-fidelity 3D assets (e.g., coffee_cup_pro.ply, cowboy_boots_pro.ply) with independent internal geometry.

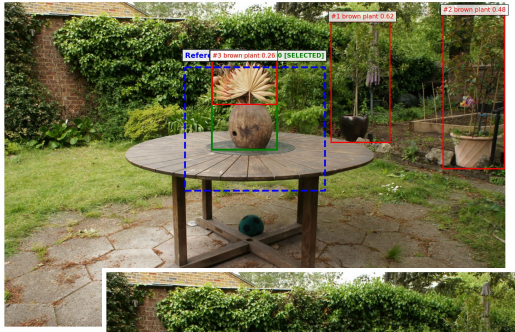
2. The Placement Module (Internal):

- **Centroid Alignment:** We calculate the centroid of the *removed* ROI to determine the target placement coordinates [x, y, z].
- **Affine Transformation:** We apply specific scaling factors (e.g., 0.06 for Cup) and spatial offsets defined in the config to align the object with the ground plane.
- **Merging:** The new Gaussians are mathematically appended to the optimized scene checkpoint, creating a unified renderable volume.

Experiment I - Validation of Volumetric Selection

Objective: Validate the translation of 2D semantic priors into coherent 3D volumes.

Visual Pipeline: Progression from Input View to Semantic Box, Instance Mask, and Projected 3D ROI.

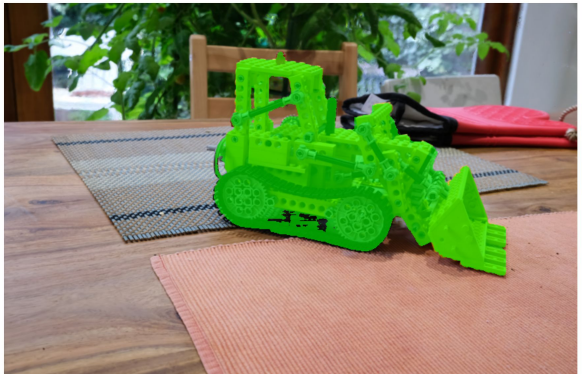


Case A: Spatial Disambiguation (Garden Scene)

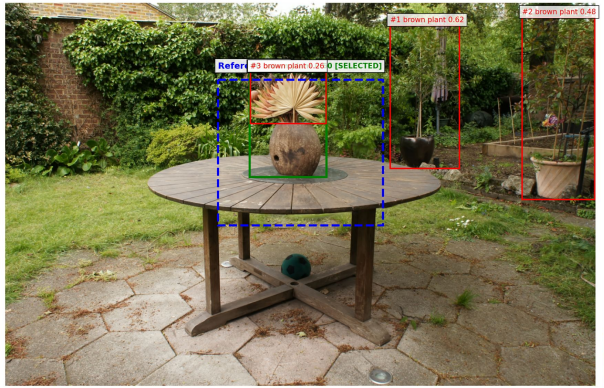
- **Strategy:** Implemented a **Reference Box** to strictly limit the inference window.
- **Result:** Successfully rejected false positives (surrounding foliage), isolating only the target plant.

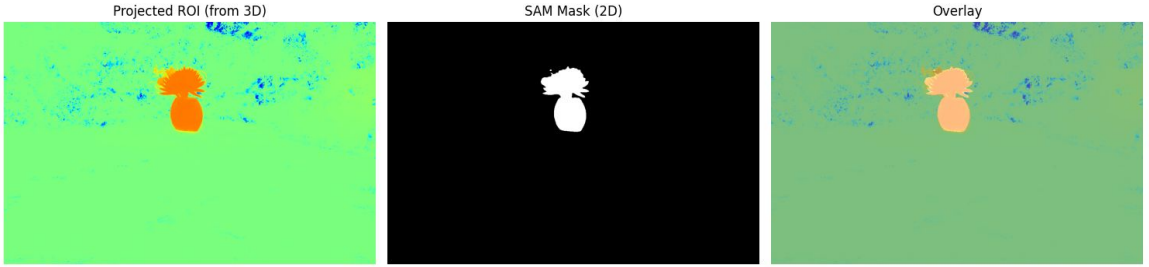
Case B: Zero-Shot Lifting (Kitchen Scene)

- **Strategy:** Utilized **Text-Only Detection** relying on unique semantic features.
- **Result: Depth-Based Voting** proved critical, successfully filtering out background floor pixels to prevent leakage.



Experiment I - Validation of Volumetric Selection





Experiment II - Removal & Inpainting Ablation

Objective: Verify geometric integrity of the removal and compare synthesis methods.

Visual Comparison: Holed Render (visualizing the deleted volume) vs. **LaMa Result** vs. **SDXL Failure**.

Hole Mask Quality: Multi-Pass Depth Comparison accurately captured the full object volume (solving the "Artichoke Problem") without leaving opaque inner cores.

Ablation Results:

- **SDXL (Baseline):** Failed due to hallucinations (inserting new objects into the void).
- **LaMa (Ours):** Successfully propagated existing floor textures, creating a geometrically consistent empty surface.

Experiment II - Removal & Inpainting Ablation

Experiment III - Generative 3D Composition

Objective: Assess the geometric consistency of external assets merged into the scene.

Visual Integration: Generated Asset (.ply) integration into the Merged Scene across multiple view angles.

Results:

- **Garden:** Replaced "Brown Plant" with **Coffee Cup**
- **Kitchen:** Replaced "Tracker Toy" with **Cowboy Boots**
- **Geometric Consistency:** Added objects exhibit true volumetric behavior. Camera rotation confirms correct perspective, occlusion, and depth relative to the original scene, distinguishing this from 2D overlays.

Experiment III - Generative 3D Composition

Conclusion

The "Disharmony" Problem: We found that chaining independent SOTA frameworks creates significant compatibility gaps.

- *Visual Mismatch:* Objects generated by `GaussianDreamerPro` (e.g., Cowboy Boots) often look "cartoonish" or have baked-in lighting that clashes with the realistic Mip-NeRF 360 scene.
- *Optimization Failure:* We could not robustly relight the inserted objects to match the environment shadows.

ROI Lifting Failure (Kitchen Scene): Our "Occlusion-Aware Lifting" algorithm is not universally robust.

- *Success (Garden):* Worked perfectly for distinct objects like the "Brown Plant."
- *Failure (Kitchen):* Failed on the "Yellow Tracker Toy" due to its complex thin geometry and surrounding clutter. The algorithm struggled to separate the object from the floor, leaving "awkward remainings" (ghostly artifacts) that even optimization could not fix.

Conclusion

Core Conclusion: Our work proves that a "patchwork" pipeline of separate 2D/3D tools is too brittle for general-purpose editing. The gap between 2D masks, 3D geometry, and generative assets is too large to bridge manually.

Future Work: We propose moving away from pipelines toward a **Unified End-to-End Model** ("Instruct-to-3D").

- *The Concept:* A single model trained to take [3D Scene + Text] as input and output [Updated Gaussian Parameters] directly.
- *Benefit:* This would natively handle geometric consistency, lighting integration, and multi-view coherence without requiring intermediate masks or manual placement logic.

Conclusion

GitHub Repository: https://github.com/jinyoonok2/3DCV-3D-Scene-Edit-with-3DGS

Q&A