Realistic Synthetic 3D Environment Generation for Scalable Vision-Based Automation

Vikram Raj Nagoor Kani	James Butler	Nathaniel Christie	Sumanth Prabhu
M.Eng Autonomy and Robotics	M.CE	M.Eng Autonomy and Robotics	M.ECE
VN18	jfbutle2	nmc7	prabhu5

CS 543 Final Report

1. Introduction

Many industries depend on automation pipelines built around traditional computer vision techniques. These legacy systems are expensive to maintain, difficult to scale, and struggle to adapt to new products or shifting environments. While modern deep learning models for instance segmentation, object detection, and reinforcement learning have demonstrated strong performance in controlled settings, deploying them in production requires robust generalization to unseen objects, layouts, and environmental conditions. A primary barrier to achieving this generalization is the shortage of diverse, accurately labeled training data, which is costly and time-consuming to collect in the real world.

This project addresses that bottleneck by developing a framework for generating realistic synthetic 3D environments that can serve as an automated source of labeled training data. The core idea is to reconstruct real-world objects and spaces as 3D digital assets using a combination of Structure-from-Motion (SfM), 3D Gaussian Splatting (3DGS), and differentiable mesh reconstruction, and then compose those assets into editable virtual scenes within a physics simulation engine. These scenes can be rendered from arbitrary viewpoints with automatically generated ground-truth labels, enabling scalable dataset generation without manual annotation.

The framework targets four primary scene-editing capabilities: object insertion, object removal, object replacement, and object repositioning. Given scope and time constraints, this project focuses primarily on the insertion capability, establishing the end-to-end pipeline from real-world capture to Unity scene composition.

1.1 Background and Related Work

3D Gaussian Splatting (3DGS)

3D Gaussian Splatting [1] is a rasterization-based technique for real-time novel-view synthesis. Unlike Neural Radiance Fields (NeRFs), which rely on continuous implicit representations queried through expensive volumetric ray-marching, 3DGS represents a scene as an explicit collection of 3D Gaussian primitives. Each Gaussian is parameterized by a 3D position, an anisotropic covariance matrix, an opacity value, and a view-dependent color encoded via Spherical Harmonics. This explicit representation supports real-time rendering at high resolution while achieving state-of-the-art visual quality. For this project, 3DGS serves as the primary appearance representation for both object assets and scene basemaps.

Structure-from-Motion (SfM)

Structure-from-Motion is a photogrammetry technique that recovers 3D structure and camera poses from unordered 2D image collections. COLMAP [2] was used throughout this project to estimate camera intrinsics and extrinsics from overlapping photographs, producing sparse point clouds that serve as the initialization scaffold for downstream 3DGS training. Accurate camera poses are a hard prerequisite for high-quality Gaussian optimization: reprojection errors above roughly 1.0 px correlate with measurable degradation in trained splat quality.

Differentiable Mesh Reconstruction (nvdiffrec)

nvdiffrec [3] is a differentiable inverse-graphics tool that extracts a textured triangle mesh from multi-view images and camera poses. It represents the surface implicitly using DMTet and optimizes it through differentiable rasterization until the mesh silhouette and appearance match the input views. The resulting mesh with UV-mapped texture is compatible with standard game engines and physics simulators. In this pipeline, nvdiffrec provides the collision geometry for physics simulation in Unity, while the Gaussian Splat provides the photorealistic appearance layer.

Mip-Splatting and SuGaR

Vanilla 3DGS is optimized for the specific resolutions and viewing angles of its training images, and can exhibit aliased artifacts flickering, splotchy edges, missing geometry when the scene is rendered off-trajectory. Mip-Splatting addresses this by introducing a 2D image-space anti-aliasing filter and a 3D smoothing kernel into the rasterizer. SuGaR complements this by regularizing Gaussian positions toward coherent surfaces during training and then extracting an explicit mesh via Poisson surface reconstruction, reducing the prevalence of floater Gaussians that appear as visible blobs from off-axis viewpoints.

Gaussian Grouping

Gaussian Grouping [5] extends 3DGS training with a learned group-identity field. Each Gaussian is assigned a group ID by lifting 2D segmentation masks produced by the Segment Anything Model (SAM) [4] into 3D. This enables selective deletion, recoloring, or replacement of specific objects within a trained splat. Object removal was explored in this project as an extension beyond insertion.

DEVA: Tracking Anything with Decoupled Video Segmentation

DEVA [9] is a semi-online video object segmentation framework that decouples per-frame segmentation from temporal propagation. It accepts mask proposals from any image-level segmentation model and propagates consistent identities across frames using a bi-directional association mechanism. In this project, DEVA was used downstream of SAM to produce temporally consistent per-frame mask label maps across multi-view capture sequences, which were then used as supervision for Gaussian Grouping identity training.

2. Details of the Approach

The overall system is organized into three stages: (A) 3D Object Asset Acquisition, (B) World Map Reconstruction and Basemap Evaluation, (C) Map Generation with MASt3R-SLAM, COLMAP, and NeRF, and (D) Simulation Evaluation and Asset Interaction. Each stage is described in detail below.

2.A 3D Object Asset Acquisition (Vikram)

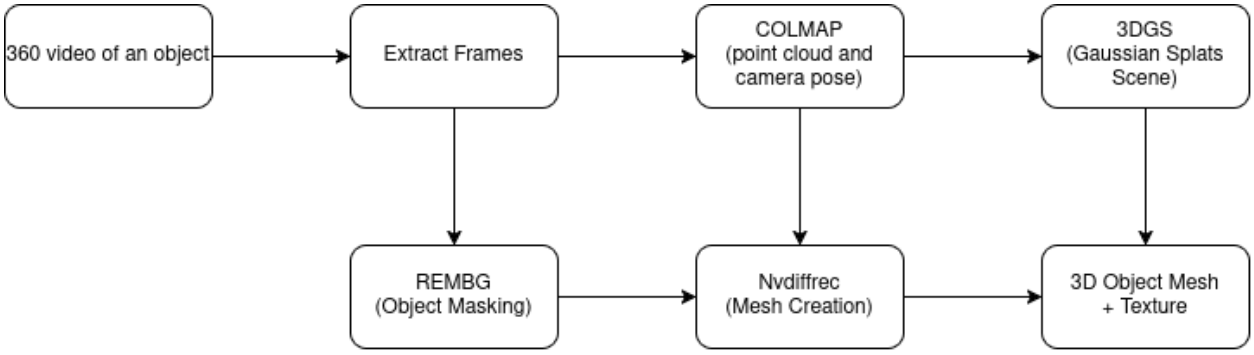

Figure: The above image showcases a pipeline to generate realistic texture and mesh for a 3D object.

2.A.1 Data Capture

A smartphone was used to capture 4K video (3840×2160, 30 fps, portrait mode) of a fire hydrant. Frames were extracted at 3 fps using FFmpeg, yielding approximately 97 images. A textured outdoor pavement background was deliberately retained during capture because smooth, textureless objects such as the hydrant provide insufficient SIFT keypoints on their own surfaces; the surrounding texture improves COLMAP feature matching and registration stability.

2.A.2 Structure-from-Motion with COLMAP

COLMAP [2] was run on the unmasked original images to maximize feature matching quality. The pipeline proceeded through SIFT feature extraction, exhaustive pairwise matching, incremental sparse reconstruction, and image undistortion to the pinhole camera model required by the 3DGS codebase. All 97 images were successfully registered (100% registration rate), producing a sparse point cloud of 86,202 points. The scene had a spatial scale of approximately 192 units and required normalization for downstream tools.

2.A.3 Background Removal with REMBG

Background removal was applied to the COLMAP-undistorted images (1920×1081) using the REMBG library [11]. Three segmentation models were evaluated: u2net, isnet-general-use, and birefnet-general, with birefnet-general producing the highest quality. Approximately 88 of 97 images (~91%) were cleanly segmented; the remaining 5–9 images with minor artifacts near the hydrant base were manually reprocessed. The final masked set was output as RGBA PNG files.

2.A.4 3DGS Training

3DGS training used the official Graphdeco-Inria implementation [9]. The final training configuration ran 30,000 iterations on an NVIDIA RTX 4070 Laptop GPU (8 GB VRAM), producing a point_cloud.ply file of optimized 3D Gaussians verified visually in the SuperSplat browser viewer.

2.A.5 Mesh Reconstruction with nvdiffrec

nvdiffrec [3] was used to generate a collision mesh from the masked images and COLMAP camera poses. Key preprocessing steps included normalizing the scene to a unit sphere, proportionally scaling camera intrinsics to match the training image resolution, and disabling pre_load to manage RAM usage. The resulting mesh broadly captures the hydrant shape and serves as the collision proxy in Unity physics simulation.

2.A.6 Object Removal: Gaussian Grouping (Experimental)

An experimental object removal and scene recomposition pipeline was implemented using Gaussian Grouping [5], which extends 3DGS by assigning each Gaussian a compact learned identity encoding that lifts 2D segmentation masks into 3D during training.

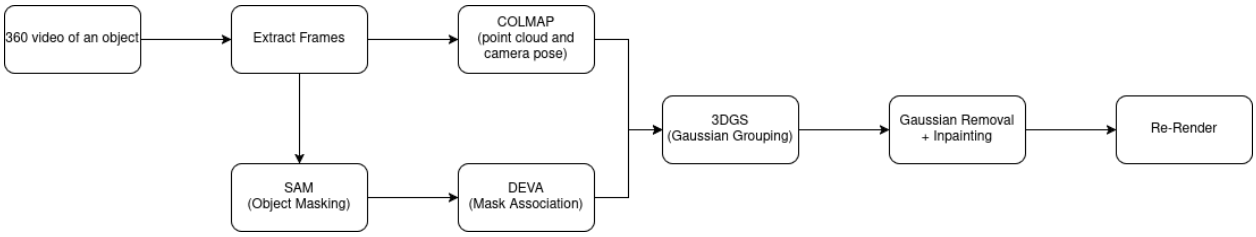

Figure: The above image showcases a pipeline to generate a map which can be used to edit objects in the scene.

Preprocessing. Multi-view images were first processed by SAM (ViT-H) in automatic everything mode, generating per-frame instance masks across all 246 captured views. DEVA [9] was then applied to propagate and associate mask IDs across the multi-view sequence, treating it as a video. This produced a unified set of per-frame PNG label maps where each pixel carries a consistent integer object ID across the entire sequence. The DEVA output was then used as supervision for the Identity Encoding during Gaussian Grouping training.

Training. Gaussian Grouping was trained for 30,000 iterations on an NVIDIA A100 GPU. Each Gaussian was augmented with a 16-dimensional learnable Identity Encoding optimized jointly with standard appearance and geometry reconstruction losses. Two additional loss terms were introduced: a 2D Identity Loss applying cross-entropy supervision from the DEVA-associated mask labels via differentiable rendering, and a 3D Regularization Loss enforcing spatial consistency by pulling the Identity Encodings of the K=5 nearest neighboring Gaussians toward agreement, improving coverage of occluded or infrequently observed Gaussians.

Object Removal. After training, object removal was performed by identifying the target object's integer ID from the rendered identity maps and deleting all Gaussians classified under that ID. This operation requires no retraining and executes in seconds. The removal was only partially successful: a portion of the target object's Gaussians was correctly identified and deleted, but residual geometry remained visible in novel-view renders.

Root cause analysis identified four contributing failure modes:

- Incomplete SAM masks across views. SAM failed to segment the target object in several frames due to viewpoint variation, lighting changes, and object-background similarity. Gaussians observed only in unsegmented frames received no positive identity signal and were incorrectly assigned to the background, causing them to survive deletion.

- Gaussian identity ambiguity at boundaries. Boundary Gaussians influenced by both masked and unmasked views received ambiguous identity assignments during training, causing them to be classified as background and retained after deletion.
- DEVA association errors on early frames. The semi-online temporal setting in DEVA requires a buffer of frames before propagation begins, resulting in the first 13 frames having no associated mask output. These frames were filled with a placeholder mask from the nearest available frame, introducing incorrect supervision for Gaussians predominantly visible in early views.
- Absence of inpainting. Object removal exposes background regions never observed by any training camera, resulting in holes or noisy geometry behind the deleted object. A complete pipeline would follow removal with a 3D inpainting step using a 2D inpainting model such as LaMa to fill hole regions, followed by fine-tuning of newly introduced Gaussians against the inpainted 2D guidance.

2.B World Map Reconstruction and Basemap Evaluation (James)

2.B.1 Basemap Representation Candidates

Because the basemap occupies the dominant portion of any rendered scene, its quality sets a ceiling on the perceived realism of the entire environment. Five reconstruction variants were evaluated on a common indoor scene (Mip-NeRF 360 room, 311 images): a pretrained INRIA Mip-NeRF 360 Bonsai checkpoint used as a sanity-check baseline, a custom vanilla 3DGS reconstruction, a Mip-Splatting variant, a SuGaR refinement pass applied on top, and the same Mip-Splatting + SuGaR pipeline applied to a synthetic capture (Replica dataset).

2.B.2 COLMAP Configuration

The custom reconstruction was trained on the Mip-NeRF 360 room scene using a 272/39 train/test split (every 8th image held out per the dataset convention). COLMAP was configured for sub-pixel keypoint localization, registering all 311 cameras with a mean reprojection error of 0.948 px, below the 1.0 px threshold considered healthy for 3DGS training. The 3DGS training run (30,000 iterations, 1557×1038 resolution) reproduced the published benchmark of approximately 31.4 dB PSNR, confirming that upstream poses were sufficient.

2.B.3 Unity Integration and .ply Merging Pipeline

The trained basemap was imported into Unity 6.4 (URP) via aras-p's GaussianSplatAsset plugin [7]. The plugin renders each Gaussian Splat asset with its own per-frame depth sort but does not provide a global sort across multiple assets, which causes alpha-erasure when two splats overlap: the basemap's accumulated transparency overwrites foreground objects.

To resolve this without modifying the renderer, a transform merging step was added to the asset preparation pipeline. The merging procedure reads each object's world-space placement from the Unity scene file, then produces a single fused .ply asset that preserves the spatial configuration set up in the editor. The merged asset imports back into Unity as a single Gaussian Splat Renderer, where a single global depth sort eliminates the alpha-erasure problem. This merging pipeline enables large-scale scene composition with correct rendering of multiple independently reconstructed objects.

2.C Map Generation with MASt3R-SLAM, COLMAP, and NeRF (Nate)

2.C.1 Structure-from-Motion Pipeline Development

Initial reconstruction efforts centered on MASt3R-SLAM, which proved highly incompatible with the target Windows deployment environment. Experimentation with Windows Subsystem for Linux confirmed that the computational and system requirements exceeded what WSL could reliably handle, necessitating a native Linux installation to run the program effectively. Rather than pivoting the deployment platform, an alternative reconstruction approach was pursued.

COLMAP [2], a Structure-from-Motion and Multi-View Stereo pipeline offering superior accuracy for object-centric reconstruction and straightforward compatibility with Windows systems, was selected as a replacement. A full ingestion pipeline was constructed around COLMAP: input video files are processed to extract individual frames, which are then passed through the SfM pipeline to produce per-frame positional data, a 3D

Gaussian Splat derived from that data, and a Poisson mesh extracted from the splat geometry. Initial results showed strong recovery of surface detail in high-frequency regions such as printed graphics, but the broader geometry exhibited the rough, melted surface quality characteristic of direct Poisson surface reconstruction from Gaussian splat representations.

2.C.2 NeRF-Based Surface Refinement

To address the surface quality limitations of direct Poisson reconstruction, Nerfstudio [10ref] was adopted for neural radiance field refinement. This framework accepts Gaussian splat data as input and trains a neural radiance field to reconstruct the scene, effectively smoothing surface geometry and suppressing reconstruction artifacts through learned scene representation. The Nerfstudio model was trained on output data from the COLMAP pipeline, producing renderings with noticeably improved surface continuity relative to the raw Poisson mesh.

The refined NeRF scene can be exported as a 3DGS for downstream meshing. To improve the quality and compactness of the final mesh, the COLMAP pipeline was extended to incorporate background removal using REMBG [11]. In this modified workflow, background-free frame variants are generated for training, while the original frames with full context are retained solely for camera pose estimation. Nerfstudio is then trained on the background-removed images using these poses, yielding a more isolated object representation. Post-export filtering was applied to remove geometric fragments remaining from incomplete background separation. While this approach produced meshes with less surface melting than the baseline Poisson reconstruction, detail recovery was reduced and persistent holes in the mesh geometry remained unresolved.

2.C.3 Comparative Object Evaluation

To assess reconstruction robustness across object types, a second subject, an oil canister with high surface variation and dense printed text was evaluated through the same pipeline. The more geometrically distinct surface provided a richer set of keypoints for the meshing algorithm, yielding improved reconstruction quality relative to the initial bottle subject. Additionally, a scene-level reconstruction of the bottle within its surrounding environment was performed, producing results qualitatively superior to the isolated object model, consistent with SfM pipelines benefiting from additional background geometry to constrain camera pose estimation.

2.C.4 Discussion

This stage established a functional end-to-end pipeline from video capture through textured mesh generation, validating the combined COLMAP and Nerfstudio approach for object-centric reconstruction on Windows-compatible hardware. Key limitations identified include the significant additional training time introduced by the NeRF refinement stage, incomplete surface closure in isolated-object meshes, and a reduction in fine detail relative to the raw splat representation.

2.D Simulation Evaluation and Asset Interaction (Team Member 4)

2.D.1 Unity Environment Integration

The simulation and interaction phase focused on integrating reconstructed assets into Unity to evaluate their utility as editable entities within a realistic 3D environment. Unity was selected for its robust scene-integration workflow, offering native support for spatial transforms, illumination, camera models, Rigidbody dynamics, and colliders, alongside plugins for Gaussian Splat visualization. A key objective was the decoupling of an object's visual and physical representations: the Gaussian splat serves as the photorealistic appearance layer, while a mesh or simplified proxy provides the collision geometry required for physics simulation.

Following the import of scene and object assets into Unity, an object-insertion workflow was validated. The fire hydrant asset was instantiated within the reconstructed world map and manipulated using translation, rotation, and scaling operations through Unity's editor controls. Simplified proxy geometry and colliders were employed to validate the interaction framework independently of mesh fidelity, enabling testing of whether inserted assets could maintain stable poses, respond to gravitational forces, maintain ground-plane contact, and interact with surrounding scene elements through the Unity physics engine.

2.D.2 Object Insertion and Scene Editing

After importing the reconstructed scene and object assets into Unity, the fire hydrant asset was inserted into the environment and manipulated using translation, rotation, and scaling operations. Since the reconstructed meshes were still under refinement, placeholder colliders were used to validate the interaction workflow independently of

final mesh quality. A lightweight transform-control script was implemented to automate object placement and scaling within the scene.

2.D.3 Physics and Collision Testing

To support interaction within the simulation environment, simplified colliders and Rigidbody components were attached to inserted objects. These proxy colliders enabled gravity simulation, collision handling, and stable placement inside the scene. Diverse physics configurations were evaluated, including static colliders for environmental geometry and Rigidbody components for dynamic inserted assets. The evaluation involved iterative adjustment of object scale, collider bounds, mass, and collision parameters to ensure physical behavior remained aligned with the visual placement of the Gaussian-rendered asset. Interaction testing confirmed that inserted assets could maintain stable poses, respond correctly to gravity, and interact with the environment while remaining visually aligned with the Gaussian-rendered representation.

2.D.4 Discussion

This stage demonstrates that reconstructed Gaussian assets can function as editable scene objects rather than passive visualizations. Although the current implementation relies on simplified collision proxies, the workflow can directly support higher-quality meshes generated by tools such as nvdiffrec or SuGaR. The resulting pipeline provides a foundation for synthetic dataset generation, robotics simulation, and interactive scene-editing applications.

3. Results

3.1 Object Asset Acquisition Results

COLMAP Reconstruction

COLMAP successfully registered all 97 images (100% registration rate). The sparse point cloud contained 86,202 points, and mean reprojection error was within acceptable bounds. The 100% registration rate and dense point cloud confirm that the capture methodology textured background, 360-degree coverage, 3 fps extraction was effective for this class of object.

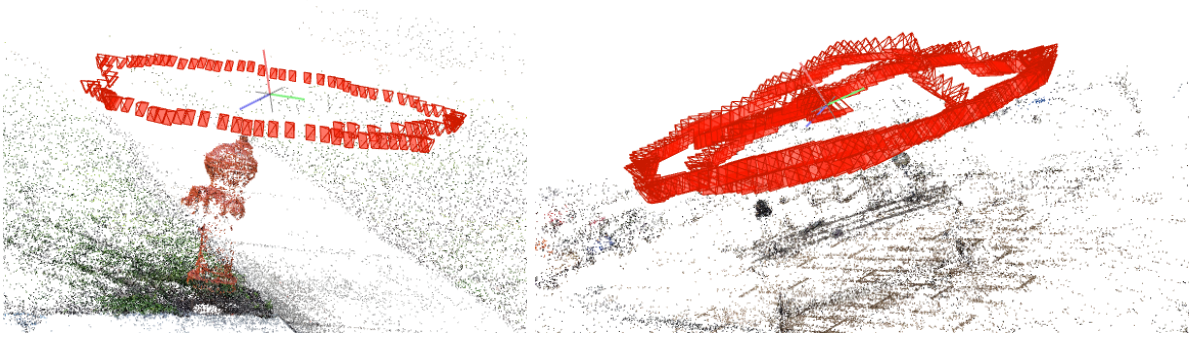

Figure: The above image showcases a point cloud and camera poses for each frame used to create that map using COLMAP of a Fire Hydrant and a Table with an object on it.

Background Removal

REMBG achieved clean segmentation on approximately 88 of 97 images (~91%). The 5–9 images exhibiting minor artifacts (pavement streaks near the hydrant base) were manually reprocessed using a higher-quality model. Final mask quality was sufficient for both 3DGS and nvdiffrec training.

Figure: The above image showcases masked out fire hydrant images.

Gaussian Splatting

3DGS training converged after 30,000 iterations. The resulting Gaussian splat rendered the fire hydrant with high photorealism when viewed in SuperSplat. Background Gaussians were present but negligible for the intended use case, where only the foreground object is composited into the target scene.

Figure: The above image showcases a Gaussian Splat created from the COLMAP generated above.

Mesh Reconstruction (nvdiffrec)

Nvdiffrec did not produce a mesh which was usable and while looking into optimizing the results we saw another approach which produced a mesh better and that's shown below .

*Figure: The above image showcases mesh generated by **nvdiffrec** vs Ground Truth.*

Mesh Reconstruction via Poisson Surface Reconstruction and NeRF Refinement

The following meshes were produced from running Poisson surface reconstruction on 3DGS represented as point clouds. Results progressed from a direct COLMAP Poisson surface reconstruction, through NeRF reconstruction, to a Poisson surface reconstruction on the NeRF-refined 3DGS. The NeRF refinement stage produced measurably smoother surfaces and reduced floater artifacts, though persistent mesh holes remained in regions of limited multi-view coverage. The oil canister subject, possessing greater surface texture density, yielded superior reconstruction quality relative to the hydrant across all pipeline variants.

Figure: The above image showcases the meshes created using the refined method.

Automatic Segmentation and Object Classification

SAM was applied in automatic everything mode across all captured views, generating per-frame instance masks without manual prompting. DEVA was subsequently applied to propagate and associate mask identities temporally across the multi-view sequence. The combined pipeline achieved consistent object-level segmentation across the majority of frames, with degraded performance in early frames prior to DEVA's propagation buffer being established. The resulting label maps provided the identity supervision used in Gaussian Grouping training.

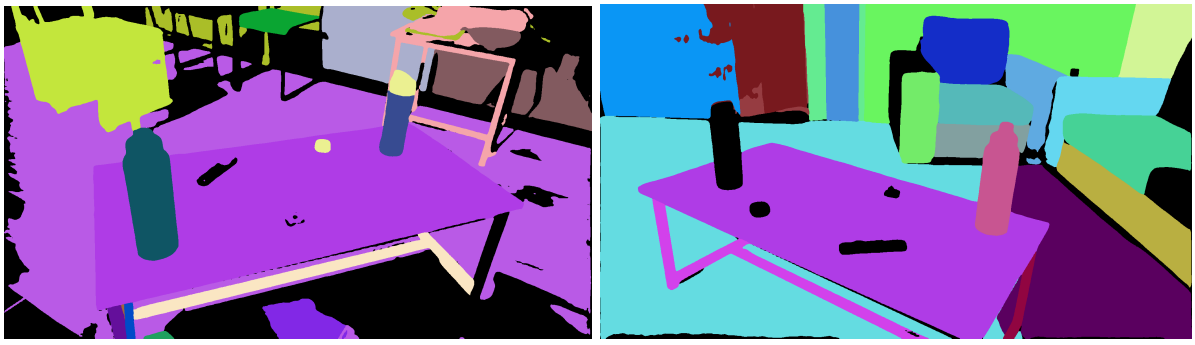

Figure: The above image showcases the masks generated using SAM.

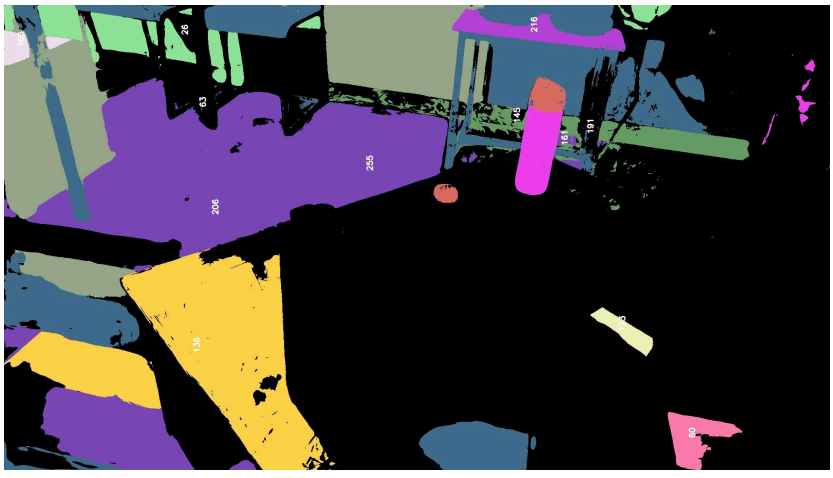

Figure: The above image showcases the Ids associated with the masks for Gaussian Splat Removal using DEVA.

Object Removal Results

Object removal via Gaussian Grouping achieved partial success: the ttarget object's Gaussian primitives were correctly identified and partially deleted from the scene. However, residual geometry remained visible in novel-view renders due to the failure modes described in Section 2.A.6. The removal operation itself was near-instantaneous

once training completed, confirming the efficiency advantage of identity-based deletion over scene retraining. Full, artifact-free removal would require temporally consistent video-mode segmentation and a 3D inpainting pass to fill the exposed background geometry.

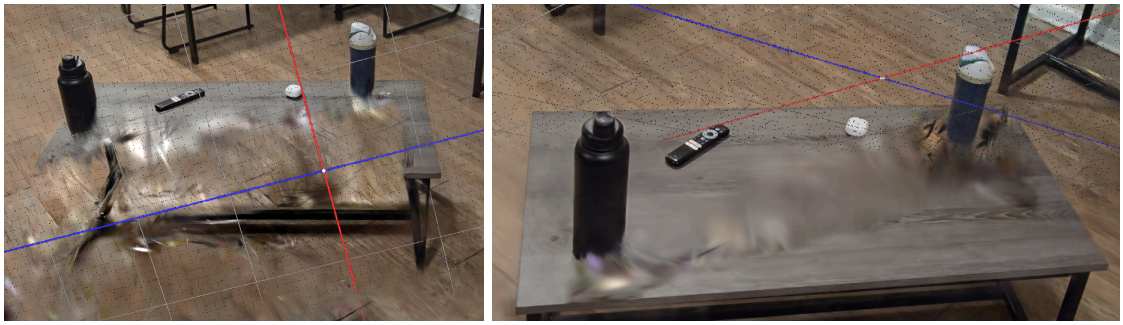

Figure: The above image showcases two separate attempts in removing the bottle object from the scene.

3.2 Basemap Evaluation Results

Table 1. Reports quantitative metrics across reconstruction variants on the Mip-NeRF 360 room scene. Metrics are PSNR (dB, higher is better) for pixel fidelity, SSIM (0–1, higher is better) for structural similarity, and LPIPS (lower is better) for learned perceptual distance.

Variant	Test PSNR	SSIM	LPIPS	Gaussians
Vanilla 3DGS	32.00 dB	0.9327	0.1874	1.16 M
Mip-Splatting	32.13 dB	0.9358	0.1778	1.20 M

Mip-Splatting improves on every metric relative to vanilla 3DGS, but the improvement margin (approximately +0.13 dB PSNR) is within noise on a 39-image test split. This is consistent with Mip-Splatting's mode of action: its primary benefit is in off-trajectory rendering, which is under-represented by single-viewpoint metrics on held-out training-trajectory images.

3.3 Dataset Quality as the Dominant Factor

The most consequential finding of the basemap evaluation was that input dataset quality dominates reconstruction quality across all algorithms tested. Table 2 reports results for the same Mip-Splatting pipeline applied to three different datasets spanning a range of capture conditions.

Table 2. Same Mip-Splatting pipeline applied to three different capture conditions.

Dataset	Setting	Test PSNR	SSIM	LPIPS
Replica	Indoor synthetic	39.17 dB	0.9762	0.0708
Mip-NeRF 360	Indoor natural photo	32.13 dB	0.9358	0.1778
Tanks & Temples	Outdoor natural photo	26.10 dB	0.8927	0.1579

The three datasets span approximately 13 dB PSNR on identical algorithms and parameters. SSIM rises from 0.89 to 0.98 and LPIPS falls from 0.18 to 0.07 as the capture moves from outdoor natural to indoor synthetic. This 13 dB inter-dataset spread is roughly two orders of magnitude larger than the 0.13 dB cross-variant spread on a single dataset. The practical implication is that deliberate capture planning — ceiling angles, overlap on textureless surfaces, uniform capture distances — represents a far greater investment return than further algorithmic tuning.

Note: Tanks & Temples was trained with tighter densification caps due to training-time constraints. A portion of its performance gap reflects the less-expressive Gaussian model rather than purely dataset effects.

Figure. Visual evidence that dataset quality dominates optimization, three captures rendered with the same Mip-Splatting + SuGaR pipeline. Top: Replica synthetic indoor, with an inserted mesh on the coffee table to demonstrate the merge pipeline. Middle: Mip-NeRF 360 room natural indoor. Bottom: Tanks & Temples outdoor truck.

3.4 Unity Integration Results

The fire hydrant Gaussian Splat was successfully imported into Unity using the aras-p GaussianSplatAsset plugin [7] and composited onto the Bonsai basemap as a single fused .ply asset. The alpha-erasure artifact caused by per-asset depth sorting was resolved by the merging pipeline described in Section 2.B.3. Basic interaction was validated using primitive colliders, with mesh-based colliders available as a higher-fidelity option once the nvdiffrec mesh is further refined. Physics testing confirmed stable object placement, correct gravity response, and reliable ground-plane contact for inserted assets.

Figure. Merged hydrant scene (Unity integration). The fire-hydrant splat composited onto the Bonsai basemap as a single fused .ply asset

4. Discussion and Conclusions

4.1 Key Findings

The central technical contribution of this project is a validated end-to-end pipeline from real-world video capture to composited Gaussian Splat scenes in a physics engine. Several important insights emerged:

- Dataset quality is the binding constraint on reconstruction quality. Across every algorithm variant tested, input capture conditions explained roughly two orders of magnitude more variance in PSNR than the choice of algorithm. Future work should prioritize systematic capture protocols over algorithmic improvements.

- The dual-representation approach (Gaussian Splat for appearance, mesh for physics) is viable on consumer hardware. The full pipeline from 4K smartphone video to Unity-ready assets was demonstrated on an RTX 4070 Laptop GPU with 8 GB VRAM, establishing that the framework does not require high-end server hardware.
- The .ply merging pipeline resolves the alpha-erasure artifact in multi-asset Unity scenes. By fusing independently reconstructed assets into a single Gaussian Splat Renderer prior to import, correct global depth sorting is achieved without any modification to the renderer plugin.
- NeRF-based refinement improves Poisson mesh surface continuity but introduces training overhead and reduces fine detail. The trade-off is most favourable for objects with complex geometry, such as the oil canister, where the additional smoothing outweighs the detail loss.
- Object removal via Gaussian Grouping requires temporally consistent 2D segmentation. Incomplete SAM masks and early-frame DEVA association errors were identified as the primary failure modes, both of which are addressable with video-mode SAM 2.

4.2 Limitations and Negative Results

- Object removal was only partially successful. Residual geometry remained after Gaussian deletion due to boundary identity ambiguity and missing mask coverage in early DEVA frames. A clean removal would require temporally consistent video segmentation and a 3D inpainting pass.
- Mesh quality was limited by VRAM and training resolution. The nvdiffrec and Poisson meshes broadly capture object silhouettes but lack fine surface detail. Higher-resolution training or a more expressive mesh representation would be needed for appearance-quality mesh rendering.
- Object replacement and repositioning were reserved as future work, as they require diffusion-model-based inpainting to fill voids left by removed objects.
- MASt3R-SLAM was found incompatible with the Windows deployment environment, and WSL deployment was computationally unstable. COLMAP was selected as the practical alternative, with no significant loss of reconstruction quality.

4.4 Conclusions

This project successfully demonstrated the core asset acquisition and scene composition stages of a synthetic data generation pipeline for vision-based automation. A fire hydrant was reconstructed as both a photorealistic 3D Gaussian Splat and a collision mesh from smartphone video alone, composited into a virtual environment in Unity, and rendered with correct multi-asset depth sorting via a custom merging pipeline. Quantitative evaluation of basemap reconstruction variants confirmed that input dataset quality is the dominant determinant of reconstruction quality, a finding with direct implications for future capture protocol design. Object removal via Gaussian Grouping was explored and partially realized, with incomplete SAM segmentation and early-frame DEVA errors identified as the primary failure modes and a clear path to resolution via video-mode segmentation. The NeRF-based refinement pipeline demonstrated improved surface continuity over direct Poisson reconstruction, at the cost of additional training time and some detail loss. Unity simulation testing 4 confirmed that Gaussian Splat assets can function as fully interactive scene objects when paired with physics proxy colliders. Together these contributions establish a practical foundation for scalable, automated synthetic dataset generation for vision-based industrial automation systems.

5. Statement of Individual Contribution

This project was a collaborative effort among four team members, coordinated via Discord for continuous communication and weekly Zoom sync meetings to review progress, align on tasks, and address challenges.

Vikram Raj Nagoor Kani: Developed the end-to-end asset acquisition pipeline including 4K video capture, COLMAP SfM reconstruction, REMBG background removal, 3DGS training, and nvdiffrec mesh generation. Investigated object removal using Gaussian Grouping with DEVA-assisted mask propagation and conducted detailed failure analysis. Responsible for the full object-level reconstruction pipeline from raw video to game-engine-ready assets.

James Butler: Investigated simulation engine integration and basemap reconstruction. Evaluated multiple 3DGS variants (vanilla 3DGS, Mip-Splatting, SuGaR) on multiple datasets and produced the quantitative basemap comparison. Developed the Unity integration workflow including the .ply merging pipeline to resolve alpha-erasure artifacts in multi-asset scenes. Responsible for the full basemap-to-Unity pipeline and quantitative evaluation.

Nathaniel Christie: Investigated map generation using MASt3R-SLAM and, upon identifying compatibility constraints, developed an alternative COLMAP-based reconstruction pipeline. Built and evaluated a COLMAP-to-NeRF-to-mesh pipeline using Nerfstudio for surface refinement, and assessed reconstruction robustness across multiple object types including a fire hydrant and an oil canister. Responsible for the NeRF-based refinement pipeline and comparative object evaluation.

Sumanth Prabhu: Investigated simulation evaluation and asset interaction in Unity. Developed and tested the workflow for inserting, positioning, scaling, and physically interacting with Gaussian Splat objects in a Unity scene. Implemented simplified proxy colliders to validate stability and collision behavior independently of final mesh fidelity. Responsible for the Unity physics integration and interactive scene-editing validation.

6. References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," ACM SIGGRAPH, 2023. https://arxiv.org/abs/2308.04079

[2] J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," CVPR, 2016. https://github.com/colmap/colmap

[3] J. Munkberg et al., "Extracting Triangular 3D Models, Materials, and Lighting From Images," CVPR, 2022. https://github.com/NVlabs/nvdiffrec

[4] A. Kirillov et al., "Segment Anything," ICCV, 2023. https://arxiv.org/abs/2304.02643

[5] M. Ye et al., "Gaussian Grouping: Segment and Edit Anything in 3D Scenes," 2023. https://arxiv.org/abs/2312.00732

[6] T. Ke et al., "Segment Anything in High Quality," NeurIPS, 2024.

[7] aras-p, "UnityGaussianSplatting," GitHub. https://github.com/aras-p/UnityGaussianSplatting

[8] Open3D, "Open3D Documentation." https://www.open3d.org/docs/release/

[9] H.-K. Chen et al., "Tracking Anything with Decoupled Video Segmentation (DEVA)," GitHub. https://github.com/hkchengrex/Tracking-Anything-with-DEVA

[10] E. Dexheimer, "MASt3R-SLAM." https://edexheim.github.io/mast3r-slam/

[11] D. Gatis, "REMBG: Background Removal Tool." https://github.com/danielgatis/rembg

[12] Nerfstudio, "Nerfstudio Documentation." https://docs.nerf.studio/

[13] P. Bourke, "PLY File Format." https://paulbourke.net/dataformats/ply/

[14] Scikit-learn, "DBSCAN Clustering." https://scikit-learn.org/stable/modules/clustering.html#dbscan

[15] Graphdeco-Inria, "3D Gaussian Splatting (Official Implementation)." https://github.com/graphdeco-inria/gaussian-splatting

[16] "Computer Vision Training Dataset Generation for Robotic Environments Using Gaussian Splatting." https://arxiv.org/abs/2512.13411

[17] K. Wei, "Gaussian Splatting Notes." https://github.com/kwea123/gaussian_splatting_notes

[18] "Mathematical Supplement for the gsplat Library." https://arxiv.org/pdf/2312.02121

[19] NVIDIA Research, "Asset Harvester." https://research.nvidia.com/labs/sil/projects/asset-harvester/

[20] Towards Data Science, "A Python Engineer's Introduction to 3D Gaussian Splatting (Part 2)." https://towardsdatascience.com/a-python-engineers-introduction-to-3d-gaussian-splatting-part-2-7e45b270c1df/