CS 4644/7643 Milestone Report: Sparse view 3D reconstruction with Gaussian splatting

Sri Siddarth Chakaravarthy
sp313@gatech.edu

Jinchu Li
jinchu.li@gatech.edu

Kunal Aneja
kaneja6@gatech.edu

Sohan Malladi
smalladi32@gatech.edu

Abstract

Sparse-view 3D reconstruction is a significant challenge in novel-view synthesis because of the limited availability of input images. This results in degraded reconstruction quality and incomplete geometry. With such a limitation, the practicality of advanced techniques such as Gaussian Splatting is drawn into question. These advanced techniques require extensive multi-view datasets in order to achieve accurate results. Our project aims to address this issue by introducing a novel three-stage architecture that is applicable to sparse-view scenarios. The first stage is a coarse 3D Gaussian Splatting model that is trained to capture the basic geometry of the scene from limited views. The seconds stage is a reconstruction model to generate pseudo-views in order to enhance the structural details and overall accuracy. The third stage is where Fisher Information is employed to identify and correct pose errors, which allows targeted parameter updates for further improvements.

Through our approach, we employ various key advancements such as using object structure priors and a Gaussian Repair model. Through our approach's performance against diverse datasets, it can be seen that our method significantly improves reconstruction quality. Our approach outperforms existing baselines in metrics such as LPIPS, SSIM, and PSNR. Through addressing the limitations caused by sparse-views, our approach expands the applicability of Gaussian Splatting to a variety of fields. Areas that require 3D reconstruction workflows like gaming, augmented reality, and filmmaking could see an improvement through our work.

1. Introduction

Novel-view synthesis has garnered significant attention in recent years due to its wide-ranging applications in various fields, including gaming, filmmaking, and augmented reality (AR). One of the leading techniques in this domain is Gaussian Splatting, known for its ability to generate high-quality 3D reconstructions efficiently. Unlike methods such as Neural Radiance Fields (NeRFs) [8], Gaussian Splatting projects 3D Gaussian representations onto 2D splats, making it more resource-efficient and suitable for real-time rendering scenarios.

1.1. Problem Statement and Importance

Despite its advantages, Gaussian Splatting faces significant challenges in sparse-view scenarios where limited input images are available. In real-world applications, collecting extensive multi-view datasets can be impractical or time-consuming, which poses a substantial barrier to the widespread adoption of these techniques. This limitation is particularly critical in contexts where rapid 3D content generation is required. Addressing the sparse-view issue is essential to enhance the efficiency and accessibility of 3D reconstruction processes, enabling broader applicability across various industries and research fields.

Current literature, including foundational works like 3D Gaussian Splatting [6] and recent advancements such as GaussianObject, has made considerable progress in novel-view synthesis and 3D reconstruction. However, these methods typically rely on a large number of input views to achieve high-quality reconstructions. In scenarios where data acquisition is constrained, such as limited access to multi-view images, the performance of these methods deteriorates, leading to artifacts and incomplete geometry. The inability to effectively handle sparse-view inputs restricts the broader applicability of Gaussian Splatting techniques. Therefore, improving Gaussian Splatting's performance in sparse-view conditions is crucial to overcome these limitations and facilitate more versatile and efficient 3D reconstruction workflows.

The primary aim of this project is to enhance the performance of Gaussian Splatting under sparse-view conditions. To achieve this, we propose a three-stage architecture. The first stage involves training a coarse 3D Gaussian Splatting model to capture the basic geometry of the scene from limited views. The second stage employs a large reconstruction model to generate pseudo-views, enhancing structural details and improving overall accuracy. Finally, in the third stage, Fisher Information is utilized to identify and correct errors in the novel view poses, enabling targeted parameter updates in the model. By integrating these stages, our approach seeks to produce high-quality 3D reconstructions from sparse-view inputs, thereby expanding the applicability of Gaussian Splatting in practical scenarios.

2. Related Literature

Gaussian Splatting has emerged as a powerful technique for novel-view synthesis and 3D reconstruction. In this section, we discuss the foundational work relevant to our research, focusing on two significant contributions in the field.

2.1. 3D Gaussian Splatting for Real-Time Radiance Field Rendering [6]

The concept of 3D Gaussian Splatting was introduced to address the need for efficient and high-quality rendering of radiance fields [6]. This method represents scenes using 3D Gaussians characterized by parameters such as position, opacity, anisotropic covariance, and spherical harmonic coefficients for color representation. Starting with sparse points from camera calibration, the method initializes the scene with 3D Gaussians. This enables a continuous volumetric radiance field representation, while avoiding computational overhead in empty spaces. Furthermore, the technique dynamically adjusts the density of Gaussians through processes like densification and splitting based on positional gradients, ensuring a compact yet precise representation of the scene. It also employs a tile-based rasterizer for efficient real-time rendering by sorting 3D Gaussians in screen space and performing α-blending, leveraging GPU acceleration for fast rendering and backpropagation. The optimization process minimizes a combined loss function that incorporates loss L_1 and a structural similarity term, ensuring that the rendered images are consistent with training views. This method achieves high visual quality with competitive training times and allows real-time rendering at 1080p resolution.

Despite its strengths, this approach faces limitations in scenarios with sparse input views. The method relies on sufficient multiview coverage to optimize the 3D Gaussians accurately. When the number of input images is limited, it struggles to reconstruct unseen regions, leading to artifacts or incomplete geometry. Additionally, it may not accurately capture complex view-dependent lighting effects in sparse-view settings, affecting the realism of rendered images. These limitations highlight the need for enhanced techniques that can improve Gaussian Splatting's performance under sparse-view conditions.

Our research aims to address these limitations by introducing strategies that enhance reconstruction quality from limited inputs. By incorporating pseudo-view generation and error correction mechanisms, we build upon the foundational concepts of the original 3D Gaussian Splatting method, aiming to extend its applicability to real-world sparse-view scenarios.

2.2. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis [8]

NeRF or Neural Radiance Fields is a framework that is able create high quality novel views of scenes from a limited set of input images [8]. The unique approach of NeRF is that it represents a scene as a continuous 5D function. This function maps spatial coordinates (x, y, z) and viewing directions (θ, ϕ) to emitted radiance and volume density. It is able to accomplish this by using a fully connected neural network. In order to render an image from this representation, NeRF uses volume rendering. It samples points along the camera rays, queries the network for radiance and density, and combines the outputs to create a 2D image. By decreasing the difference between created and input images, NeRF's rendering pipeline allows various levels of optimization. NeRF also uses two techniques to improve the quality of the outputs and the efficiency in which the outputs are produced. The first technique is positional encoding, which maps input coordinates to higher-dimensional spaces. The second technique is hierarchal sampling which distributes computational resources to areas with visible content. Through these various approaches and techniques, NeRF achieves state-of-the art results and outperforms other models and approaches for view generation. Despite these various strengths, NeRF has some limitations. The first limitation is that training is extremely computationally expensive and inference is very slow because of the dense sampling along rays. NeRF also relies on precise camera poses, which can make it unreliable in a lot of real world use cases where the data will not be as fleshed out and have clean pose data.

For our project, NeRF provides many beneficial approaches that align with our goals. The first is NeRF's ability to produce detailed geometric and appearance information for complex scenes. This aligns with out goal of trying to generate 3D representations with sparse views. The second beneficial approach is NeRF's hierarchal sampling and positional encoding which align with our goal of capturing fine details.

2.3. SparseNeRF: Distilling Depth Ranking for Few-shot Novel View Synthesis

SparseNeRF enables neural radiation fields (NeRFs) to create novel high-quality views from sparse input views [11]. This is able to combat the traditional pitfalls of NeRF. SparseNeRF is able to do this because a variety of advancements that it made. The first is to use local depth ranking regularization, which employs relative depth comparisons instead of absolute depth values. This ensures that robot supervision is used even when there is noisy input data. Another advancement that SparseNeRF has made is introducing spatial continuity regularization to keep geometric coherence. This distills depth-continuity priors from the input maps. These advancements offer a plethora of practical advancements that give reason for SparseNeRF's popularity. The first is that it does not increase the inference time because the depth priors are only used during training. In addition, the modular design allows it to integrate perfectly with the different NeRF implementations. Another advancement is the general idea of the project, which is that in real-world circumstances, dense-view data are not always available, thus SparseNeRF thrives with any density of data.

Although SparseNeRF provides an array of positives, there are some limitations. The first is that in areas that are completely blocked or unobserved in the sparse input views, the created novel views will not have the correct geometric details. This is a pitfall because the model depends on the information in the training data to produce these novel views. Another limitation is that due to SparseNeRF relying on depth ranking regularization, highly noisy or inconsistent depth inputs can lead to inaccuracies that are hard to detect. SparseNeRF also assumes that there is spatial consistency amongst neighboring pixels, which is not always the case. This can lead to oversmoothing in areas with fine details.

In the context of our project, SparseNeRF's approach to depth ranking and spatial continuity aligns with our project's goals heavily. By employing the techniques described above, our project aims to further enhance the quality of sparse-view 3D reconstruction while ensuring that we do not have the same limitations as SparseNeRF. For example, we aimed to address the presence of unobserved regions by integrating more advanced estimation model or combining SparseNeRF with additional scene priors.

2.4. SparseGS: Real-Time 360° Sparse View Synthesis Using Gaussian Splatting

SparseGS extends Gaussian Splatting to enable real-time novel-view synthesis in 360-degree unbounded scenes from sparse input views [**?**]. The method addresses challenges inherent in few-shot settings, such as background collapse and the presence of floaters—artifacts that occur due to overfitting to limited training views. To mitigate these issues, SparseGS integrates depth priors obtained from pre-trained monocular depth estimation models and employs explicit constraints to enhance scene consistency from unseen viewpoints. A novel floater pruning technique is introduced, which directly identifies and removes artifacts in the 3D Gaussian representation by leveraging differences between mode-selected and alpha-blended depth maps. Additionally, the method utilizes score distillation sampling with diffusion models to refine regions with insufficient coverage in the training data. Experimental results demonstrate that SparseGS significantly outperforms baseline Gaussian Splatting and NeRF-based methods in terms of perceptual quality metrics such as LPIPS and PSNR, while also reducing training and inference times. This work underscores the potential of combining explicit scene representations with depth and generative priors to improve sparse-view reconstruction, aligning closely with our project's goals to enhance Gaussian Splatting in sparse-view scenarios.

2.5. InstantSplat: Sparse-View SfM-Free Gaussian Splatting in Seconds

InstantSplat presents an efficient method for novel-view synthesis from sparse, unposed images by eliminating the dependency on traditional Structure-from-Motion (SfM) preprocessing [3]. The framework integrates multi-view stereo (MVS) predictions with point-based scene representations to rapidly construct 3D Gaussians. By generating densely populated surface points across all training views and estimating initial camera parameters through pixel alignment, InstantSplat addresses the limitations posed by SfM in sparse-view scenarios, where feature matching is often unreliable. To handle the excessive number of Gaussians resulting from pixel-wise predictions—which can compromise both training speed and accuracy—the method employs a grid-based, confidence-aware farthest point sampling strategy. This approach strategically reduces redundancy by positioning point primitives at representative locations in parallel. Additionally, InstantSplat refines pose estimation and scene parameters through a gradient-based joint optimization framework under self-supervision. This leads to a significant reduction in training time—from hours to mere seconds—while maintaining robust performance across various datasets and numbers of views. InstantSplat's ability to efficiently handle sparse-view inputs without relying on SfM aligns closely with our project's objectives to enhance Gaussian Splatting in sparse-view conditions.

3. Methodology

We propose the idea of leveraging physical information to improve the notion of scene structure. In particular, we focus on introducing object structure priors, e.g. the basic/rough geometry of the object, to help build multi-view consistency, including visual hull to locate Gaussians within the object outline and floater elimination to remove outliers. This has helped us to tackle the sparse view problem, and propose a framework to reconstruct high-quality 3D objects from as few as 3-4 input images.

To erase artifacts caused by omitted or highly compressed object information, a Gaussian repair model is proposed which is driven by 2D large diffusion models that translate corrupted rendered images into high-fidelity ones. Normal diffusion models are unable to reconstruct corrupted images we use a self-generating pair of images strategy to construct the images to tune more advanced diffusion models and adding 3D noises to Gaussian attributes. This repair model is used to refined the 3D Gaussians optimized with structure priors, where the rendering quality can be further improved.

To overcome the issue of limited 3D information for reconstruction from sparse training data. (SfM points are often absent). We adapt two techniques to initially optimize the 3D Gaussian representation, which take full advantage of structure priors from the limited views and result in a satisfactory outline of the object.

Initialization with Visual Hull

1. Utilize the view frustums and object masks to create a visual hull that serves as a geometric scaffold for initializing 3D Gaussians.
2. The visual hull provides structure priors that assist in building multi-view consistency by excluding unreasonable Gaussian distributions. It is acquired via SAM.
3. Points are randomly initialized within the visual hull using rejection sampling (i. Uniformly sample random 3D points and project them onto image planes ii. Retain points that fall within the intersection of all image-space masks.)
4. Assign colors to points by averaging bilinearly interpolated pixel colors across reference image projections.
5. Transform these 3D points into 3D Gaussians for the initialization of the model.

Floater Elimination

1. Recognize that the visual hull provides a coarse estimation of the object geometry, which may include regions that do not belong to the object due to limited reference image coverage.
2. These extraneous regions, known as floaters, degrade the quality of novel view synthesis.
3. To address this, utilize the K-Nearest Neighbors (KNN) algorithm to calculate the average distance to the nearest $\sqrt{P}$ Gaussians for each Gaussian in the set G_c.
4. Apply adaptive thresholding (τ) based on mean and standard deviation distances to exclude Gaussians classified as floaters.
5. Periodically repeat the thresholding process throughout optimization, gradually reducing τ linearly to zero to refine the scene representation.

3.1. Gaussian Repair Model ($\mathcal{R}$)

To generate information of regions that have inadequate Gaussian distributions in the scene we initally test a repair model $\mathcal{R}$ by 2D diffusion models, which takes corrupted rendered images $x'(\mathcal{G}_c, \pi^{\text{nov}})$ as input and outputs photorealistic and **high-fidelity images** $\hat{x}$.

Figure 1. Repaired views using ControlNet

3.2. Fisher Information for Reconstruction Gain

In our method we adapt fisher information for active view selection and quantification by employing Fisher Information in Radiance Fields [2, 5, 10] with this approach, we can tell how much informational value a possible new view will provide. This will enable efficient selection of views and ensure that the most amount of data is used. This helps us to select the next best view and pixel-wise uncertainty quantification. This approach combats the problems of the inability to quickly and efficiently select the next view, while also taking into account pixel-wise uncertainty management. It helped us identify which new views will provide the best value in our sparse data situations.

4. Baselines & Experiments

Baseline 1: Vanilla Gaussian Splatting (3DGS)

The vanilla 3DGS focuses on three key components: 3D Gaussian representation, adaptive density control, and fast tile-based rasterization. Starting with sparse points from camera calibration, the method initializes the scene with

3D Gaussians, preserving the properties of continuous volumetric radiance fields while avoiding computationally expensive regions devoid of content [4, 9]. Gaussians are characterized by parameters such as position, opacity α, anisotropic covariance, and spherical harmonic (SH) coefficients for color representation. The adaptive density control mechanism dynamically adds or removes 3D Gaussians during optimization to achieve a compact yet precise representation of the scene. This process includes two primary strategies: densification and splitting of Gaussians based on positional gradients. Densification targets under-reconstructed areas, while splitting is applied to regions of high variance, ensuring that both geometry and density adapt to the scene's structure [1].

The tile-based rasterizer enables efficient real-time rendering by sorting 3D Gaussians in screen space and performing α-blending in a visibility-aware manner. This approach leverages fast GPU sorting and is inspired by tile-based rasterization [7]. During the forward pass, the rasterizer projects 3D Gaussians to 2D splats, enabling real-time rendering with low memory consumption. The optimization process minimizes a combined loss function, which incorporates L_1 loss and a **D-SSIM** term:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}$$

where $\lambda = 0.2$ is used in all tests. This loss function ensures that the rendered images from novel views are consistent with training views, capturing both pixel-level and structural differences [12].

Baseline 2: CoR-GS

CoR-GS uses co-pruning and pseudo-view co-regularization to improve on 3D Gaussian Splatting [13]. Co-pruning identifies Gaussians with large discrepancies between independently trained radiance fields and removes these inconsistent Gaussians. Pseudo-view co-regularization, on the other hand, creates interpolated views from the current training views. Using these pseudo-views, it measures the rendering disagreements between the two radiance fields. Inaccuracies are characterized by large discrepancies in rendering, which are suppressed by incorporating them into the training loss function. This strategy ensures that reconstructed geometry is consistent and non-conflicting. CoR-GS's approach to refined geometry aligns with our three-stage approach.

The CoR-GS loss function combines supervised color reconstruction at the training view with pseudo-view regularization, represented by rendering differences at pseudo-views:

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \lambda_p \mathcal{R}_{\text{pcolor}}, \tag{1}$$

where $\mathcal{L}_{\text{color}}$ is a combination of L_1 loss and D-SSIM for image reconstruction at training views, and $\mathcal{R}_{\text{pcolor}}$ is the pseudo-view co-regularization term. The weight λ_p balances the supervised and pseudo-view losses. This loss function encourages consistency between radiance fields across different views, suppressing errors that may arise in sparse-view scenarios.

These works provide us with great dive into our approach into enhancing 3D Gaussian Splatting for sparse-view 3D construction. Our methodology can be executed by incorporating visual hull techniques for initial geometry estimation, using MVS for dense point cloud creation, implementing Fisher Information for active view selection, and utilize co-pruning and pseudo-view co-regularization from CoR-GS. Taking inspiration from various implementations will enable us to produce high quality 3D reconstructions from sparse-view inputs.

Our Method

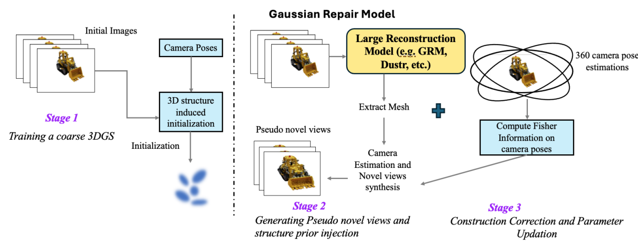

Figure 2. Proposed Method

To construct the visual hull representation of objects in 3D scenes, we utilize the Segment Anything Model (SAM) to generate high-quality binary masks from input images. The visual hull relies on silhouettes extracted from multiple viewpoints, making precise and consistent mask generation critical to capturing the underlying object geometry. SAM's ability to generalize across diverse objects and scenes, combined with its promptable architecture, allows it to adapt seamlessly to the varied image inputs in our pipeline. By leveraging SAM, we automate the extraction of object boundaries from images, significantly reducing the need for manual intervention or dataset-specific fine-tuning.

The extracted masks from SAM are employed as inputs to the visual hull algorithm, where they define the spatial regions occupied by the object in each viewpoint. These silhouettes are intersected across views to reconstruct an approximate 3D shape, serving as a robust initialization for our Gaussian-based representation. Furthermore, SAM's precise segmentation capabilities ensure minimal noise in the masks, which directly improves the fidelity of the resulting visual hull. This approach not only streamlines the preprocessing pipeline but also enhances the quality and efficiency of downstream 3D reconstruction tasks, particularly in scenes with complex geometries or occlusions. The use of SAM for mask generation thus plays a pivotal role in achieving accurate and scalable 3D modeling, bridging the gap between 2D image data and high-quality 3D representations. These masks can be seen in the figure below.

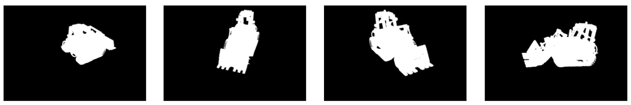

Figure 3. Object masks generated from SAM

We propose using **Fisher Information** to guide active view selection for repairing 3D reconstructions. By quantifying the information content of candidate views via the Fisher Information matrix, we identify views that maximize uncertainty reduction. The Fisher Information matrix is defined as:

$$I(w) = -\mathbb{E}_{y|x}\left[\frac{\partial^2 \log p(y|x,w)}{\partial w^2}\right], \tag{2}$$

where w are model parameters, and y, x denote the image and camera pose, respectively. It is approximated as:

$$H''[y|x,w^*] \approx \nabla_w f(x;w^*)^\top \nabla_w f(x;w^*), \tag{3}$$

enabling efficient computation without requiring ground truth.

View Repair with Gaussian Splatting

After selecting the optimal view, reconstruction proceeds through a Gaussian-based pipeline inspired by **GaussianObject**. A coarse 3D Gaussian representation is initialized using visual hull priors, parameterized as:

$$G_i = \{\mu_i, q_i, s_i, \sigma_i, sh_i\}, \tag{4}$$

where $\mu_i, q_i, s_i, \sigma_i, sh_i$ denote the center, rotation, scale, opacity, and spherical harmonics, respectively.

The rendering equation is then used to refine the representation:

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t),d)\,dt, \tag{5}$$

where $T(t)$ is the accumulated transmittance, σ the opacity, and $c(r(t),d)$ the radiance given direction d.

Depth estimation is a crucial step in reconstructing accurate 3D geometry, and our trained model provides precise depth predictions as an intermediate output. Leveraging the learned parameters of the model, depth values are estimated for each pixel by evaluating the geometry and appearance consistency across multiple views. The depth map is derived from the Gaussian-based representation or the optimized radiance field, where each Gaussian or voxel contributes to the overall scene depth based on its density and opacity along a ray. By integrating depth cues from all available views, the model effectively resolves ambiguities caused by occlusions or sparse inputs.

The resulting depth maps are used to refine the 3D reconstruction further, ensuring alignment between the reconstructed geometry and the input images. These maps not only capture fine-grained surface details but also provide a global understanding of scene structure, which is vital for downstream tasks such as mesh extraction and rendering. Additionally, the robustness of the model allows it to handle challenging scenarios, such as textureless regions or varying illumination, delivering high-quality depth predictions. This depth estimation capability enhances the pipeline's ability to reconstruct detailed and realistic 3D models while maintaining computational efficiency. Depth estimations of views have been shown in the image below.

Figure 4. Depth maps generated from trained model

Our approach integrates Fisher Information with the GaussianObject framework, ensuring that selected views provide maximal information gain. This combination allows efficient initialization, repair, and iterative refinement of 3D representations, achieving high-quality reconstructions even with sparse input views.

4.1. Evaluation Metrics

4.1.1 Datasets

Since there are no sparse-view 3D object datasets, we leverage existing 3D reconstruction datasets that have been used for benchmarking in prior research. We sample images randomly from these datasets to create a sparse-view version of the scenes for training. We incorporate different dataset versions with varying numbers of training views: 5 views, 10 views, 15 views, etc., enabling ablation studies.

- **Mip-NeRF 360**: Contains 9 high-resolution scenes (5 outdoor, 4 indoor) with 100–330 images each, featuring central objects with detailed backgrounds.

- **NeRF Synthetic Dataset**: Multiple 3D scenes rendered from synthetic objects with simple backgrounds, offering a variety of views per object.

4.1.2 Evaluation Metrics

When running our experiments we used the below metrics to evaluate our method against the baselines.

- **PSNR**: Measures reconstruction quality by comparing predicted and reference images.
- **SSIM**: Assesses structural similarity, focusing on luminance, contrast, and structure.
- **LPIPS**: Evaluates visual similarity using deep features.

4.2. Benchmarks

For the benchmarks we compare our method with one of the SOTA methods in sparse-view reconstruction along with the vanilla gaussian splatting method.

Below are tables which demonstrate our method's performance when compared to other SOTA methods in sparse-view reconstruction.

Model	PSNR (dB)	SSIM	LPIPS
Vanilla 3DGS	20.3	0.69	0.09
Cor-GS	23.1	0.73	0.12
Ours	**23.6**	**0.74**	**0.12**

Table 1. Performance metrics on the kitchen scene (Mip-Nerf360 dataset)

Model	PSNR (dB)	SSIM	LPIPS
Vanilla 3DGS	15.3	0.49	0.06
Cor-GS	21.1	0.53	0.11
Ours	**21.4**	**0.56**	**0.12**

Table 2. Performance metrics on the lego scene(lego synthetic dataset)

From the results we can see that by leveraging Fisher Information to guide the selection of views for refining 3D reconstructions we are able to quantify the information content of candidate views. This helps us to maximize the reduction of uncertainty in model parameters.

4.3. Mesh Extraction using SDFs

To enable editable mesh extraction, we incorporate SDF-based surface alignment. The density function derived from Gaussians is:

$$d(p) = \sum_g \alpha_g \exp\left(-\frac{1}{2}(p-\mu_g)^T \Sigma_g^{-1}(p-\mu_g)\right), \quad (6)$$

where $\alpha_g, \mu_g, \Sigma_g$ are the Gaussian opacity, center, and covariance. The corresponding SDF is:

$$f(p) = \pm s g^* \sqrt{-2\log(d(p))}, \quad (7)$$

where g^* is the Gaussian contributing most to $d(p)$. Points on the surface level set $d(p) = \lambda$ are sampled to extract a mesh using Poisson reconstruction. The extracted mesh is refined by binding new Gaussians aligned with the mesh triangles. Gaussian parameters such as scale and orientation are optimized iteratively:

$$R = \frac{1}{|P|}\sum_{p\in P} |f(p) - f_{\text{target}}(p)|, \quad (8)$$

where P is the set of sampled points. This strategy allows efficient rendering, editing, and high-quality reconstruction.

5. Conclusion

We present a framework combining Fisher Information for active view selection with Gaussian-based reconstruction and SDF-guided mesh extraction. This pipeline enables efficient 3D scene reconstruction and high-quality, editable mesh generation, suitable for rendering, editing, and animation. By leveraging Fisher Information for optimal view selection and SDF alignment for mesh refinement, our method bridges the gap between sparse input data and realistic 3D models, with potential applications in computer graphics, AR/VR, and robotics. Future work may extend this to dynamic scenes and real-time refinement.

Team Contributions

Team Member	Contributions
Kunal Aneja	Implemented core methods, handled active view selection and writing report..
Sri Siddarth Chakaravarthy	Developed algorithms, integrated visual hull techniques and writing report..
Jinchu Li	Environment setup, conducted baseline experiments and writing the report.
Sohan Malladi	Conducted experiments, analyzed performance metrics, designed poster and writing report.

Table 3. Summary of each team member's contributions to the project.

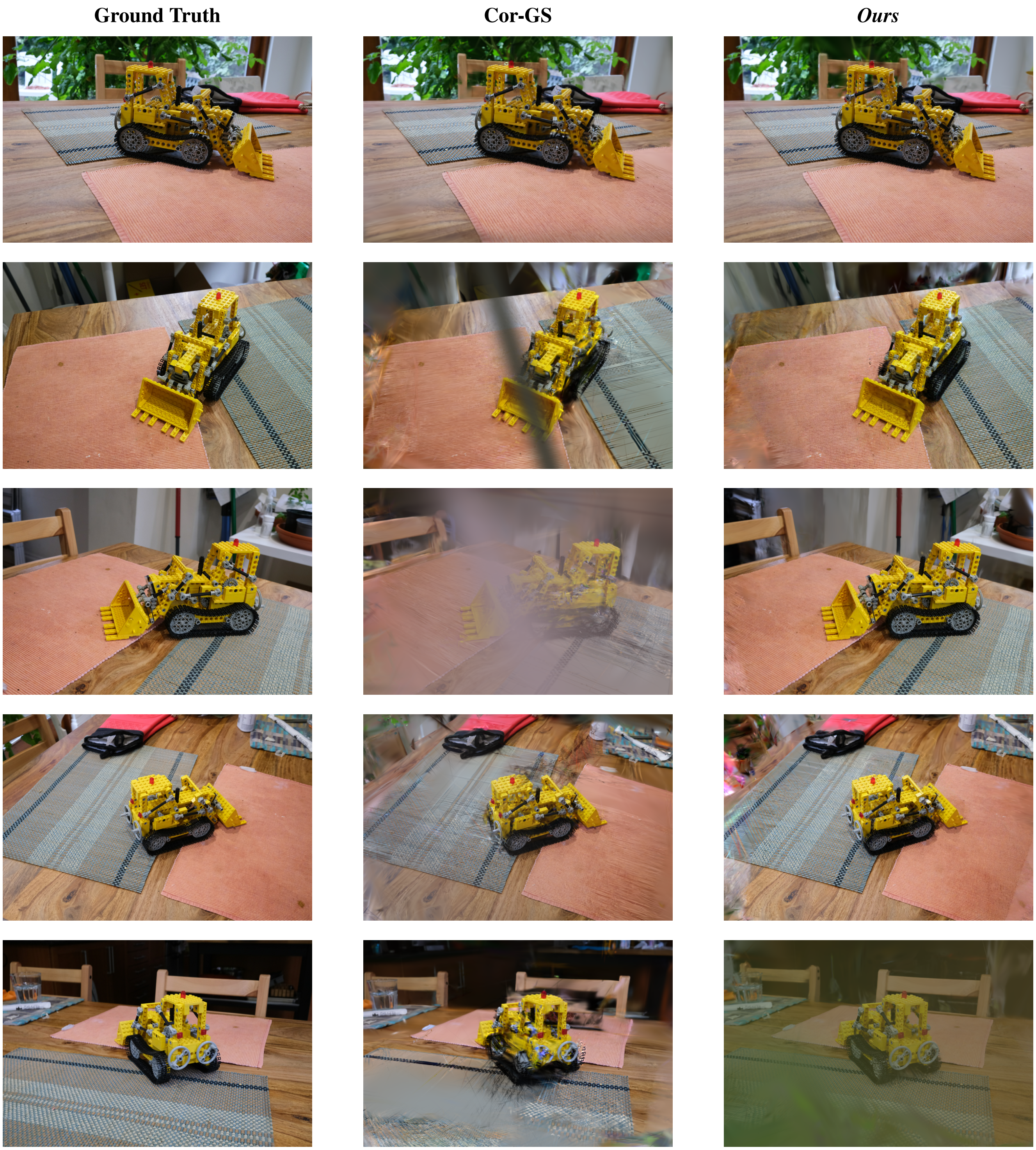

Figure 5. Qualitative results comparing baseline Cor-GS against our method on the mip-nerf 360 kitchen scene. This is a sparse-view setting with 15 training views.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2022.

[2] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024.

[3] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Sparse-view sfm-free gaussian splatting in seconds. *arXiv preprint arXiv:2403.20309*, 2024.

[4] Sierra Fridovich-Keil, Alex Yu, et al. Plenoxels: Radiance fields without neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[5] Kaiwen Jiang, Yang Fu, Yash Belhe, Xiaolong Wang, Hao Su, Ravi Ramamoorthi, et al. A construct-optimize approach to sparse view synthesis without camera pose. *arXiv preprint arXiv:2405.03659*, 2024.

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.

[7] Christoph Lassner and Michael Zollhöfer. Pulsar: Efficient sphere-based neural rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–10, 2021.

[8] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[9] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022.

[10] Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. Coherentgs: Sparse novel view synthesis with coherent 3d gaussians. *arXiv preprint arXiv:2403.19495*, 2024. Available: `https://arxiv.org/abs/2403.19495`.

[11] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. *arXiv preprint arXiv:2303.16196*, 2023.

[12] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[13] Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. Cor-gs: Sparse-view 3d gaussian splatting via co-regularization. *arXiv preprint arXiv:2405.12110*, 2024.