Gaussian Splatting in Mirrors: Reflection-Aware Rendering via Virtual Camera Optimization

Zihan Wang[1]
zihan.1.wang@aalto.fi
Shuzhe Wang[1]
shuzhe.wang@aalto.fi
Matias Turkulainen[1,3]
matias.turkulainen@aalto.fi
Junyuan Fang[1]
junyuan.fang@aalto.fi
Juho Kannala[1,2]
juho.kannala@aalto.fi

[1] Aalto University
[2] University of Oulu
[3] ETH Zurich

Abstract

Recent advancements in 3D Gaussian Splatting (3D-GS) have revolutionized novel view synthesis, facilitating real-time, high-quality image rendering. However, in scenarios involving reflective surfaces, particularly mirrors, 3D-GS often misinterprets reflections as virtual spaces, resulting in blurred and inconsistent multi-view rendering within mirrors. Our paper presents a novel method aimed at obtaining high-quality multi-view consistent reflection rendering by modelling reflections as physically-based virtual cameras. We estimate mirror planes with depth and normal estimates from 3D-GS and define virtual cameras that are placed symmetrically about the mirror plane. These virtual cameras are then used to explain mirror reflections in the scene. To address imperfections in mirror plane estimates, we propose a straightforward yet effective virtual camera optimization method to enhance reflection quality. We collect a new mirror dataset including three real-world scenarios for more diverse evaluation. Experimental validation on both Mirror-Nerf and our real-world dataset demonstrate the efficacy of our approach. We achieve comparable or superior results while significantly reducing training time compared to previous state-of-the-art. We release our code as open-source at: https://github.com/rzhevcherkasy/BMVC24-GSIM.

1 Introduction

3D Gaussian Splatting (3D-GS) [7] has recently made significant advancements in the field of novel view synthesis (NVS) [13, 14, 28, 29, 34, 35] and scene reconstruction [1, 10, 11, 15, 25, 30]. The method employs an explicit Gaussian based scene representation and novel differentiable rasterization algorithm, enabling high fidelity rendering quality that rivals that

of Neural Radiance Field (NeRF) [17] based methods, while significantly reducing rendering times. Although 3D-GS performs well in NVS, it encounters difficulties in handling specular reflections, particularly when mirrors or other reflective objects are present in the scene. The method tends to misinterpret reflections as virtual spaces behind mirrors, leading to multi-view inconsistencies. This results in blurry and disordered renderings of mirrors and their edges, which compromises the overall quality of the novel view synthesis.

Prior work based on NeRF has explored reflection-aware rendering to tackle specular reflection. Ref-NeRF [24] substitutes the orginal NeRF's ray-marching parametrization of view-dependent radiance with a representation that includes reflected radiance, aiming to directly account for reflective surfaces during optimization. MS-NeRF [31] represents mirror reflections as a group of feature fields in a parallel sub-space, enhancing the model's ability to handle complex reflective scenarios. NeRFReN [3] instead models both transmitted and reflective components in a scene by two separate radiance fields, providing a more nuanced depiction of light interactions. Among these advancements, Mirror-NeRF [32] stands out as the closest state-of-the-art method. It estimates the direction of reflected rays in the scene and employs Whitted Ray Tracing [27] for a more physically accurate ray-tracing model for reflective scenes. Despite these innovations, NeRF-based approaches are severely limited by their long training times and inability to achieve real-time rendering speeds on common devices.

In 3D-GS literature, the issue of reflections has not been extensively addressed. Concurrent work Mirror-3DGS [16], adopts a virtual camera to render mirror regions, but Mirror-3DGS requires additional depth supervision tailored specifically for mirrors. Another concurrent work, MirrorGaussian [12], leverages dual-rendering strategy to render both real-world Gaussians and the virtual ones from the mirror space. In contrast, our method solves the reflection problem using virtual camera rendering. Furthermore, current research into mirror reflections encounters significant challenges due to a lack of diverse real-world scene data. Although Mirror-NeRF includes three real-world indoor scenes featuring mirrors, the dataset is limited as the mirrors have the same size and shape, which does not fully represent the complexity of diverse real-world environments. This limitation highlights the need for more varied datasets to better understand and address the challenges in reflection rendering in real-world settings.

To address the issue of mirror reflections in 3D Gaussian Splatting (3D-GS), we develop a progressive training pipeline for mirror rendering that utilizes virtual camera rendering inspired by [19]. We first predict a mirror plane equation in world coordinates through depth and normal estimations in 3D-GS, which enables us to further derive the virtual camera pose. A rendered image of the 3D-GS scene containing a mirror in view is then created as a fusion of a reflected image obtained from the virtual camera render and a non-reflected image which is obtained by traditional 3D-GS rasterization into the current camera view. To enhance the quality of the virtual camera rendering, the predicted mirror plane is refined by virtual camera pose optimization during optimization. To address the problem of insufficient real-world scene data, we collected a new dataset containing three real-world scenes with mirrors. Our dataset encompasses scenes of varying scales, incorporating mirrors of different sizes and shapes. We believe this new dataset can progressively provide training and testing environments ranging from simple to complex, thereby better facilitating the evaluation of method effectiveness.

Our contributions can be summarized as follows:

- We propose a rendering method based on 3D-GS for scenes containing mirror reflec-

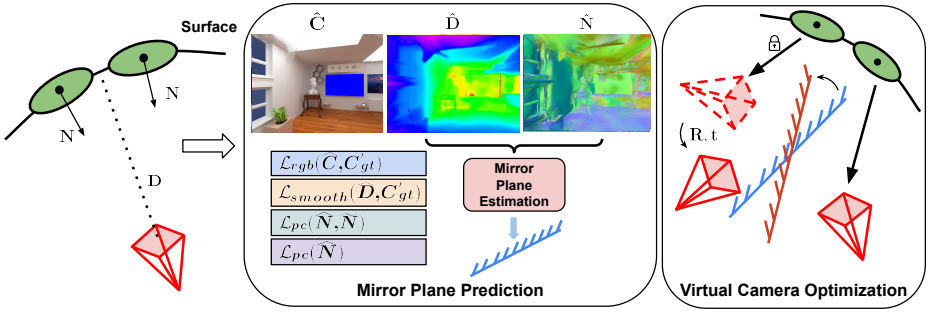

Figure 1: Pipeline Overview: We extend 3D-GS with depth and normal supervision to initialize a mirror plane in Sec. 2.3. Blue region in $\hat{C}$ indicates the mirror. We render both real and virtual camera viewpoints and combine them into a single image in Sec 2.2. We further refine virtual camera positions during optimization to achieve photo-realistic mirror reflections in Sec. 2.4.

tions. Mirror reflections are explained by renders onto virtual cameras placed around a mirror plane. Output images of camera views containing mirrors are a fusion of virtual camera renders and traditional 3D-GS renders. Our method achieves high-quality mirror reflection rendering and maintains the real-time rendering capabilities of 3D-GS with minimal added memory or computation needs.

- We present a dataset featuring real-world scenes with mirrors, encompassing various scales and mirror sizes. Unlike existing datasets, ours progressively includes mirror scenes of increasing complexity, offering challenging scenes for evaluation of the method's effectiveness.
- Experimental results on both synthetic and real-world scenes demonstrate that our method matches or even exceeds state-of-the-art techniques across multiple metrics while preserving the high rendering quality of 3D-GS.

2 Method

This section details the method of the proposed pipeline. We first review the 3D Gaussian Splatting [7] in Sec. 2.1. Sec 2.3 introduces the mirror plane calculation from the estimated depth and normal maps. Sec 2.2 defines the virtual camera pose by the estimated mirror plane equation and Sec 2.4 refine the mirror plane via the virtual camera optimization. The training objective is presented in Sec 2.5. The main pipeline is illuminated in Fig. 1.

2.1 3D Gaussian Splatting

We build our method on 3D Gaussian Splatting [7] which represents a scene by a collection of differentiable Gaussian distributions $\mathcal{G}_i = \{(\mu_i, \Sigma_i, O_i, \theta_i)\}_i$, with means and covariance matrices (μ_i, Σ_i), opacities O_i, and view-dependent colours $C_i \in \mathcal{C}^{N_{\text{sh}}}$ represented via spherical harmonic coefficients N_{sh}. To render individual views from the Gaussian scene represen-

tation, Gaussians are projected into screen space as 2D Gaussians using the current camera view matrix, sorted by z-depth, and alpha-blended to produce pixel colors $\hat{C}$:

$$\hat{C} = \sum_{i \in N} c_i \alpha_i T_i, \text{ where } T_i = \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{1}$$

where T_i is the accumulated transmittance at the rendered pixel p and α_i consists of the i^{th} Gaussian's blending term located in view space with position $\hat{\mu}_i$:

$$\alpha_i = O_i \cdot \exp\left(\frac{1}{2}(p - \hat{\mu}_i)\Sigma_i^{-1}(p - \hat{\mu}_i)\right). \tag{2}$$

The process of projection and rendering is parallelized allowing for real-time rendering.

2.2 Virtual Camera Definition

Prior work [32] address mirror reflections in the NeRF context using Whitted-Style Ray Tracing. However, performing ray tracing on a 3D-GS scene is computationally expensive, necessitating the exploration of alternative methods. Inspired by [19], we observe that a region of an image from a real camera containing a reflection from a mirror can be explained by an alternative image obtained by a virtual camera placed symmetrically about a mirror plane. In fact, with perfect mirror reflections, the image seen of a reflection from the real camera and the image seen from that of the virtual camera are identical. The virtual camera is symmetrically posed behind the mirror as shown in Fig. 2 and views the same Gaussian scene.

Given a real camera viewpoint with the extrinsic matrix $[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3\times4}$ and the mirror plane $[\mathbf{n}, o]$ (discussed in Sec. 2.3), we obtain the extrinsic matrix of virtual camera $[\mathbf{R}'|\mathbf{t}'] \in \mathbb{R}^{3\times4}$ following [19]:

$$\begin{bmatrix} \mathbf{R}' & \mathbf{t}' \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} - 2\mathbf{n}\mathbf{n}^T & 2o\mathbf{n} \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}.$$

Once the virtual camera pose is determined, we render the mirror-reflection image using this virtual camera and standard 3D-GS forward rasterization. Non-reflective regions are rendered by the corresponding real camera. We fuse both regions using segmented mirror masks [9] to generate the output image $\hat{C}_{\text{pred,f}}$:

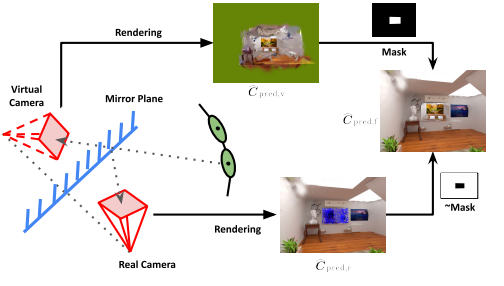

Figure 2: The process of virtual camera rendering in Sec. 2.2.

$$\hat{C}_{\text{pred,f}}(k) = \begin{cases} \hat{C}_{\text{pred,v}}(k), & \text{if } k \in \mathcal{P}_{cam} \cap \mathcal{P}_M \\ \hat{C}_{\text{pred,r}}(k), & \text{if } k \in \mathcal{P}_{cam} \cap \overline{\mathcal{P}_M} \end{cases}, \tag{3}$$

where $\hat{C}_{\text{pred,r}}$ is the rendered region obtained from the real camera, $\hat{C}_{\text{pred,v}}$ is the rendered with the virtual camera. $\mathcal{P}_{\text{cam}}$ denotes all the pixels in the rendered image, and $\mathcal{P}_M$ is the set of pixels belonging to the masked mirror region.

2.3 Mirror Plane Prediction

To obtain the mirror plane in world coordinates, we extend the original 3D-GS rendering with with depth and normal rasterization. Following DN-Splatter [23], we render depth and normal maps from the 3D-GS scene for each camera view.

Color optimization. During this stage, we only render RGB image $\hat{C}_{\text{pred,r}}$ from real camera. In order to reconstruct the mirror surface, we replace the mirror region color on gt image with a constant color to obtain C'_{gt}. We use $\mathcal{L}_{rgb}(\hat{C}_{\text{pred,r}}, C'_{gt})$ to optimize the color where $\mathcal{L}_{rgb}$ is the regularization loss proposed in 3D-GS [7].

Depth estimation and regularization. Given a camera view with an extrinsic matrix $[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3\times4}$ and intrinsics $\mathbf{K} \in \mathbb{R}^{3\times3}$, we estimate depth maps from the center of each 3D Gaussians μ_i with the following equation:

$$\mu'_i = [x_i, y_i, d_i]^T = \mathcal{T}(\mu_i, \mathbf{R}, \mathbf{t}, \mathbf{K}), \tag{4}$$

where $\mathcal{T}(\cdot)$ is the projection equation. We then use the z-depth d_i in $\mu'_{\mathbf{i}}$ and render per-pixel depths $\hat{D}$ using alpha-compositing:

$$\hat{D} = \sum_{i\in N} d_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{5}$$

where α_i is the blending coefficient for the i-th Gaussian. We primarily concentrate on indoor scenes, which typically feature numerous planar surfaces. Therefore, depth estimates for adjacent pixels are expected to be similar. To enforce this consistency, we employ a smoothness prior to depth estimation [8, 23]. Specifically, we use an RGB-weighted depth smoothness regularization loss that penalizes dissimilar predictions in textureless areas:

$$\mathcal{L}_{smooth} = \sum_{k\in\mathcal{P}_{cam}} \sum_{q\in\mathcal{N}(k)} exp(-\gamma \left|C'_{gt}(k) - C'_{gt}(q)\right|) \left|\hat{D}_{pred,r}(k) - \hat{D}_{pred,r}(q)\right|, \tag{6}$$

where k and q represent pixels, $\mathcal{N}(k)$ represents the 4 horizontal/vertical neighbour pixels around k, and γ is a hyper-parameter.

Normal estimation and regularization. 3D Gaussians do not inherently contain a normal direction. However, previous methods [8, 23] have observed that during 3D-GS optimization process, Gaussians gradually become flatter and approach a planar shape. Therefore, the shortest scaling axis of a Gaussian's covariance matrix can be considered as a good estimate of a normal direction. Similar to depth rendering, we obtain per-pixel normal maps following:

$$\hat{N} = \sum_{i\in N} \mathbf{n}_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{7}$$

where $\mathbf{n}_i \in \mathbb{R}^3$ is the normalized vector representing the direction of the shortest axis of the i-th Gaussian. We leverage the above depth estimates for supervision to ensure consistent normal estimates at nearby pixels. We encourage consistency between the rendered normal $\hat{N}$ and the pseudo-normal $\tilde{N}$, computed from the gradient of rendered depths $\hat{D}$ under the local planarity assumption. The normal consistency is quantified as an L1 loss:

$$\mathcal{L}_n = \left\|\hat{\mathbf{N}} - \tilde{\mathbf{N}}\right\|, \tag{8}$$

where $\tilde{\mathbf{N}}$ is the pseudo ground truth normal estimate derived from the gradient of the depth map [4]:

$$\tilde{\mathbf{N}}(x,y) = \frac{\nabla_x d \times \nabla_y d}{|\nabla_x d \times \nabla_y d|}, \tag{9}$$

and d are nearby depth points of pixel (x,y).

In addition to the regularization with pseudo ground truth normals, we enforce that segmented mirror regions to have the same normal direction with a planar assumption. Specifically, during training we randomly sample p pixels from the segmented mirror region and penalize differences in angular normal estimates with the planar-constraint loss:

$$\mathcal{L}_{pc} = \frac{1}{\mathbf{N}_p^2} \sum_{k=1}^{\mathbf{N}_p} \sum_{q=1}^{\mathbf{N}_p} (1 - \cos(\langle \mathbf{n_k}, \mathbf{n_q} \rangle)), \tag{10}$$

where $\mathbf{n_k}$ and $\mathbf{n_q}$ are per-pixel normal, and $\mathbf{N}_p$ is the amount of sampled normals.

Mirror Plane Estimation. Following [6], we define the mirror plane $\hat{\pi} = [\mathbf{n}, o]$ using a unit normal vector $\mathbf{n}$ and offset o giving the canonical plane equation $\hat{\pi}^T[x,y,z,-1] = 0$. After optimizing the 3D-GS scene with depth and normals regularization for a few thousand steps, we randomly pick a camera view containing a mirror within its view frustum and backproject the depth $\hat{D}$ and normal $\hat{\mathbf{N}}$ maps of the mirror region to the world coordinate to get 3D mirror coordinates $\hat{C}$ and normal $\hat{\mathbf{N}}_w$. The average value of the normals $\hat{\mathbf{N}}_w$ is considered as the normal of the mirror plane $\mathbf{n}$. We utilize RANSAC [2] and the plane definition to estimate a robust plane equation $\hat{\pi}^T[x,y,z,-1] = 0$. The detailed RANSAC process is presented in the supplementary. Note that we only calculate the mirror plane once during the full training procedure.

2.4 Virtual Camera Optimization

We initialize the mirror plane estimate in Sec. 2.3, and use it to compute the virtual camera pose as stated in Sec. 2.2. However, due to the lack of ground truth depth or normal supervision during training, this estimated mirror plane is imperfect. This may lead to a discrepancy between the computed pose of the virtual camera and the ideal pose, resulting in misaligned reflection rendering images. To address this issue, we implemented a simple yet effective method to optimize the estimated mirror plane using virtual camera pose optimization. Specifically, in this stage, we optimize the $\hat{C}_{\text{pred,f}}$ obtained from Sec.2.2 with the photometric loss between $\hat{C}_{\text{pred,f}}$ and ground truth image C_{gt}:

$$\mathcal{L}_{vco} = \mathcal{L}_{rgb}(\hat{C}_{\text{pred,f}}, C_{gt}). \tag{11}$$

During the virtual camera rendering process, we do not compute gradients to the Gaussian attributes (μ, Σ, O, θ). Instead, following iComMa [22], we extend 3D-GS by explicitly deriving gradient flow to the virtual camera poses $[\mathbf{R}'|\mathbf{t}']$. The pose is then optimized during training. Since the virtual camera pose is derived from the mirror plane and the real camera pose, the optimization of the virtual camera pose using the chain rule naturally leads to the refinement of the mirror plane equation as well. This method ensures that both the virtual camera pose and the mirror plane equation are both effectively optimized with the same photometric loss defined above.

We believe that the main bias in virtual camera rendering at this stage comes from the deviation of the virtual camera pose. Additionally, the optimization speed of Gaussians is much faster than that of the camera pose. Consequently, if we optimize the Gaussians' attributes

and the virtual camera pose simultaneously, the optimization process prefers duplicating more Gaussians to accommodate the imperfections in the virtual camera pose, rather than correcting the pose itself. This could lead to artifacts in the rendered images. By focusing solely on calculating and optimizing the gradient of the virtual camera pose, we ensure that the optimization process prioritizes the accuracy of the virtual camera pose and the mirror plane equation, thereby enhancing the overall quality of the rendering.

2.5 Progressive Training

We partition our training into various stages. At the beginning of optimization, we train on the full training image dataset with no mirror masking using the original regularization loss proposed in 3D-GS [7]. This ensures that the 3D-GS scene is initialized well. We then activate our depth regularization with Eq. (5), (6) and normal regularization with Eq. (8), (10) still using the full training images. During this stage, we train with the following mirror plane prediction loss:

$$\mathcal{L}_{mpp} = \mathcal{L}_{rgb}(\hat{C}_{\text{pred,r}}, C'_{gt}) + \lambda_n \mathcal{L}_n + \lambda_s \mathcal{L}_{smooth} + \lambda_{pc} \mathcal{L}_{pc}. \tag{12}$$

After this stage, we estimate the mirror plane and enable the virtual camera optimization illustrated in Sec. 2.4 and use $\mathcal{L}_{vco}$ in Eq. (11) to optimize the virtual camera pose, hence refine the mirror plane. In this stage, we disable gradients to the Gaussian attributes. In the final stage, we assume that the current mirror plane equation and virtual camera poses have converged to accurate estimates. Therefore, we disable virtual camera pose optimization and optimize the Gaussian scene attributes. During this phase, we fine-tune both non-reflective and reflective regions, leading to photo-realistic novel-view synthesis.

3 Experiments

3.1 Training Details

Most of the experiments conducted in this paper are on a single V100 GPU and using the pytorch framework (Only the rendering speed (FPS) is tested on a single RTX 4090 GPU), with a total of 60,000 training steps. For all the experiments, the regularization weights are set as follows: $\lambda_s = 0.01$, $\lambda_n = 0.005$, $\lambda_{pc} = 0.01$, and $\gamma = 0.1$. We employed the Adam Optimizer [8] for virtual camera optimization, setting the learning rate at 0.0005. The total number of steps for the mirror plane prediction stage was 1000 for experiments conducted on Mirror-NeRF's synthetic dataset and our dataset, and 1500 for Mirror-NeRF's real dataset. The virtual camera optimization stage consisted of 10,000 steps. Please refer to the supplementary for the implementation details.

3.2 Experimental Results

Dataset. We evaluate our method using the publicly available Mirror-Nerf dataset [32], which includes three synthetic scenes (Washroom, Living room, Office) and three real scenes (Market, Lounge, Discussion room). Each scene is captured within a room containing mirror, with approximately 300 frames taken from a 360-degree perspective. The dataset also includes a mirror reflection mask in each image. To enhance the evaluation of our method in real-world scenarios, we provide an additional dataset containing three scenes, Recovery

Room, Work Space, and Corridor, with the mirror size and shape in each scene being different from others. The dataset comprises 250 frames per scene, with mirror reflection masks extracted using SAM [9] and camera poses estimated by COLMAP [20]. We follow [32] to split the training and test dataset and each set includes a mix of views with and without mirror. We show the example images of the mirror scenes from simple to complex in the supplementary.

Baselines and Evaluation Protocol. We consider the following methods as baselines for comparison (a) start-of-the-art Mirror-NeRF [32]; two fast-rendering methods (b) Instant-NGP [18] and (c) DVGO [21] and (d) the official 3DGS [7]. Following the standard practice in novel view synthesis, we evaluate the rendering quality using the metrics PSNR, SSIM [26] and LPIPS [33].

3.2.1 Mirror-NeRF Dataset Results

Table. 1 presents the experimental results for Mirror-Nerf [32] synthetic and real datasets. On the synthetic dataset, our method outperforms the best method, Mirror-NeRF, in both PSNR and SSIM metrics, achieving state-of-the-art performance. This success is largely due to our mirror plane prediction described in Section 2.3, which effectively initializes the plane equation on synthetic datasets. The subsequent virtual camera optimization further enhances the rendering quality. Compared to the vanilla 3D-GS, our method achieves significant improvement with the average values across three scenes. Specifically, in the office scene, our method shows gains of +8.65 in PSNR, +0.057 in SSIM, and -0.067 in LPIPS.

On real datasets, our method approaches the performance of Mirror-NeRF, with significantly faster rendering speed (FPS 128.22 v.s. 0.71). Compared to more efficient rendering methods like InstantNGP [18] and DVGO [21], the proposed approach leads in all metrics with a large margin, indicating that our method can provide both high-quality and efficient image rendering. We provide the qualitative comparison in Fig. 3, and our method achieves high rendering quality with less blurred regions.

3.2.2 Our Real-World Dataset Results

Table. 2 presents the comparative results of vanilla 3D-GS and our method on the proposed dataset. In the Corridor and Recovery Room scene which contains large mirrors and planer surface, our method significantly outperforms 3D-GS on all the metrics. 3D-GS estimates mirror reflections as virtual spaces behind the mirror plane, necessitating extensive optimization of additional Gaussians within this virtual space. This results in blurry details and textures, since the resulting mirror regions are not multi-view consistent. Conversely, our method does not require the optimization of additional Gaussians to explain mirror reflections and facilitates more physically accurate mirror rendering using virtual cameras resulting in better detail preservation in the scene.

However, in the Work Space scene, our method exhibits a slight decline relative to 3D-GS. This scene is characterized by a large, cluttered space filled with numerous items, which poses challenges for our method due to the absence of planer surfaces for depth or normal supervision. Consequently, it struggles to derive an effective mirror equation, leading to discrepancies in the virtual camera pose estimation and less effective virtual camera rendering compared to 3D-GS. This highlights areas where further refinement of our method is needed to handle complex scenarios effectively.

Method	Synthetic				Real		
	PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑	PSNR ↑	SSIM ↑	LPIPS ↓
InstantNGP [23]	23.54	0.71	0.42	-	10.51	0.20	0.71
DVGO [21]	28.05	0.82	0.29	-	22.18	0.67	0.33
Mirror-NeRF [32]	38.08	0.958	**0.028**	0.71	**25.31**	**0.789**	**0.082**
3D-GS [8]	36.52	0.953	0.064	**480.29**	23.39	0.715	0.268
Ours	**39.87**	**0.979**	0.038	128.22	24.19	0.759	0.234

Table 1: Results on the Mirror-NeRF's synthetic and real-world dataset, our method approaches or achieves state-of-the-art performance on these two datasets, and significantly improves upon 3D-GS. The FPS values are measured on an RTX 4090 GPU. The best result is in **bold**, and the second best is underlined.

Method	**Corridor** (easy)			**Recovery Room** (medium)			**Work Space** (complex)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3D-GS [8]	25.9	0.845	0.332	28.84	0.917	0.166	**26.88**	**0.884**	**0.219**
Ours	**29.14**	**0.874**	**0.291**	**32.21**	**0.938**	**0.139**	24.89	0.849	0.288

Table 2: Results on the our real-world mirror dataset. Our proposed method outperforms vanilla 3D-GS on two of the scenes. The best result is marked with **bold**.

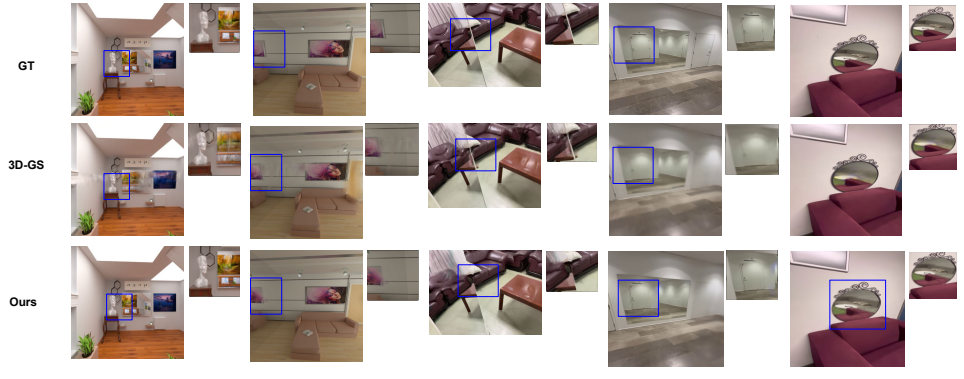

Figure 3: Qualitative comparisons of ground truth, 3D-GS, and our method on the Living Room, Office, Lounge, Corridor, and Recovery Room scenes from Mirror-NeRF dataset. The smaller image in the upper right corner of each main images is an enlargement of a mirror region.

3.3 Ablation Study

We present the ablation study on loss function selection and virtual camera optimization strategies in Table. 3. The experiments demonstrate that removing any loss function related to the depth map or normal map in plane prediction significantly reduces the accuracy of rendering quality. Removing the normal loss shows the worst performance, indicating the effectiveness of the proposed depth and normal optimization. Additionally, without virtual

Settings	PSNR ↑	SSIM ↑	LPIPS ↓
w/o $\mathcal{L}_n$	32.30	0.964	0.071
w/o $\mathcal{L}_{pc}$	32.51	0.956	0.069
w/o $\mathcal{L}_{smooth}$	34.49	0.967	**0.037**
w/o Camera Optimization	30.79	0.943	0.102
Camera + Gaussians Optimization	31.30	0.940	0.103
Full Model	**37.30**	**0.977**	0.042

Table 3: Ablation studies of various components of our proposed approach using the Office scene form Mirror-NeRF dataset. The highest score is in **bold**.

camera optimization leads to a notable decline in rendering quality. This underscores the critical role of camera optimization in refining both the mirror plane and the virtual camera pose, especially when the initial plane prediction is imperfect. Furthermore, we conduct ablation study on the joint virtual camera and Gaussians optimization. The results demonstrate that this joint optimization detracts from the rendering quality, confirming our assumptions discussed in Sec. 2.4.

4 Conclusion

We introduce an improved 3D-GS based method for novel view synthesis in scenes that contain mirror reflections. Our method solves mirror reflections by estimating a mirror plane and modeling reflections using virtual cameras. We extend the 3D-GS representation with depth and normal supervision to accurately estimate the mirror plane in world coordinates. Following this, we employ virtual camera optimization to improve the quality of reflection rendering. To enhance the evaluation of our method, we collect a dataset that includes different mirror sizes in real-world settings. We outperform vanilla 3D-GS and exceed previous NeRF methods on both synthetic and real-world datasets while offering faster rendering speeds.

Limitations. It is important to note that our method assumes the presence of planar surfaces in indoor scenes. The absence of these planar surfaces could lead to a drop in performance due to the inaccurate depth and normal estimates, indicating a need for further methodological enhancements to address more diverse and complex scene geometries. Although our dataset provides more real-world scenes with mirror reflections, it has some limitations. For instance, the placement angles of the mirrors in the scenes are not sufficiently diverse. Future work could consider introducing more challenging real-world datasets to better validate the robustness of the method.

References

[1] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. *arXiv preprint arXiv:2312.12337*, 2023.

[2] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[3] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022.

[4] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. *arXiv preprint arXiv:2403.17888*, 2024.

[5] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023.

[6] Linyi Jin, Shengyi Qian, Andrew Owens, and David F Fouhey. Planar surface reconstruction from sparse views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12991–13000, 2021.

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

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[10] Mengtian Li, Shengxiang Yao, Zhifeng Xie, Keyu Chen, and Yu-Gang Jiang. Gaussianbody: Clothed human reconstruction via 3d gaussian splatting. *arXiv preprint arXiv:2401.09720*, 2024.

[11] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, et al. Vastgaussian: Vast 3d gaussians for large scene reconstruction. *arXiv preprint arXiv:2402.17427*, 2024.

[12] Jiayue Liu, Xiao Tang, Freeman Cheng, Roy Yang, Zhihao Li, Jianzhuang Liu, Yi Huang, Jiaqi Lin, Shiyong Liu, Xiaofei Wu, et al. Mirrorgaussian: Reflecting 3d gaussians for reconstructing mirror reflections. *arXiv preprint arXiv:2405.11921*, 2024.

[13] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. *arXiv preprint arXiv:2312.00109*, 2023.

[14] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

[15] Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting. *arXiv preprint arXiv:2404.00409*, 2024.

[16] Jiarui Meng, Haijie Li, Yanmin Wu, Qiankun Gao, Shuzhou Yang, Jian Zhang, and Siwei Ma. Mirror-3dgs: Incorporating mirror reflections into 3d gaussian splatting. *arXiv preprint arXiv:2404.01168*, 2024.

[17] 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.

[18] 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.

[19] Rui Rodrigues, Joao P Barreto, and Urbano Nunes. Camera pose estimation using images of planar mirror reflections. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 382–395. Springer, 2010.

[20] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[21] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Superfast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022.

[22] Yuan Sun, Xuan Wang, Yunfan Zhang, Jie Zhang, Caigui Jiang, Yu Guo, and Fei Wang. icomma: Inverting 3d gaussians splatting for camera pose estimation via comparing and matching. *arXiv preprint arXiv:2312.09031*, 2023.

[23] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv preprint arXiv:2403.17822*, 2024.

[24] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.

[25] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

[26] 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.

[27] Turner Whitted. An improved illumination model for shaded display. In *ACM Siggraph 2005 Courses*, pages 4–es. 2005.

[28] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023.

[29] Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-scale 3d gaussian splatting for anti-aliased rendering. *arXiv preprint arXiv:2311.17089*, 2023.

[30] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023.

[31] Ze-Xin Yin, Jiaxiong Qiu, Ming-Ming Cheng, and Bo Ren. Multi-space neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12407–12416, 2023.

[32] Junyi Zeng, Chong Bao, Rui Chen, Zilong Dong, Guofeng Zhang, Hujun Bao, and Zhaopeng Cui. Mirror-nerf: Learning neural radiance fields for mirrors with whitted-style ray tracing. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4606–4615, 2023.

[33] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[34] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. *arXiv preprint arXiv:2312.02155*, 2023.

[35] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00451*, 2023.