Ray-Tracing Enhanced 2D Gaussian Splatting for Transparent Object Reconstruction

Jinrui Zhang, Hengxu Yu, Zihan Yi

Abstract

Transparent object reconstruction presents fundamental limitations in 3D Gaussian Splatting (3DGS) frameworks due to missing surface normal information and unmodeled light refraction effects. We propose a breakthrough solution through these innovations: First, a differentiable ray-tracing module integrated with 2D Gaussian representation that physically simulates refraction/reflection at transparent interfaces. Second, a VGGSfM-based reconstruction pipeline leveraging deep feature tracking to overcome traditional SfM failures on transparent objects. Finally, adaptive spherical harmonics (SH) optimization and covariance regularization techniques that maintain geometric stability during training. Validated on a dedicated transparent object dataset, our framework demonstrates unprecedented capability in reconstructing complex glassware and liquid surfaces, achieving photorealistic rendering quality while preserving real-time performance advantages of Gaussian splatting paradigms.

Introduction

The complex light transport phenomena (e.g., multi-bounce refraction, Fresnel reflection, subsurface scattering) in transparent objects fundamentally challenge conventional 3D reconstruction:

1. Problem Analysis

Traditional 3D Gaussian Splatting (3DGS) aggregates scene radiance through discrete Gaussian primitives, but continuous light field variations in transparent media cause primitive misalignment, manifesting as phantom artifacts on reconstructed surfaces.

2. Technical Barrier

Although ray tracing physically models light transport, its requirement for explicit surface normals fundamentally conflicts with 3DGS's implicit geometric representation.

3. Key Innovations

We use 2D Gaussian Splatting with ray-tracing. By integrating ray tracing physics with 2D Gaussian primitives, we establish differentiable light transport mechanisms in image space, fundamentally bypassing traditional dependency on 3D surface normals. We also improve the loss function, SH function and add geometric regularization modules preserving physical plausibility of Gaussian primitives. This approach achieves unified modeling of refraction, reflection and scattering effects within Gaussian splatting frameworks for the first time, opening new avenues for real-time high-fidelity transparent scene reconstruction.

4. Contributions

This work achieves the first high-quality transparent object reconstruction within the Gaussian splatting framework, bridging a critical research gap while delivering an open-source solution compatible with existing ecosystems and practical technical guidelines.

Related works

Traditional Transparent Object Reconstruction

Ray Tracing(Zhang et al. 2021) requires known geometry, failing with unknown shapes

Polarization Imaging(Ye et al. 2022) depends on specialized hardware, limiting scalability

Neural Implicit Representations

NeRF Variants(e.g.TransNeRF) model transparency via volume rendering but require long training hours

3DGS Extensions(GS-Water) attempt refraction constraints, yet suffer from normal estimation errors due to spherical Gaussian assumptions

Our work

First work integrating ray tracing physics priors with 2D Gaussian splatting, achieving physically-accurate reconstruction.

Methodology

1.. Differentiable Ray-Tracing Integration

Integrate differentiable ray-tracing into 3DGS to model refraction and reflection. Jointly optimize geometry and material parameters via gradient propagation.

2.. Gaussian Parameter Control

Progressively increase SH maximum order to prevent early-stage overfitting. Enforce minimum eigenvalues to avoid spherical degeneration.

Table 1. Comparative analysis of current SOTA methods

Method	Results
3D Gaussian Splatting	
2D Gaussian Splatting	
3DGRT	

Training

$$\mathcal{L}_{\text{smooth}} = \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} \frac{\log\left(\frac{2}{|\langle \mathbf{n}_i, \mathbf{n}_j \rangle|+1}\right)}{d_{ij}+\epsilon}$$

We follow the training framework established in the original 2D Gaussian Splatting (2DGS) method, optimizing the model parameters to reconstruct geometrically accurate and visually coherent surfaces. However, our approach introduces modifications to enhance geometric fidelity, particularly targeting surface smoothness and accurate normal alignment. Specifically, we integrate three key loss terms: depth distortion, modified normal consistency, and a novel smoothness regularization.

Depth Distortion

Following the approach of Kerbl et al. [2023], we employ a depth distortion loss to mitigate inaccuracies arising from discrepancies between intersected Gaussian primitives' color and depth renderings. Inspired by the Mip-NeRF360 model, our depth distortion loss concentrates the rendering weight distribution along rays by minimizing depth differences between intersections, defined as follows:

$$\mathcal{L}_d = \sum_{j=1}^{i-1} \omega_j \omega_i |z_i - z_j|$$

Normal Consistency

Our implementation enhances the normal consistency loss by adjusting hyperparameters to better capture surface normals' orientation relative to depth gradients. We define the loss as:

$$\mathcal{L}_n = \sum_i \omega_i (1 - n_i^\top N)$$

where n_i is the oriented normal vector of the i-th Gaussian primitive, and N is computed from depth gradients obtained via neighboring points. Unlike the original formulation, we have fine-tuned the weight ω_i and to improve convergence stability and normal accuracy**.**

Smoothness Regularization

To address potential surface roughness and ensure spatial coherence of normals across adjacent Gaussian splats, we introduce a smoothness loss defined as:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} \frac{\log\left(\frac{2}{|\langle \mathbf{n}_i, \mathbf{n}_j \rangle|+1}\right)}{d_{ij}+\epsilon}$$

This smoothness loss penalizes sharp deviations between adjacent normals, thus fostering locally smooth surfaces.

Final Objective

Combining these losses, our complete optimization objective becomes:

$$\mathcal{L} = \mathcal{L}_c + \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}}$$

Here, $\mathcal{L}_c$ denotes the RGB reconstruction loss combined with the D-SSIM loss as in Kerbl et al. [2023]. Hyperparameters λ_d, λ_n and λ_{smooth} balance the influence of each loss component and were empirically determined. In our experiments, we set λ_d=1000, λ_n=0.2, λ_{smooth}=0.3.

Experiments

We now present evaluations of our transparent object reconstruction method, including appearance with previous SOTA approaches. We then analyze the contribution of the proposed components.

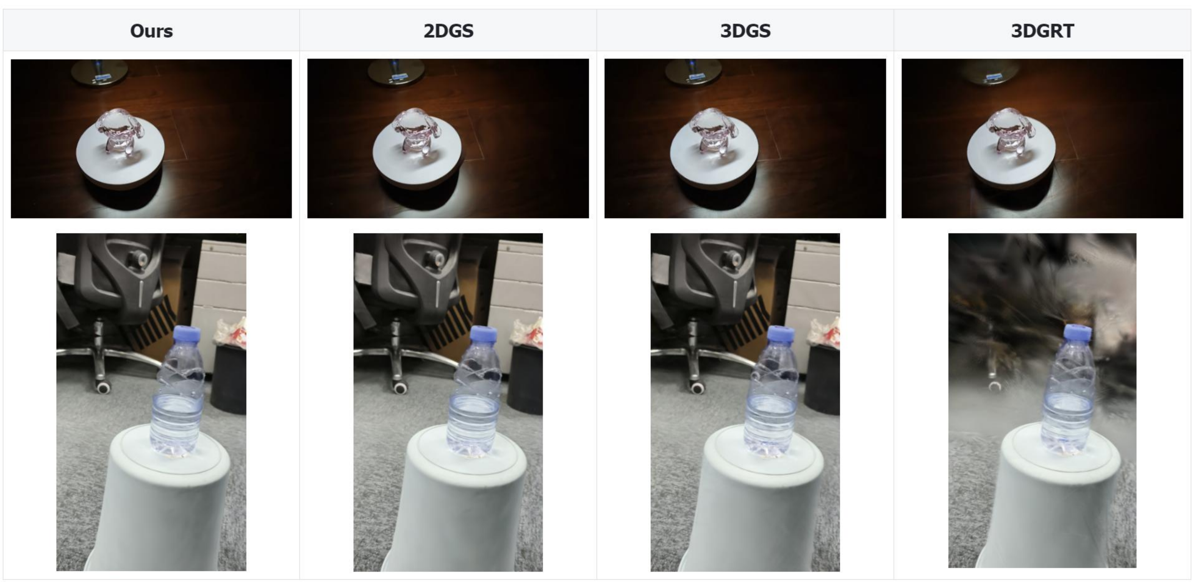
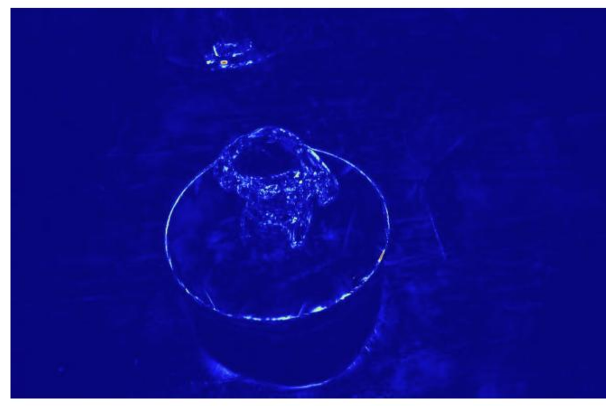


Table 2. We compare our method with current SOTA in explicit representation. We report our results based on the same colmap dataset generated by vggsfm.

	PSNR	Time(min)	Storage(MB)
3DGS	30.21	17.4	37.85
3DGRT	29.95	18.7	35.90
2DGS	31.06	**14.21**	29.1
Ours	**32.57**	15.05	**23.91**

Table 3. Performance comparison between 3DGS, 3DGRT, 2DGS, Ours. We report the reconstruction 30K iterations and the correspond error map.

	PSNR
Ours	**32.57**
Ours (No SH)	32.10
Ours (No Smooth)	31.13
Ours (with RT)	30.59



Conclusion

3D Gaussian Splatting has been a big step forward since Nerf for its speed and quality. However, it comes with some limitations due to its way it expresses and renders the scene, and the refraction of light is just one of them. We explored ways to improve the situation by 1) using VGGSfM instead of classic SfM, 2) using ray tracing with refraction to render the scene, and 3) adding a punishment to force the surfaces to be smoother to help ray tracing produces better results. Due to time and resource constraints, we are still investigating how the methods work out and what our next steps are..

References

1. Wang, Jianyuan, et al. "Vggsfm: Visual geometry grounded deep structure from motion." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024.
2. Byrski, Krzysztof, et al. "RaySplats: Ray Tracing based Gaussian Splatting." arXiv preprint arXiv:2501.19196 (2025).