

Supplementary Material for "On the Error Analysis of 3D Gaussian Splatting and an Optimal Projection Strategy"

Letian Huang[✉], Jiayang Bai[✉], Jie Guo^{*✉}, Yuanqi Li[✉], and Yanwen Guo[✉]

Nanjing University, Nanjing, China

{lthuang, jybai, yuanqili}@smail.nju.edu.cn, {guojie, ywguo}@nju.edu.cn

<https://letianhuang.github.io/op43dgs/>

A Details of Error Analysis

Unit Vector Simplification We have proved that the composition of the transformation ϖ , projecting onto the unit sphere, and the transformation φ , projecting onto the tangent plane, is equivalent to the single transformation φ directly projecting onto the tangent plane:

$$\begin{aligned}
 (\varphi \circ \varpi)(\mathbf{x}') &= \varphi\left(\mathbf{x}'(\mathbf{x}'^\top \mathbf{x}')^{-1/2}\right) \\
 &= \mathbf{x}'(\mathbf{x}'^\top \mathbf{x}')^{-1/2} \left(\mathbf{x}_0^\top \left(\mathbf{x}'(\mathbf{x}'^\top \mathbf{x}')^{-1/2}\right)\right)^{-1} \\
 &= \mathbf{x}'(\mathbf{x}_0^\top \mathbf{x}')^{-1} \\
 &= \varphi(\mathbf{x}').
 \end{aligned} \tag{1}$$

Therefore, we can simplify the derivation of the error function ϵ by projecting all relevant points onto the unit sphere as unit vectors (Equation 8 in the main paper).

Closed-Form Expression for the Error Function This section provides a comprehensive derivation of the integral expression for the error function as presented in the main paper. Upon expressing \mathbf{x}' , \mathbf{x}_0 and $\boldsymbol{\mu}'$ as unit vectors and substituting them into Equation 6 in the main paper, the Jacobian matrix is derived:

$$\mathbf{J} = \frac{\partial \varphi}{\partial \mathbf{x}'}(\boldsymbol{\mu}') = \begin{bmatrix} \frac{1}{\cos(\phi_\mu) \cos(\theta_\mu)} & 0 & -\frac{\sin(\phi_\mu)}{\cos^2(\phi_\mu) \cos(\theta_\mu)} \\ 0 & \frac{1}{\cos(\phi_\mu) \cos(\theta_\mu)} & \frac{\sin(\theta_\mu)}{\cos^2(\phi_\mu) \cos^2(\theta_\mu)} \\ 0 & 0 & 0 \end{bmatrix}. \tag{2}$$

* Corresponding author.

Substituting this Jacobian matrix into Equation 5 in the main paper yields the derivation of the Taylor expansion remainder term:

$$R_1(\mathbf{x}') = \begin{bmatrix} -\frac{\sin(\phi - \phi_\mu) \cos(\theta)}{\cos^2(\phi_\mu) \cos(\theta_\mu)} + \tan(\phi) - \tan(\phi_\mu) \\ \frac{\sin(\theta)}{\cos(\phi_\mu) \cos(\theta_\mu)} - \frac{\sin(\theta_\mu) \cos(\phi) \cos(\theta)}{\cos^2(\phi_\mu) \cos^2(\theta_\mu)} + \frac{\tan(\theta_\mu)}{\cos(\phi_\mu)} - \frac{\tan(\theta)}{\cos(\phi)} \\ 0 \end{bmatrix}. \quad (3)$$

And substituting this expression into Equation 7 in the main paper results in the integral expression (Equation 9 in the main paper). The main paper provides the integral expression (Equation 9 in the main paper) and the function graph (Figure 2 in the main paper) for the error function, without presenting the closed-form expression after integration. We provide the closed-form expression as follows:

$$\begin{aligned} \epsilon(\theta_\mu, \phi_\mu) = & \frac{1}{16 \cos^4(\phi_\mu) \cos^4(\theta_\mu) \tan(\phi_\mu + \frac{\pi}{4}) \tan(\theta_\mu + \frac{\pi}{4})} \\ & \left\{ 8 \left(\left(2 \tan\left(\theta_\mu + \frac{\pi}{4}\right) - \pi \right) \tan\left(\theta_\mu + \frac{\pi}{4}\right) + 2 \right) \cos^4(\phi_\mu) \cos^4(\theta_\mu) \tan^2\left(\phi_\mu + \frac{\pi}{4}\right) + \right. \\ & (-4 \cdot (2 \cos(2\theta_\mu) + \pi) \sin^2(\phi_\mu) + \pi(2 \cos(2\theta_\mu) + \pi) + 4 \cos(2\theta_\mu) + 2\pi) \\ & \sin^2(\theta_\mu) \tan\left(\phi_\mu + \frac{\pi}{4}\right) \tan\left(\theta_\mu + \frac{\pi}{4}\right) + 8 \log \frac{(\sin(\phi_\mu + \frac{\pi}{4}) + 1)(\cos(\phi_\mu + \frac{\pi}{4}) + 1)}{(1 - \sin(\phi_\mu + \frac{\pi}{4}))(1 - \cos(\phi_\mu + \frac{\pi}{4}))} \\ & \log \frac{(1 - \sin(\phi_\mu + \frac{\pi}{4}))(1 - \cos(\phi_\mu + \frac{\pi}{4})) \exp(2\sqrt{2} \cos(\theta_\mu))}{(\sin(\phi_\mu + \frac{\pi}{4}) + 1)(\cos(\phi_\mu + \frac{\pi}{4}) + 1)(\tan^2 \sin(\theta_\mu)(\theta_\mu + \frac{\pi}{4}))} \\ & \cos^3(\phi_\mu) \cos^3(\theta_\mu) \tan\left(\phi_\mu + \frac{\pi}{4}\right) \tan\left(\theta_\mu + \frac{\pi}{4}\right) + \\ & 4\pi \left(-4 \log\left(\tan\left(\phi_\mu + \frac{\pi}{4}\right)\right) \tan(\phi_\mu) + \pi \tan^2(\phi_\mu) + 2 \tan\left(\phi_\mu + \frac{\pi}{4}\right) - \pi \right) \\ & \cos^4(\phi_\mu) \cos^4(\theta_\mu) \tan\left(\phi_\mu + \frac{\pi}{4}\right) \tan\left(\theta_\mu + \frac{\pi}{4}\right) + \\ & \left(2\pi \left((4 + 2\pi + 16\sqrt{2}) \sin^2(\theta_\mu) - 2 + \pi \right) \cos^2(\phi_\mu) + (2\pi - 4) \cos(2\theta_\mu) - 2\pi + \pi^2 \right) \\ & \cos^2(\theta_\mu) \tan\left(\phi_\mu + \frac{\pi}{4}\right) \tan\left(\theta_\mu + \frac{\pi}{4}\right) + \\ & 16\sqrt{2} \log \frac{(1 - \sin(\phi_\mu + \frac{\pi}{4}))(1 - \cos(\phi_\mu + \frac{\pi}{4})) \exp(2\sqrt{2} \cos(\theta_\mu))}{(\sin(\phi_\mu + \frac{\pi}{4}) + 1)(\cos(\phi_\mu + \frac{\pi}{4}) + 1)} \\ & \cos^3(\phi_\mu) \cos^4(\theta_\mu) \tan\left(\phi_\mu + \frac{\pi}{4}\right) \tan\left(\theta_\mu + \frac{\pi}{4}\right) + \\ & 64 \sin^2(\phi_\mu) \cos^2(\phi_\mu) \cos^4(\theta_\mu) \tan\left(\phi_\mu + \frac{\pi}{4}\right) \tan\left(\theta_\mu + \frac{\pi}{4}\right) - \\ & 16\sqrt{2} \cdot (1 + \sqrt{2}) \sin(\theta_\mu) \sin(2\theta_\mu) \cos^2(\phi_\mu) \cos(\theta_\mu) \tan\left(\phi_\mu + \frac{\pi}{4}\right) \tan\left(\theta_\mu + \frac{\pi}{4}\right) + \\ & \left. 8\pi \cos^4(\phi_\mu) \cos^4(\theta_\mu) \tan\left(\theta_\mu + \frac{\pi}{4}\right) + \frac{8(-2\pi \cos^2(\theta_\mu) + \pi + 4) \cos^4(\phi_\mu) \cos^2(\theta_\mu)}{(\tan(\theta_\mu) - 1)^2} \right\}. \end{aligned} \quad (4)$$

B Details of Adaptation for Various Camera Models

In the main paper, it is highlighted that our projection, independent of perspective image plane, allows for the adaptation to various camera models by modifying the transformation from image space to camera space in the rasterization based on the unit sphere (Equation 16 in the main paper). This section provides a detailed explanation of our method's adaptation to various camera models.

Fisheye The design models for fisheye cameras can generally be categorized into four types: equidistant projection model, equisolid angle projection model, orthographic projection model, and stereographic projection model. We explain our projection's adaptation to fisheye camera models, taking the equidistant projection model as an example.

For a pixel (u, v) on the image, we cast a ray. According to the transformation between the image space and camera space for the equidistant projection model, we obtain:

$$\mathbf{x}_{2D} = \varphi_P \left(\begin{array}{c} \frac{(u-c_x) \sin \left(\sqrt{\frac{(u-c_x)^2}{f_x^2} + \frac{(v-c_y)^2}{f_y^2}} \right)}{f_x \sqrt{\frac{(u-c_x)^2}{f_x^2} + \frac{(v-c_y)^2}{f_y^2}}} \\ \frac{(v-c_y) \sin \left(\sqrt{\frac{(u-c_x)^2}{f_x^2} + \frac{(v-c_y)^2}{f_y^2}} \right)}{f_y \sqrt{\frac{(u-c_x)^2}{f_x^2} + \frac{(v-c_y)^2}{f_y^2}}} \\ \cos \left(\sqrt{\frac{(u-c_x)^2}{f_x^2} + \frac{(v-c_y)^2}{f_y^2}} \right) \end{array} \right) \quad (5)$$

where c_x, c_y, f_x, f_y denote the intrinsic parameters of the camera model.

Panorama For a pixel (u, v) on the image, we cast a ray. According to the transformation between the image space and camera space for panoramic images, we obtain:

$$\mathbf{x}_{2D} = \varphi_P \left(\begin{array}{c} \left[\sin \left(\frac{\pi(-W+2u)}{W} \right) \cos \left(\frac{\pi(-\frac{H}{2}+v)}{H} \right) \right] \\ \sin \left(\frac{\pi(-\frac{H}{2}+v)}{H} \right) \\ \left[\cos \left(\frac{\pi(-\frac{H}{2}+v)}{H} \right) \cos \left(\frac{\pi(-W+2u)}{W} \right) \right] \end{array} \right) \quad (6)$$

where H, W represent the height and width of the image, respectively.

Then, similar to Equation 17 in the main paper, the 2D Gaussian function values can be obtained for alpha blending to generate the image. The alpha blending process after obtaining the 2D Gaussian is the same as in the original 3D-GS [5].

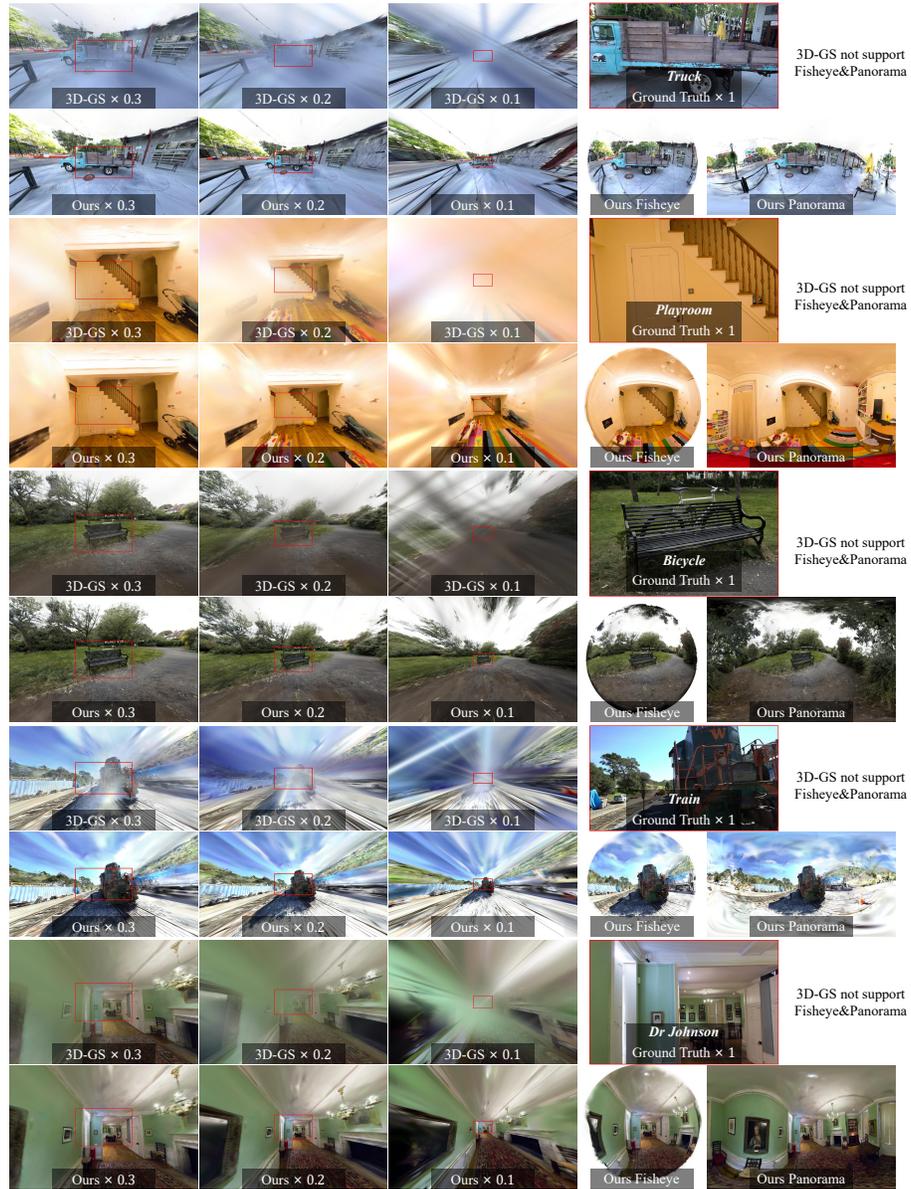


Fig. 1: We show comparisons of our method to the original 3D-GS [5] under various camera models and different focal lengths.

C Details of Experiments

Additional Results Table 1-2 list PSNR for our evaluation over all considered techniques and real-world scenes, corresponding to Table 1 in the main paper. Figure 2 illustrates additional qualitative comparisons between our method and other approaches.

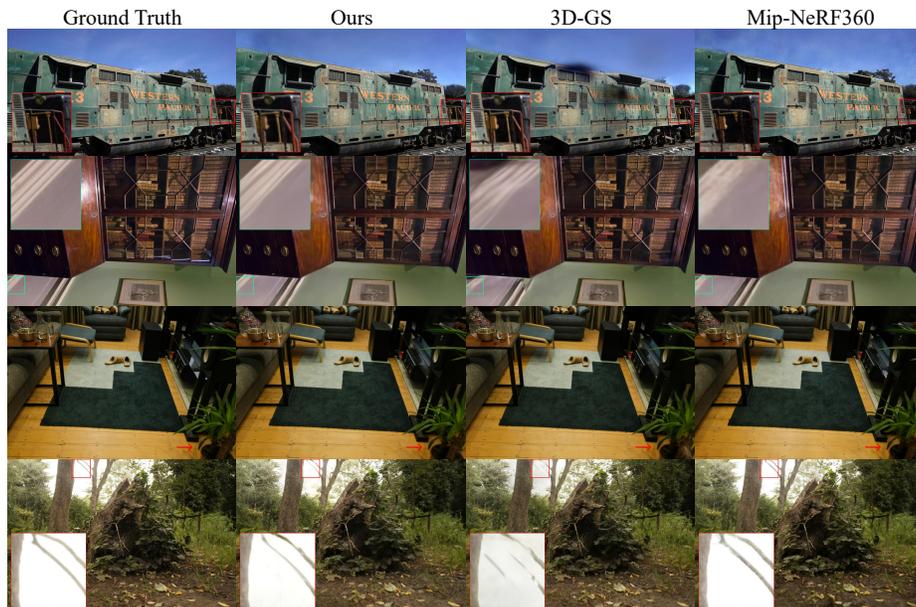


Fig. 2: We show comparisons of our method to previous methods and the corresponding ground truth images from held-out test views. The scenes are, from the top down: TRAIN from Tanks&Temples [6]; DRJOHNSON from the Deep Blending dataset [4] and ROOM, STUMP from Mip-NeRF360 dataset [1]. Differences in quality highlighted by arrows/insets.

Impacts of Decreasing Focal Length In the quantitative comparison experiments about the impact of reducing focal length, we utilized a focal length mask methodology due to the absence of ground truth for wide-angle images. The detailed procedures are illustrated in Figure 3. The focal mask selects central parts of the rendering with corresponding ground truth based on the focal length scaling factor for metric calculation. For rendering, we crop the corresponding part with available ground truth, then resize the ground truth to match the size of this patch. Finally, we compute the metrics for these two images.

For qualitative experiments, we show additional results in Figure 1. In addition to the original focal length reduction factors of 0.3 and 0.2, we introduce

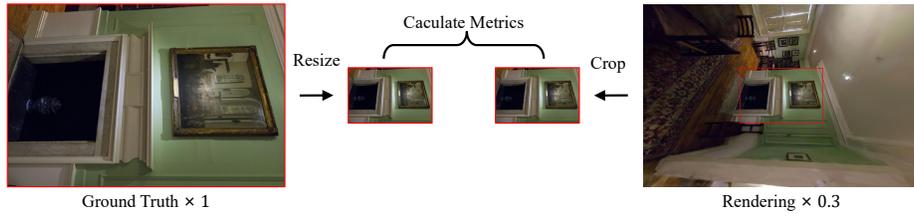


Fig. 3: Illustration of details of metrics calculation in the quantitative comparisons on the impact of decreasing focal length.

0.1 times the focal length. Additionally, we generate images with corresponding perspectives using other camera models, while 3D-GS [5] is not presented as it lacks support for these camera models.

Comparison with the Other 3D-GS Methods We choose Mip-Splatting [9] and Scaffold-GS [7] to compare in the same setting as in Figure 6 of the main paper. For Mip-Splatting [9] and Scaffold-GS [7] still employ conventional projection of 3D-GS, the approximation error remains larger compared to ours, as shown in Figure 4. Additionally, our method can be seamlessly integrated with theirs to reduce the projection error of these methods.

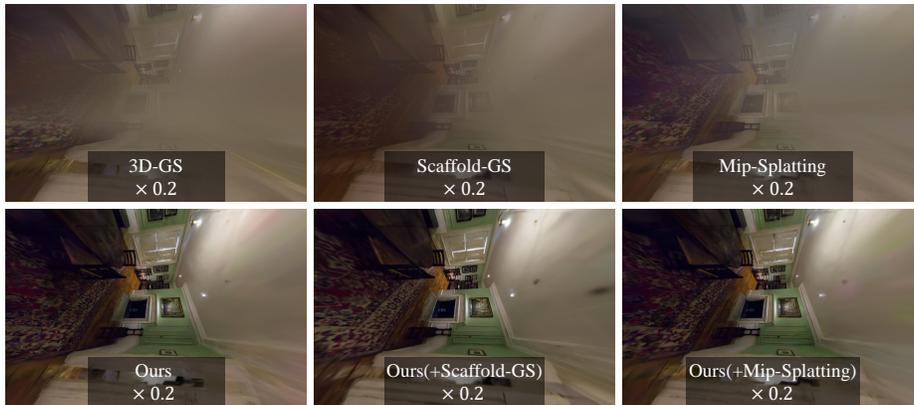


Fig. 4: We show comparisons of our method to the original 3D-GS [5], Mip-Splatting [9] and Scaffold-GS [7] with a large FOV camera. In the figure, Ours(+X) indicates the combination of our method with the X method.

Adaptability to Various Camera Models Since our method is based on differentiable rasterizer, it can also be used to train on the non-pinhole camera dataset. We illustrate the results obtained from training on the non-pinhole cam-

era dataset MATTERPORT [2] directly, where 3D-GS fails entirely to reconstruct the scene, as shown in Figure 5.



Fig. 5: We show the results of our method trained on the non-pinhole camera dataset MATTERPORT [2].

Table 1: PSNR scores for Mip-NeRF360 scenes.

	MipNeRF360 Indoor					MipNeRF360 Outdoor					Avg.	
	Room	Counter	Kitchen	Bonsai	Avg.	Bicycle	Flowers	Garden	Stump	Treehill		Avg.
Plenoxels [3]	27.59	23.62	23.42	24.67	24.83	21.91	20.10	23.49	20.66	22.25	21.68	23.08
INGP-Base [8]	29.27	26.44	28.55	30.34	28.65	22.19	20.35	24.60	23.63	22.36	22.63	25.30
INGP-Big [8]	29.69	26.69	29.48	30.69	29.14	22.17	20.65	25.07	23.47	22.37	22.75	25.59
M-NeRF360 [1]	31.63	29.55	32.23	33.46	31.72	24.37	21.73	26.98	26.40	22.87	24.47	27.69
3D-GS [5]	30.63	28.70	30.32	31.98	30.41	25.25	21.52	27.41	26.55	22.49	24.64	27.21
Ours	31.58	29.05	31.23	32.31	31.04	25.07	21.54	27.15	26.57	22.77	24.62	27.48

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)
2. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: 2017 International Conference on 3D Vision (3DV). pp. 667–676. IEEE Computer Society (2017)
3. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5501–5510 (2022)

Table 2: PSNR scores for Tanks&Temples and Deep Blending scenes.

	Tanks&Temples			Deep Blending			Avg.
	Truck	Train	Avg.	Dr Johnson	Playroom	Avg.	
Plenoxels [3]	23.22	18.93	21.07	23.14	22.98	23.06	22.07
INGP-Base [8]	23.26	20.17	21.72	27.75	19.48	23.62	22.67
INGP-Big [8]	23.38	20.46	21.92	28.26	21.67	24.96	23.44
M-NeRF360 [1]	24.91	19.52	22.22	29.14	29.66	29.40	25.81
3D-GS [5]	25.19	21.10	23.14	28.77	30.04	29.41	26.27
Ours	25.19	21.67	23.43	29.14	29.89	29.51	26.47

- Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)* **37**(6), 1–15 (2018)
- Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* **42**(4) (2023)
- Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)* **36**(4), 1–13 (2017)
- Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 20654–20664 (2024)
- Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* **41**(4), 1–15 (2022)
- Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2024)