3DGUT: Enabling Distorted Cameras and Secondary Rays in Gaussian Splatting
Supplementary Material

In this supplementary material, we present an extension to generalized Gaussian particles (Sec. A), derive a numerically stable scheme for computing the partial derivative through the proposed 3D particle evaluation (Sec. B, cf. Sec. 4.2), and provide further ablations of the proposed UT-based rasterization (Sec. C). We also include details on autonomous vehicle dataset reconstructions (Sec. D). Finally, we summarize the Gaussian rasterization algorithm and demonstrate that our method serves as a drop-in replacement for a small part of it (Sec. E). Please refer to the ACCOMPANYING VIDEO for more qualitative results.

A. Generalized Gaussian Particles

In 3DGRT [34] the authors propose to use particles with different kernel functions and their most efficient approach is based on a *generalized Gaussians of degree* 2. In Tab. 4 we demonstrate that our approach supports different particles as well. Different to [34], we define a generalized Gaussians kernel function of degree n as

$$\rho(\boldsymbol{x}) = \exp(-\lambda((\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}))^{\frac{n}{2}}) \tag{12}$$

with $\lambda = \frac{r^2}{r^n}$ a scale factor defined to get the same kernel response at a given distance r as the reference Gaussian kernel (we use $r = 3$). Note that 3DGRT *generalized Gaussians of degree* 2 corresponds to our generalized Gaussians kernel of degree 4.

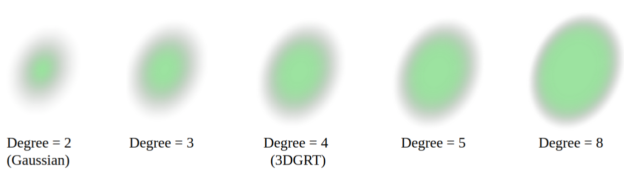

Figure 10. Rendering the same generalized Gaussian particle with different degrees. Higher degree particles are denser and have a steeper and narrower fall-off.

Fig. 10 illustrates the effect of using a different kernel function on the particle extent and density. Tab. 4 shows how the degree of the generalized Gaussian kernel function permits to better control the trade-off between the rendering quality and speed.

B. Derivation of Backward Gradients

In the following, we provide a step-by-step derivation of $\partial\alpha/\partial\boldsymbol{\mu}$. The derivations of $\partial\alpha/\partial\boldsymbol{S}$ and $\partial\alpha/\partial\boldsymbol{R}$ follow analogously.

Table 4. Quality and speed tradeoffs computed on MipNERF360 [1] (excluding *flower* and *treehill* for fair comparison with 3DGRT) for various particle generalized Gaussian kernel functions. Note that our kernel of degree= 4 corresponds to the generalized Gaussian of degree= 2 proposed in 3DGRT [34].

	MipNERF360			
	Ours (sorted)		3DGRT	
Kernel function	PSNR↑	FPS↑	PSNR↑	FPS↑
Degree = 2 (Gaussian)	28.77	207	28.69	55
Degree = 3	28.71	217	/	/
Degree = 4 (3DGRT)	28.46	233	28.71	78
Degree = 5	28.33	238	/	/
Degree = 8	27.63	243	/	/

Remember that $\alpha = \sigma\rho(\boldsymbol{o} + \tau_{\text{max}}\boldsymbol{d})$ and consider that τ_{max} can be defined in the canonical Gaussian space as

$$\tau_{\text{max}_g} = -\boldsymbol{o}_g^T\frac{\boldsymbol{d}_g}{||\boldsymbol{d}_g||}, \tag{13}$$

where $\boldsymbol{o}_g = \boldsymbol{S}^{-1}\boldsymbol{R}^T(\boldsymbol{o}-\boldsymbol{\mu})$ and $\boldsymbol{d}_g = \boldsymbol{S}^{-1}\boldsymbol{R}^T\boldsymbol{d}$ denote the ray origin and ray direction expressed in Gaussian canonical space, respectively. An illustration of the geometric relationship between values is provided in Fig. 11.

Let $\omega_g^2 = ||\boldsymbol{o}_g + \tau_{\text{max}_g}\frac{\boldsymbol{d}_g}{||\boldsymbol{d}_g||}||^2$ denote the squared distance from the Gaussian particle center to the point of maximum response such that $\alpha = \sigma e^{-0.5\omega_g^2}$. The partial derivatives can be computed as

$$\frac{\partial\alpha}{\partial\omega_g^2} = -0.5\sigma e^{-0.5\omega_g^2} \tag{14}$$

$$\frac{\partial\omega_g^2}{\partial\boldsymbol{o}_g} = 2\boldsymbol{o}_g + 2\tau_{max_g}\frac{\boldsymbol{d}_g}{||\boldsymbol{d}_g||} \tag{15}$$

$$\frac{\partial\boldsymbol{o}_g}{\partial\boldsymbol{\mu}} = -\boldsymbol{S}^{-1}\boldsymbol{R}^T \tag{16}$$

C. Gaussian Projection Quality

While Monte Carlo sampling (cf. Fig. 2) is expensive to compute, it provides accurate reference distributions for assessing the quality of both EWA and the proposed UT-based projection methods. This assessment can be quantified using the Kullback–Leibler (KL) divergence between both 2d distributions, where lower KL values indicate the projected Gaussians better approximate the reference projections. In Fig. 12, we evaluate the KL divergence for a fixed reconstruction (`MipNERF360 bicycle` [1]). Specifically, for

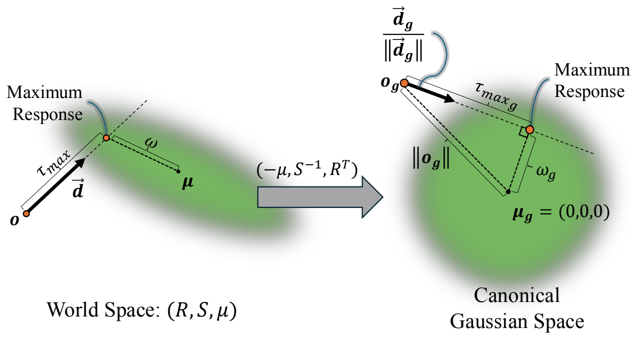

Figure 11. An illustration of the geometric transformation of a Gaussian from world space to canonical Gaussian space.

each visible Gaussian, we compare the projections obtained using either method under different camera and pose configurations against MC-based references (using 500 samples per reference). The resulting KL divergence distributions are visualized in the histograms at the bottom.

While both distributions of divergences are consistent for the static pinhole camera case (first column), UT-based projections are more accurate compared to EWA-based estimates for the static fisheye camera case (third column), indicating that UT yields a better approximation in case of higher non-linearity of the projection. For rolling-shutter camera poses (second and fourth columns), RS-aware UT-based projections still approximate the RS-aware MC references well. In contrast, RS-unaware EWA linearizations break down and fail to approximate this case (histogram domains are capped to 0.04 for clearer visualization, but the EWA-based projections have a long tail distribution of larger KL values still). The tearing artifacts observed in EWA-based RS renderings arise from these inaccurate projections, leading to incorrect pixel-to-Gaussian associations during the volume rendering step.

D. Waymo Autonomous Vehicle Dataset

For comparison on the Waymo Open Perception dataset [46], we follow [34] and select 9 static scenes. Images in the dataset are captured using a distorted camera with rolling shutter sensor, mounted on the front of the vehicle. To adapt to this dataset, we incorporated additional losses for lidar depth and image opacity, combining them as a weighted sum: the L1-loss $\mathcal{L}_1^{\text{depth}}$ for depth and the L2-loss $\mathcal{L}_2^{\text{opacity}}$ for opacity, such that $\mathcal{L}^{\text{waymo}} = \mathcal{L} + 0.001\mathcal{L}_1^{\text{depth}} + 0.05\mathcal{L}_2^{\text{opacity}}$, where $\mathcal{L}$ is the loss function defined in Sec. 4.4. We initialized scenes using a colored point cloud generated by combining screen-projected lidar points with camera data. For the case of 3DGS [18], we rectify the images and ignore the rolling shutter effects following [6]. For 3DGRT [34] and our method, we make use of the full camera model and

Table 5. On the Waymo [46] autonomous vehicles dataset that was captured with distorted camera model and rolling-shuter sensor, our method achieves better quality compared to 3DGRT [34]. Note that 3DGS [18] requires the training and evaluation to be done on rectified images without rolling shutter effects and is hence not directly comparable.

Method\Metric	Waymo	
	PSNR↑	SSIM↑
3DGS [18]	29.83	0.917
3DGRT [34]	29.99	0.897
Ours (sorted)	30.16	0.900

compute the rolling shutter effect correctly. The quantative results are reported in Tab. 5 and qualitative visualizations are available in Fig. 13.

E. Gaussian Rasterization Algorithm

To show that our proposed UT-based projection can be used as a drop-in replacement to the 3DGS rasterization pipeline, we summarize their pipelines in terms of pseud-code in Algs. 1 and 2. Note that we keep the Alg. 1 intact and only adapt the the ESTIMATE2DGAUSSIAN function in Alg. 1.

Algorithm 1 RASTERIZE

Input: Gaussian parameters: $\{\boldsymbol{\mu}_i, \boldsymbol{R}_i, \boldsymbol{S}_i, \sigma_i\}_{i=1}^N$, camera extrinsic $\boldsymbol{W}$, camera intrinsic $\boldsymbol{D}$
Output: 2D Means: $\boldsymbol{v}_{\boldsymbol{\mu}_i}$, 2D AABBs: $\boldsymbol{r}_i$

for i in $1 \dots N$ **do** ▷ *iterate over the particles*
  $\boldsymbol{v}_{\boldsymbol{\mu}_i}, \boldsymbol{\Sigma}'_i = \text{Estimate2DGaussian}(\boldsymbol{\mu}_i, \boldsymbol{R}_i, \boldsymbol{S}_i, \boldsymbol{W}, \boldsymbol{D})$
  $\boldsymbol{h}_i = \text{Extent}(\boldsymbol{\Sigma}'_i, \sigma_i)$ ▷ *use opacity to compute a tighter 2D extent*
  $\boldsymbol{r}_i = \text{ComputeRectangle}(\boldsymbol{h}_i, \boldsymbol{v}_{\boldsymbol{\mu}_i})$ ▷ *2D rectangle used for tile-based rasterization*

Algorithm 2 ESTIMATE2DGAUSSIAN

Input: Gaussian parameters: $\boldsymbol{\mu}, \boldsymbol{R}, \boldsymbol{S}$, camera extrinsic $\boldsymbol{W}$, camera intrinsic $\boldsymbol{D}$, α, β, κ
Output: 2D Mean: $\boldsymbol{v}_{\boldsymbol{\mu}}$, 2D Covariance: $\boldsymbol{\Sigma}'$

$\lambda = \alpha^2(3+\kappa) - 3$
$\boldsymbol{x} = \text{SampleSigmaPoints}(\boldsymbol{\mu}, \boldsymbol{R}, \boldsymbol{S}, \lambda)$ ▷ *Eq. (6)*
$\boldsymbol{w} = \text{ComputeWeights}(\alpha, \beta, \lambda)$ ▷ *Eqs. (7) and (8)*
$\boldsymbol{v}_{\boldsymbol{x}} = \text{ProjectPoints}(\boldsymbol{x}, \boldsymbol{W}, \boldsymbol{D})$ ▷ *evaluate $g(\boldsymbol{x})$*
$\boldsymbol{v}_{\boldsymbol{\mu}} = \text{EstimateMean}(\boldsymbol{v}_{\boldsymbol{x}}, \boldsymbol{w})$ ▷ *Eq. (9)*
$\boldsymbol{\Sigma}' = \text{EstimateCovariance}(\boldsymbol{v}_{\boldsymbol{\mu}}, \boldsymbol{v}_{\boldsymbol{x}}, \boldsymbol{w})$ ▷ *Eq. (10)*
return $\boldsymbol{v}_{\boldsymbol{\mu}}$ $\boldsymbol{\Sigma}'$

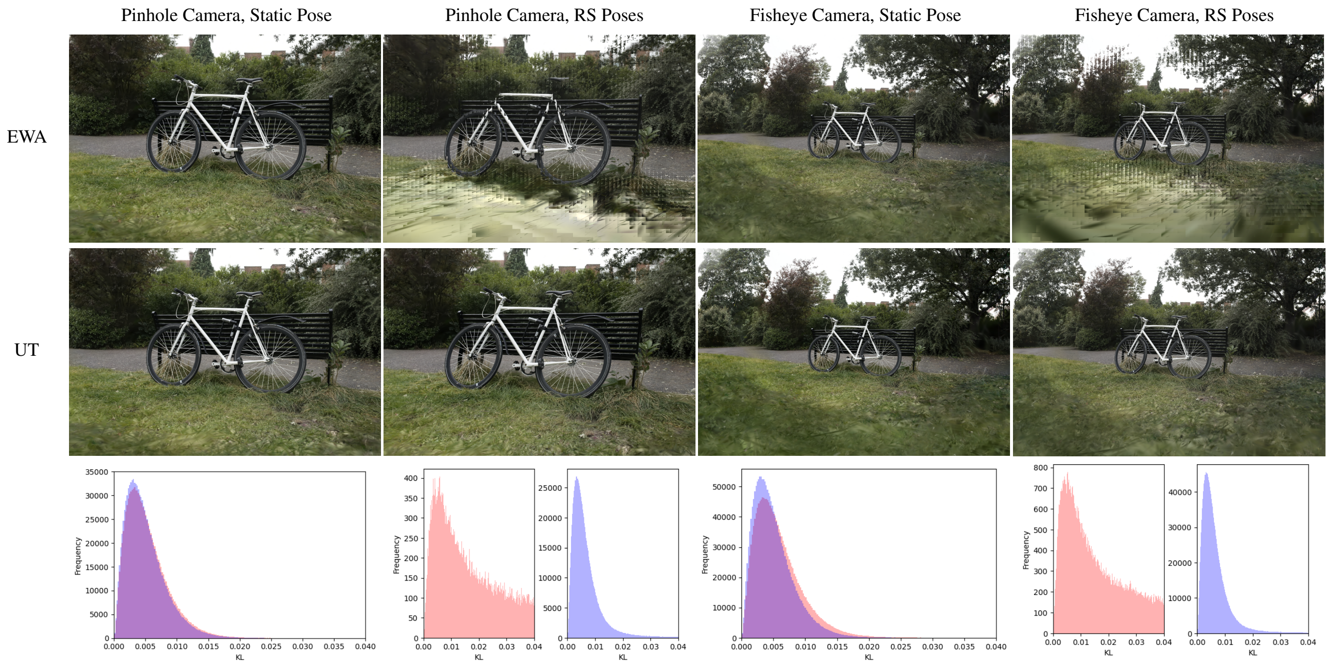

Figure 12. Gaussian Projection Quality: for both distortion-free pinhole and fisheye camera models, as well as static and rolling-shutter (RS, top-top-bottom shutter direction) poses, we evaluate the Kullback–Leibler (KL ↓) divergence of each Gaussian projected using either EWA (●) or UT-based (●) projections against *Monte-Carlo*-based reference projection. The distribution of KL-divergences for each rendering is shown in the histograms below

F. Additional Qualitative Results

In the main paper, Fig. 4 showcased a qualitative comparison of our model against various baselines on the MipNeRF360 dataset [1]. Expanding on this, Fig. 14 provides an additional comparison using a different dataset (Tanks & Temples [21]). This figure highlights the qualitative performance of our method alongside the baseline approaches: 3DGS [18], 3DGRT [34], and StopThePop [37]. The results demonstrate that our approach delivers comparable or superior rendering quality.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 5, 6, 1, 3

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 3, 6

[3] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 2

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[5] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *The Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[6] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024. 2

[7] Jorge Condor, Sebastien Speierer, Lukas Bode, Aljaz Bozic, Simon Green, Piotr Didyk, and Adrian Jarabo. Don't Splat your Gaussians: Volumetric Ray-Traced Primitives for Modeling and Rendering Scattering and Emissive Media, 2024. 2, 3, 5, 8

[8] Daniel Duckworth, Peter Hedman, Christian Reiser, Peter Zhizhin, Jean-François Thibert, Mario Lučić, Richard Szeliski, and Jonathan T. Barron. Smerf: Streamable memory efficient radiance fields for real-time large-scene exploration, 2023. 2

[9] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *arXiv preprint arXiv:2103.10380*, 2021. 2

[10] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024. 2

[11] Antoine Guédon and Vincent Lepetit. Gaussian frosting:

Figure 13. Qualitative comparison of our novel-view synthesis results against 3DGRT on the Waymo dataset [46]. Images are sampled from 8 different scenes.

Figure 14. Qualitative comparison of our novel-view synthesis results against the baselines on the Tanks & Temples [21] dataset.

Editable complex radiance fields with real-time rendering. *ECCV*, 2024. 2

[12] Fredrik Gustafsson and Gustaf Hendeby. Some relations between extended and unscented kalman filters. *IEEE Transactions on Signal Processing*, 60(2):545–555, 2012. 2

[13] Florian Hahlbohm, Fabian Friederichs, Tim Weyrich, Linus Franke, Moritz Kappel, Susana Castillo, Marc Stamminger, Martin Eisemann, and Marcus Magnor. Efficient perspective-correct 3d gaussian splatting using hybrid transparency, 2024. 5

[14] Letian Huang, Jiayang Bai, Jie Guo, Yuanqi Li, and Yanwen Guo. On the error analysis of 3d gaussian splatting and an optimal projection strategy. *arXiv preprint arXiv:2402.00752*, 2024. 2, 3, 4

[15] Faris Janjoš, Lars Rosenbaum, Maxim Dolgov, and J. Marius Zöllner. Unscented autoencoder, 2023. 3

[16] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the kalman filter to nonlinear systems. In *Defense, Security,*

and Sensing, 1997. 2, 3

[17] Simon J Julier, Jeffrey K Uhlmann, and Hugh F Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of 1995 American Control Conference-ACC'95*, pages 1628–1632. IEEE, 1995. 3

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2, 3, 4, 5, 6, 7, 8

[19] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4), 2024. 2

[20] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Jeff Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *arXiv preprint arXiv:2404.09591*, 2024. 2, 5

[21] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 5, 6, 3, 4

[22] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)*, 40(4), 2021. 1

[23] Christoph Lassner and Michael Zollhofer. Pulsar: Efficient sphere-based neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1440–1449, 2021. 2

[24] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21719–21728, 2024. 2

[25] Zimu Liao, Siyan Chen, Rong Fu, Yi Wang, Zhongling Su, Hao Luo, Linning Xu, Bo Dai, Hengjie Li, Zhilin Pei, et al. Fisheye-gs: Lightweight and extensible gaussian splatting module for fisheye cameras. *arXiv preprint arXiv:2409.04751*, 2024. 3, 4, 6, 7

[26] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, and Wenming Yang. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *CVPR*, 2024. 2

[27] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2

[28] Yang Liu, He Guan, Chuanchen Luo, Lue Fan, Junran Peng, and Zhaoxiang Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians, 2024. 2

[29] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2, 5

[30] Alexander Mai, Peter Hedman, George Kopanas, Dor Verbin, David Futschik, Qiangeng Xu, Falko Kuester, Jonathan T. Barron, and Yinda Zhang. Ever: Exact volumetric ellipsoid rendering for real-time view synthesis, 2024. 2, 3, 5, 6, 8, 9

[31] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Francisco Vicente Carrasco, Markus Steinberger, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources, 2024. 2

[32] Marilena Maule, João Comba, Rafael Torchelsen, and Rui Bastos. Hybrid transparency. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, page 103–118, New York, NY, USA, 2013. Association for Computing Machinery. 5

[33] 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. In *ECCV*, 2020. 1, 2

[34] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics and SIGGRAPH Asia*, 2024. 2, 3, 4, 5, 6, 7, 8, 9, 1

[35] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 1, 2

[36] Steven G. Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, and Martin Stich. Optix: A general purpose ray tracing engine. *ACM Trans. Graph.*, 29(4), 2010. 5

[37] Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. StopThePop: Sorted Gaussian Splatting for View-Consistent Real-time Rendering. *ACM Transactions on Graphics*, 4 (43), 2024. 2, 4, 5, 6, 3

[38] Christian Reiser, Richard Szeliski, Dor Verbin, Pratul P. Srinivasan, Ben Mildenhall, Andreas Geiger, Jonathan T. Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *SIGGRAPH*, 2023. 2

[39] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *European Conference on Computer Vision*, 2020. 1

[40] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. *ACM Transactions on Graphics (ToG)*, 41(4):1–14, 2022. 1

[41] Marco Salvi and Karthikeyan Vaidyanathan. Multi-layer alpha blending. *Proceedings of the 18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2014. 5

[42] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[43] Otto Seiskari, Jerry Ylilammi, Valtteri Kaatrasalo, Pekka Rantalankila, Matias Turkulainen, Juho Kannala, and Arno Solin. Gaussian splatting on the move: Blur and rolling shutter compensation for natural camera motion, 2024. 3

[44] Gopal Sharma, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. Volumetric rendering with baked quadrature fields. *arXiv preprint arXiv:2312.02202*, 2023. 2

[45] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2

[46] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5, 7, 2, 4

[47] Haithem Turki, Vasu Agrawal, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Deva Ramanan, Michael Zollhöfer, and Christian Richardt. Hybridnerf: Efficient neural rendering via adaptive volumetric surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19647–19656, 2024. 2

[48] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000. 3, 4

[49] Ziyu Wan, Christian Richardt, Aljaž Božič, Chao Li, Vijay Rengarajan, Seonghyeon Nam, Xiaoyu Xiang, Tuotuo Li, Bo Zhu, Rakesh Ranjan, et al. Learning neural duplex radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8307–8316, 2023. 2

[50] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 1

[51] Zian Wang, Tianchang Shen, Merlin Nimier-David, Nicholas Sharp, Jun Gao, Alexander Keller, Sanja Fidler, Thomas Müller, and Zan Gojcic. Adaptive shells for efficient neural radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(6):1–15, 2023. 2

[52] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 1

[53] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P. Srinivasan, Richard Szeliski, Jonathan T. Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. *arXiv*, 2023. 1, 2

[54] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details for 3d gaussian splatting, 2024. 2

[55] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 5, 7

[56] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, 2024. 5

[57] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002. 1, 3