Clear-Splatting: Learning Residual Gaussian Splats for Transparent Object Depth Estimation

Aviral Agrawal[1*], Ritaban Roy[1*], Bardienus P. Duisterhof[1*],
Keerthan Bhat Hekkadka[1*], Hongyi Chen[1*], Jeffrey Ichnowski[1]

***Abstract*— Grasping and manipulating transparent objects poses a significant challenge for robots. Recent work showed neural radiance fields (NeRFs) work well for depth perception in scenes with transparent objects, and these depth maps can be used to grasp transparent objects with high accuracy. NeRF-based depth reconstruction can still struggle with challenging transparent objects and lighting conditions. In this work, we study the performance of Gaussian Splatting (3DGS) for depth perception of transparent objects. We compare 3DGS to existing NeRF-based methods, and contribute a new method – Clear-Splatting. This method draws inspiration from Residual-NeRF to leverage a scene prior, since robots often operate in the same area, by first learning *background Splats* of the scene without transparent objects to be manipulated. It then learns *residual Splats* to complete the scene. Our experiments on synthetic dataset show that Clear-Splatting results in competitive depth maps with up to 67.09% lower RMSE and a 87.80% lower MAE for depth estimation compared to NeRF-based baselines. We also discuss challenges faced by Gaussian splatting for transparent objects, such as floaters and slower training.**

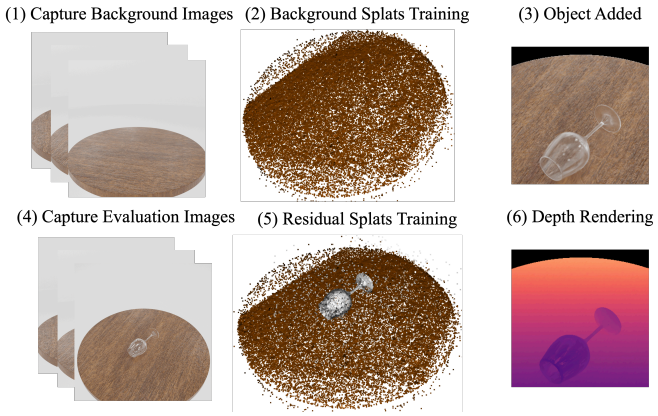

Fig. 1: Clear-Splatting leverages mostly static scenes to improve depth perception. It begins by training *background Splats* (2) of the entire scene without transparent objects (1). The transparent object is then added on to the background scene (3). Then we learn *residual Splats* (5) to complement the *background Splats* from (4). The rendered depth is shown in (6).

I. Introduction

Enabling robots to dextrously manipulate transparent objects can be put to use in various downstream applications. Robots often use depth images of objects to decide what action (e.g., pull, lift, or drop) to perform. However, common depth sensors struggle to capture depth images for arbitrary transparent objects [1], [2], [3], [4] and the same is true for monocular depth estimators [5]. Learning-based approaches for transparent object depth estimation work well in-distribution, but can struggle to generalize outside their training data [1]. The lack of surface features on transparent objects also makes it challenging to retrieve depth maps using approaches such as COLMAP [6].

Neural Radiance Fields (NeRFs) [7] are implicit neural network scene representations trained on multiple views of the same scene and capable of state-of-the-art novel view synthesis. Dex-NeRF [1] and Evo-NeRF [8] showed that NeRFs can perceive depth of transparent objects to grasp them. However, these methods also showed that NeRFs tend to struggle with transparent objects, such as wine glasses or kitchen foil with challenging lighting conditions. Dex-NeRF, while achieving high grasp success rates, was slow to compute. To address this, Residual-NeRF [9] contributed a method which uses a *background NeRF*, a *Residual-NeRF*, and a *Mix-Net* to speed up training and improve depth maps.

In this work, we study the performance using Gaussian Splatting [10] (3DGS) for transparent object depth perception. We propose Clear-Splatting (Figure 1), a method to leverage a strong scene prior to improve the depth perception of transparent objects using 3DGS. In many scenarios, the geometry of the robot's work area is mostly static and opaque, e.g., shelves, desks, and tables. Inspired by Residual-NeRF[9], Clear-Splatting leverages the static and opaque parts of the scene as a prior, to reduce ambiguity and improve depth perception. Clear-Splatting first learns *background Splats* of the entire scene by training on images without transparent objects present. Clear-Splatting then uses images of the full scene with the transparent objects to learn *residual Splats*. It additionally uses a depth-based pruning technique to remove potential 'floaters', which are floating Gaussians of high opacity irregularly positioned through the scene, and consequently outputs a cleaner depth map.

We evaluate Clear-Splatting on four photo-realistic synthetic scenes and compare its performance to other relevant NeRF algorithms. We compare depth reconstruction quality and learning speed. The results suggest that Clear-Splatting improves on the NeRF-based approaches with a 67.09% lower RMSE and an 87.80% lower MAE in depth estimation. The results suggest NeRF-based approaches converge significantly faster, although their final reconstruction quality is lower compared to 3DGS. We highlight several research opportunities by using 3DGS for transparent depth mapping, such as speeding up training and removing 'floaters'.

* Authors contributed equally to this work.
[1]Carnegie Mellon University, The Robotics Institute
Authors' email: {avirala, ritabanr, bduister, kbhathek, hongyic, jichnows}@andrew.cmu.edu

II. Related Work

Neural Rendering for Novel View Synthesis: Clear-Splatting builds on prior work in novel view synthesis to render depth maps from the 3D reconstruction. A popular novel-view synthesis approach is NeRF [7], which uses neural networks to learn a mapping from a 3D point and view angle to a density and an RGB radiance. NeRF renders pixels using existing volume rendering techniques. Subsequent works improve NeRF along several axes: e.g., speeding up training and inference time via novel representations and system optimizations [11], [12], [13], [14], [15], [16], [17], [11], [18], or depth supervision [19], [20], [21], [22], [23]. Other works extend NeRF to more challenging conditions, such as sparser camera views [24], [25], [26], [27], fewer extrinsic camera calibrations [28], [29], [30], [31], transparent objects [8], [1], [9] and reflective surfaces [32].

Depth Perception of Transparent Objects: Several works have proposed methods for accurate depth perception, shape estimation, and/or pose estimation. Xie et al. [33] developed a pipeline based on transformer neural networks capable of transparent object segmentation. Phillips et al. [3] leveraged a random forest algorithm to extract the pose and shape of transparent objects. Xu et al. [4] contributed an algorithm for estimating the 6-degrees-of-freedom (DOF) pose of a transparent object using only a single RGBD image. Wang et al. [34] contributed MVTrans for depth mapping, segmentation, and pose estimation of transparent objects. Chen et al. [2] contributed a benchmark dataset for segmentation, object pose estimation, and depth completion.

Ichnowski et al. [1] showed how NeRFs can be leveraged to infer state-of-the-art depth perception of transparent objects, and unlike training depth supervision-centric approaches, did not require prior training on a set of objects.

3D Gaussian Splatting [10] proposed a differential rasterizer to render a large number of Gaussian *Splats*, each with their state including color, position, and covariance matrix. Clear-Splatting builds on 3D Gaussian Splatting for better depth rendering.

III. Problem Statement

Given a set of images of a scene without the transparent objects present, $\{I_{\text{bg}_i}\}_{i=1}^{N_{\text{bg}}}$, where N_{bg} are background images each with camera matrix P_{bg} (i.e., the intrinsics and extrinsics). In addition, we also have access to $\{I_{\text{res}_i}\}_{i=1}^{N_{\text{res}}}$, where N_{res} are the same scene images with transparent object present with P_{res}. P_{bg} and P_{res} are not necessarily the same, N_{bg} and N_{res} are also not necessarily equal.

The objective is to recover *novel view depth maps* from any given camera pose P, and then use the novel views for downstream tasks, like grasp planning [1], [35]. Perceived holes in objects, i.e., locations where geometry is not clear, can lead to gripper collisions. On the other hand, hallucinating non-existent surfaces may lead to occlusions of the object of interest, leading to no viable grasp location. Thus, the goal of depth estimation is to reduce the per-pixel error in the depth maps with RMSE and MAE error metrics as defined in section V-C.

IV. Method

Clear-Splatting, shown in Figure 1, recovers depth using multiple camera views by first learning background Splats from $\{I_{\text{bg}_i}\}_{i=1}^{N_{\text{bg}}}$, and then learn residual Splats from $\{I_{\text{res}_i}\}_{i=1}^{N_{\text{res}}}$. We build on Gaussian Splatting [10] and review preliminaries of novel-view synthesis with Gaussian Splatting (Section IV-A). Once trained, we then use Splats representation to render depth maps (Section IV-B).

A. Preliminary: Gaussian Splatting

3D Gaussian Splatting [10] learns scene representation by rendering a large set of Gaussians each defined by their mean position μ and covariance matrix Σ. Thus, for each $x \in \mathbb{R}^3$, its Gaussian $G(x)$ is

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}, \tag{1}$$

Directly optimizing the covariance matrix Σ would lead to infeasible covariance matrices, as they must be positive semi-definite to have a physical meaning. Instead, Gaussian Splatting [10] proposes decomposing Σ into a rotation R and scale S for each Gaussian,

$$\Sigma = RSS^TR^T,$$

and optimize R, S, and the mean position.

Given the transformation W of a camera, the covariance matrix can be projected into image space as

$$\Sigma' = JW\Sigma W^TJ^T,$$

where J is the Jacobian of the affine approximation of the projective transformation.

During rendering, we compute the color C of a pixel by blending N ordered Gaussians overlapping the pixel :

$$C = \sum_{i \in N} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

where c_i is the color of each point and α_i is given by evaluating a 2D Gaussian with covariance multiplied by a learned per-point opacity [36], [10].

B. Preliminary: Depth from Gaussian Splatting

Previous works like [37] and [38], calculate depth from Gaussians using an equation similar to eq. 2, with the colors c_i replaced by the distance of the Gaussian. This can be viewed as alpha-blending the depth of the ordered Gaussians.

For Clear-Splatting, we adopt an approach inspired by Ichnowski et al. [1] and Luiten et al. [39]. We set the per-pixel depth as the depth of the Gaussian center where the accumulated transmittance of the ray drops below a threshold m. If a ray does not reach this threshold it is assigned a high default depth. This method of calculating depth avoids perceiving *floaters* around depth boundaries compared to alpha-blending the depths of the Gaussians. We use $m = 0.7$ in our experiments, obtained empirically.

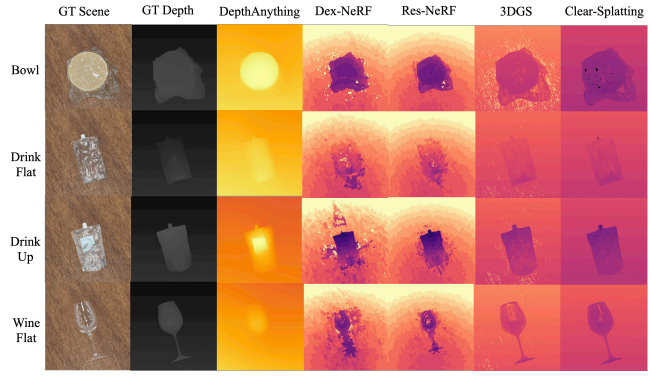

Fig. 2: From left to right, synthetic Blender scenes, depth maps for ground truth, Depth Anything (correct up to a scale), 3D geometry baselines, and Clear-Splatting. The results show that Clear-Splatting improves depth maps with fewer holes and less noise.

TABLE I: Root Mean Square Error (RMSE) in synthetic Blender Scenes.

	Bowl	Drink Flat	Drink Up	Wine Flat
Method		RMSE ↓		
Dex NeRF [1]	0.0381	0.0745	0.0699	0.0396
Res NeRF [9]	0.0226	0.0237	0.0332	0.0320
3DGS [10]	0.0200	**0.0074**	0.0137	0.0139
Clear-Splatting	**0.0139**	0.0078	**0.0130**	**0.0107**

C. Learning Residual Splats

Clear-Splatting first initializes Gaussians in random states and optimizes for view reconstruction to find *background Splats* of the environment without the transparent object present. We then reset the optimizer and introduce a new smaller set of *residual Gaussians*, and optimize the state of all Gaussians simultaneously. The *background Gaussians* are not frozen to facilitate learning of visual effects introduced by the added object, such as shadow and reflections.

In addition, we also introduce *depth pruning* where we prune the Gaussians that are within a threshold d distance in a given view, at a set frequency of training iterations and lasts until Gaussian densification [10] happens to help the remaining Gaussians adjust the learnt features. This is instrumental in dealing with occlusions in the depth domain, resulting primarily from *floaters* which are sets of unoptimized Gaussians that persist due to textureless background in the training views.

V. Experiments

Common depth sensors struggle to infer the depth of transparent objects accurately, which makes it challenging to create a real dataset with accurate depth maps of such objects. So, we evaluate Clear-Splatting against the baselines (Section V-A) on synthetic photo-realistic Blender scenes (Section V-B) with ground-truth depth. The results suggest Clear-Splatting outperforms the baselines by generating higher-quality depth maps. Finally, we also evaluate the training speed of Clear-Splatting (Section V-D) on a single NVIDIA GeForce RTX 4090 GPU with 24GB of VRAM.

A. Baselines

We evaluate the following baselines: Dex-NeRF [1] and Residual-NeRF (Res-NeRF) [9]. While other multi-view stereo (MVS) methods for transparent objects exist, to the best of our knowledge, they do not accept arbitrary poses. All NeRF-based approaches are implemented in Torch-NGP [40], which uses a multi-resolution hash encoding. Better results might be achieved without hash encoding at the cost of significantly higher training time. We also compare against 3DGS [10] which, in contrast to Clear-Splatting, uses the default alpha blending to render the depth map as in [37]. It optimizes the entire scene without any background priors, and suffers from *floaters* due to the lack of *depth-pruning*.

TABLE II: Root Mean Square Error (RMSE) in synthetic scenes [Top view].

	Bowl	Drink Flat	Drink Up	Wine Flat
Method		RMSE ↓		
Dex NeRF [1]	0.0415	0.3212	0.0627	0.0432
Res NeRF [9]	0.0172	0.0163	0.0172	0.0171
3DGS [10]	0.0291	0.0120	0.0168	0.0114
Clear-Splatting	**0.0149**	**0.0086**	**0.0083**	**0.0057**

TABLE III: Mean Absolute Error (MAE) in synthetic scenes.

	Bowl	Drink Flat	Drink Up	Wine Flat
Method		MAE ↓		
Dex NeRF [1]	0.0203	0.0156	0.0195	0.0248
Res NeRF [9]	0.0147	0.0295	0.0203	0.0163
3DGS [10]	0.0070	**0.0029**	0.0038	0.0044
Clear-Splatting	**0.0040**	0.0036	**0.0038**	**0.0043**

B. Synthetic Blender Data

Clear-Splatting cannot be evaluated on existing datasets such as ClearPose [2], which captures 63 transparent objects, due to the lack of background images for training the background Splats. Therefore, we use four of the scenes made available in Residual-NeRF [9], as shown in Figure 2. The dataset was rendered using Blender [41]. Residual-NeRF [9] used poses in the hemisphere from the original NeRF datasets [7] to render train, test, and validation images. Background NeRF and residual NeRF receive images taken from the same 100 train poses.

C. Blender Depth Results

We evaluate the inferred depth maps by Clear-Splatting and the baselines by comparing them against the ground truth provided by Blender.

1) Quantitative Comparison: We compare Clear-Splatting against Dex-NeRF, Res-NeRF and 3DGS by computing the MAE (Equation 3) and RMSE (Equation 4).

$$\text{MAE} = \frac{\sum_{(i,\mathbf{r})\in\Omega_r} \|\hat{D}_i(\mathbf{r}) - D_i(\mathbf{r})\|_1}{n}, \tag{3}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{(i,\mathbf{r})\in\Omega_r} \|\hat{D}_i(\mathbf{r}) - D_i(\mathbf{r})\|^2}{n}}, \tag{4}$$

Here $i \in [0, ..., N]$ is the frame number, $\mathbf{r}$ is the pixel location, and Ω_r is the set of all pixel locations across frames. $\hat{D}(\mathbf{r})$ is the inferred depth in meters, $D(\mathbf{r})$ is the GT depth in meters. We crop each image before evaluation to focus on the transparent object and not bias the results with the

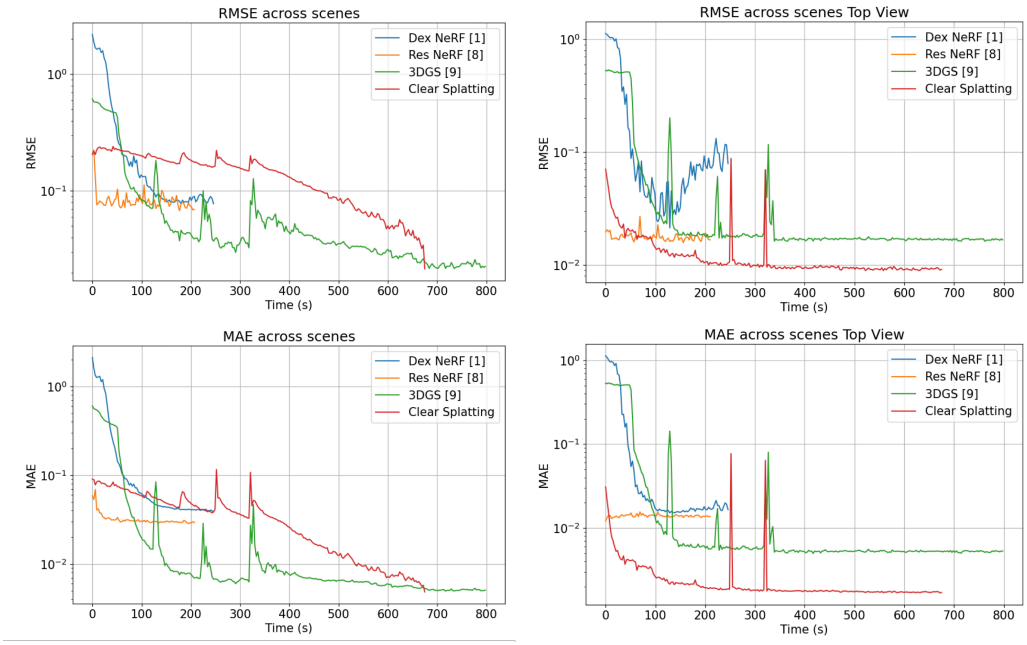

Fig. 3: Left column: For each method in tables I, III, we average RMSE/MAE across scenes and plot them across training time. These plots show that Clear-Splatting gives the best performance across methods at the cost of increased convergence time. Right column: For each method in tables II, IV, we average values across scenes and plot those values across training time. For top view, Clear-Splatting performs the best in the least time across methods.

TABLE IV: Mean Absolute Error (MAE) in synthetic scenes [Top view].

	Bowl	Drink Flat	Drink Up	Wine Flat
Method		MAE ↓		
Dex NeRF [1]	0.0226	0.0368	0.0215	0.0154
Res NeRF [9]	0.0155	0.0139	0.0150	0.0141
3DGS [10]	0.0110	0.0032	0.0044	0.0034
Clear-Splatting	**0.0022**	**0.0017**	**0.0015**	**0.0015**

background. In each crop, the entire transparent object is visible, while the background is partially cropped out.

Table I shows the RMSE and Table III the MAE for Clear-Splatting compared against the baselines. The NeRF-based baselines use hash-encoding which considerably speeds up NeRF training, required for our evaluation, at the cost of quality reduction. We also evaluate the metrics for a single top-view, shown in Table II for RMSE and Table IV for MAE. This is important since it is a good potential view for the gripper. The tables suggest that Clear-Splatting outperforms the baselines except for the 'Drink Flat' scene MAE.

2) Qualitative Comparison: Figure 2 shows the depth maps inferred by Clear-Splatting and the relevant baselines. The depth maps resulting from 3DGS-based approaches appear less 'blobby' and contain less noise, which explains the quantitative results. The results could be further improved by tuning m for each scene, we have opted for setting $m = 3$ for NeRF-based approaches and $m = 0.7$ for Clear-Splatting. Ichnowski et al. [1] found $m = 15$ to work best, evaluating different scenes and using a NeRF implementation without multi-resolution hash encoding.

D. Training Speed

To evaluate the quality of depth reconstruction over time, we log the RMSE and MAE from the predicted depths averaged over all synthetic scenes during training and is shown in Figure 3, assuming a pre-trained background 3DGS for Clear-Splatting. Compared to the Gaussian splatting based methods, Res-NeRF converges significantly faster at the expense of a much higher RMSE/MAE. We also show the same plots for the top view separately, as it is the most representative view for grasping.

The results show that Clear-Splatting utilizes the background Splats to speed up training initially but is unable to leverage the learnt prior to speed up the entire training process. However, with slightly higher training time, Clear-Splatting outperforms 3DGS [10] and the NeRFs in terms of both RMSE and MAE across all scenes and views. The performance improvement becomes more significant for the top view of the scenes, which is essential for grasping.

VI. Conclusion and Discussion

In this work, we study the performance of using Gaussian Splatting [10] (3DGS) for transparent object depth perception. We propose Clear-Splatting (Figure 1), a method to leverage a strong scene prior to improving depth perception of transparent objects using 3DGS. Clear-Splatting begins by learning *background Splats* of the entire scene without transparent objects. Following this, *residual Splats* are trained to complement the *background Splats*. The results suggest that Clear-Splatting learns a competitive depth reconstruction.

This work could be improved by comparing against more MVS methods non-specific to transparent objects. Future work may also include combining Clear-Splatting with recent advances in depth map completion. Future research could explore the performance across different transparent objects and scene conditions.

REFERENCES

[1] J. Ichnowski*, Y. Avigal*, J. Kerr, and K. Goldberg, "Dex-NeRF: Using a neural radiance field to grasp transparent objects," in *Conference on Robot Learning (CoRL)*, 2020.

[2] X. Chen, H. Zhang, Z. Yu, A. Opipari, and O. C. Jenkins, "Clearpose: Large-scale transparent object dataset and benchmark," in *European Conference on Computer Vision*, 2022.

[3] C. Phillips, M. Lecce, and K. Daniilidis, "Seeing glassware: from edge detection to pose estimation and shape recovery," 06 2016.

[4] C. Xu, J. Chen, M. Yao, J. Zhou, L. Zhang, and Y. Liu, "6dof pose estimation of transparent object from a single rgb-d image," *Sensors*, vol. 20, no. 23, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/23/6790

[5] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," *arXiv preprint arXiv:2401.10891*, 2024.

[6] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[8] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 353–367. [Online]. Available: https://proceedings.mlr.press/v205/kerr23a.html

[9] B. P. Duisterhof, Y. Mao, S. H. Teng, and J. Ichnowski, "Residual-nerf: Learning residual nerfs for transparent object manipulation," in *ICRA*, 2024.

[10] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[11] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[12] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," *CoRR*, vol. abs/2103.13744, 2021. [Online]. Available: https://arxiv.org/abs/2103.13744

[13] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *NeurIPS*, 2020.

[14] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "PlenOctrees for real-time rendering of neural radiance fields," in *ICCV*, 2021.

[15] C. Sun, M. Sun, and H. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *CVPR*, 2022.

[16] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, "Fastnerf: High-fidelity neural rendering at 200fps," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, oct 2021, pp. 14 326–14 335. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.01408

[17] S. Lombardi, T. Simon, G. Schwartz, M. Zollhoefer, Y. Sheikh, and J. Saragih, "Mixture of volumetric primitives for efficient neural rendering," *ACM Trans. Graph.*, vol. 40, no. 4, jul 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459863

[18] M. H. Mubarik, R. Kanungo, T. Zirr, and R. Kumar, "Hardware acceleration of neural graphics," 2023.

[19] K. Deng, A. Liu, J. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2022, pp. 12 872–12 881. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01254

[20] B. Attal, E. Laidlaw, A. Gokaslan, C. Kim, C. Richardt, J. Tompkin, and M. O'Toole, "Törf: Time-of-flight radiance fields for dynamic scene view synthesis," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[21] Y. Wei, S. Liu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo," in *ICCV*, 2021.

[22] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. S. Kaplanyan, and M. Steinberger, "DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks," *Computer Graphics Forum*, vol. 40, no. 4, 2021. [Online]. Available: https://doi.org/10.1111/cgf.14340

[23] E. Sucar, S. Liu, J. Ortiz, and A. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[24] J. Y. Zhang, G. Yang, S. Tulsiani, and D. Ramanan, "NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild," in *Conference on Neural Information Processing Systems*, 2021.

[25] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, "Stereo radiance fields (srf): Learning view synthesis from sparse views of novel scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021.

[26] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, "Sparf: Neural radiance fields from sparse and noisy poses." IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023.

[27] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[28] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[29] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "iNeRF: Inverting neural radiance fields for pose estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[30] Y. Chen, X. Chen, X. Wang, Q. Zhang, Y. Guo, Y. Shan, and F. Wang, "Local-to-global registration for bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8264–8273.

[31] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park, "Self-calibrating neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 5846–5854.

[32] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-NeRF: Structured view-dependent appearance for neural radiance fields," *CVPR*, 2022.

[33] E. Xie, W. Wang, W. Wang, P. Sun, H. Xu, D. Liang, and P. Luo, "Segmenting transparent objects in the wild with transformer," 08 2021, pp. 1194–1200.

[34] Y. R. Wang, Y. Zhao, H. Xu, S. Eppel, A. Aspuru-Guzik, F. Shkurti, and A. Garg, "Mvtrans: Multi-view perception of transparent objects," 2023.

[35] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *6th Annual Conference on Robot Learning*, 2022.

[36] W. Yifan, F. Serena, S. Wu, C. Öztireli, and O. Sorkine-Hornung, "Differentiable surface splatting for point-based geometry processing," *ACM Transactions on Graphics*, vol. 38, no. 6, p. 1–14, Nov. 2019. [Online]. Available: http://dx.doi.org/10.1145/3355089.3356513

[37] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and W. Xinggang, "4d gaussian splatting for real-time dynamic scene rendering," *arXiv preprint arXiv:2310.08528*, 2023.

[38] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," *arXiv preprint arXiv:2309.13101*, 2023.

[39] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," in *3DV*, 2024.

[40] J. Tang, "Torch-ngp: a pytorch implementation of instant-ngp," 2022, https://github.com/ashawkey/torch-ngp.

[41] B. O. Community, "Blender - a 3d modelling and rendering package," 2018. [Online]. Available: http://www.blender.org