Enhancing Sparse-View 3D Reconstruction with Unknown Poses via Geometric Prior Guided Refinement

Chunghwan Lee
NAVER LABS
South Korea
jhlee612@naverlabs.com

Deokhwa Kim
NAVER LABS
South Korea
deokhwa.kim@naverlabs.com

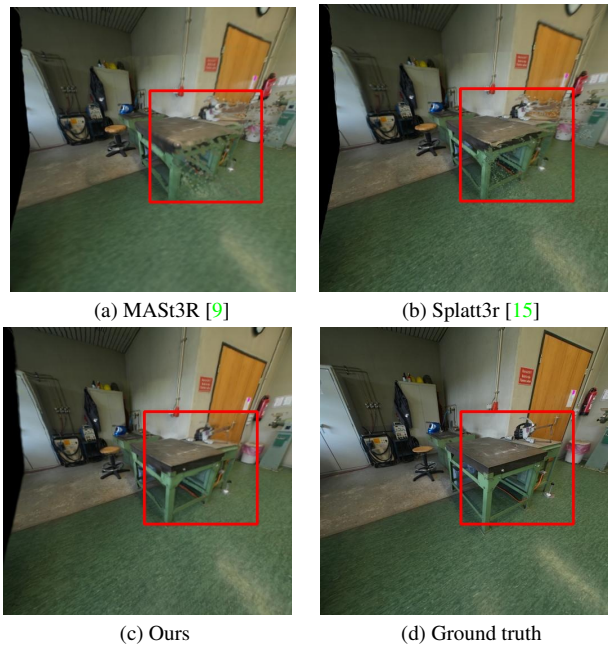
Somi Jeong*
NAVER LABS
South Korea
somi.jeong@naverlabs.com

Abstract

High-fidelity 3D reconstruction from sparse-view images without prior camera poses remains a significant challenge in computer vision. Recently, there has been significant interest in methods for reconstructing 3D representations from unposed sparse views, with active research leveraging approaches such as DUSt3R and MASt3R to successfully estimate relative poses and generate pixel-aligned 3D Gaussians from sparse views. However, these methods often suffer from relatively poor camera pose accuracy and misguidance of points, leading to artifacts that degrade the overall reconstruction accuracy of the scene. In this paper, we propose a point transformer-based 3DGS refinement module to address the challenges of uncontrolled misguided Gaussians. Our model leverages the geometrical inductive bias from a pre-trained point transformer to refine 3D Gaussians by injecting the differences in Gaussian features based on geometric distributions that the standard shallow Gaussian estimation header fails to capture. Our empirical investigations reveal that the proposed method refines 3D Gaussians by incorporating geometrical information.

1. Introduction

Restoring realistic 3D scenes from sparse-view images has been a long-standing and challenging problem in 3D computer vision due to the difficulty of reconstructing detailed geometry and appearance. With the advancements in Neural Radiance Field (NeRF) [12] and 3D Gaussian Splatting (3DGS) [7], 3D reconstruction and novel view synthesis (NVS) have been attracted attention for their ability to achieve real-time and high-fidelity results. However, these methods heavily depend on dense image inputs and require pre-computed pose obtained from pipelines such as Structure-from-Motion (SfM) [14]. Under sparse view conditions (*i.e.* 2–10 input images), SfM often suffers from degraded geometry quality. Moreover, NeRF and 3DGS basically requires iterative per-scene optimization, which is time-consuming and impractical for real-world usage.



(a) MASt3R [9] (b) Splatt3r [15] (c) Ours (d) Ground truth

Figure 1. Qualitative comparison of novel view synthesis (NVS) results of MASt3R [9], Splatt3r [15], our proposed method, and ground truth (GT). Our approach effectively suppresses floating artifacts and improves reconstruction quality.

To address these problems, recent methods [1, 3, 4, 6, 10, 15, 18] introduce *feed-forward and generalizable* 3D reconstruction approach from *sparse views* in a learning-based manner. These approaches eliminate the dependency on SfM [14] pipelines and pre-calculated camera poses, enabling their use in unconstrained, real-world settings. However, these methods have yet to attempt leveraging the geometric prior of the point transformer [17] to generate 3D Gaussians that account for the point distribution.

In this paper, we introduce a framework that aligns geometric prior from a pre-trained point transformer in generalizable sparse-view scene reconstruction through *coarse-*

*Corresponding author.

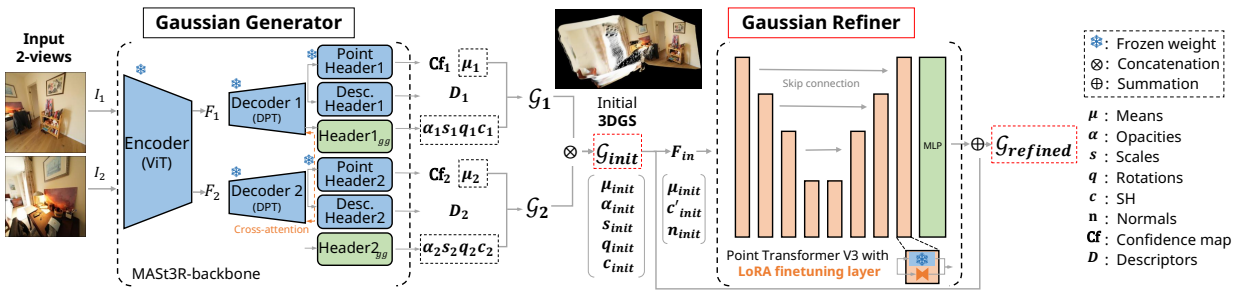


Figure 2. Overall pipeline for Gaussian generator and Gaussian Refiner. The Gaussian generator takes two paired unposed views as input and initially generates 3D Gaussians. Then, our point transformer [17]-based Gaussian refiner with geometric prior (described in § 2.3) cleanse the initial Gaussian to geometrically aligned Gaussians. Pre-trained MASt3R backbone [9] are frozen and pre-trained point transforemr backbone is fine-tuned via additional LoRA [5] layers.

to-refine pipeline. By comparing NVS results Splatt3r [15] and MASt3R [9] in Fig. 1, we demonstrate the validity of our coarse-to-refine design.

In summary, our contributions are as follows:

+ A novel coarse-to-fine model architecture for refining the output of a generalizable feed-forward 3DGS model,
+ Comprehensive experiments to effectively inject geometric priors and exploration for optimal training methods,
+ Competitive experimental evaluation of quantitative and qualitative novel view synthesis.

This work takes one more step toward feed-forward generalizable sparse view scene generation, providing a feasible route for realistic 3D reconstruction and rendering in unconstrained environments.

2. Method

To properly reconstruct a 3D scene from sparse unposed images, we choose to benefit coarse-to-refine pipeline described in Fig 2 using 3DGS [11], and unconstrained stereotype reconstruction methods [9, 16]. We first describe these technologies preliminarily to lay the groundwork.

2.1. Preliminaries: 3DGS

3D Gaussian Splatting [11] is a recent work in the limelight that models a 3D scene as a collection of anisotropic 3D Gaussian kernels distributed in 3D space. Unlike voxel or mesh-based representations, 3DGS relatively allows for continuous modeling of geometry and appearance, making it particularly well-suited for sparse-view scenarios. In our work, we define each Gaussian feature by a mean $\mu \in \mathbb{R}^3$, a scale factor $\mathbf{s} \in \mathbb{R}^3$, a quaternion rotation $\mathbf{q} \in \mathbb{R}^4$, an opacity values $\alpha \in \mathbb{R}$, and color values $\mathbf{c} \in \mathbb{R}^{3\times d}$ via spherical harmonics (SH) [20], where d denotes the degree of SH. Following commonly used notation, we also represent the covariance matrix $\Sigma = RSS^TR^T$ of a Gaussian as a combination of rotation and scale components, while storing scale matrices and rotation matrices as scale vector $\mathbf{s}$ and rotation quaternion $\mathbf{q}$. 3DGS achieves efficient rendering compared to NeRF [21] while maintaining decent rendering quality, thus providing an effective option to tackle the difficulties of restoring intricate scenes from sparse input views.

2.2. Preliminaries: Gaussian Generator

To obtain 3D point clouds with camera parameters, we employ the overall pipeline from Splatt3r [15], which stems from the unconstrained stereo 3D reconstruction methods DUSt3R [16] and MASt3R [9]. The unconstrained stereo 3D reconstruction methods have shown remarkable precision in reconstructing dense 3D reconstruction and performing pixel-wise matching simultaneously.

Given a pair of unposed images $\mathbf{I}_1, \mathbf{I}_2 \in \mathbb{R}^{H\times W\times 3}$, where H is the height and W is the width of the input images, it first encodes the images using a ViT-based encoder [2]. Each header predicts pairwise local dense point maps $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^{3\times H\times W}$ with their corresponding confidence maps $\mathbf{Cf}_1, \mathbf{Cf}_2 \in \mathbb{R}^{H\times W}$. Regressing the two pointmaps in the same camera coordinate system effectively estimates the geometry and cameras parameters. Compared to its baseline method DUSt3R [16], MASt3R [9] outputs two additional dense feature maps $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{R}^{d\times H\times W}$ with a matching header to improve the pixel matching capabilities and enhance pointmap regression. Considering the center μ_1, μ_2 of the Gaussian features as the pointmaps $\mathbf{X}_1, \mathbf{X}_2$, it aims to estimate the remaining features. The model takes decoder output $\mathcal{F}_1, \mathcal{F}_2$ as inputs and estimates the Gaussian features: an opacity value α, a scale factor $\mathbf{s}$, a quaternion rotation $\mathbf{q}$, and color values $\mathbf{c}$.

$$\begin{aligned} \mathcal{F}_1 = \text{Decoder}_1(\mathbf{F}_1), \quad \mathcal{F}_2 = \text{Decoder}_2(\mathbf{F}_2), \\ \alpha_1, \mathbf{s}_1, \mathbf{q}_1, \mathbf{c}_1 = \text{Header}^1_{gg}(\mathcal{F}_1), \\ \alpha_2, \mathbf{s}_2, \mathbf{q}_2, \mathbf{c}_2 = \text{Header}^2_{gg}(\mathcal{F}_2). \end{aligned} \tag{1}$$

Finally, the Gaussian features $\mathcal{G}_1, \mathcal{G}_2$ for each view are constructed and these features are combined to form the initial Gaussians $\mathcal{G}_{init}$. Here, $\mathcal{G}_1 := \{\mu_1, \alpha_1, \mathbf{s}_1, \mathbf{q}_1, \mathbf{c}_1\}, \mathcal{G}_2 := \{\mu_2, \alpha_2, \mathbf{s}_2, \mathbf{q}_2, \mathbf{c}_2\}$ are expressed in the same coordinate frame of $\mathbf{I}_1$, thus yielding the Gaussian features of entire

scene by concatenating them: $\mathcal{G}_{init} = \mathcal{G}_1 \oplus \mathcal{G}_2$.

We also adopt photometric loss with masking M from Splatt3r [15]. The objective function comprises a weighted sum of mean square error (MSE) and LPIPS [22] loss. MSE loss ensures photometric consistency across target views and rendered images from Gaussians $\mathcal{G}_{init}$ and LPIPS loss improves visual quality. Additionally, to ensure effective supervision, masking M includes valid pixels (non-negative, non-zero), pixels within the target frustum, and those where the rendered depth matches the ground truth. For instance, pixels not visible in any context image are excluded by unprojecting points from the target image and reprojecting them onto the context images.

$$\begin{aligned} L_{photo} = \lambda_{\text{MSE}} L_{\text{MSE}}(M * \mathcal{R}(\mathcal{G}_{init}), M * I) \\ + \lambda_{\text{LPIPS}} L_{\text{LPIPS}}(M * \mathcal{R}(\mathcal{G}_{init}), M * I), \end{aligned} \tag{2}$$

where $\mathcal{R}$ is Gaussian rasterizer from [11] and I represents ground truth target images.

2.3. Gaussian Refiner

While the Gaussian generator produces coarse 3D Gaussians from unposed stereo images, it still suffers from geometrically incorrect floating points. These floating points, stemming from MASt3R [9]'s point map alignment on 2D depth maps, often result in spatial discontinuities. To address these issues, we introduce a *Gaussian Refiner* module that facilitates improving the initial Gaussian representations by estimating the residuals for Gaussian features. We design this module based on point transformer V3 (PTv3) [17], a foundational model known for its robustness and efficiency in point cloud data processing. As our backbone model for the refiner, PTv3 is trained on the Scannet++ dataset [19] for point cloud semantic segmentation using cross-entropy loss. Through this process, the PTv3 model learns an inductive bias to identify classes based on the distribution of points, enabling it to effectively encode geometric priors.

To train our Gaussian refiner, firstly, we leverage a Low-Rank Adaptation (LoRA) [5] to finetune the PTv3 [17] for feed-forward Gaussian refinement, adding a LoRA layer to point transformer's MLP and embedding layers. To ensure the model's input remains consistent with the previously trained input, we align the input representation to be identical by transforming the mean value μ from Gaussian features and SH $\mathbf{c}$ into a 6-channel representation. This representation combines normalized color $\mathbf{c}' \in [-1, 1]^3$ and normal vector $\mathbf{n} \in \mathbb{R}^3$. We calculate the normal vector hybrid KNN and radius KD-tree search, as the pointmap from MASt3R [9] is dense enough. The PTv3 backbone model, initialized with pre-trained weights for scene segmentation task on Scannet++ [19] datatset, is frozen during training to retain its geometrical recognition capability. The output features from the backbone are transformed using an MLP header to match the dimensions of the Gaussian features. We note that after several attempts, we decided only to update the Gaussian features for scale $\mathbf{s}$, opacity α, rotation $\mathbf{q}$, and SH $\mathbf{c}$, while keeping the mean value fixed. This decision was made as it yielded positive results. Finally, the refined features $\{\mathbf{s}, \alpha, \mathbf{q}, \mathbf{c}\}$ are added to the original Gaussian features in a residual manner, enhancing the input representation without requiring extensive retraining of the transformer layers.

We use a simple MSE photometric loss without masking in the objective function, as the refining process does not involve abrupt changes.

$$L_{photo} = L_{\text{MSE}}(\mathcal{R}(\mathcal{G}_{init}), I). \tag{3}$$

Even with a Photometric MSE loss, the model's ability to minimize floating artifacts and predict geometrically accurate Gaussian features originates from the greater capabilities of the point transformer with a geometric prior compared to the DPT header [13]. Specifically, the DPT header lacks the inductive bias to sufficiently learn and distinguish geometry-related information, failing to reduce the opacity or scale of Gaussians considered as floating artifacts. In contrast, using a point transformer with pre-trained weights frozen, the model naturally learns to identify and mitigate geometrically incorrect floating artifacts through its inherent geometric inductive bias. Accordingly, Gaussian features are updated to become consistent among similar classes of structures, enhancing the geometric accuracy.

3. Experiment

3.1. Implementation details.

Datasets. Specifically, we trained and tested our model with Scannet++ [19] dataset, which features diverse indoor scenes with ground truth depth. We used the whole paired dataset following the preprocessing from DUSt3R [16]. We used 3461 test image pairs with overlap ratio—0.9/0.9 and 4900 pairs for the others to evaluate our approach.

Training & validation details. Our Gaussian generator model, Splatt3r [15], is retrained on the preprocessed Scannet++ [19] dataset. Specifically, during the retraining of the model from Splatt3r [15], we set λ_{mse} and λ_{lpips} to 1.0 and 0.25, respectively, while applying masking. The normalized depth loss from DUSt3R [16] and MASt3R [9] was excluded. Additionally, to enable the model to update the fixed Gaussian positions μ predicted by the frozen pre-trained MASt3R model, we set the Gaussian header to predict an offset Δ, which is added to μ ($\mu = \mu + \Delta$). The model was trained using Adam optimizer [8] with a learning rate of 0.00001 for 5 epochs. We also set the degree of SH to 1 like [15]. After retraining, we freeze the Gaussian generator part and train the Gaussian refiner part with MSE loss without updating the mean value of Gaussians.

Method	Overlap	Large (0.9 / 0.9)			Medium (0.5 / 0.5)			Small (0.3 / 0.3)		
	Metric	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Mast3r [9]		19.3702	0.7589	0.3169	19.3971	0.7790	0.2668	19.1822	0.7940	0.2515
Splatt3r (PT) [15]		19.4712	0.7493	0.2530	19.7363	**0.7854**	0.2152	19.5451	**0.8062**	**0.2086**
Ours		**20.9749**	**0.7606**	**0.2307**	**20.4672**	0.7778	**0.2098**	**19.9308**	0.7921	0.2127
Splatt3r (RT) [15]		21.6228	0.7843	0.2053	21.5017	0.8138	0.1735	21.2606	0.8346	0.1654

Table 1. Comparison of NVS quality on Scannet++ [19]. We note that PT is pre-trained weights which authors from [15] provide and RT is retrained weights by us. We divide the test set via overlap ratio ($O_{context}$, O_{target}): the first representing the minimum overlap proportion between context images, and the second between the context and target images.

(a) Context views (b) GT target views (c) Ours (d) Splatt3r (RT) [15] (e) Splatt3r (PT) [15] (f) Mast3r [9]

Figure 3. Qualitative comparison of NVS results of MASt3R [9], Splatt3r [15], our proposed method, and GT with overlap ratios of 0.3, 0.5, and 0.9 (top/middle/bottom respectively). Our approach effectively suppresses floating artifacts and improves reconstruction quality.

Baselines. We evaluated Splatt3r [15] using the pre-trained model provided by the authors. The baseline framework extends MASt3R [9] with a simple header for projecting Gaussians for view synthesis, optimized with MSE and LPIPS losses. To render views efficiently, we set a unity covariance with a scale of 0.00001.

Metrics. We calculate PSNR, SSIM, and LPIPS between a target image and rendered images using GT pose following Splatt3r [15]. We demonstrate the results according to the overlap ratio of two values: the first represents the minimum overlap proportion between context images, and the second indicates the overlap threshold between the context and target images.

Discussion. In Fig. 3, the results qualitatively show a reduction in floating artifacts and improved geometric alignment, demonstrating the model's ability to refine Gaussian features. The results in Tab. 1, quantitatively show that our overall pipeline convergently improves the geometrical consistency of the original Splatt3r [15]. Moreover, while the high NVS quality of the retrained Splatt3r stems from Gaussians expanding to fill empty spaces, this often causes Gaussians at the edges, beyond the target view, to grow excessively large, resulting in noticeable blocking artifacts outside the input views. The refiner effectively identifies and mitigates these artifacts as well. Lastly, the low SSIM in low-overlapping views may result from the geometric prior producing plausible yet slightly different structures compared to the ground truth.

4. Conclusion

To conclude, this work presents a new Gaussian refining pipeline for sparse-view 3D scene reconstruction and novel view synthesis, enhancing the baseline Splatt3r framework by injecting geometrical prior with a point transformer-based refinement.

Our study was limited to examining the possibility of refining Gaussians from unposed sparse views and conducting a simple comparison due to the constraints of available GPU resources. However, given that the model using only pre-trained weights provided sufficient guidance, designing a geometric loss and training with additional target viewpoints could yield even better results.

References

[1] Zequn Chen, Jiezhi Yang, and Heng Yang. PreF3R: Pose-free feed-forward 3D gaussian splatting from variable-length image sequence. *arXiv preprint arXiv:2411.16877*, 2024. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2

[3] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large Spatial Model: End-to-end unposed images to semantic 3d. *arXiv preprint arXiv:2410.18956*, 2024. 1

[4] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. PF3plat: Pose-free feed-forward 3D gaussian splatting. *arXiv preprint arXiv:2410.22128*, 2024. 1

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 3

[6] Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangpil Kim, Eunbyung Park, et al. SelfSplat: Pose-free and 3D prior-free generalizable 3D gaussian splatting. *arXiv preprint arXiv:2411.17190*, 2024. 1

[7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[8] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[9] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2025. 1, 2, 3, 4

[10] Hao Li, Yuanyuan Gao, Dingwen Zhang, Chenming Wu, Yalun Dai, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. GGRt: Towards generalizable 3D gaussians without pose priors in real-time. *arXiv preprint arXiv:2403.10147*, 2024. 1

[11] Xi Liu, Chaoyi Zhou, and Siyu Huang. 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. *arXiv preprint arXiv:2410.16266*, 2024. 2, 3

[12] 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. 1

[13] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 3

[14] 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. 1

[15] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 1, 2, 3, 4

[16] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 2, 3

[17] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer V3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 1, 2, 3

[18] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3D gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024. 1

[19] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 3, 4

[20] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2(3):6, 2021. 2

[21] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. Pixelnerf: Neural radiance fields from one or few images. 2021 IEEE. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2020. 2

[22] 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. 3