Geometry Constrained Gaussian Splatting

Alexander Swerdlow Ayush Jain Neham Jain
Carnegie Mellon University
{aswerdlo,ayushj2,nhjain}@andrew.cmu.edu

Abstract

Recent state-of-the-art novel view synthesis methods represent a 3D scene as a set of 3D Gaussians which can be rendered to produce high-quality images. However, these methods typically assume access to numerous, densely captured, RGB images of a scene and often find degenerate solutions when supplied with only a few images. To mitigate this issue, we introduce two geometric constraints to regularize the optimization process. First, we propose to explicitly supervise the underlying geometry with depth maps obtained from state-of-the-art monocular depth estimation models. Secondly, we incorporate an epipolar constraint between images generated from distinct, randomly chosen poses. This constraint ensures the proximity of 3D coordinates for a keypoint in a specific view and its corresponding keypoint along the epipolar line in another view. Extensive experiments on popular LLFF dataset show that these constraints help improve the perceptual quality of few-shot view synthesis with Gaussian Splatting. We provide additional visualizations and video results at https://gcgsplatting.github.io/.

1. Introduction

In recent years, significant advancements have been made in learned implicit representations for 3D. More recently, 3D Gaussians [8] have been popularized as an explicit scene representation that has useful properties in both efficient differentiable rendering and interpretability, achieving state-of-the-art performance on novel view synthesis. In the novel view synthesis task, a model is given multiple posed RGB images of a scene as input and is evaluated on the quality of the image rendered from a novel camera location for the scene. Gaussian Splatting encodes a scene as a set of 3D gaussians and generates a novel view using volumetric rendering. Typically, the gaussian splatting model optimizes each scene individually and requires many calibrated training views and a considerable amount of training time.

During optimization, the internal representation—the set of gaussians—is typically only supervised by photometric loss on the rendered image. This loss ensures that the underlying representation can render the correct image from all provided views, thus incentivizing the representation to approximate the true underlying 3D geometry. Empirically, this loss is often sufficient to generate realistic novel views. However, using a photometric loss as the sole guidance to learn a 3D representation does not always take full advantage of our priors on scene structure and the image formation process. Consequently, when only few input images are available during optimization, these models typically end up finding degenerate solutions, resulting in poor quality rendered images from images outside the training distribution.

Similar to Gaussian splatting, neural radiance fields also face similar issues in few-shot settings. DietNeRF [7] tackles this problem by introducing a consistency loss across various views for a particular scene for which ground truth information is not available. The key idea here is to leverage the fact that semantic content of a scene across different views stays the same (even for novel views). During training, DietNeRF extracts the semantic content of a novel view using CLIP [16] and a consistency loss is applied in this feature space between the reconstructions of novel views and known views. Another stream of work [4, 18] renders depth maps from the neural fields and supervises them with ground-truth depth or depth estimated by monocular depth estimation methods. While these methods have been significantly impactful in few-shot settings with NeRF models, they have not been extensively explored in the context of gaussian splatting.

In this work, we aim to improve performance in few-shot settings with Gaussian splatting models with the help of geometry-based constraints. Similar to DS-NeRF [4] and URF [18], we render depth maps from the Gaussian splatting model in given views and supervise them from depth obtained from state-of-the-art monocular depth estimation models [17]. Since monocular depth estimation models have scale ambiguity, we introduce a relaxed relative loss which quantifies the distribution difference between the depth maps.

However, this depth constraint focuses solely on given camera poses. Drawing inspiration from DietNeRF and aiming to constrain renderings from novel camera poses as well

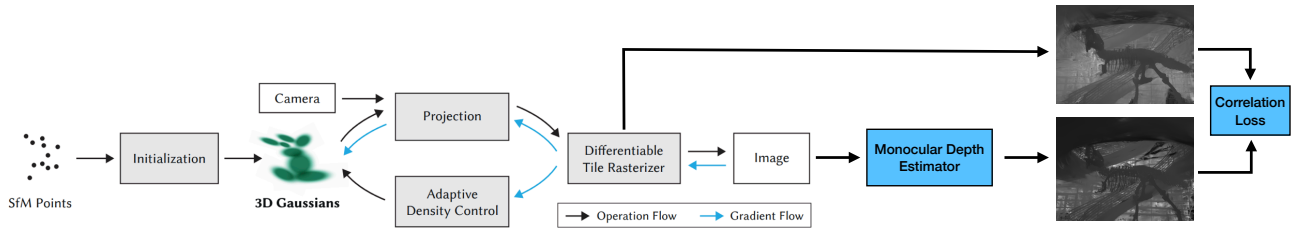

Figure 1. **Pipeline used for enforcing depth constraint**. Figure extended from [8]. In addition to vanilla perceptual loss, we additionally render a depth map from the Gaussian splatting model and supervise it with depth from a monocular depth estimation model with correlation loss.

during training, we propose an epipolar constraint between the images rendered from the novel and the given camera poses. In each training iteration, we randomly sample a novel camera, generate an RGB image, and identify matching points between the new and existing renderings. For any given point in the initial image, its corresponding point in the secondary image must align with the epipolar line originating from the first point in the secondary image. Additionally, these corresponding points should exhibit semantic similarity within the feature space. We then enforce a constraint to ensure the close proximity of their actual 3D locations in space.

We test our approach in the novel view synthesis benchmark of LLFF dataset [11] in a few-shot setting and observe significant quantitative and qualitative improvement with the help of our proposed constraints. In summary, our contributions are:

- We introduce an additional depth loss to leverage depth maps from monocular models which makes it particularly useful for sparse view reconstruction where ambiguities are frequent. This integrates prior knowledge about scene structure, enhancing the accuracy of the learned 3D representation.
- Using the camera poses, we construct epipolar lines in novel views. We enforce 3D consistency between corresponding key points in known and novel views, extending the depth constraint beyond the training set. This provides additional self-supervision, further improving the quality of novel view synthesis.

2. Related Work

Implicit Neural Representations: Using neural networks to perform inverse rendering has a long history, but progress in this area was significantly accelerated by NeRF [12]. Given a dense-set of input images, this work used lightweight MLPs to approximate view-dependent color and positional density of a scene and thus used this function to later synthesize novel views. Later works have applied this technique to various tasks, such as reconstructing 3D meshes, estimating lighting and material properties, and optimizing camera parameters [9, 25, 27]. Some of the challenges and limitations of method include its high computational cost, its sensitivity to noise and occlusion, and its difficulty in handling dynamic scenes and complex geometry. To address this, many followup works adopted a similar forward-rendering scheme but replaced the MLP-representation with more explicit alternatives such as voxel-grids or tri-plane representations [2, 23].

Gaussian Splatting: More recently, 3D Gaussian Splatting [8] replaced this entire pipeline with a set of 3D Gaussians that are optimized to represent parts of a scene and are efficiently splatted—projected—onto the image plane. This work uses the same general concept of learning a view-dependent color and positional density but has an entirely different underlying representation and rendering mechanism using a tile-based rasterizer. This approach supports real-time rendering and faster training compared to most NeRF approaches but maintains high visual quality and editability.

Few-Shot Radiance Fields: A lot of recent works try to address the heavy data requirements of NeRFs. These can be grouped into four clusters: 1) Latent code conditioning: These approaches [19, 24] first learn an image encoder and radiance field decoder at train time. At test time, they use the learned encoder conditioned on scene encoding from the available few shot viewpoints. 2) Meta-Learning: These approaches [5, 20] typically perform additional gradients updates at test time to the given few-shot instances and have been shown to improve the performance substantially. 3) Dense supervision via auxiliary tasks: The key idea here is to add additional decoding heads to the NeRF model: instead of only decoding the density and color, they additionally supervise the model to explicitly decode depth [1, 4, 22]. 4) Providing indirect supervision to intermediate views: DietNeRF [7] supervises intermediate pixels through semantic consistency using a pre-trained CLIP model [16] and shows promising results. In this work, we explore the ideas of adding denser supervision with depth constraint and supervising intermediate views with epipolar constraint.

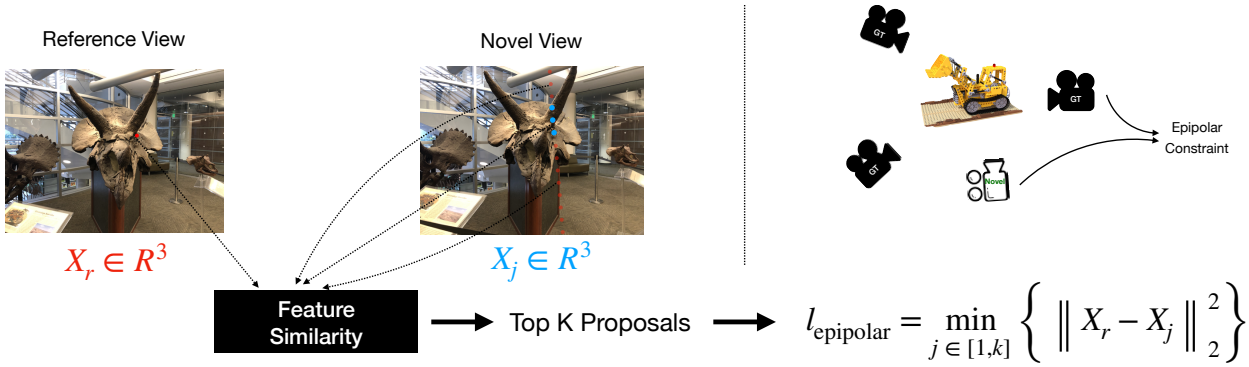

Figure 2. **Pipeline for enforcing epipolar constraint** For a point in reference image, we find top-k matching points in the novel view by enforcing that matches lie on the epipolar line and that they are close to the reference point in the feature space. We then obtain the 3D coordinates of the matches, and supervise the reference 3D point to be close to at least one of the 3D matches in novel view.

3. Method

3.1. Gaussian Splatting

3D Gaussian splatting [8] centers around optimizing a set of n anisotropic Gaussians, each of which have a center $\mu \in \mathbb{R}^3$, covariance $\Sigma \in \mathbb{R}^{3\times3}$, color $c \in \mathbb{R}^3$, and opacity $\alpha \in \mathbb{R}^1$. Each Gaussian influences a point x in 3D space following the 3D Gaussian distribution:

$$G(\boldsymbol{x}) = \frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \quad (1)$$

To render an image, we compute the color and density of the 3D Gaussians using rasterization. We first project the 3D Gaussians to 2D using camera projection. Point-based approaches exploit a similar equation to NeRF-style volume rendering, rasterizing a pixel color with ordered points that cover that pixel.

$$C = \sum_{i\in\mathcal{N}} c_i\sigma_i \prod_{j=1}^{i-1}(1-\sigma_j), \quad \sigma_i = \alpha_i G(x_i) \quad (2)$$

$$D = \sum_{i\in\mathcal{N}} d_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \quad (3)$$

α_i is the learned opacity of each gaussian point, c_i is obtained by a set of learned spherical harmonic coefficients per-gaussian and d_i is obtained by projecting the center μ_i onto the camera plane: $d_i = (R_i\mu_i + T_i)$. Here C and D denote the color and depth of each pixel respectively.

3.2. Depth Supervision

When only a few images are available for supervision, the underlying geometry of the Gaussian splatting model can be underconstrained, resulting in poor novel view synthesis. As an additional supervision for the underlying geometry, we render depth maps from the Gaussian splatting model ($\boldsymbol{D}_{\text{ren}}$) and supervise them with depths obtained from a monocular depth estimation model ($\boldsymbol{D}_{\text{est}}$). This pipeline is shown in Fig. 1.

An immediate issue with depth supervision with monocular depth estimation models is the inherent scale ambiguity between the depth estimates produced by it and the scene scale learned by the Gaussian splatting model. To address this, instead of directly supervising the depth magnitudes, we compute the Pearson correlation between the estimated and rendered depth maps and supervise the correlation to be high. This loss metric is designed to quantify the distribution difference between these depth maps and is represented by the following function:

$$\text{Corr}\left(\hat{\boldsymbol{D}}_{\text{ren}}, \hat{\boldsymbol{D}}_{\text{est}}\right) = \frac{\text{Cov}\left(\hat{\boldsymbol{D}}_{\text{ras}}, \hat{\boldsymbol{D}}_{\text{est}}\right)}{\sqrt{\text{Var}\left(\hat{\boldsymbol{D}}_{\text{ren}}\right)\text{Var}\left(\hat{\boldsymbol{D}}_{\text{est}}\right)}} \quad (4)$$

This softened constraint enables the alignment of depth structures without being impeded by discrepancies in absolute depth values.

3.2.1 Differentiable Depth Rasterization

To facilitate backpropagation from depth information to guide Gaussian training, we use a differentiable depth rasterizer [6]. This extension of the original rasterizer allows us to receive the error signal by comparing the rendered depth $\boldsymbol{D}_{\text{ren}}$ with the estimated depth $\boldsymbol{D}_{\text{est}}$. Specifically, our approach leverages alpha-blending rendering in the 3D-GS framework for depth rasterization. In this technique, the z-buffer from the ordered Gaussians contributing to a pixel is accumulated to produce the final depth value, denoted as d:

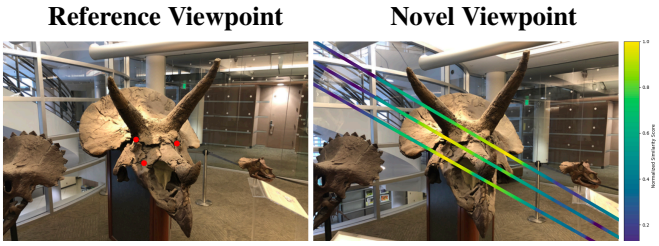

Figure 3. **Epipolar Feature Correspondence using DINOv2.**

$$d = \sum_{i=1}^{n} d_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{5}$$

Here d_i represents the z-buffer of the i^{th} Gaussians and α is identical to that in Eq. 2. The fully differentiable implementation enables the depth correlation loss, further improving the similarity between the rendered and estimated depths.

3.3. Epipolar Supervision

Given a reference input image I_r, we render a novel view I_n from a randomly sampled camera that is not part of the input set. Each of these views has known intrinsics and poses, $(K_r, R_r, t_r), (K_n, R_n, t_n)$. As shown in Fig. 2, our goal is to find matching points between the two images and constrain them to be close in the 3D space. Next, we describe the procedure to obtain the matches.

We first run the SIFT detector [10] on the reference image, I_r to obtain a set of p keypoints. We aim to find the corresponding matching points of these keypoints in the novel image. From the epipolar geometry, we know that the corresponding point must lie on the epipolar line of the keypoint in the novel image. Thus, given a keypoint x_r in the reference image I_r, we compute the epipolar line in the novel image, I_n:

$$l_n = F x_r \tag{6}$$

$$F = K_n^{-\top} [t_r]_\times R_r K_r^{-1} \tag{7}$$

We use feature similarity to narrow down the location of the corresponding point on the epipolar line. Specifically, we pass each image through an image encoder to obtain $E \in \mathbb{R}^{h \times w \times d}$ where h, w are the spatial dimensions of the feature map and d is the feature dimension. We found DINOv2 [14] to work particularly well. We take m points along l_n that are inside the novel image. Given our keypoint, x_r and these points, $x^i_{\text{query}} \in \mathbb{R}^2$ where $i \in [1, m]$, we perform bilinear sampling on E, obtaining $f_r, f^i_{\text{query}} \in \mathbb{R}^d$. We compute the cosine similarity between f_r and f^i_{query} where $i \in [1, m]$. We then have a set of similarity scores, $\{S \in \mathbb{R}^m : -1 \leq S_i \leq 1\}$, and take the top-$k$ values.

Algorithm 1: Epipolar Feature Correspondence Algorithm

Data: Input image I_r, Novel view I_n, Intrinsics and poses $(K_r, R_r, t_r), (K_n, R_n, t_n)$
Result: A set of filtered keypoints in the novel view with their 3D coordinates

Initialize $L^{\text{init}}_{\text{epipolar}} = \{\}$, S=$\{\}$;
Run SIFT detector on I_r to get p keypoints;
for *Each of the p keypoints,* x_r **do**
  Find epipolar line l_n in I_n using $l_n = F x_r$ and $F = K_n^{-\top} [t_r]_\times R_r K_r^{-1}$;
  Pass I_r and I_n through image encoder to obtain feature maps E;
  Select m points along l_n within the image, x^i_{query};
  Perform bilinear sampling on E for x_r and x^i_{query} to get features f_r, f^i_{query};
  Compute cosine similarity between f_r and f^i_{query};
  Obtain similarity scores and select top k values;
  Add maximum similarity score to S ;
  Filter k points in I_n to get x^j_{filtered};
  Unproject x_r and x^j_{filtered} to 3D coordinates X_r, X_j;
  Compute $l_{\text{epipolar}} = \min_{j \in [1,k]} \{\|X_r - X_j\|_2^2\}$;
  Add l_{epipolar} to L_{epipolar};
end
Compute σ of S;
Set $L_{\text{epipolar}} = \bar{L}^{\text{init}}_{\text{epipolar}}$

This gives top-k candidates for point matching. We find k to be an important hyperparameter and ablate this in Sec. 4.3. We also show the similarity score of the features along the epipolar line in Fig. 3.

Given that the novel view has a different perspective, a point visible in one image may be occluded in another image. Thus, a descriptor in the reference image may not have any valid match in the novel image. To deal with this, we drop the descriptors and their epipolar matches if the highest top-k similarity value does not lie in one standard deviation in the distribution of the highest similarities. Thus, by filtering out the less confident candidates, we can effectively ignore these points. Furthermore, we drop the descriptors near the image boundaries, as they are more likely to lie outside the novel camera's field of view.

Finally, given a filtered set of k points in the novel image $x^j_{\text{filtered}} \in \mathbb{R}^2$ where $j \in [1, k]$ and our corresponding keypoint, x_r, we obtain their 3D coordinates, X_i where $j \in [1, k]$, along with X_r, the reference keypoint. We want to supervise X_i to be near at least one of the top-k matches

X_i in the novel image. Thus, given these 3D coordinates, we find the closest unprojected point in the novel view to the reference point:

$$l_{\text{epipolar}} = \min_{j \in [1,k]} \left\{ \|X_r - X_j\|_2^2 \right\} \tag{8}$$

Finally, we normalize the 3D points to lie between $[-1, 1]$ and supervise the L2 (Euclidean) distance between the closest point and the reference point, constraining the novel view to match the geometry in the given views. We repeat this process for each of the p keypoints, obtaining a set $L_{\text{epipolar}}^{\text{init}}$.

Finally, we average the loss of this set, obtaining L_{epipolar} as an additional loss term. We only add this loss after a set number of warm-up steps to allow the Gaussian splatting to learn a reasonable geometry. An overview of this algorithm is shown in Algorithm 1.

4. Experiments

4.1. Implementation Details

Depth We use the depth rasterizer from [6] which is an extension of the original rasterizer and has support for depth rendering and backward pass on the rendered depth value. We make use of a pretrained monocular depth predictor, DPT [17] on torchhub as our monocular depth estimator.

Epipolar Constraint To obtain novel camera views, we take a random input view and generate a spiral path with a random radius—with a predefined maximum bound—and sample a view along this path. For our image features, we use the output of the last attention layer in a `ViT-B/14` DINOv2 [14] model trained with registers [3]. We reduce the scale of the images and add padding to obtain a 224×224 image, resulting in a feature map dimension of 16×16. Experimentally, we found point-wise sampling to perform better than patch descriptors such as SIFT [10] or HardNet [15]. We only start supervision with L_{epipolar} after 2000 steps.

Training We ran all our experiments on a single Nvidia RTX 4090 GPU, training for 10k steps which took roughly 10 minutes per scene. We used the same learning rate and gaussian densification and pruning hyperparameters as the original Gaussian Splatting model [8]. We used a depth loss weight, $\lambda_{\text{depth}} = 5\text{e}{-2}$ and epipolar weight $\lambda_{\text{epipolar}} = 1\text{e}{-2}$.

Datasets We use the LLFF dataset [11] which consists of eight forward-facing real-world scenes. Following RegNeRF [13], we select every eighth image as the test set. We use 3 views to train all the methods and evaluate them at resolutions of 1008×756. For qualitative comparison of various methods, we create a virtual camera trajectory which follows an elliptical path.

We also collected a multiview RGB dataset in Smith Hall at Carnegie Mellon University and ran colmap to obtain their camera poses.

Figure 4. **Qualitative Results using Depth Regularization (3 shot setting).** We see that the fossil in the top row is better distinguished from the background with depth constraint. We also see a similar improvement on the right side of the tree in the bottom row.

Metrics We evaluate our method on standard benchmarks used in prior novel-view synthesis works. Specifically, we use Structural Similarity Index Measure (SSIM) [21] and Peak Signal to Noise Ratio (PSNR) to compare the rendered images with the ground truth test imgaes with the same viewpoints. We also calculate the Learned Perceptual Image Patch Similarity (LPIPS) [26] and report our findings on validation datasets. Note that higher SSIM and PSNR values are better, whereas a lower LPIPS value is better.

Metric	SSIM ↑	PSNR ↑	LPIPS ↓
Without Depth	0.552	17.43	0.307
+ Depth Constraint	0.566	17.60	0.303
+ Depth & Epipolar Constraint (Ours)	**0.567**	**17.69**	**0.301**

Table 1. **Quantitative Results on NeRF LLFF dataset (3 shot setting)**

4.2. Depth Constraint

We perform qualitative evaluation and compare our method against vanilla Gaussian splatting. Looking at the RGB rendering in Fig. 4, we see that the fossil in the top row is better distinguished from the background with depth constraint. We see a similar improvement on the right side of the tree in the bottom row. Additionally, we separately visualize the rendered depth maps in Fig. 5 which shows an even marked improvement with the conference table and fossil demonstrating far incorrect floating Gaussians. Please see our website for videos of these results. We also qualitatively compare the effect of depth supervision on a custom-collected dataset

Figure 5. **Depth Visualization.** From left to right: Given RGB, Rendered depth with depth constraint, Rendered depth without depth constraint.

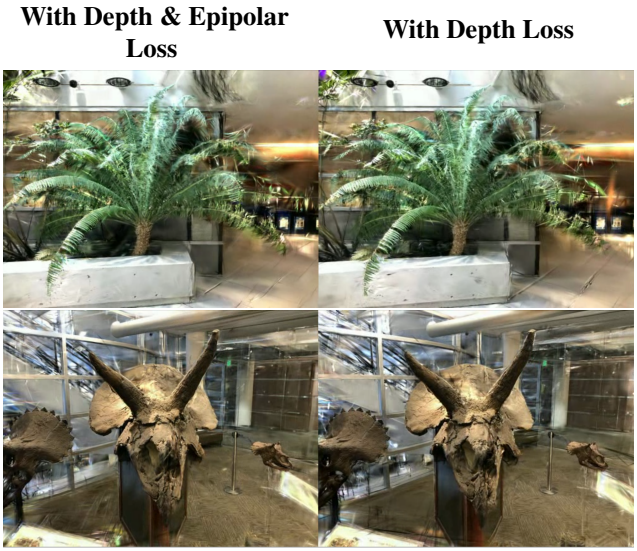

Figure 6. **Qualitative Results using Epipolar Regularization (3 shot setting).**

on our website.

In Tab. 1, we quantitatively compare our method on standard image evaluation metrics such as PSNR, SSIM and LPIPS score. Our method shows some improvements over vanilla Gaussian splatting.

4.3. Epipolar Constraint

We conduct a qualitative assessment comparing our method against our depth-constrained gaussian splatting method. The results of this qualitative assessment can be found in Fig. 6. We find a marginal improvement over the depth-constrained method of gaussian splatting. Additional qualitative results are on our website. We also provide quantitative results for our experiments in Tab. 1. We observe only a very small improvement over the baseline. Additionally, we perform an ablation on k which is the number of the most similar points that we consider for our 3D consistency loss in Eq. (8). The quantitative results of this experiment can be found in Tab. 2. We observe that using a k value of 4 gives the best results and thus we use $k = 4$ in all our experiments.

Topk	SSIM ↑	PSNR ↑	LPIPS ↓
4	**0.777**	**20.50**	**0.217**
8	0.768	19.99	0.220
16	0.775	20.18	0.218
32	0.773	20.31	0.218

Table 2. **Quantitative Results for top-k Experiment on the room scene of the NeRF LLFF dataset**

5. Conclusion

In this project, we explore approaches for introducing geometric constraints for Gaussian splatting. We find that uti-

lizing monocular depth priors and epipolar consistency between the given and novel-rendered views results in higher-quality novel view synthesis. The results are preliminary but promising and point to interesting directions for further investigations. Investigating strategies to standardize geometry across diverse datasets, especially in challenging areas like the sky where depth estimation may pose difficulties, represents a promising direction for future research. Exploring ways to strengthen our epipolar constraint by designing better strategies for matching is another important future direction. Moreover, it would be interesting to investigate other ways to constrain the underlying geometry of these models.

References

[1] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14104–14113, Montreal, QC, Canada, 2021. IEEE. 2

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial Radiance Fields, 2022. 00108 arXiv:2203.09517 [cs]. 2

[3] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers, 2023. arXiv:2309.16588 [cs]. 5

[4] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer Views and Faster Training for Free, 2022. 00398 arXiv:2107.02791 [cs]. 1, 2

[5] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait Neural Radiance Fields from a Single Image, 2021. arXiv:2012.05903 [cs]. 2

[6] Ingra14m. Depth diff gaussian rasterization. https://github.com/ingra14m/depth-diff-gaussian-rasterization/tree/depth, 2023. 3, 5

[7] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis, 2021. 00224 arXiv:2104.00677 [cs]. 1, 2

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering, 2023. 00005 arXiv:2308.04079 [cs]. 1, 2, 3, 5

[9] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-Adjusting Neural Radiance Fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5721–5731, 2021. Conference Name: 2021 IEEE/CVF International Conference on Computer Vision (ICCV) ISBN: 9781665428125 Place: Montreal, QC, Canada Publisher: IEEE. 2

[10] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1150–1157 vol.2, 1999. 4, 5

[11] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines, 2019. arXiv:1905.00889 [cs]. 2, 5

[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, 2020. 01563 arXiv:2003.08934 [cs]. 2

[13] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs, 2021. arXiv:2112.00724 [cs]. 5

[14] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, 2023. arXiv:2304.07193 [cs]. 4, 5

[15] Milan Pultar. Improving the HardNet Descriptor, 2020. arXiv:2007.09699 [cs]. 5

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021. arXiv:2103.00020 [cs]. 1, 2

[17] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 1, 5

[18] Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban Radiance Fields. pages 12932–12942, 2022. 00016. 1

[19] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis, 2021. arXiv:2007.02442 [cs]. 2

[20] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned Initializations for Optimizing Coordinate-Based Neural Representations, 2021. arXiv:2012.02189 [cs]. 2

[21] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. Conference Name: IEEE Transactions on Image Processing. 5

[22] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. NerfingMVS: Guided Optimization of Neural Radiance Fields for Indoor Multi-view Stereo, 2021. 00165 arXiv:2109.01129 [cs]. 2

[23] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields without Neural Networks, 2021. 00062 arXiv:2112.05131 [cs]. 2

[24] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields from One or Few Images, 2021. 00454 arXiv:2012.02190 [cs]. 2

[25] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. NeRF-Editing: Geometry Editing of Neural Radiance Fields, 2022. arXiv:2205.04978 [cs]. 2

[26] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018. 04554. 5

[27] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination. *ACM Transactions on Graphics*, 40(6):1–18, 2021. arXiv:2106.01970 [cs]. 2