Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)

VGOS: Voxel Grid Optimization for View Synthesis from Sparse Inputs

Jiakai Sun, **Zhanjie Zhang**, **Jiafu Chen**, **Guangyuan Li**, **Boyan Ji**, **Lei Zhao***, **Wei Xing***
Zhejiang University
{csjk, cszzj, chenjiafu, ji_by, cszhl, wxing}@zju.edu.cn, lgy1428275037@163.com

Abstract

Neural Radiance Fields (NeRF) has shown great success in novel view synthesis due to its state-of-the-art quality and flexibility. However, NeRF requires dense input views (tens to hundreds) and a long training time (hours to days) for a single scene to generate high-fidelity images. Although using the voxel grids to represent the radiance field can significantly accelerate the optimization process, we observe that for sparse inputs, the voxel grids are more prone to overfitting to the training views and will have holes and floaters, which leads to artifacts. In this paper, we propose VGOS, an approach for fast (3-5 minutes) radiance field reconstruction from sparse inputs (3-10 views) to address these issues. To improve the performance of voxel-based radiance field in sparse input scenarios, we propose two methods: (a) We introduce an incremental voxel training strategy, which prevents overfitting by suppressing the optimization of peripheral voxels in the early stage of reconstruction. (b) We use several regularization techniques to smooth the voxels, which avoids degenerate solutions. Experiments demonstrate that VGOS achieves state-of-the-art performance for sparse inputs with super-fast convergence. Code will be available at https://github.com/SJoJoK/VGOS.

1 Introduction

In 3D vision, novel view synthesis is a long-standing task that aims to synthesize a target image with an arbitrary target camera pose from given source images and their camera poses. Recently, Neural Radiance Fields (NeRF) [Mildenhall *et al.*, 2020], a learning-based neural implicit representation, have emerged as a powerful tool yielding high-fidelity results on this task. However, NeRF requires tens to hundreds of dense inputs and hours to days of training time to get high-quality results. When considering real-world applications such as autonomous driving, AR/VR, and robotics that lack dense data and require real-time performance, NeRF's limitations of relying on dense input views and lengthy optimization time are even more magnified.

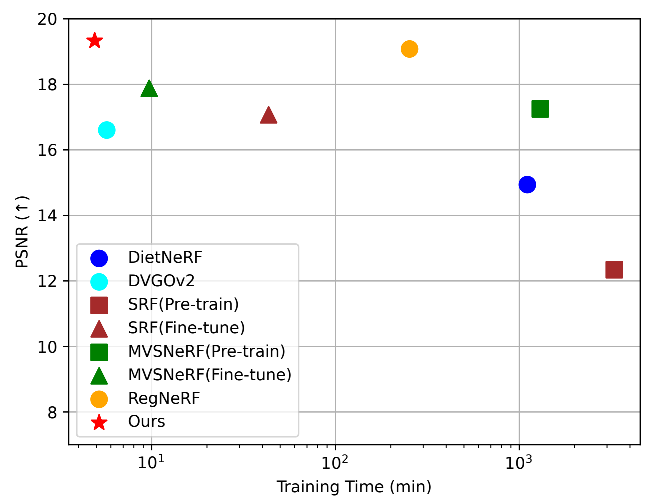

Figure 1: The comparison between VGOS and previous methods on the LLFF dataset in 3-view settings. Note: For a fair comparison, the training time of each method is measured on our machine with a single NVIDIA RTX 3090 GPU using respective official implementations. Our model outperforms previous methods both in reconstruction speed (training time) and quality of results (PSNR) for sparse inputs.

To speed up the optimization, recent works [Sun *et al.*, 2022a; Chen *et al.*, 2022a; Müller *et al.*, 2022; Yu *et al.*, 2022] utilize explicit data structures to represent the radiance field, reducing the training time to minutes. However, these data structures designed to shorten the optimization process of the radiance field do not consider the performance for sparse inputs and still require dense inputs to obtain high-quality results.

To improve NeRF's performance on sparse inputs, several works [Chen *et al.*, 2021; Yu *et al.*, 2021; Liu *et al.*, 2022] first pre-train a model on the multi-view images dataset of many scenes and then use the pre-trained model and the optional per-scene fine-tuning process to synthesize novel views for sparse inputs. Although these works have obtained promising results, acquiring pre-training data may be expensive, and the pre-training time is also very long. In addition, these methods may not generalize well for domains not covered

*Corresponding Authors

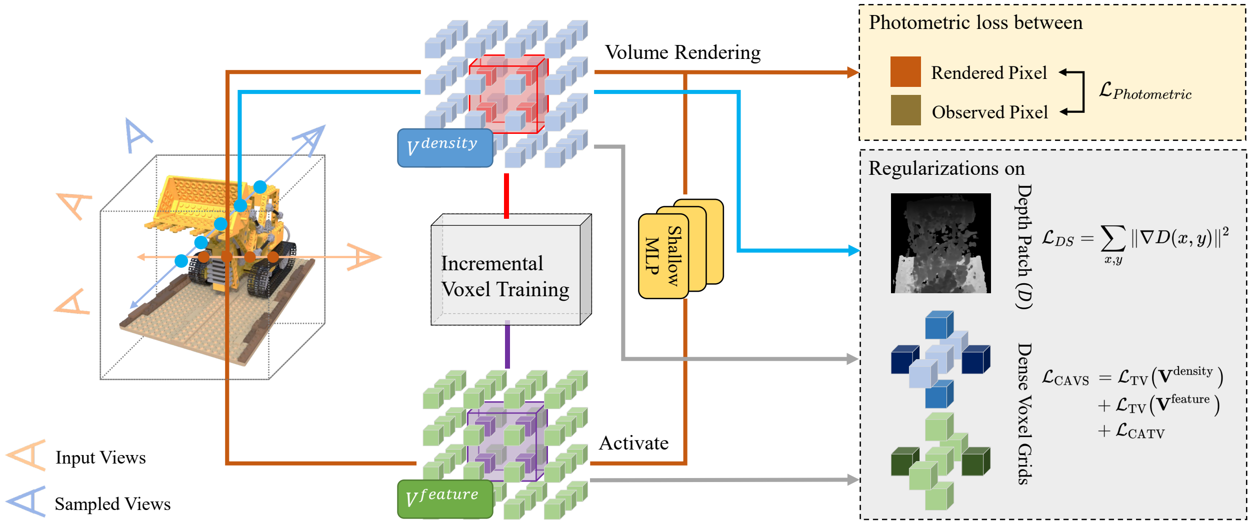

Figure 2: Overview of VGOS architecture. Except for the photometric loss (Equation (4)) from a given set of input images (orange views), the depth smoothness loss (Equation (12)) is imposed on the rendered depth patches (Equation (11)) from sampled views (blue views), and the voxel girds are regularized by the proposed color-aware voxel smoothness loss (Equation (10)). Moreover, An incremental voxel training strategy is utilized to prevent overfitting by expanding the range of optimized voxels (red and purple voxels) incrementally.

by the pre-training data. Other works [Jain *et al.*, 2021; Niemeyer *et al.*, 2022; Deng *et al.*, 2022; Xu *et al.*, 2022] train the model from scratch for every new scene. To enhance the performance for sparse inputs, some works [Jain *et al.*, 2021; Niemeyer *et al.*, 2022] regularize appearance or semantics by introducing models pre-trained on large-scale image datasets. Although these methods can generate high-quality rendering results, their results suffer from incorrect geometry, and the pre-trained model increases the method's complexity. Besides, some works leverage depth maps to supervise the optimization [Deng *et al.*, 2022] or augment training images [Xu *et al.*, 2022]. The addition of depth information helps the model obtain relatively correct geometry from sparse inputs, but depth maps are not as easy to obtain as RGB images.

To overcome the aforementioned shortcomings and limitations, we present an approach for fast radiance field reconstruction from sparse inputs, namely VGOS. As shown in the Fig. 1, our model achieves on-par high-quality results after minutes of training time compared with the previous state-of-the-art approaches, which take hours of per-scene optimization or days of generalizable pre-training. Specifically, we directly optimize the voxel grids representing the radiance field [Sun *et al.*, 2022a]. However, for sparse inputs, the reconstruction of the radiance field (a) is more prone to overfitting to the training views, and (b) the voxel grids will have holes and floaters. In order to solve these two problems, we propose two methods: (a) incremental voxel training strategy and (b) voxel smoothing method. With the improvement of the training strategy and the new regularization method, our model achieves state-of-the-art performance for sparse input without any pre-trained model and with only RGB images as input.

Specifically, the incremental voxel training strategy is to freeze the optimization of peripheral voxels at the early stage of training and gradually thaw the peripheral voxels as the training progresses. This strategy prevents the voxels close to the cameras' near planes from overfitting to the training views, thus boosting the quality of radiance field reconstruction. The voxel smoothing method helps prevent degenerate solutions by regularizing the depth maps rendered from unobserved viewpoints [Niemeyer *et al.*, 2022] and penalizing the sharpness inside the voxel grids with the proposed color-aware voxel smoothness loss.

In summary, the main contributions of our work can be summarized as follows:

- We propose an incremental voxel training strategy to prevent the voxels from overfitting to the training views by suppressing the optimization of peripheral voxels in the early stage of radiance field reconstruction.
- We propose a voxel smoothing method to avoid incorrect geometry by regularizing the dense voxel grids and utilizing depth smooth loss , which eliminates holes and floaters in the voxel grids, thus improving the quality of radiance field reconstruction in sparse input scenarios.
- Extensive experiments on different datasets demonstrate that our proposed model, even without any pre-trained model and extra inputs, achieves one to two orders of magnitude speedup compared to state-of-the-art approaches with on-par novel view synthesis quality.

2 Related Work

2.1 Novel View Synthesis

Novel view synthesis is a time-honored problem at the intersection of computer graphics and computer vision. Previous works use light field [Levoy and Hanrahan, 1996; Shi *et* *al.*, 2014] and lumigraph [Gortler *et al.*, 1996; Buehler *et al.*, 2001] to synthesize novel views by interpolating the input images. Moreover, explicit representations, such as meshes [Debevec *et al.*, 1996; Hu *et al.*, 2021], voxels [Sitzmann *et al.*, 2019a], and multiplane images [Mildenhall *et al.*, 2019; Flynn *et al.*, 2019], are introduced into this task. Recently, several works [Sitzmann *et al.*, 2019b; Niemeyer *et al.*, 2020; Yariv *et al.*, 2020; Mildenhall *et al.*, 2020] have introduced implicit representation and corresponding differentiable rendering methods due to their convenient end-to-end optimization and high-quality results. Among these works, Neural Radiance Fields [Mildenhall *et al.*, 2020] (NeRF) achieve photo-realistic rendering results by representing the radiance field as a multi-layer perceptron (MLP) and differentiable volume rendering method. Subsequent works have improved the performance of NeRF in many aspects, such as training on multi-resolution images [Barron *et al.*, 2021], unconstrained images [Martin-Brualla *et al.*, 2021; Chen *et al.*, 2022b], unbounded scenes [Barron *et al.*, 2022], dark scenes [Mildenhall *et al.*, 2022], and deforming scenes [Park *et al.*, 2021a; Park *et al.*, 2021b]. However, NeRF and these variants require dense inputs to generate high-quality results, which is not always available in real-world applications.

2.2 Fast Radiance Field Reconstruction

Although NeRF can achieve high-fidelity rendering, it takes hours to days of training to reconstruct the radiance field for new scenes. Several works [Sun *et al.*, 2022a; Yu *et al.*, 2022; Chen *et al.*, 2022a; Müller *et al.*, 2022] use explicit or hybrid radiance field representations to reduce training time to a few minutes. DVGO [Sun *et al.*, 2022a] uses dense voxel grids and a shallow MLP to represent the radiance field, while DVGOv2 [Sun *et al.*, 2022b] re-implements some operations in CUDA to achieve improved performance. Plenoxels [Yu *et al.*, 2022] uses a sparse voxel grid and coefficients of spherical harmonic for view-dependent colors to realize a fully explicit representation. TensoRF [Chen *et al.*, 2022a] achieves efficient radiance field reconstruction by decomposing the volume field and modeling the low-rank components. Instant-NGP [Müller *et al.*, 2022] represents the radiance field as a multiresolution hash table and small neural networks, achieving convincing acceleration using C/C++ and fully-fused CUDA kernels. However, these acceleration methods do not reduce the dependence of radiance field reconstruction on dense inputs, while our approach performs high-quality novel view synthesis from sparse inputs in minutes of optimization.

2.3 Sparse Input Radiance Field Reconstruction

Many methods have been proposed to overcome the NeRF's dependence on dense inputs. Several works [Yu *et al.*, 2021; Chen *et al.*, 2021; Chibane *et al.*, 2021] compensate for information scarcity from sparse inputs by pre-train a conditional model of the radiance field. PixelNeRF [Yu *et al.*, 2021] and SRF [Chibane *et al.*, 2021] train convolutional neural network (CNN) encoders to extract features of the input images. MVSNeRF [Chen *et al.*, 2021] uses a 2D CNN to get 2D image features from the input images and then uses plane sweeping to obtain a cost volume which will be processed by a 3D CNN. These methods get promising results, but pre-training on multi-view image datasets is expensive and time-consuming. Besides, most of these methods require fine-tuning on new scenes, and the performance of these methods will decline when the data domain changes at test time.

On the other hand, a line of works [Jain *et al.*, 2021; Niemeyer *et al.*, 2022; Deng *et al.*, 2022; Xu *et al.*, 2022] use models pre-trained on large-scale image datasets and depth maps to train the radiance field from scratch. Diet-NeRF [Jain *et al.*, 2021] uses prior knowledge about scene semantics learned by pre-trained CLIP ViT [Radford *et al.*, 2021] to constrain a 3D representation. RegNeRf [Niemeyer *et al.*, 2022] uses pre-trained Real-NVP [Dinh *et al.*, 2017] to regularize the colors predicted at unseen viewpoints. DS-NeRF [Deng *et al.*, 2022] takes depth maps as input to supervise the reconstruction of the radiance field. Besides, Sin-NeRF [Xu *et al.*, 2022] uses global structure prior provided by pre-trained DINO-ViT [Caron *et al.*, 2021] and augments data using depth maps.

In addition, InfoNeRF [Kim *et al.*, 2022] is a prior-free model without any extra inputs, which regularizes the reconstruction of the radiance field by minimizing ray entropy and reducing information gain. However, this scheme requires the weights of all sampled points on rays. Therefore, reducing the number of sampling points is difficult, which is commonly used in NeRF acceleration approaches.

In contrast, our approach is $10\times$-$100\times$ faster than state-of-the-art approaches with comparable high-quality results without expensive and time-consuming pre-train process and without additional input or pre-trained model to increase complexity.

3 Method

Our approach, which builds upon DVGOv2 [Sun *et al.*, 2022b] (Sec. 3.1), performs fast radiance field reconstruction from sparse RGB input images without any pre-trained model. We find that unexpected overfitting and holes and floaters of the voxel grids lead to degenerate solutions for sparse inputs. To prevent the radiance field from overfitting to the input views, we introduce an incremental voxel training strategy (Sec. 3.2) that suppresses the optimization of peripheral voxels. Moreover, we smooth the voxels (Sec. 3.3) by regularizing the predicted geometry from sampled views and the shape of the explicit radiance field. We depict an overview of our approach in Fig. 2.

3.1 Background

Neural Radiance Fields

A radiance field is a function that maps a 3D position $\mathbf{x}$ and a viewing direction $\mathbf{d}$ to the corresponding view-dependent emission color $\mathbf{c}$ and volume density σ. NeRF [Mildenhall *et al.*, 2020] uses MLP to parameterize this function:

$$\mathrm{MLP}_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma), \tag{1}$$

where Θ is the learnable MLP parameters. Note that the positional encoding [Tancik *et al.*, 2020] is applied to x and d before the MLP to enable the MLP to represent higher frequency functions.

To synthesize novel views, NeRF uses volume rendering techniques. To be specific, the rendered color $C(\mathbf{r})$ of a target pixel is obtained by integrating colors and densities between near and far bounds t_n and t_f along a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ from the camera center $\mathbf{o}$ through the pixel along direction $\mathbf{d}$:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \tag{2}$$

where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$ is the accumulated transmittance along the ray from t_n to t, and $\sigma(\cdot)$ and $\mathbf{c}(\cdot,\cdot)$ indicate the density and color prediction of the radiance field F_Θ, respectively.

In practice, the integral is approximated by quadrature:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i, \tag{3}$$

where $T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right)$, N is the number of samples points along the ray $\mathbf{r}$; σ_i, $\mathbf{c}_i$ are the density and color of the i^{th} sampled point, and $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples.

NeRF's MLP can be optimized over a set of input images and their camera poses by minimizing the photometric MSE between the ground truth pixel color $C_{GT}(\mathbf{r})$ and the rendered color $\hat{C}(\mathbf{r})$:

$$\mathcal{L}_{\text{Photometric}} = \frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}} \left\| \hat{C}(\mathbf{r}) - C_{GT}(\mathbf{r}) \right\|^2, \tag{4}$$

where $\mathcal{R}$ denotes a set of rays.

Direct Voxel Grid Optimization

It is time-consuming to query the color and density of each sampled point through MLP, so DVGO [Sun *et al.*, 2022a] is proposed to accelerate this process by representing the radiance field as voxel grids. Such an explicit scene representation is efficient to query color $\mathbf{c}$ and density σ for any 3D position $\mathbf{x}$ with trilinear interpolation:

$$\begin{aligned} \ddot{\sigma} &= \text{interp}\left(\mathbf{x}, \mathbf{V}^{\text{density}}\right), \\ \mathbf{c} &= \text{interp}\left(\mathbf{x}, \mathbf{V}^{\text{rgb}}\right), \\ \sigma &= \log(1 + \exp(\ddot{\sigma} + b)), \end{aligned} \tag{5}$$

where the shift $b = \log\left((1 - \alpha_{\text{init}})^{-\frac{1}{s}} - 1\right)$ is the bias term determined by hyperparameter α_{init} and voxel size s, $\mathbf{V}^{\text{density}}$ and $\mathbf{V}^{\text{rgb}}$ are the voxel grids storing raw density $\ddot{\sigma}$ before applying the density activation and color, respectively.

In practice, DVGO uses a coarse-to-fine training strategy. In the fine stage, a shallow MLP is used to process viewing-direction $\mathbf{d}$ and feature f from a feature voxel grid $\mathbf{V}^{\text{feature}}$ to model view-dependent color emission.

$$\mathbf{c} = \text{MLP}_\Theta\left(\text{interp}(\mathbf{x}, \mathbf{V}^{\text{feature}}), \mathbf{x}, \mathbf{d}\right), \tag{6}$$

where Θ is the learnable MLP parameters.

Subsequent work, namely DVGOv2 [Sun *et al.*, 2022b], improve DVGO by re-implementing part of the Pytorch operations with CUDA and extending it to support forward-facing and unbounded inward-facing capturing.

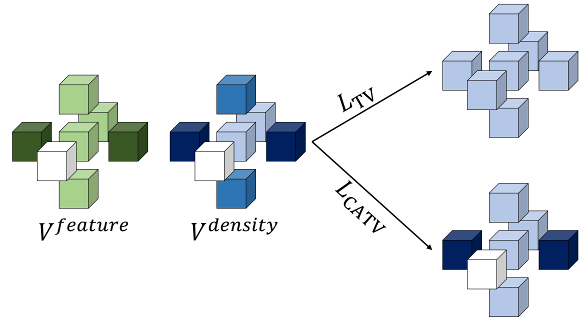

Figure 3: Visual comparisons between $\mathcal{L}_{\text{TV}}$ and $\mathcal{L}_{\text{CATV}}$ regularization. $\mathcal{L}_{\text{TV}}$ doesn't utilize the information of $\mathbf{V}^{\text{feature}}$, while $\mathcal{L}_{\text{CATV}}$ takes advantage of the correlation between $\mathbf{V}^{\text{density}}$ and $\mathbf{V}^{\text{feature}}$.

3.2 Incremental Voxel Training

Although DVGO uses various techniques to avoid degenerate solutions, the radiance field will overfit to input views for sparse scenarios. Specifically, we find that for sparse inputs, the peripheral voxels close to the camera near planes have high density values at the initial stage of training to reproduce the input views. However, the high density value of the outer voxels hinders the optimization of the inner voxels, which makes it difficult for the radiance field to converge to the correct geometry so that the quality of rendering results at novel views will decline.

We propose a simple yet non-trivial incremental voxel training strategy to solve the above-mentioned problem. For the voxel grids $\mathbf{V} \in \mathbb{R}^{C\times N_x\times N_y\times N_z}$ representing the radiance field, where C is the dimension of the modality, $N_x \cdot N_y \cdot N_z$ is the total number of voxels, we define an expanding bounding box $\mathbb{B}$ whose corner points are $P_{min}, P_{max} \in \mathbb{R}^3$:

$$\begin{aligned} P_{min} &= (P_{min\_init} \times (1 - r(i))) \odot (N_x, N_y, N_z), \\ P_{max} &= (P_{max\_init} \times (1 - r(i)) + r(i)) \\ &\quad \odot (N_x, N_y, N_z), \end{aligned} \tag{7}$$

where $P_{min\_init} \in [0,1]^3$ and $P_{max\_init} \in [0,1]^3$ are the initial ratio of the expanding bounding box $\mathbb{B}$, and $r(i) = min(\frac{i}{M}, 1)$ determine the range of the bounding box $\mathbb{B}$, where i is the current training iteration and M is the pre-defined max iteration steps of the increment process. We only optimize the voxels inside the bounding box $\mathbb{B}$; this training strategy freezes the optimization of the peripheral voxels in the early training, avoiding overfitting and leading to better rendering results at novel views. We set $M = 256$ for all scenes, $P_{min\_init} = (0.2, 0.2, 0.2), P_{max\_init} = (0.8, 0.8, 0.8)$ for bounded inward-facing scenes and $P_{min\_init} = (0, 0, 0.995), P_{max\_init} = (1, 1, 1)$ and $M = 256$ for forward-facing scenes in our experiments.

3.3 Voxel Smoothing

Although we use the incremental voxel training strategy to alleviate the overfitting, if we only use the photometric MSE loss (Equation (4)) to supervise the training from sparse inputs, the radiance field will still overfit to the input views.

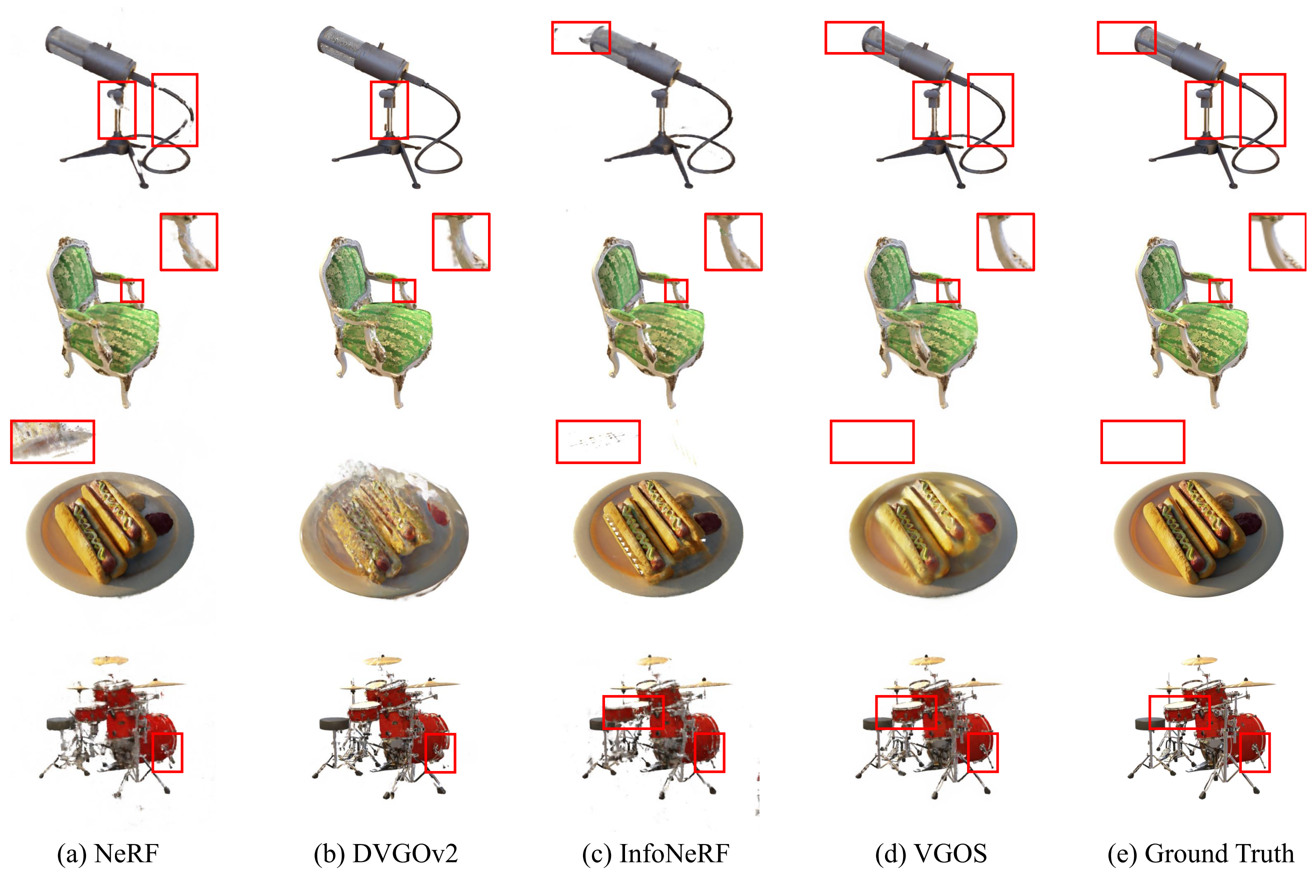

Figure 4: Qualitative comparison on Realistic Synthetic 360° in the 4-view setting. All the experiments are performed with the same inputs. Please zoom in for more details.

To solve this problem, we propose a novel color-aware voxel smoothness loss on the dense voxel grids and utilize the depth smoothness loss on the sampled views to smooth the voxels.

Regularization on Dense Voxels

To prevent the outliers and noises in the explicit model, previous works [Yu *et al.*, 2022; Chen *et al.*, 2022a; Sun *et al.*, 2022b] utilize total variation (TV) loss [Rudin and Osher, 1994]:

$$\mathcal{L}_{\text{TV}}(\mathbf{V}) = \sum_{\mathbf{v}\in\mathbf{V}} \Delta(\mathbf{v}) \,, \tag{8}$$

with $\Delta(\mathbf{v})$ shorthand for the mean of the loss (L1, L2, or Huber Loss) between the value in voxel $\mathbf{v}$ and its six nearest-neighbor voxels and $\mathbf{V}$ indicates the voxel grids storing density, color or feature, which is indeed effective. However, these works calculate the TV loss of density and color separately, not taking advantage of the correlation between density and color in the explicit radiance field.

We observe that, in the radiance field, the density change is not smooth where the color changes sharply. According to the above observation, we propose color-aware total variance (CATV) loss, which uses the activated value in the color voxel grid to guide TV loss of the density voxel grid and is formulated as:

$$\begin{aligned} F_{\text{CA}}(\mathbf{V},\mathbf{v}) &= \Delta_{\text{activate}}(\mathbf{v}), \mathbf{v}\in\mathbf{V} \,, \\ \mathcal{L}_{\text{CATV}} &= \sum_{\mathbf{v}\in\mathbf{V}^{\text{density}}} e^{-F_{\text{CA}}(\mathbf{V}^{\text{feature}},\mathbf{v})}\Delta(\mathbf{v}) \,, \end{aligned} \tag{9}$$

with $\Delta_{\text{activate}}(\mathbf{v})$ indicates that the activated values are used calculating $\Delta(\mathbf{v})$. In practice, we use L1 loss in F_{CA}, and Huber Loss is in $\mathcal{L}_{\text{CATV}}$. Sigmoid is used in F_{CA} to normalize the feature values to $[0, 1]$ and align the choices of activation functions in DVGO. In Fig. 3, we show the differences between using $\mathcal{L}_{\text{TV}}$ and $\mathcal{L}_{\text{CATV}}$ to regularize $\mathbf{V}^{\text{density}}$.

To ensure flexibility, we use both $\mathcal{L}_{\text{TV}}$ and $\mathcal{L}_{\text{CATV}}$. Therefore, the color-aware voxel smoothness (CAVS) loss which is used to regularize the dense voxels is formulated as:

$$\begin{aligned} \mathcal{L}_{\text{CAVS}} = {} & \lambda_{\text{TVF}}\mathcal{L}_{\text{TV}}(\mathbf{V}^{\text{feature}}) + \lambda_{\text{TVD}}\mathcal{L}_{\text{TV}}(\mathbf{V}^{\text{density}}) \\ & + \lambda_{\text{CATV}}\mathcal{L}_{\text{CATV}} \,, \end{aligned} \tag{10}$$

where λ_{TVF}, λ_{TVD} and λ_{CATV} are the corresponding weights. Since computing $\mathcal{L}_{\text{CAVS}}$ is time-consuming, we implement it in CUDA kernel to speedup the process. Besides, we only backpropagate the gradient of $\mathcal{L}_{\text{CATV}}$ to $\mathbf{V}^{\text{density}}$. We set $\lambda_{\text{TVD}} = 5\cdot 10^{-4}$, $\lambda_{\text{TVF}} = \lambda_{\text{CATV}} = 5\cdot 10^{-5}$ in coarse-stage training and $\lambda_{\text{TVD}} = 5\cdot 10^{-5}$, $\lambda_{\text{TVF}} = 10^{-5}$, $\lambda_{\text{CATV}} = 5\cdot 10^{-6}$ in fine-stage training for bounded inward-facing scenes. For forward-facing scenes which only need fine-stage training, we set $\lambda_{\text{TVD}} = 5 \cdot 10^{-5}$,and $\lambda_{\text{TVF}} = \lambda_{\text{CATV}} = 5 \cdot 10^{-6}$.

Regularization on Sampled Viewpoints

The piecewise-smooth of geometry is a classic hypothesis in depth and disparity estimation [Scharstein and Szeliski, 2002]. Hence we utilize the depth smoothness (DS) loss introduced by RegNeRF [Niemeyer *et al.*, 2022] on the unseen views to improve scene geometry.

To get unobserved views, we sample camera pose $p \sim \pi$ where π is the distribution of camera poses if π is available. For bounded inward-facing scenes such as one from the Realistic Synthetic 360° dataset [Mildenhall *et al.*, 2020], π is the uniform distribution over the hemisphere with the known radius. For forward-facing scenes like those from the LLFF dataset [Mildenhall *et al.*, 2019], π is the uniform distribution over a 2D plane with given boundaries. If π is not available, we generate new poses by interpolating between input poses.

We can estimate depth $\hat{d}$ along the ray $\mathbf{r}$ cast from the sampled camera pose the similar way we render color in Equation (3):

$$\hat{d}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i)) \ . \tag{11}$$

By estimating depth from sets of neighboring rays, we can render depth patches and regularize them by the DS loss:

$$\mathcal{L}_{\text{DS}} = \frac{\lambda_{\text{DS}}}{|\mathcal{R}|} \sum_{\mathbf{r}_c \in \mathcal{R}} \sum_{(x,y)} \|\nabla D(x, y)\|^2 \ , \tag{12}$$

where $\mathcal{R}$ indicates a set of rays cast from the sampled poses, D is the depth patch centered at $\mathbf{r}_c$, and λ_{DS} is the loss weight. In practice, finite difference formula is used to compute ∇D. We set $\lambda_{\text{DS}} = 5 \cdot 10^{-4}$ in coarse-stage training and $\lambda_{\text{DS}} = 10^{-5}$ in fine-stage training for bounded inward-facing scenes. For forward-facing scenes, we set $\lambda_{\text{DS}} = 5 \cdot 10^{-4}$.

3.4 Total Loss Function

The total loss function of our model is given by:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Photometric}} + \mathcal{L}_{\text{CAVS}} + \mathcal{L}_{\text{DS}} \ , \tag{13}$$

note that the hyperparameters to balance the regularization terms have been included in Equation (10) and Equation (12). Besides, $\mathcal{L}_{\text{Photometric}}$ uses the rays from the input views, and $\mathcal{L}_{\text{CAVS}}$ utilizes the rays from the sampled views.

4 Experiments

4.1 Datasets and Evaluations

We perform experiments on inward-facing scenes from the Realistic Synthetic 360° dataset [Mildenhall *et al.*, 2020] and forward-facing scenes from the LLFF dataset [Mildenhall *et al.*, 2019].

Realistic Synthetic 360° Dataset. The Realistic Synthetic 360° dataset contains pathtraced images of 8 synthetic scenes with complicated geometry and realistic non-Lambertian materials. Each scene has 400 images rendered from inward-facing virtual cameras with different viewpoints. Following the protocol of InfoNeRF [Kim *et al.*, 2022], we randomly sample 4 views out of 100 training images as sparse inputs and evaluate the model with 200 testing images.

Model	PSNR↑	SSIM↑	LPIPS↓	Training Time↓
NeRF	15.93	0.780	0.320	2 hrs*
DietNeRF	16.06	0.793	0.306	19 hrs
PixelNeRF	16.09	0.738	0.390	3-4 days⋆+10 hrs
DVGOv2	17.19	0.767	<u>0.223</u>	<u>4 mins</u>
InfoNeRF	<u>18.62</u>	<u>0.811</u>	0.230	4 hrs
VGOS(ours)	**18.91**	**0.825**	**0.205**	**3 mins**

Table 1: Quantitative comparison on Realistic Synthetic 360° in the 4-view setting. The asterisk (∗) denotes that early-stopping is used instead of the default setting. The star (⋆) denotes the generalizable pre-training time. Bold and underline indicate the best and the second-best values for each metric.

LLFF Dataset. The LLFF Dataset consists of 8 complex real-world scenes captured by a handheld cellphone. Each scene has 20 to 62 forward-facing images. We hold out 1/8 of the images as test sets following the standard protocol [Mildenhall *et al.*, 2020] and report results for 3 input views randomly sampled from the remaining images.

Metrics. We measure the mean of peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) [Wang *et al.*, 2004], and learned perceptual image patch similarity (LPIPS) [Zhang *et al.*, 2018] to evaluate our model.

4.2 Implementation Details

We implement our model on the top of DVGOv2 codebase using Pytorch [Paszke *et al.*, 2019]. Following DVGO, We use the Adam [Kingma and Ba, 2015] to optimize the voxel grids with the initial learning rate of 0.1 for all voxels and 10^{-3} for the shallow MLP and exponential learning rate decay is applied.

For scenes in the Realistic Synthetic 360° dataset, we train the voxel grids for 5K iterations with a batch size of 2^{13} rays for input views and 2^{14} rays for sampled views in both stages.

For scenes in the LLFF dataset, we train the voxel grids for 9K iterations with a batch size of 2^{12} rays for input views and 2^{14} rays for sampled views in only one stage.

Please refer to the supplementary material for more details.

4.3 Comparisons

Following InfoNeRF [Kim *et al.*, 2022], the presented metrics for comparisons are the average score of five experiments with different viewpoint samples.

Realistic Synthetic 360° Dataset

We compare our model with NeRF [Mildenhall *et al.*, 2020], DietNeRF [Jain *et al.*, 2021], PixelNeRF [Yu *et al.*, 2021], InfoNeRF [Kim *et al.*, 2022], and DVGOv2 [Sun *et al.*, 2022a; Sun *et al.*, 2022b] on the Realistic Synthetic 360° dataset in the 4-view setting. Since PixelNeRF is pre-trained on the DTU dataset, we fine-tuned it for 20K iterations similar to [Deng *et al.*, 2022] for improved performance.

Tab. 1 presents the overall quantitative results, and Fig. 4 shows the qualitative results. As the baseline, NeRF has degenerate solutions for sparse inputs. DietNeRF and PixelNeRF outperform the baseline relatively by introducing pretrained models. Although DVGOv2 aims to accelerate the reconstruction process, it achieves superior results to NeRF in

Model	PSNR↑	SSIM↑	LPIPS↓	Training Time↓
PixelNeRF	16.17	0.438	0.512	3-4 days*+10 hrs
SRF	17.07	0.436	0.529	2-3 days*+43 mins
MVSNeRF	17.88	0.584	**0.327**	1-2 days*+10 mins
Mip-NeRF	14.62	0.351	0.495	14 hrs
DietNeRF	14.94	0.370	0.496	18 hrs
DVGOv2	16.60	0.566	0.422	<u>6 mins</u>
RegNeRF	<u>19.08</u>	<u>0.587</u>	<u>0.336</u>	4 hrs
VGOS(ours)	**19.35**	**0.620**	0.432	**5 mins**

Table 2: Quantitative comparison on LLFF in the 3-view setting. The star ($\star$) denotes the generalizable pre-training time. Bold and underline indicate the best and the second-best values for each metric. Note that the training time of Mip-NeRF can be reduced to 3 hours if early-stopping is applied, but we follow the baseline implement of RegNeRF's official code which trains Mip-NeRF for 250K iters with a batch size of 2^{12} rays.

Variant	*Inc.*	$\mathcal{L}_{\text{DS}}$	$\mathcal{L}_{\text{CAVS}}$	PSNR↑	SSIM↑	LPIPS↓
Baseline	×	×	×	14.47	0.5684	0.5830
w/ Inc.	✓	×	×	15.86	0.6303	0.5281
w/ $\mathcal{L}_{\text{DS}}$	×	✓	×	16.91	0.6756	0.5104
w/ $\mathcal{L}_{\text{CAVS}}$	×	×	✓	<u>20.08</u>	<u>0.7939</u>	0.4258
w/o Inc.	×	✓	✓	18.57	0.7688	0.4404
w/o $\mathcal{L}_{\text{DS}}$	✓	×	✓	17.84	0.7357	0.4730
w/o $\mathcal{L}_{\text{CAVS}}$	✓	✓	×	19.79	0.7663	<u>0.4183</u>
Full Model	✓	✓	✓	**21.82**	**0.8220**	**0.3869**

Table 3: Ablation study on the *room* scene in the 3-view setting.

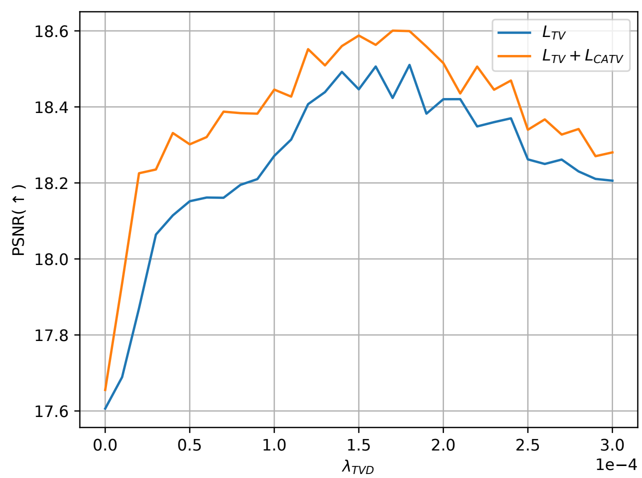

Figure 5: Comparison of $\mathcal{L}_{\text{TV}}$ and $\mathcal{L}_{\text{TV}} + \mathcal{L}_{\text{CATV}}$ on the *flower* scene in the 3-view setting, with λ_{CATV} set to 0 and $5 \cdot 10^{-6}$ separately.

sparse input scenarios, which we observe as another advantage of explicit models. InfoNeRF outperforms the previous methods in terms of all image quality metrics. However, it takes twice the training time than NeRF and is unsuitable for common acceleration approaches since it requires the weights of all sampled points on rays. Our model achieves state-of-the-art performance with an outstanding convergence speed.

LLFF Dataset

We compare our model with Mip-NeRF [Barron *et al.*, 2022], DietNeRF [Jain *et al.*, 2021], PixelNeRF [Yu *et al.*, 2021], SRF [Chibane *et al.*, 2021], MVSNeRF [Chen *et al.*, 2021], RegNeRF [Niemeyer *et al.*, 2022] and DVGOv2 [Sun *et al.*, 2022a; Sun *et al.*, 2022b] on the LLFF Dataset in the 3-view setting. Similar to the experiments on the Realistic Synthetic 360° Dataset, we fine-tune PixelNeRF, SRF, and MVSNeRF on each scene of the LLFF dataset to handle the domain shift issue since these methods are pre-trained on the DTU dataset.

The overall quantitative results are presented in Tab. 2. Besides, we provide the qualitative results in the supplementary material. Our model is superior to previous works in each metric except for LPIPS, which measures human perception. However, pre-trained model extracting high-level information is not used in our approach, which is a trade-off between complexity and performance, leading to relatively higher LPIPS on our model's evaluation results.

4.4 Ablation Study

Effectiveness of Proposed Components

We conduct ablation studies on the *room* scene to evaluate the contributions of each component of our proposed model. Tab. 3 report the results with various combinations of the incremental voxel training strategy (*Inc.*), $\mathcal{L}_{\text{DS}}$ and $\mathcal{L}_{\text{CAVS}}$. Besides, we provide more results and vision comparisons in the supplementary material.

With (PSNR+X) indicating the PSNR improvement compared to the baseline, we can observe the following:

w/ Inc. improve performance (PSNR+1.39), which demonstrates the effectiveness of *Inc.*. Similarly, *w/* L_{CAVS} results in a substantial boost in performance (PSNR+5.61), highlighting the effectiveness of the $\mathcal{L}_{\text{CAVS}}$ regularization term.

The effects of these three components are not strictly orthogonal. For instance, while combining $\mathcal{L}_{\text{DS}}$ and *Inc.* leads to improved performance (*w/o* $\mathcal{L}_{\text{CAVS}}$ vs. *w/ Inc.*), adding $\mathcal{L}_{\text{DS}}$ to a model that only utilizes $\mathcal{L}_{\text{CAVS}}$ results in a decrease in performance (*w/o Inc.* vs. *w/* $\mathcal{L}_{\text{CAVS}}$)

The full model, which combines all three components, achieves the best performance overall. This indicates the effectiveness of our proposed model.

Effectiveness of $\mathcal{L}_{\text{CATV}}$

We conduct ablation studies on the *flower* scene to evaluate the contribution of $\mathcal{L}_{\text{CATV}}$. As illustrated in the Fig. 5, the performance of the model is improved by adding $\mathcal{L}_{\text{CATV}}$, regardless of λ_{TVD}. This demonstrate the effectiveness of the $\mathcal{L}_{\text{CATV}}$ regularization term. Note that the gradient of $\mathcal{L}_{\text{CATV}}$ is only backpropagated to $\mathbf{V}^{\text{density}}$ in our implementation (Equation (9)), so λ_{TVF} remains fixed at $5 \cdot 10^{-6}$.

5 Conclusion

NeRF suffers from a long training time and the requirement of dense inputs. To overcome the above shortages, we propose VGOS, an approach to improve the performance of the voxel-based radiance field from sparse inputs. By directly optimizing voxel grids, the incremental voxel training strategy, and the voxel smoothing method, VGOS is $10\times - 100\times$ faster than previous few-shot view synthesis methods with state-of-the-art render quality while avoiding the degenerate solutions for explicit radiance field methods in sparse input scenarios.

Acknowledgements

This work was supported in part by the projects No. 2020YFC1523202-2, 2021YFF0900604, 19ZDA197, LY21F020005, 2021009, 2019011,2022C01222, Research and Application Demonstration of Key Technologies in Digital Archaeology of Ningbo Prehistoric Sites, Key Technologies and Product Research and Development for Cultural Relics Protection and Trading Circulation, MOE Frontier Science Center for Brain Science & Brain-Machine Integration (Zhejiang University), National Natural Science Foundation of China (62172365).

References

[Barron *et al.*, 2021] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, pages 5855–5864, 2021.

[Barron *et al.*, 2022] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022.

[Buehler *et al.*, 2001] Chris Buehler, Michael Bosse, Leonard McMillan, Steven J. Gortler, and Michael F. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, 2001.

[Caron *et al.*, 2021] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021.

[Chen *et al.*, 2021] 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 *ICCV*, 2021.

[Chen *et al.*, 2022a] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022.

[Chen *et al.*, 2022b] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *CVPR*, pages 12943–12952, 2022.

[Chibane *et al.*, 2021] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *CVPR*, pages 7911–7920, 2021.

[Debevec *et al.*, 1996] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH*, 1996.

[Deng *et al.*, 2022] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, June 2022.

[Dinh *et al.*, 2017] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *ICLR*, 2017.

[Flynn *et al.*, 2019] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. *CVPR*, 2019.

[Gortler *et al.*, 1996] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH*, 1996.

[Hu *et al.*, 2021] Ronghang Hu, Nikhila Ravi, Alexander C. Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *ICCV*, 2021.

[Jain *et al.*, 2021] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, pages 5885–5894, October 2021.

[Kim *et al.*, 2022] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*, pages 12912–12921, 2022.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Levoy and Hanrahan, 1996] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996.

[Liu *et al.*, 2022] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *CVPR*, 2022.

[Martin-Brualla *et al.*, 2021] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

[Mildenhall *et al.*, 2019] 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. *ACM Trans. Graph.*, 38(4):1–14, 2019.

[Mildenhall *et al.*, 2020] 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.

[Mildenhall *et al.*, 2022] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *CVPR*, pages 16190–16199, 2022.

[Müller *et al.*, 2022] 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, July 2022.

[Niemeyer *et al.*, 2020] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, pages 3504–3515, 2020.

[Niemeyer *et al.*, 2022] 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. In *CVPR*, 2022.

[Park *et al.*, 2021a] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *CVPR*, pages 5865–5874, 2021.

[Park *et al.*, 2021b] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), dec 2021.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021.

[Rudin and Osher, 1994] Leonid I Rudin and Stanley Osher. Total variation based image restoration with free local constraints. In *ICIP*, volume 1, pages 31–35. IEEE, 1994.

[Scharstein and Szeliski, 2002] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002.

[Shi *et al.*, 2014] Lixin Shi, Haitham Hassanieh, Abe Davis, Dina Katabi, and Frédo Durand. Light field reconstruction using sparsity in the continuous fourier domain. *ACM Trans. Graph.*, 2014.

[Sitzmann *et al.*, 2019a] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *CVPR*, pages 2437–2446, 2019.

[Sitzmann *et al.*, 2019b] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

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

[Sun *et al.*, 2022b] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved direct voxel grid optimization for radiance fields reconstruction. *arXiv preprint arXiv:2206.05085*, 2022.

[Tancik *et al.*, 2020] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, 2020.

[Wang *et al.*, 2004] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004.

[Xu *et al.*, 2022] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. *arXiv preprint arXiv:2204.00928*, 2022.

[Yariv *et al.*, 2020] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020.

[Yu *et al.*, 2021] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

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

[Zhang *et al.*, 2018] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.