Vanishing Point Aided Hash-Frequency Encoding for Neural Radiance Fields (NeRF) from Sparse 360°Input

Kai Gu, Thomas Maugey, Sebastian Knorr, Christine Guillemot

Vanishing Point Aided Hash-Frequency Encoding for Neural Radiance Fields (NeRF) from Sparse 360° Input

Kai Gu* Thomas Maugey* Sebastian Knorr†,‡ Christine Guillemot*

*INRIA, France; †Ernst-Abbe University of Applied Sciences Jena, Germany;
‡Technical University of Berlin, Germany.

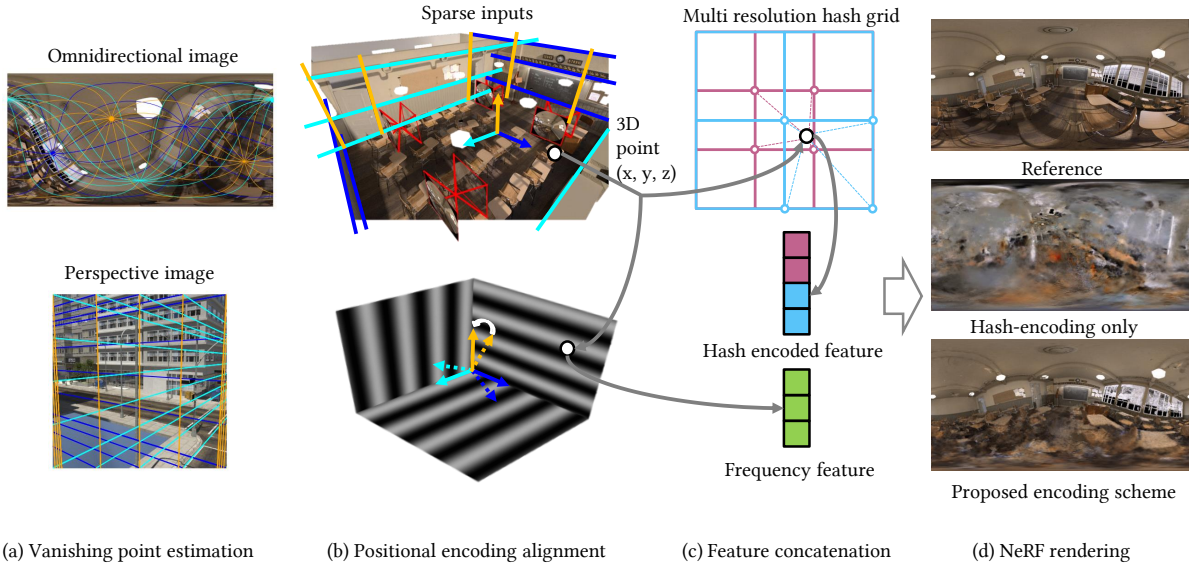


Figure 1: The proposed method improves the sparse input view synthesis of the hash-encoding-based NeRF method. (a) Vanishing point extraction from the perspective or omnidirectional images. (b) Scene-positional encoding alignment according to estimated vanishing points. (c) Hash-frequency encoding compositing (d) NeRF rendering

Abstract

Neural Radiance Fields (NeRF) enable novel view synthesis of 3D scenes when trained with a set of 2D images. One of the key components of NeRF is the input encoding, i.e. mapping the coordinates to higher dimensions to learn high-frequency details, which has been proven to increase the quality. Among various input mappings, hash encoding is gaining increasing attention for its efficiency. However, its performance on sparse inputs is limited. To address this limitation, we propose a new input encoding scheme that improves hash-based NeRF for sparse inputs, i.e. few and distant cameras, specifically for 360° view synthesis. In this paper, we combine frequency encoding and hash encoding and show that this combination can increase dramatically the quality of hash-based NeRF for sparse inputs. Additionally, we explore scene geometry by estimating vanishing points in omnidirectional images (ODI) of indoor and city scenes in order to align frequency encoding with scene structures. We demonstrate that our vanishing point-aided scene alignment further improves deterministic and non-deterministic encodings on image regression and NeRF tasks where sharper textures and more accurate geometry of scene structures can be reconstructed.

*{firstname.lastname}@inria.fr;
†{firstname.lastname}@eah-jena.de

Index Terms: Computing methodologies—Artificial intelligence—Computer vision—3D imaging

1 Introduction

Virtual reality (VR) and augmented reality (AR) have become increasingly popular in recent years, creating a demand for high-quality and efficient virtual scene representation to provide an immersive experience for users. To achieve this, 6-degree-of-freedom (6DoF) navigation within a scene and continuous view synthesis from a set of real captures are essential. The light field camera is one promising solution, which uses a structured camera array to capture densely packed ray directions, enabling continuous view synthesis. Omnidirectional image (ODI) is also a widely used format as it offers 3DoF inherently from a single capture. To accurately represent a 3D scene and generate photorealistic novel views, a dense sampling of views from various positions and directions is desirable. However, it can be challenging due to limited space and camera equipment availability. Recent advancements in image-based view synthesis techniques have made it possible to reconstruct scenes with sparse image captures from commercial devices, i.e. making it more convenient and cost-effective for users.

Neural Radiance Fields (NeRF) [27] technique has been introduced as one powerful technique for view synthesis and scene reconstruction from a set of input views. The implicit neural scene representation of NeRF enables the modeling of view-dependent radiance and complex geometry. This is achieved by mapping the spatial positions and viewing directions (represented as 5D coordinates) to

corresponding color and opacity values through a multi-layer perceptron (MLP). Positional encoding is a crucial component of NeRF as it enables coordinate-based MLPs to efficiently learn high-frequency features. The original NeRF method uses a sinusoidal input mapping which is also known as frequency encoding or frequency feature. Several studies have shown that different encoding schemes can further improve the performance and quality of NeRF [40], some even utilize special activation functions without the need for explicit positional encoding [9,35,38]. The most recent study, Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, also known as Instant NGP [29], has enabled fast training and real-time rendering due to its input hashing and sparse feature grid. However, these methods may suffer from heavy degradation,when the input views are distant and sparse.

In the domain of sparse view synthesis, we observed the limitation of the hash-based NeRF technique, which gets trapped easily in local minima due to the inherent ambiguity of the problem. Our motivation is to enable sparse novel view synthesis with the NeRF method while still benefiting from the efficiency provided by the hash-based method. Different approaches have been proposed to address this issue, such as exploiting stable features from 2D images and cross-view consistency [7, 8, 20, 45] to generalize NeRF with few or even single input views, and exploring the sparsity of scene geometry and regularizing NeRF with depth smoothness [31] and ray entropy [23]. However, these methods typically require a large pre-trained model for extracting and aggregating local features, or some extra view sampling phases to regularize unseen views, and none of them focused on the 360° scenes.

Our focus is on the reconstruction of 360° scenes with limited captures from a sparse array of omnidirectional cameras and enabling large-scale 6 DoF navigation in the virtual scene. To tackle the sparse view synthesis problem, particularly for 360° content, we introduce a new hash-frequency encoding that combines the benefits of frequency encoding with the efficiency of hash encoding. Our approach explicitly aligns the frequency encoding with the vanishing points estimated from the planar image or the ODI. This new pipeline enables hash-based NeRF to approximate accurate geometry and radiance from sparse inputs while enhancing the reconstruction of structures aligned with the orthogonal coordinate system. As a result, our proposed method achieves superior quality compared to the baseline methods.

In this paper, we demonstrate that frequency encoding has better generalization performance over hash encoding when dealing with limited inputs and how frequency encoding can assist hash encoding by conducting experiments on synthetic and real datasets. We follow the finding in [40], which shows that the frequency encoding is biased towards data with more frequency details along the axes. We present the improvements, which the scene-encoding alignment brings to 3D-NeRF as well as to 2D image regression task. In summary, our contributions are:

- We propose a new hash-frequency composited encoding scheme for sparse 360 reconstruction tasks. We demonstrate that it dramatically improves the robustness of hash-based NeRF.
- We align the frequency encoding with the scene geometry of indoor and city scenes by estimating vanishing points from planar and spherical images, which further improves the reconstruction quality from 2D and 3D coordinate-based MLPs.

2 Related Work

2.1 Neural Scene Representation

As a new way to model visual signals, the method represents the scene as an MLP that takes the spatial and temporal coordinates as input. The use of coordinate-based MLPs has been gaining increasing attention in the field of 3D vision, largely thanks to the representative work of Mildenhall et al. [27], the Neural Radiance Fields (NeRF). The NeRF technique models the 3D scene by incorporating view-dependent radiance estimation for individual 3D points. The parameters of NeRFs are optimized by minimizing the photometric errors of the predicted color. The proposed method enables high-fidelity view synthesis and has become a new foundation in different computer vision domains, such as simultaneous localization and mapping (SLAM) [49], camera calibration [22, 24, 43], dynamic scene modeling [32, 33, 36], 3D generative modeling [19, 28], and high-quality VR display [13]. The major drawbacks of the NeRF method, such as time-consuming per-scene optimization and slow rendering, have been addressed in many different ways. Some studies address this problem by utilizing partial or full explicit scene representation and a more efficient input encoding scheme [14, 29, 39]. The novel hash-encoding and multi-resolution voxel grid techniques proposed in InstantNGP [29] enable on-the-fly training and real-time inference. However, the results of these methods degrade when inputs become sparser, especially when the cameras are as distant as a few meters from each other. Our proposed encoding scheme improves the robustness of hash-encoding when dealing with sparse input.

2.2 Omnidirectional Imaging

ODIs are usually captured by cameras or camera arrays with a field of view that covers nearly the entire sphere, therefore they are also known as 360° cameras. These cameras allow users to capture their entire surroundings with a single device, providing an immersive experience for viewers. The raw captures from the cameras can have various formats depending on the projection and number of lenses used. Fisheye lenses are the most commonly employed in 360° cameras as they have a large field of view. One widely used format of ODIs is the equirectangular-projection (ERP) image, which maps the sphere onto a rectangular with an aspect ratio of 2:1. It is challenging to apply algorithms for planar images directly on ODIs because of the distortion from the special projection. Therefore, many novel view synthesis pipelines have been developed specifically for ODIs. Multi-sphere [3] and multi-cylinder [15] images allocate blending weights and colors onto discrete surfaces, enabling the synthesis of free viewpoint panoramas. The Omni-NeRF [16] extends NeRF to be trained with raw fisheye captures. Some works [18, 37] focus on photo-realistic renderings from a single ERP image with pre-acquired depth information. Our pipeline models the ERP geometry explicitly, enabling accurate vanishing point estimation and scene-encoding alignment, and therefore achieving novel view synthesis from ODIs with higher quality and allowing large-scale 6 DoF navigation within the virtual scene.

2.3 Sparse Input View Synthesis

One of the most challenging tasks for NeRF approaches is view synthesis from sparse inputs. The MVSNeRF [7] utilizes multi-view stereo (MVS) constraints and creates a cost volume to effectively generalize the NeRF. The PixelNeRF [45] and SRF [8] leverage local features extracted from 2D images by Convolutional Neural Networks (CNN) and thus are able to predict the 3D geometry from limited input views. Higher-level features such as cross-view semantic consistency, are also considered in later studies. DietNeRF [20] utilizes the semantic latent code from CLIP [34] to regularize the appearance in close views. By incorporating pre-acquired depth

information, DS-NeRF [12] improves the reconstruction performance with less image supervision. The RegNeRF [31] regularizes the geometry and appearance of image patches from unseen viewpoints by enforcing the depth smoothness and maximizing the log-likelihood predicted from a pre-trained flow model. Likewise, DiffusioNeRF [44] employs a similar approach, utilizing a pretrained denoising diffusion model to effectively regulate color and depth patches. Meanwhile, InfoNeRF [23] and MIP-NeRF-360 [4] focus on refining the density distribution along individual rays, effectively compacting the representation of the scene.

Regularity of indoor and city scenes is also exploited to improve the accuracy of reconstruction. Guo et al. [17] classify orthogonal surfaces and enhanced low-texture surface reconstruction by enforcing their orthogonality. However, these existing methods for view synthesis from sparse inputs may require costly pre-training or additional regularization steps. In this work, we extract the global scene information by estimating the vanishing points from 2D images and our approach aligns the deterministic input encoding with the scene structure and combines the aligned frequency features with efficient hash-encoding without adding overheads on the NeRF or requiring additional depth information.

2.4 Vanishing Point Estimation

Vanishing point estimation is an important task in computer vision and has been extensively studied in recent years. Traditional methods [25] for vanishing point estimation rely on geometric constraints and line detection algorithms. These methods detect line segments first, and possible vanishing points are proposed with line clustering algorithms based on geometric cues. These methods are often sensitive to noise and require manual tuning of parameters.

Recently, deep learning-based approaches have shown promising results in vanishing point estimation [5, 6]. These approaches formulate vanishing point detection as a classification problem that detects in-image vanishing points. Zhou et al. [47] were inspired by deformable convolution networks [11] and proposed a conic convolution operation to extract features by explicitly enforcing the kernels to follow structural lines, followed by a vanishing point scan on the hemisphere. This method can detect potential vanishing points outside the image. However, this method and the conic convolution only work for perspective images. Our adaptation makes this method compatible with ODIs.

3 Background

Our method is built upon the recent state-of-the-art NeRF [27] pipeline, Nerfacto [41]. In this section, we first introduce the NeRF and its key component: input encoding, as well as the integrated improvements in Nerfacto. To accurately estimate vanishing points, which is a crucial step for encoding alignment, we give an overview of the conic convolution-based approach NeurVPS [47], which is the basis for our extension for ODIs.

3.1 Neural Radiance Fields

Given an input 5D coordinate (x, y, z, θ, ϕ) in 3D space with (θ, ϕ) being the viewing direction, the MLP F_Θ predicts the volume density σ and the view-dependent color $\mathbf{c} = [r, g, b]$, with Θ being the weights of the MLP

$$[\mathbf{c}, \sigma] = F_\Theta(x, y, z, \theta, \phi). \tag{1}$$

To synthesize individual pixels, the points along the projected rays are discretely sampled from the MLP and aggregated through the volume rendering function. The weights of the MLP are learned by minimizing the photometric loss of the predicted pixel color and the ground truth from input images.

Positional Encoding: The NeRF MLP takes the 5-dimensional spatial coordinates (x, y, z, θ, ϕ) as input. In more general cases, feeding the d-dimensional input $\mathbf{u} \in \mathbb{R}^d$ directly to the coordinate-based MLP may result in an underfitting to the low-frequency content of the scene. One effective way is to map the input coordinates to a higher dimensional hyperspace before passing them into an MLP. The proposed mapping $\gamma(\mathbf{u})$ in NeRF uses sinusoids with logarithmically-spaced axis-aligned frequencies, therefore it is also known as frequency encoding:

$$\gamma(\mathbf{u}) = \left[\ldots, \cos\left(2\pi\sigma^{j/m}\mathbf{u}\right), \sin\left(2\pi\sigma^{j/m}\mathbf{u}\right), \ldots\right]^{\mathrm{T}} \quad \text{for } j = 0, \ldots, m-1. \tag{2}$$

the input coordinate $\mathbf{u}$ is scaled to $[0, 1)$, m and σ are hyperparameters which can be tuned for specific tasks. The in-depth study in the paper [40] has shown the behavior of frequency encoding when different parameter selections are applied, they proposed a new frequency encoding called Gaussian random Fourier feature (Gaussian RFF or RFF) that samples frequencies from a normal distribution $\mathcal{N}(0, \sigma^2)$.

Hash Encoding: Instant-NGP [29] is regarded as a major advance in the field of neural representation. The method encodes input coordinates with a multi-resolution hash table to a trainable feature vector. Such an input encoding scheme enables trainable encoding parameters ψ. For the input coordinate $\mathbf{u}$, integer indices are assigned to the neighboring voxels at different grid levels by hashing their corner coordinates. The spatial hash function is expressed as:

$$h(\mathbf{u}) = \left(\bigoplus_{i=1}^{d} u_i \pi_i\right) mod\ T, \tag{3}$$

where $\bigoplus$ is the bit-wise XOR operation, π_i are large prime numbers, and T is the size of the hash table. Next, the learnable features are retrieved by performing a lookup in the hash table. The features located at the corners of different levels are then d-linearly interpolated and concatenated to generate the encoded feature vector $\gamma(\mathbf{u}; \psi)$.

Nerfacto: Tancik et al. [41] developed a novel NeRF framework, which combines various features from the state-of-the-art NeRF methods and is optimized for real data captures. The pipeline incorporates hash-encoding to ensure efficiency. Nerfacto further improves the reconstruction quality for unbounded 360° scenes with the scene contraction and proposal sampling techniques of MIP-NeRF 360 [4]. It contracts the scene by mapping the infinity to a cube with sides of length 2. The proposal sampler is optimized by online distillation to sample occupied spatial regions efficiently without dense sampling along the rays.

3.2 Vanishing Point Estimation

NeurVPS [47] utilizes a canonical conic space to locally compute the global geometric information of vanishing points. It introduces a novel operator called conic convolution, which facilitates feature extraction and aggregation along structural lines (Fig. 4).

Given a vanishing point candidate $\hat{\mathbf{v}}$, the conic convolution network predicts whether there exists a real vanishing point $\mathbf{v}$, where the angle between the corresponding directions of $\hat{\mathbf{v}}$ and $\mathbf{v}$ is less than a predefined threshold μ.

The output $\mathbf{y}$ of a 3×3 conic convolution, which takes the feature map $\mathbf{x}$ as input, can be expressed as follows:

$$\mathbf{y}(\mathbf{p}) = \sum_{\delta x=-1}^{1} \sum_{\delta y=-1}^{1} \mathbf{w}(\delta x, \delta y) \cdot \mathbf{x}\left(\mathbf{p} + \delta x \cdot \tilde{\mathbf{t}} + \delta y \cdot \mathbf{R}_{\frac{\pi}{2}} \tilde{\mathbf{t}}\right), \tag{4}$$

$$\tilde{\mathbf{t}} := \frac{\hat{\mathbf{v}} - \mathbf{p}}{\|\hat{\mathbf{v}} - \mathbf{p}\|_2} \in \mathbb{R}^2, \tag{5}$$

where $\mathbf{p} \in \mathbb{R}^2$ is the pixel coordinates, $\mathbf{w}$ represents a 3×3 trainable convolution filter, $\mathbf{R}_{\frac{\pi}{2}}$ denotes the rotational matrix that performs a

counterclockwise 90° rotation on a 2D vector, $\tilde{\mathbf{t}}$ is the normalized direction vector pointing from the output pixel coordinate $\mathbf{p}$ to the convolution center $\hat{\mathbf{v}}$. This applies specifically to images with pinhole projection.

After 4 consecutive conic convolution and max-pooling layers, the extracted feature map of corresponding vanishing point candidates is then flattened and passed to a sigmoid classifier with binary cross entropy loss. The final prediction is made by a hierarchical sampling of vanishing points candidates on the hemisphere.

4 METHOD

The proposed pipeline consists of two major parts. The first part involves the compositing of hash-frequency encoding where we incorporate the hash-encoding from InstantNGP [29] by modifying the input encoding scheme. The second part focuses on scene-encoding alignment by estimating vanishing points with NeurVPS [47]. To adapt NeurVPS for ODIs, we first explain the 3D geometry within the ERP format. We then demonstrate various transformations applied to the input coordinates for 2D and 3D frequency encodings to ensure alignment with the scene geometry.

4.1 Composition of Hash-Frequency Encoding

Here, we introduce auxiliary features in our encoding scheme to improve the generalizability of the hash encoding. These features are m-level frequency-encoded input coordinates $\gamma_{FE}(\mathbf{u}) \in \mathbb{R}^{2md}$. In the meantime, the input coordinates are hash-encoded (Sec. 3.1) in a k levels multi-resolution feature grid of feature size f. These encoded coordinates are linearly interpolated and concatenated with the frequency features. This concatenation forms a feature vector $\gamma(\mathbf{u}) \in \mathbb{R}^{kf+2md}$. The use of auxiliary features can be found in different applications such as encoded view direction and textures used in neural radiance caching [30].

The hashing process is a pseudo-random permutation, and it is independent for each of the k levels. Due to the multi-resolution design of the feature grid, the encoded features are able to preserve low and high-frequency details without overfitting. However, our experiments indicate that the hash encoding fails when the input becomes extremely sparse (as few as 8 inputs that are 1 meter apart). In contrast, NeRF with sinusoidal frequency encoding can still approximate the scene geometry.

Due to the explicit design of the multi-resolution hash grid, proper centering of the scene and scaling are crucial for achieving optimal performance. Despite the implementation of scene coordinate contraction [4], which reparameterizes infinite coordinates to ensure that the unbounded scene always fits into the scene box with a side length of 2, it is still essential to rescale and position the 3D region or object of interest within the uncontracted region. However, these centering and rescaling processes are not always perfect, particularly when reconstructing 360° scenes with randomly posed cameras. Our proposed encoding scheme introduces more spatial consistency, enhancing the performance of hash encoding. Moreover, these deterministic features also improve the robustness of the system when dealing with imperfect scene positioning and scaling, as they are not reliant on the learnable features from the explicit structure.

4.2 NeurVPS for Omnidirectional Images

In order to align the positional encoding with the scene structure, we need to extract the vanishing points from images. We adapted NeurVPS [47] to enable vanishing point estimation from ERP images. Following the intuition of NeurVPS, the conic convolutional network exploits the geometric prior and learns effectively vanishing point-related features, where the conic kernels should follow the distorted "straight lines" leading toward the convolution center, i.e. the vanishing points.

Parallel Lines and Vanishing Points in ERP: To determine the corresponding kernel rotation for a given vanishing point candidate,

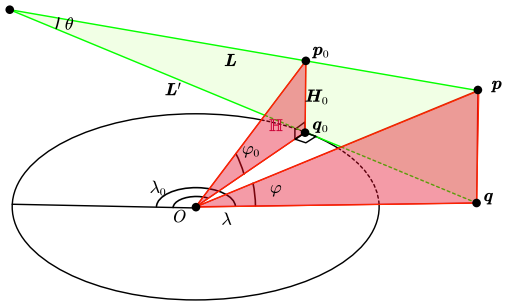


Figure 2: Angular coordinates of a generic spatial line that is specified by $\phi_0, \lambda_0, \theta$. [2].

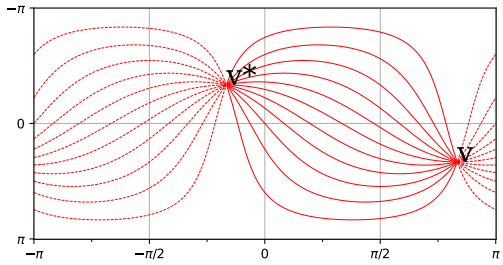


Figure 3: A group of parallel lines (red) and their great circle (dashed) at $\pi/12$ intervals has an incline of $\pi/6$, with vanishing point $\mathbf{v}$ at $(7\pi/12, \pi/6)$.

we first look at the projection of spatial lines on the sphere and show how to find the tangent vector of the projected curves on an ERP image. This involves defining the line equation in 3D and mapping it onto the sphere using its longitude (λ) and latitude (ϕ) coordinates.

Let $\mathbf{L}$ be a spatial line defined in the angular coordinate, $\mathbb{H}$ being the vertical plane where it lies. As shown in Fig. 2, θ is the incline of $\mathbf{L}$ to the horizontal reference plane, λ_0 is the angle from the reference direction to the plane $\mathbb{H}$, and ϕ_0 is the elevation of $\mathbf{p}_0$. $\mathbf{L}$ has its projection $\mathbf{L}'$ on the reference plane, $\mathbf{oq}_0$ is the line segment perpendicular to $\mathbf{L}'$, and $\mathbf{p}_0$ is the point on the vertical plane going through $\mathbf{oq}_0$ that intersects $\mathbf{L}$. We determine the expression of the latitude $\phi(\lambda) = F(\lambda|\lambda_0, \phi_0, \theta)$ of point $\mathbf{p}$ in terms of its longitude λ. Let $\mathbf{q}$ be the orthogonal projection of $\mathbf{p}$ onto the equatorial plane. Let $\Delta\mathbf{X} = |\mathbf{q}_0\mathbf{q}|, \mathbf{D}_0 = |\mathbf{oq}_0|, \mathbf{H}_0 = |\mathbf{q}_0\mathbf{p}_0|$. Then

$$\begin{aligned}\phi(\lambda) &= \arctan\left(\frac{|\mathbf{pq}|}{|\mathbf{qo}|}\right) = \arctan\left(\frac{\mathbf{H}_0 + \tan(\theta)\Delta\mathbf{X}}{\sqrt{\mathbf{D}_0^2 + \Delta\mathbf{X}^2}}\right) \\ &= \arctan\left(\frac{\tan(\phi_0) + \tan(\theta)\tan(\lambda - \lambda_0)}{\sqrt{1 + \tan(\lambda - \lambda_0)}}\right).\end{aligned} \tag{6}$$

Based on the derived line expression in Eq. 6, a fixed λ_0 and θ determine a set of parallel lines with varying elevations ϕ_0. These lines converge at a shared vanishing point at $\mathbf{v}(\lambda_0 + \pi/2, \theta)$ and at its antipode $\mathbf{v}^*(\lambda_0 - \pi/2, -\theta)$. This enables us to express 3D lines on the ERP image plane using the specified vanishing point and ϕ_0, as illustrated in Fig. 3.

Conic Convolution Operators in ERP: Fig. 4 shows the different rotations that should be applied to kernels of the conic convolution for ERP images.

To calculate the rotation of corresponding kernels at any point $\mathbf{p} = (\lambda, \phi)$ towards the vanishing point $\mathbf{v}(\lambda_0 + \pi/2, \theta)$, we must ascertain the tangent vector $\mathbf{t}$ of the projected spatial line $\mathbf{L}$ on the 2D plane. Given a point $\mathbf{p}$ on the spatial lines, where ϕ is a function

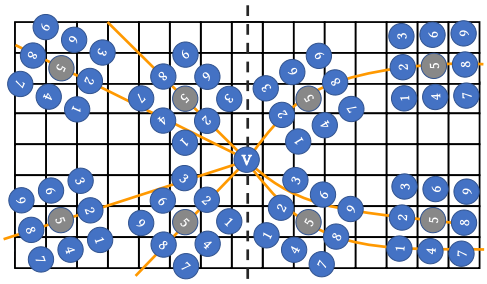


Figure 4: Sampled pixels of 3 × 3 conic convolution kernels for perspective image (left), and ERP image (right). For **v** being the vanishing point, grey circles are the output pixels. Yellow lines indicate the structural lines converging at **v**.

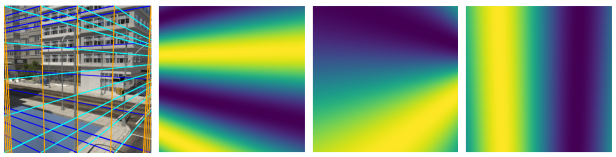


Figure 5: Image of a city scene with structural lines in x (blue), y (cyan), z (orange) directions (left). Encoded 2D coordinates with respect to x (mid left), y (mid right), z (right) directions.

of λ, we can determine $\mathbf{t} = (1, \frac{d\phi}{d\lambda})$ with the slope calculated as follows:

$$\frac{d\phi}{d\lambda} = \frac{\tan(\phi_0)\sin(\lambda_0-\lambda)+\tan(\theta)\cos(\lambda_0-\lambda)}{(\tan(\phi_0)\cos(\lambda_0-\lambda)-\tan(\theta)\sin(\lambda_0-\lambda))^2+1}. \quad (7)$$

The value of ϕ_0 is uniquely determined by the vanishing point **v** and the point **p** through which the spatial line **L** passes. This value can be calculated as follows:

$$\phi_0(\lambda,\phi) = \arctan\left(\tan(\phi)\sqrt{1+\tan(\lambda-\lambda_0)} - \tan(\theta)\tan(\lambda-\lambda_0)\right). \quad (8)$$

And the normalized direction vector $\tilde{\mathbf{t}}$ can be expressed as:

$$\tilde{\mathbf{t}} = \frac{\mathbf{t}}{||\mathbf{t}||_2}. \quad (9)$$

We modify the conic convolution operator by replacing the direction vector $\tilde{\mathbf{t}}$ in Eq. (5). Finally, we adopt the network design and hierarchical sampling strategy from NeurVPS to train our model, which we refer to as NeurVPS-ERP.

4.3 Vanishing point aligned frequency encoding

Once we have extracted the vanishing points from the 2D image, our objective is to align the sinusoidal positional encoding with the scene structures. This alignment ensures that points belonging to the same spatial line share the same mapping of specific encoding levels. However, the process of alignment differs between image regression and NeRF tasks.

Perspective Image Regression: given one of the estimated vanishing points $\mathbf{v}_i$, we establish a polar coordinate system with the origin located at $\mathbf{v}_i = (x_i, y_i)$. Any pixel $\mathbf{p} = (x, y)$ can then be represented in polar coordinates as follows:

$$\begin{aligned} r_i &= \sqrt{(x-x_i)^2+(y-y_i)^2} \\ \alpha_i &= \arctan 2(y-y_i, x-x_i). \end{aligned} \quad (10)$$

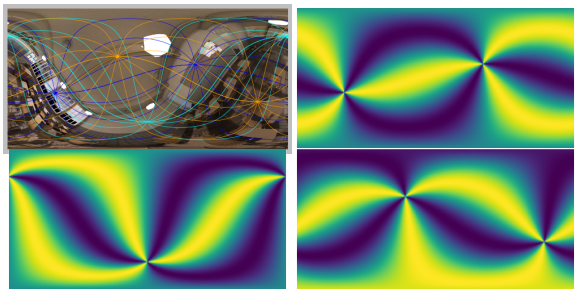


Figure 6: The ERP image of an indoor scene with structural lines in x (blue), y (cyan), z (orange) directions (top left). Encoded 2D coordinates with respect to x (top right), y (bottom left), z (bottom right) directions.

With this parameterization, points on the radial lines originating from the vanishing point will share the same α value. For all n vanishing points, we can then modify the input **u** of the positional encoding as follows:

$$\mathbf{u}_{aligned} = (\alpha_1, \ldots, \alpha_n)^\top, \quad (11)$$

with $\mathbf{u}_{i,aligned} = \alpha_i$ being the polar angles of pixels **p** with respect to the vanishing point $\mathbf{v}_i$. Fig. 5 shows the sinusoidal encoded α in 3 vanishing point directions estimated from the image.

ERP Image Regression: Eq. (6) shows that for a vanishing point $\mathbf{v}_i = (\lambda_0^i + \pi/2, \theta^i)$, values of $\phi_0^i \in (-\pi, \pi)$ define all the parallel lines intersecting in $\mathbf{v}_i$. For any pixel $\mathbf{p} = (\lambda, \phi)$, ϕ_0^i is a function of λ, ϕ as indicated in Eq. (8). Thus, all pixels on the same spatial line have the same ϕ_0 values. Similarly we express the elements of **u** by ϕ_0^i with respect vanishing point $\mathbf{v}_i$ as follows:

$$\mathbf{u}_{aligned} = (\phi_0^1, \ldots, \phi_0^n)^\top. \quad (12)$$

Fig. 6 illustrates the encoded ϕ_0 in 3 orthogonal directions of ERP images.

3D NeRF: The extracted vanishing point in a 2D image can be converted to Cartesian coordinates $\mathbf{v}_i \in \mathbb{R}^3$.

$$\mathbf{v}_i = \left(\sin(\theta^i)\cdot\sin(\lambda_0^i+\frac{\pi}{2}), \cos(\theta^i), -\sin(\theta^i)\cdot\cos(\lambda_0^i+\frac{\pi}{2})\right)^\top. \quad (13)$$

According to the Manhattan world assumption and constrained estimation from NeurVPS, we can estimate 3 vanishing points which form a new orthonormal basis **V**:

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]^\top. \quad (14)$$

The coordinates are sampled on the casted rays from the individual cameras. The camera is associated with a rotation **R** that maps the coordinates from camera space to world space. With the orthonormal basis **V**, estimated from the image with corresponding camera rotation **R**, the global transform should have the form:

$$\mathbf{u}_{aligned} = \mathbf{V}\mathbf{R}^\top\mathbf{u}. \quad (15)$$

5 Experiments

In this section, we first demonstrate the effectiveness of hash-frequency compositing. We then evaluate our vanishing point estimation pipeline by comparing it with the original NeurVPS [47] implementation. Through experimental analysis, we demonstrate that our proposed encoding alignment technique not only enhances our method but also improves different state-of-the-art encoding

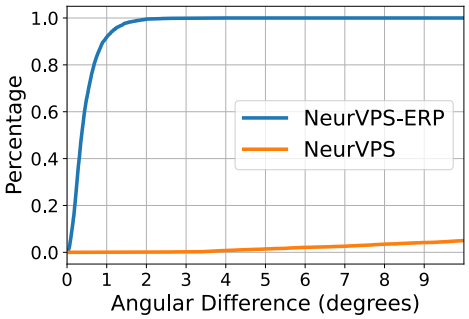


Figure 7: Angle accuracy [0,10) of NeurVPS and NeurVPS-ERP on ERP datasets

schemes. Furthermore, we present competitive results of our pipeline compared to other sparse input NeRF methods. Finally, we illustrate how our alignment technique extends to 2D image regression tasks. For detailed information regarding the implementation of the model, please refer to the supplementary material.

5.1 Hash-Frequency Encoding

We first want to find the optimal settings for combining the hash-encoding (HE) and frequency-encoding (FE) levels. To achieve this, we performed a parameter sweep across different frequency levels (2,4,8,16) while keeping the levels of the hash grid to be fixed at the default value of 16. We conduct our experiment on the synthetic datasets of ODIs from the 2 Blender demos ("Classroom", "Lone Monk") [1]. The scenes are sampled with two different camera setups, sparse ($2\times2\times2$) and normal ($3\times3\times2$). The virtual cameras are placed on the vertices of a subdivided cuboid with a minimal distance of 1 meter between adjacent cameras. For validation, we further render 20 views that are evenly distributed along a path traversing the scene.

We quantitatively evaluate how the frequency features enhance the performance of hash encoding when dealing with sparse inputs. As quality metrics, we report the average Peak Signal to Noise Ratio (PSRN), Structural Similarity (SSIM) [42], and Learned Perceptual Image Patch Similarity (LPIPS) [46] on the validation dataset. We evaluate our model after 30000 iterations with a ray batch size of 4096.

The results in Tab. 1 show that the combination of hash encoding with 8-level frequency encoding (HE + FE8) provides the most stable outcomes for both sparse and normal setups as well as for different scenes.

5.2 Vanishing Point Estimation for ERP images

We train our model NeurVPS-ERP to detect vanishing points from ODIs. For training, we render ERP images of various positions and orientations from 3 synthetic scenes ("Classroom", "Lone Monk", and "City"). For testing, we use a new scene "Barbershop" [1], and a held-out subset from the training views. To compare our extended NeurVPS-ERP model with the original NeurVPS [47], we trained both models on the ERP datasets. The models are evaluated with the angle accuracy(AA) metric proposed in NeurVPS. The AA [0,10) curve (Fig. 7), shows the percentage of predictions whose angle difference is within the thresholds under 10 degrees. The results indicate that the above 95% of the predictions from our model fall within the 2-degree threshold. In contrast, the original NeurVPS implementation struggles to effectively learn features from distorted images, resulting in significantly lower performance.

5.3 Scene Encoding Alignment

We evaluate the effectiveness of our full pipeline, i.e. the vanishing point aligned hash-frequency encoding. We compare our results against the baseline method Nerfacto with hash encoding and MIP-NeRF 360 [4] with frequency encoding. The MIP-NeRF 360 models are evaluated after 200000 iterations with a ray batch size of 1024. Additionally, we demonstrate the effectiveness of our vanishing point (VP) encoding alignment for our method as well as Nerfacto and MIP-NeRF 360. The results are reported on 360° synthetic and perspective real-world datasets. For the synthetic scenes, we introduced a global rotation of 30° along the x-axis to intentionally misalign the encodings. Subsequently, we aligned both the synthetic and real scenes using the estimated vanishing points obtained from the 2D images. As the scene configuration is crucial to the hash-based methods, we conducted experiments on scenes with initial imperfect scaling and fine-tuned (FT) ones. The initial scaling factors are calculated by ensuring all the input cameras are bounded in the scene box of side length 1, i.e. the uncontracted region. However, it's important to note that the fine-tuning of the scale factor is conducted specifically for sparse camera setups. It may not necessarily be optimal for normal camera setups.

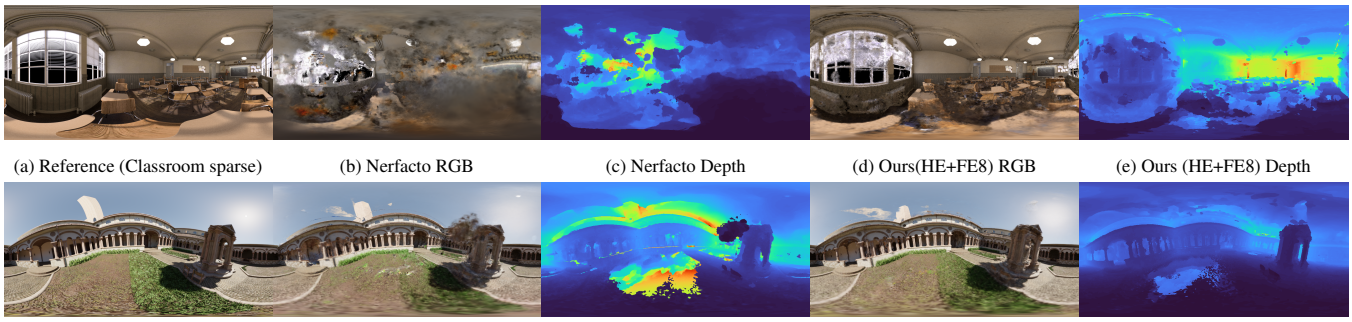
Tab. 2 illustrates that our model outperformed the baseline method Nerfacto and achieves quantitative results comparable to or better than the MIP-NeRF 360. Nerfacto, with its hash-encoding, degrades heavily when input views are sparse (see Fig. 8), while MIP-NeRF 360, with its frequency-encoding, can approximate the scene geometry well. On the other hand, our method shows stable results dealing with sparse inputs even with imperfect scene configurations. In the classroom scene, MIP-NeRF 360 shows the best results due to the high-contrast axis-aligned structures like the blinds (see 9). However, it's worth noting that the training of Nerfacto and our method is completed in approximately 5 minutes, while MIP-NeRF 360 requires around 2 hours of training time on a single NVIDIA A40. The results in Tab. 2 also demonstrate that our vanishing point scene-encoding alignment further improves the encoding approaches, especially for MIP-NeRF 360 (see also Fig. 9) by enhancing the reconstruction quality of axis-aligned structures.

We additionally compare our method against other state-of-the-art sparse input NeRF methods (PixelNeRF [45], MVSNeRF [7], RegNeRF [31]) on forward facing scenes. The evaluation is performed on the LLFF dataset [26] using 3-view setups, following the same experimental settings as in RegNeRF. The averaged results are reported across the entire dataset, which consists of 8 different scenes. The conditional models MVSNeRF and PixelNeRF have trained on the DTU [21] dataset and optimized per scene. Similarly, for hash-based methods, we compare the results of initial imperfect scene scaling and centering with the fine-tuned (FT) configurations.

The results in Tab. 3 show that the proposed method outperforms other conditional models (PixelNeRF, MVSNeRF) even with imperfect scene configurations. The proposed method achieves comparable results to the RegNeRF and better results after a fine-tuning of scene scaling and positioning. The comparison with the hash-encoding baseline method Nerfacto shows the composition of frequency encoding brought improvement to the hash-encoding and enhanced the robustness of Nerfacto while dealing with suboptimal scene scaling (see also Fig. 10).

5.3.1 Encoding Alignment for Image Regression

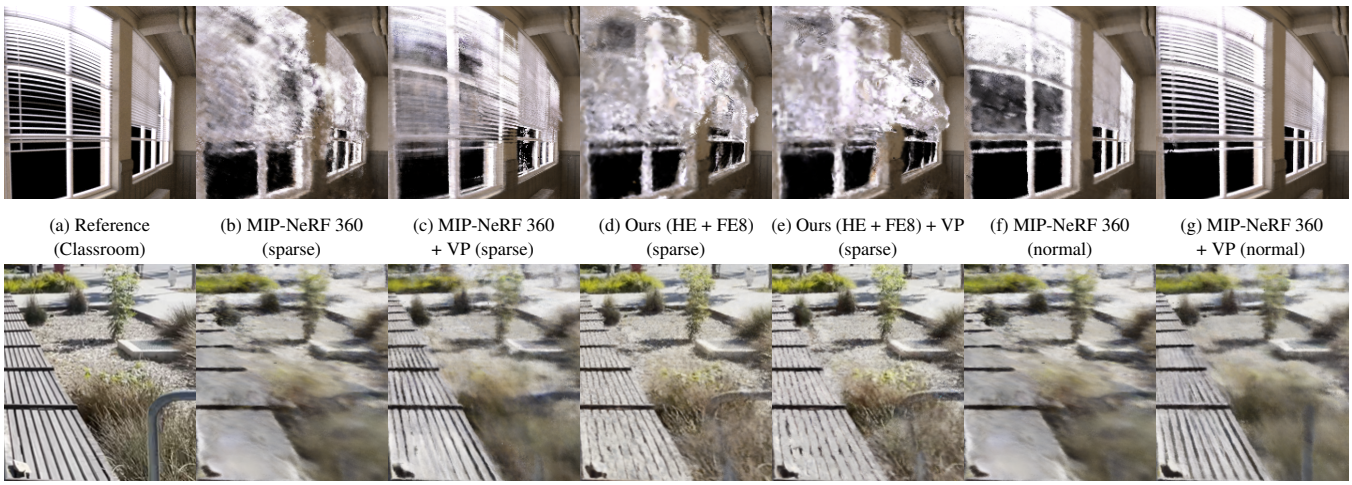
We experimentally demonstrate that the vanishing point encoding-alignment technique also applies to 2D image regression. We followed the test routine described in [40]. Like the 3D NeRF, an MLP is trained to regress from pixel coordinate to RGB value for each image. The training is done from pixels sampled from a regularly-spaced grid containing 1/4 of the pixels, and the test error on the full image is evaluated. We use 256 frequency levels for both the vanilla frequency encoding (FE256) and the Gaussian Fourier feature



(a) Reference (Classroom sparse) (b) Nerfacto RGB (c) Nerfacto Depth (d) Ours(HE+FE8) RGB (e) Ours (HE+FE8) Depth

(f) Reference (Lone Monk sparse) (g) Nerfacto + FT RGB (h) Nerfacto + FT Depth (i) Ours (HE+FE8) + FT RGB (j) Ours (HE+FE8) + FT Depth

Figure 8: Nerfacto degrades or fails when the input data becomes sparse (incorrect geometry and floaters in (b) and (g)). Our combination of hash and frequency encoding enhances the consistency of scene geometry and improves the final rendering significantly.



(a) Reference (Classroom) (b) MIP-NeRF 360 (sparse) (c) MIP-NeRF 360 + VP (sparse) (d) Ours (HE + FE8) (sparse) (e) Ours (HE + FE8) + VP (sparse) (f) MIP-NeRF 360 (normal) (g) MIP-NeRF 360 + VP (normal)

(h) Reference (BWW-entrance) (i) MIP-NeRF 360 (15 views) (j) MIP-NeRF 360 + VP (15 views) (k) Ours (HE + FE8) (15 views) (l) Ours (HE + FE8) + VP (15 views) (m) MIP-NeRF 360 (30 views) (n) MIP-NeRF 360 + VP (30 views)

Figure 9: The alignment of vanishing points improves various encodings by enhancing the on-axis structures in both sparse and normal setups. Among the encoding methods, a visually more significant improvement is observed for the frequency encoding compared to the hash encoding methods.

Reference Nerfacto RegNeRF Ours (HE + FE8) Reference Nerfacto RegNeRF Ours (HE + FE8)

Figure 10: Results on LLFF 3-view setup with suboptimal scene scaling. Our method demonstrates a dramatic improvement over the baseline hash encoding method Nerfacto, i.e. achieving state-of-the-art performance.

Table 1: Quantitative Results of Composited Encoding with Different Levels of Frequency Features

	Classroom (sparse / normal)			Lone Monk (sparse / normal)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
HE + FE2	15.76 / 15.91	0.546 / 0.520	0.591 / 0.552	15.76 / 24.19	0.541 / 0.761	0.378 / 0.145
HE + FE4	21.09 / 25.49	0.669 / 0.820	0.253 / 0.137	21.15 / 24.26	0.657 / 0.764	0.229 / 0.140
HE + FE8	**21.22 / 25.50**	**0.671 / 0.830**	**0.251 / 0.129**	21.72 / 24.33	0.677 / 0.768	0.219 / 0.136
HE + FE16	14.23 / 25.43	0.524 / 0.826	0.587 / 0.132	**21.89 / 24.44**	**0.683 / 0.771**	**0.213 / 0.132**

Table 2: Quantitative Results of Different Methods on 360° Synthetic and Real scenes

	Classroom sparse			Lone Monk sparse			BWW-Entrance 15 views		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MIP-NeRF 360	21.56	0.671	0.289	20.29	0.610	0.279	19.56	0.563	0.417
MIP-NeRF 360 + VP	**22.94**	**0.720**	**0.228**	20.76	0.651	0.235	20.32	0.606	0.358
Nerfacto	14.30	0.487	0.592	10.85	0.380	0.615	13.90	0.472	0.402
Nerfacto + FT	16.46	0.540	0.474	20.19	0.641	0.244	19.75	0.662	0.211
Nerfacto + FT + VP	16.85	0.560	0.434	20.21	0.643	0.242	20.16	0.681	0.199
Ours (HE + FE8)	19.17	0.605	0.314	18.50	0.612	0.308	20.28	0.688	0.189
Ours (HE + FE8) + FT	20.14	0.657	0.285	21.71	0.672	**0.217**	20.40	0.703	0.177
Ours (HE + FE8) + FT + VP	21.22	0.671	0.251	**21.72**	**0.677**	0.219	**20.88**	**0.708**	**0.174**

Table 3: Quantitative Results of Different Sparse-input NeRF Methods on Forward-Facing Scenes

	LLFF (3 views)		
Method	PSNR ↑	SSIM ↑	LPIPS ↓
PixelNeRF	16.17	0.438	0.512
MVSNeRF	17.88	0.584	0.327
RegNeRF	19.08	0.587	0.336
Nerfacto	14.84	0.393	0.389
Nerfacto + FT	17.94	0.576	0.206
Ours (HE + FE8)	18.91	0.598	0.181
Ours (HE + FE8) + FT	**19.09**	**0.604**	**0.176**

Source: results of other sparse-NeRF methods are obtained from the experiments reported in the RegNeRF paper [31].

Table 4: Comparison of Image Regression with Different Encodings.

	SU3	ScanNet	ERP2D
FE256	29.30	38.94	25.23
RFF256	29.50	39.04	25.19
FE255 + VP	**29.74**	**39.30**	**26.70**

(RFF256). For vanishing-point-aligned frequency encoding, the frequency levels should be a multiple of the number of vanishing points. For a fair comparison, we ensure that the vanishing-point-aligned encoding has levels equal to or less than 256. In our experiment, 255 levels of vanishing point aligned frequency features (FE255 + VP) are utilized.

We report the result on the synthetic dataset SceneCity Urban 3D (SU3) [48] and the real dataset ScanNet [10] in Tab. 4. For the ODI regression task, we use the rendering from datasets "Classroom" and "Lone Monk" (ERP2D). We did a hyperparameter sweep to find the best σ in Eq.(2) for each encoding on different data distributions. We report the average PSNR of reconstructed images on different datasets.

Tab. 4 shows that aligned frequency encoding improves the 2D image reconstruction for both perspective and ERP images over the vanilla frequency encoding and the non-deterministic encoding (Gaussian RFF) [40].

6 Conclusion

We addressed the problem of synthesizing 360° views using sparse datasets of hash-encoding-based NeRF by leveraging frequency encoding. In our experiments, we demonstrate that our composite hash-frequency encoding scheme enhances the robustness of hash encoding under imperfect scene scaling and improves the quality of view synthesis with sparse omnidirectional as well as perspective inputs. Additionally, we utilize vanishing point information extracted from 2D images to further improve different encoding schemes through scene-encoding alignment, thereby enhancing the reconstruction of on-axis content.

Limitation and Future Work: Our additional experiments show that as inputs become dense and adequate, the effectiveness of auxiliary frequency features diminishes (see the supplementary material). It will potentially prevent hash-encoding from fitting to fine details and impact performance. Future work in this area could focus on finding more suitable frequency levels for different data and improving the compositing approach without increasing the size of the features. Moreover, in our investigation, aligned positional encoding has proven effective for synthetic scenes with clear structures like Manhattan-world-like scenes, but its effectiveness may be reduced in real natural image captures due to the presence of misaligned objects, motion blur, and noise commonly found in datasets. Aligning scene structures using methods that are not limited to orthogonal vanishing points can be a potential solution.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956770.

References

[1] Blender demo. https://www.blender.org/download/demo-files/, 2023. Accessed: May 22, 2023.

[2] A. Araújo. Drawing equirectangular vr panoramas with ruler, compass, and protractor. *Journal of Science and Technology of the Arts*, 10, 2018. doi: 10.7559/citarj.v10i1.471

[3] B. Attal, S. Ling, A. Gokaslan, C. Richardt, and J. Tompkin. MatryODShka: Real-time 6DoF video view synthesis using multi-sphere images. In *European Conference on Computer Vision (ECCV)*, Aug. 2020.

[4] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5470–5479, 2022.

[5] A. Borji. Vanishing point detection with convolutional neural networks. *ArXiv*, abs/1609.00967, 2016.

[6] C.-K. Chang, J. Zhao, and L. Itti. Deepvp: Deep learning for vanishing point detection on 1 million street view images. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4496–4503, 2018. doi: 10.1109/ICRA.2018.8460499

[7] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14124–14133, 2021.

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

[9] S.-F. Chng, S. Ramasinghe, J. Sherrah, and S. Lucey. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds., *European Conference on Computer Vision (ECCV)*, pp. 264–280. Springer Nature Switzerland, Cham, 2022.

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 764–773, 2017. doi: 10.1109/ICCV.2017.89

[12] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12882–12891, 2022.

[13] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, pp. 1–11, 2022. doi: 10.1109/TVCG.2022.3203102

[14] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5501–5510, 2022.

[15] R. Gadgil, R. John, S. Zollmann, and J. Ventura. Panosynthvr: View synthesis from a single input panorama with multi-cylinder images. In *ACM SIGGRAPH*, SIGGRAPH '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3450618.3469144

[16] K. Gu, T. Maugey, S. Knorr, and C. Guillemot. Omni-nerf: Neural radiance field from 360° image captures. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2022. doi: 10.1109/ICME52920.2022.9859817

[17] H. Guo, S. Peng, H. Lin, Q. Wang, G. Zhang, H. Bao, and X. Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5511–5520, 2022.

[18] C. Hsu, C. Sun, and H. Chen. Moving in a 360 world: Synthesizing panoramic parallaxes from a single panorama. *CoRR*, abs/2106.10859, 2021.

[19] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole. Zero-shot text-guided object generation with dream fields. *arXiv*, 2021.

[20] A. Jain, M. Tancik, and P. Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5885–5894, 2021.

[21] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 406–413. IEEE, 2014.

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

[23] M. Kim, S. Seo, and B. Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12912–12921, 2022.

[24] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[25] G. McLean and D. Kotturi. Vanishing point detection by line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 17(11):1090–1095, 1995. doi: 10.1109/34.473236

[26] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[27] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12346 LNCS:405–421, 2020. doi: 10.1007/978-3-030-58452-8_24

[28] N. Müller, , Y. Siddiqui, L. Porzi, S. Rota Bulò, P. Kontschieder, and M. Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. arxiv, 2022.

[29] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4):102:1–102:15, 2022. doi: 10.1145/3528223.3530127

[30] T. Müller, F. Rousselle, J. Novák, and A. Keller. Real-time neural radiance caching for path tracing. *ACM Transactions on Graphics (TOG)*, 40(4):36:1–36:16, 2021. doi: 10.1145/3450626.3459812

[31] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[32] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 40(6), 2021.

[33] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[34] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, eds., *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

[35] S. Ramasinghe and S. Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *European Conference on Computer Vision (ECCV)*, p. 142–158. Springer-Verlag, Berlin, Heidelberg, 2022. doi: 10.1007/978-3-031-19827-4_9

[36] Sara Fridovich-Keil and Giacomo Meanti, F. R. Warburg, B. Recht, and A. Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[37] P. Y. Shreyas Kulkarni and S. Scherer. 360fusionnerf: Panoramic neural radiance fields with joint guidance. *arXiv preprint arXiv:2209.14265*, 2022.

[38] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In

H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473. Curran Associates, Inc., 2020.

[39] C. Sun, M. Sun, and H.-T. Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5459–5469, 2022.

[40] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547. Curran Associates, Inc., 2020.

[41] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023.

[42] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861

[43] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu. NeRF$--$: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[44] J. Wynn and D. Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4180–4189, June 2023.

[45] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4578–4587, 2021.

[46] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068

[47] Y. Zhou, H. Qi, J. Huang, and Y. Ma. Neurvps: Neural vanishing point scanning via conic convolution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds., *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[48] Y. Zhou, H. Qi, Y. Zhai, Q. Sun, Z. Chen, L.-Y. Wei, and Y. Ma. Learning to reconstruct 3d manhattan wireframes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[49] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.