

Cross-Spectral Neural Radiance Fields

Matteo Poggi* Pierluigi Zama Ramirez* Fabio Tosi*
Samuele Salti Stefano Mattocchia Luigi Di Stefano
CVLAB, Department of Computer Science and Engineering (DISI)
University of Bologna, Italy

{m.poggi, pierluigi.zama, fabio.tosi5}@unibo.it

Project page: <https://cvlab-unibo.github.io/xnerf-web>

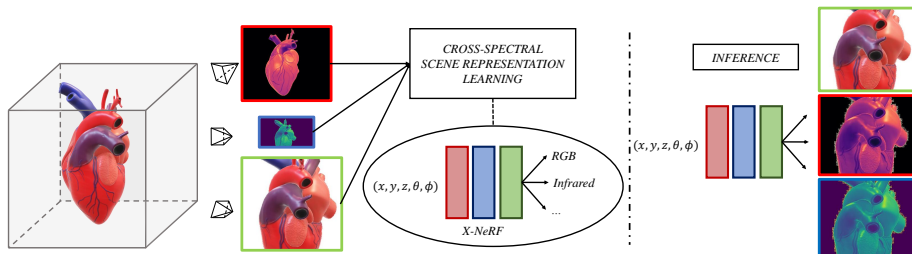


Figure 1: **Cross-Spectral rendering with X-NeRF.** Given a set of images acquired from sensors with different light spectrum sensitivity – such as infrared (red frame), RGB (green frame), multi-spectral (blue frame) – resolution and field of view, we learn a shared cross-spectral scene representation, allowing for novel view synthesis across spectra.

Abstract

We propose *X-NeRF*, a novel method to learn a Cross-Spectral scene representation given images captured from cameras with different light spectrum sensitivity, based on the Neural Radiance Fields formulation. *X-NeRF* optimizes camera poses across spectra during training and exploits Normalized Cross-Device Coordinates (NXDC) to render images of different modalities from arbitrary viewpoints, which are aligned and at the same resolution. Experiments on 16 forward-facing scenes, featuring color, multi-spectral and infrared images, confirm the effectiveness of *X-NeRF* at modeling Cross-Spectral scene representations.

1. Introduction

Novel view synthesis, the task of synthesizing new images of an object or scene observed from arbitrary viewpoints, represents a long-standing problem at the intersection between vision and graphics. It enables several applications: video/image editing, virtual reality and so on.

In the last few years, a popular trend in novel view synthesis is to model scenes as *implicit representations*. On this track, Neural Radiance Fields (NeRF) [31] represents nowadays the most prominent paradigm to render images

from arbitrary viewpoints, which yielded tremendous improvements in the quality of results. NeRF learns a scene representation as a 5D vector-valued function, modeled by a Multi-Layer-Perceptron (MLP), that outputs the emitted color (R, G, B) and volume density σ given as input a 3D location (x, y, z) and a 2D viewing direction (θ, ϕ) .

However, we argue that representing a scene only through RGB colors may be limiting, as it fails to capture the richness of the spectral information around us. For instance, by capturing the surrounding visual information with only a single RGB camera, we cannot perceive natural phenomena that would require analysing the visible spectrum with a finer wavelength granularity – i.e. not only the classic red, green or blue channels – or to go beyond the visible range. Such information could instead be gathered by sensors featuring different spectral sensitivity, such as multi-spectral (MS) or infrared (IR) cameras. Moreover, finding and analysing the correlations between spectra may help to gain a more in-depth understanding of natural processes. Consequently, to be able to reason on multi-spectral data, we would need to obtain a unified Cross-Spectral scene representation – allowing for querying, for any single point, any of the information sensed across spectra.

Based on the above observations, we propose for the first time a Cross-Spectral NeRF (**X-NeRF**), which can model scenes acquired from cameras featuring different spectral sensitivities. Collecting images with a Cross-Spectral rig,

* Joint first authorship.

we extend vanilla NeRF by training a single, shared network across spectra and learning a channel for each spectral band.

However, this straightforward extension alone is not enough to properly model a unified, Cross-Spectral representation. Indeed, two main challenges arise when considering this peculiar setting. The first concerns the need for knowing exact camera poses for any of the images acquired from the different cameras and used to train NeRF. While this information can be obtained effortlessly when dealing with RGB images [40], it is not trivial to obtain camera poses according to a common reference system across the different image modalities. The second is linked to the marked differences between sensors – resolution, field of view (FoV) – which needs to be taken into account when casting rays across 3D space, to ensure that the very same point observed in the scene is reached by rays traced from the corresponding pixel in each camera. For instance, when processing forward-facing scenes, the standard Normalized Device Coordinates convention (NDC) [31, 60] fails at this.

Both the above challenges are addressed by our X-NeRF. As for the former, we obtain camera poses from RGB images [40] and let X-NeRF learn, during the training process, only the relative poses of the other cameras – which are supposed to be constant across views, since we assume cameras being rigidly mounted on a common rig – to obtain the viewpoints for any modality starting from RGB images. Concerning the latter, we propose Normalized Cross-Device Coordinates (NXDC) to align the ray sampling strategy across cameras, taking into account the different resolutions and FoVs so as to correctly map a single point perceived by any of the cameras to the very same 3D location.

As outcome, X-NeRF enables novel view synthesis across spectra and, more importantly, rendering of aligned spectral information from any viewpoint, as shown in Fig. 1. We feel this latter aspect to be one major achievement of X-NeRF, since it yields the following appealing results: i) during rendering, it realizes a *virtual* Cross-Spectral camera, sensing a multitude of spectra from the very same viewpoint – which does not occur when sensing the scene with the different cameras together, ii) given a real image acquired from a specific viewpoint, we can render the remaining modalities aligned to the real image itself, avoiding to address a non-trivial cross-modal matching problem [64, 49], and iii) thanks to its continuous formulation, X-NeRF allows for super-solving low-resolution spectral data, e.g. so as to render MegaPixel MS data whereas existing MS cameras feature a dramatically lower resolution (~ 0.1 Mp).

To evaluate the effectiveness of X-NeRF, we built a custom rig with a high-resolution RGB camera and two low-resolution IR and MS cameras, and used it to acquire a total of 16 forward-facing scenes with ~ 30 different viewpoints for each modality, for a total of 90 views per scene, available in our project page. Our main contributions are:

- We are the first to explore the problem of learning a Cross-Spectral scene representation using the Neural Radiance Field paradigm.
- To obtain camera poses, we learn the relative transformation between different sensors while training X-NeRF itself, thus avoiding non-trivial across spectra calibration/matching.
- We propose Normalized Cross-Device Coordinates (NXDC), to deal with coordinate system misalignment between modalities in forward-facing scenes.
- We propose a dataset of 16 forward-facing scenes acquired with sensors featuring three modalities (RGB, MS, and IR) used to train and validate our proposal.

2. Related Work

Neural Radiance Fields. Novel view synthesis has a rich history within the computer vision and computer graphics fields. Recent explicit methods based on deep learning train CNNs for this very purpose [65, 11, 30, 46, 22, 51, 26, 44, 15]. Nowadays, NeRF has become the dominant scene representation for view synthesis. It allows for reconstructing photo-realistic novel views by means of a continuous volumetric function parameterized as a fully connected neural network, optimized by using a sparse set of input views. NeRF has inspired many subsequent works that extend its continuous neural volumetric representation in order to deal with different setups, e.g. dynamic scenes [27, 37, 21, 56, 13], relighting [45, 62, 3], imperfect camera poses [23, 54], multi-resolution images [2], deformable agents [34, 50, 12, 33, 35] or to realize generative models [41, 5, 20]. Despite the impressive capability to represent realistic appearance, these works usually suffer from notable limitations such as 1) a long training process, 2) a slow rendering phase and 3) the requirement to perform a standalone training from scratch for any scene. This makes the aforementioned representations impractical for use in most applications that require real-time rendering.

Faster NeRF Rendering. Different approaches pursue speeding up of the volume rendering process run by MLP-based representations. Recent works combine a dense 3D grid of MLPs with empty space skipping and early termination [38], build and dynamically update an octree structure to avoid redundant MLP queries in free space [24] or leverage explicit volumetric representations [55, 58, 14, 16]. Although the rendering speed up, gradient-based optimization cannot be used to directly optimize the data structures that are necessary for fast rendering. This means that a conversion step, from a trained model to the final representation that allows real-time rendering, is still needed.

Faster NeRF Training. Other recent works that focus on fewer input views bring faster convergence and, thus, a

faster training process. Such methods typically rely on pre-training aimed at achieving generalization [59, 53], traditional Multi-View Stereo (MVS) approaches [7, 6], neural rays [25], exploiting explicit representations [1] or combining them with implicit ones [47, 32].

Multi-Spectral Imaging. There exist several works in the field of multi-spectral (MS) imaging in the most diverse areas, ranging from robotics to automotive and from biometrics to surveillance. These applications demand a combination of visible and non-visible wavelengths ranges such as Near infrared (NIR), short-wave infrared (SWIR) and mid-wave infrared (MWIR). To name a few, some works adopt RGB-NIR for scene parsing [9] and recognition [4] while others deploy NIR images for color enhancement [61] and dehazing [10]. Other sensors, such as thermal cameras, can directly measure long-wave infrared radiation of objects regardless of an external light source, and have been deployed in pedestrian detection applications [17, 57]. Moreover, cross-spectral matching represents another challenging task that consists in recovering depth by finding correspondences between images with different spectra, in most cases by matching RGB-MS[49], RGB-IR[8, 29], RGB-thermal[36] and RGB-NIR modalities [64, 42, 18, 19].

3. Method

In this section, we present our novel X-NeRF framework. We first introduce the NeRF background, then dig into the main novelties featured by X-NeRF.

3.1. Background: Neural Radiance Field

Given an observed scene, NeRF [31] allows for novel view synthesis from arbitrary vantage points. This is achieved by training a neural network, i.e. a Multi Layer Perceptron (MLP), on a set of sparse images collected from different viewpoints. The MLP parametrises the *Radiance Field* of the scene, i.e. a function of continuous 5D values (x, y, z, θ, ϕ) , where $\mathbf{x} = (x, y, z)$ are 3D coordinates in space and (θ, ϕ) are viewing angles, function of camera pose Π . The direction can be also expressed as a 3D Cartesian unit vector \mathbf{d} . Such a function produces a 4D output R, G, B, σ , encoding the color ($\mathbf{c} = (R, G, B)$) and volume density (σ) of each 3D point in the scene. Specifically, the vanilla NeRF estimates color and density by means of two MLPs as $(\sigma, \mathbf{e}) = \text{MLP}^{(\text{pos})}(\mathbf{x})$, $\mathbf{c} = \text{MLP}^{(\text{rgb})}(\mathbf{e}, \mathbf{d})$, with σ being interpreted as the differential probability of a ray terminating at (x, y, z) , and \mathbf{e} being a feature embedding.

Volume Rendering. According to [28], the color $C(\mathbf{r})$ rendered from a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ is obtained by solving the following integral:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))c(\mathbf{r}(t), \mathbf{d})dt \quad (1)$$

with $T(t)$ being the accumulated transmittance from t_n to t along ray r . The value of the integral is estimated via quadrature, by sampling $[t_n, t_f]$ in N evenly-spaced bins, with t_n and t_f being the near and far plane respectively.

$$C(\mathbf{r}) = \sum_{i=1}^N T_i(1-\exp(-\sigma_i\delta_i))c_i, \quad T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right) \quad (2)$$

with δ_i being the distance between adjacent samples t_{i+1} and t_i . This procedure turns out to be equivalent to alpha compositing, assuming $\alpha_i = 1-\exp(-\sigma_i\delta_i)$. Terms $T(i)\alpha_i$ act as a *weight* (w_i) for each point along the ray.

Positional Encoding with Fourier Features. Traditionally, neural networks excel at learning low-frequency representations at the expense of high-frequency ones. As shown by NeRF [31], encoding 3D coordinates \mathbf{x} into a higher dimensional space allows to better recover the latter. This is achieved by applying a so-called Fourier mapping γ to each input component p independently as $\gamma(p) = (\sin(2^0\pi p), \cos(2^0\pi p), \dots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p))$.

Ray Coordinates. According to the specific scene, two main coordinate systems are used to compute 3D coordinates of points laying along rays. In case of 360° scenes framing objects with masked backgrounds, conventional world coordinate systems are used, requiring the definition of near/far bounding planes. In case of forward-facing scenes, i.e. the camera rotation between views is small or absent, Normalized Device Coordinates (NDC) [31, 60] are used as standard convention, warping an infinitely deep camera frustum into a bounded $[-1, 1]^3$ cube, where distance along the z-axis corresponds to disparity (inverse distance). This parameterization optimizes the network capacity in a way that is consistent with the geometry of perspective projection, easing the problem itself – in particular, in presence of large displacements between foreground and background [60].

3.2. X-NeRF: Cross-Spectral NeRF

Differently from the original NeRF formulation, our goal is to obtain a Cross-Spectral neural scene representation. We assume availability of N_m cameras featuring different modalities (e.g., RGB, infrared, multi-spectral), mounted on a rigid system – i.e. with fixed relative poses between cameras. Each camera with modality m has a spatial resolution of $H_m \times W_m$ and C_m number of channels, and it has been previously calibrated to estimate the intrinsic parameters $f_m^x, f_m^y, c_m^x, c_m^y$ (focal length and piercing point). For each camera, we acquire a set of N_{views} images from different viewpoints of the same scene. Thus, we gather a total of $N_m \times N_{\text{views}}$ images per scene.

We learn the Cross-Spectral scene representation as a function to map 5D coordinates (x, y, z, θ, ϕ) into a volume density (σ) and N_m output modalities, with each modality

having C_m channels, for a total of $\sum_m C_m$ channels. In our setup, we assume a shared volume density across modalities: this means rays emitted by the different sensors hit the very same 3D points and frame it in the images. This assumption would not hold in case this latter hypothesis is violated (e.g., when dealing with RGB and X-rays sensors).

At each optimization iteration, we select a training image with modality m , iterating over all the possible modalities at each step. Then, we sample a random batch of camera rays from the set of all its pixels. For each ray \mathbf{r} , we then use the volume rendering procedure described in Sec. 3.1 to estimate the response of that modality, $\hat{C}_m(\mathbf{r})$. Our loss is simply the total squared error between the rendered and true pixel values for the considered modality:

$$L_m = \sum_{\mathbf{r} \in R} \|\hat{C}_m(\mathbf{r}) - C_m(\mathbf{r})\|_2^2 \quad (3)$$

where R is the set of rays in each batch, and $C_m(\mathbf{r})$ and $\hat{C}_m(\mathbf{r})$ are the ground truth and predicted modality for ray \mathbf{r} , respectively. Since each modality is acquired from different viewpoints, for a single ray \mathbf{r} we can never compute the loss on multiple modalities. Thus, the training is carried out on the different modalities in interleaved manner.

Pose Estimation. NeRF assumes camera poses to be known beforehand during training. This information is typically retrieved by means of COLMAP [40] on the set of RGB images based on matching between image keypoints. Since in our case we have images captured by cameras with different modalities, it is extremely hard to estimate reliable keypoints and descriptors amenable to perform matching across modalities. A possible solution could be to apply COLMAP on each modality independently. However, the estimated poses would be in different reference systems (typically the first frame of the sequence) and up to different scale factors, and would be not trivial to align all cameras in a shared reference system – since it would require, again, matching across modalities.

However, as we assume cameras to be mounted on the same rigid rig, we can exploit COLMAP only on a single modality, and learn the relative poses between sensors as latent variables optimized during training, thereby avoiding the problem of matching pixel across spectra. One may argue that relative poses could be estimated offline through calibration [63]. It is however non-obvious how to perform it across distant spectra, e.g. LWIR vs RGB may need ad-hoc calibration patterns [43]. More importantly, such poses would be metric and thus require alignment with the unknown COLMAP scale, a non-trivial problem itself.

Formally, given poses $\Pi_{m_\alpha}^i, i \in \{1..N_{views}\}$ estimated by COLMAP on a reference modality m_α – RGB in our setup – we learn the relative poses $\Pi_{m_\alpha \rightarrow m}$ for any modality $m \neq m_\alpha$. These, multiplied by $\Pi_{m_\alpha}^i$, allows for obtaining poses Π_m^i for any image collected by the cam-

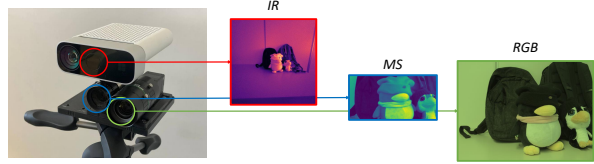


Figure 2: **Multi-modal camera rig.** We show the sensors suite used to collect our dataset.

era of modality m . Relative poses are learned by back-propagating the loss from Eq. (3) up to $\Pi_{m_\alpha \rightarrow m}$ as in [54]

$$\Pi_m^i = \Pi_{m_\alpha \rightarrow m} \times \Pi_{m_\alpha}^i, \quad \Pi_m^i = \arg \min_{\Pi_m^i} L_m(\mathbf{r}) \quad (4)$$

We highlight that each pose $\Pi_{m_\alpha \rightarrow m}$ is learnt based on the reconstruction loss of its modality m solely, without enforcing any match across modalities. We will show empirically that this is sufficient to estimate consistent poses.

NXDC: Normalized Cross Devices Coordinate. In forward-facing scenes, ray coordinates are usually expressed in Normalized Device Coordinates (NDC) [31]. However, NDC assumes that all images have been acquired by the same camera. In our case, we have several cameras with marked differences such as resolution, focal length, etc. Using every camera with its own intrinsics would cause a misalignment between ray coordinates across devices, leading X-NeRF to learn a non-registered scene representation – i.e., the rendered images from the same point of view would be misaligned. Thus, we introduce Normalized Cross-Device Coordinates (NXDC). Assuming cameras looking in the $-z$ direction¹, 3D points (in homogeneous coordinates) are projected according to perspective projection matrix M , function of near/far clipping planes n, f and top/right scene bounds r, t at near plane n

$$\pi(\mathbf{x}) : \begin{pmatrix} \frac{nx}{r} \\ \frac{ny}{t} \\ y \\ -\frac{(f+n)z}{f-n} - \frac{2fn}{(f-n)z} \\ -z \end{pmatrix} \rightarrow \begin{pmatrix} \frac{nx}{-rz} \\ \frac{ny}{-tz} \\ \frac{f+n}{f-n} + \frac{2fn}{z(f-n)} \\ -z \end{pmatrix} \quad (5)$$

The projected point is now in NDC space, where the frustum has been mapped to a $[-1,1]^3$ cube. Given a ray $\mathbf{o} + t\mathbf{d}$, we want to find the ray in NDC space that traces out the same point as the original ray (either at the same rate or not) – i.e., compute the ray origin \mathbf{o}' and direction \mathbf{d}' such that, for every sampled point with t , there exists a t' such that $\pi(\mathbf{o} + t\mathbf{d}) = \mathbf{o}' + t'\mathbf{d}'$. We define:

$$a_x = -\frac{n}{r} \quad a_y = -\frac{n}{t} \quad a_z = \frac{f+n}{f-n} \quad b_z = \frac{2fn}{f-n} \quad (6)$$

¹http://www.songho.ca/opengl/gl_projectionmatrix.html

Assuming that the far scene bound is infinity, we obtain $a_z = 1, b_z = 2n$. If we consider the standard pinhole camera math, we can rewrite as:

$$a_x = -\frac{f_x}{W/2} \quad a_y = -\frac{f_y}{H/2} \quad (7)$$

where f_x, f_y, H, W are the x and y focal lengths, height and width, respectively.

However, we employ different devices, thus we have different focal length, height and width for each camera. Simply using for each device its own parameters would lead to a different normalization across devices – i.e., different reference systems – and thus X-NeRF will not be able to learn a unified scene representation with registered modalities. To overcome this problem, we constrain ratios $\frac{f_x}{W}$ and $\frac{f_y}{H}$ to be fixed across devices. To achieve this, we select the camera modality having minimum focal/image size ratio – i.e., the largest FoV:

$$m_\beta^w \mid \frac{f_x^{m_\beta^w}}{W_{m_\beta^w}} = \min_m \left(\frac{f_x^m}{W_m} \right), \quad m_\beta^h \mid \frac{f_y^{m_\beta^h}}{H_{m_\beta^h}} = \min_m \left(\frac{f_y^m}{H_m} \right) \quad (8)$$

Following NeRF derivation², we get \mathbf{o}' , t' , and \mathbf{d}' according to fixed ratios as:

$$\mathbf{o}' = \begin{pmatrix} -\frac{m_\beta^w}{2} \frac{o_x}{o_z} \\ -\frac{m_\beta^h}{2} \frac{o_y}{o_z} \\ 1 + \frac{2n}{o_z} \end{pmatrix} \quad t' = \frac{td_z}{o_z + td_z} \quad \mathbf{d}' = \begin{pmatrix} -\frac{m_\beta^w}{2} \left(\frac{d_x}{d_z} - \frac{o_x}{o_z} \right) \\ \frac{m_\beta^h}{2} \left(\frac{d_y}{d_z} - \frac{o_y}{o_z} \right) \\ -\frac{2n}{o_z} \end{pmatrix} \quad (9)$$

In practice, this equals to padding images from sensors with lower FoV, while keeping focals unaltered. We dub the framework presented so far as **X-NeRF**.

Speeding-up X-NeRF. Finally, given the recent advances concerning fast training and rendering [47, 1, 32], we also implement a faster variant of X-NeRF to bring it closer to unconstrained use in real applications. Specifically, we exploit a mixed implicit-explicit representation to speed up both phases. Following DirectVoxGO (DVGO) [47], we implement sparse voxel grids – actually, Multi-Plane Images (MPIs) in the case of forward-facing scenes – allowing for efficient queries in 3D space. Two MPIs, $\mathbf{M}^{(\text{dens})}$ and $\mathbf{M}^{(\text{feat})}$, are built respectively to encode density and feature embeddings, from which σ is extracted by means of trilinear interpolation on the former, while color \mathbf{c} is predicted by a shallow MLP queried with features interpolated from the latter as $\sigma(\mathbf{x}) = \text{interp}(\mathbf{x}, \mathbf{M}^{(\text{dens})})$, $\mathbf{c}(\mathbf{x}, \mathbf{d}) = \text{MLP}^{(\text{rgb})}(\text{interp}(\mathbf{x}, \mathbf{M}^{(\text{feat})}), \mathbf{x}, \mathbf{d})$. Both MPIs are optimized through back-propagation during training. We dub this variant of our framework **X-DVGO**, since it extends the DVGO framework.

²https://github.com/bmild/nerf/files/4451808/ndc_derivation.pdf

4. Experimental Settings

4.1. Acquisition Setup and Dataset

Our acquisition setup consists of three devices with different spectral sensitivity: a Ximea RGB camera equipped with a Sony IMX253LQR-C 12.4 Mpx sensor; an MS camera sensitive to 10 bands within the visible spectrum, based on a IM-SM4X4-VIS2 2.2 Mpx (one MS pixel capturing the 10 bands information uses a 4×4 grid of native pixels, thus reducing the spatial resolution to 1/16); the passive infrared sensor of an Azure Kinect device, with a native resolution of 1Mpx. Accordingly, our rig perceives 14 total channels across the sensed modalities. The three cameras have been mounted on a rig, as depicted in Fig. 2, to acquire 16 indoor scenes – since the Kinect IR sensor saturates outdoor – with ~ 30 images from each sensor per scene, 5 kept out for testing and the remaining used for training. As already mentioned, in this paper we focus on scene acquired in forward-facing settings. Examples of images acquired by our rig are shown in Fig. 2, where IR and MS images are encoded with colormaps `magma` and `viridis`, respectively (with MS being averaged over channels).

4.2. Network Implementation and Training Details

We implemented our framework using PyTorch. During training and testing, all images are normalized in $[0, 1]$ over a single scene and modality, clipping intensities to the 99th percentile to filter intensity peaks in MS and IR images. Since IR images acquired by the Kinect are particularly noisy, we pre-process them by means of a 7×7 bilateral filter [48]. In the remainder of this sub-section we report implementation details concerning X-NeRF and X-DVGO, whose output layers have been extended to predict multiple modalities as described in Sec. 3.2.

X-NeRF. It is built on top of the NeRF-- codebase [54], which replicates the vanilla NeRF except for (i) not using hierarchical sampling strategy, (ii) reducing hidden layers dimension from 256 to 128 and (iii) sampling only 128 points along each ray, following [54] to pursue computational efficiency. X-NeRF is trained for 5K epochs on each scene, i.e. ~ 450 k steps (150K per modality, ~ 2.5 hours of overall training on a single 3090 RTX GPU).

X-DVGO. It is built on top of the DVGO codebase [47], using 128 depth planes for MPIs and a shallow MLP made of two hidden layers with 128 channels. Following the default settings, X-DVGO is trained for 75K steps (25K per modality, taking about 15 minutes overall on a single 3090 RTX GPU), using a total variation regularizer [39] in addition to the rendering loss. Since we observed sub-optimal results when jointly optimizing camera poses with the MLP and MPIs, leading to scarce alignment, we bootstrap X-DVGO with camera poses learned after a few steps of X-NeRF training. However, training this latter for < 10 min-

Configuration				Avg.	
Model	Train	NXDC	Test	PSNR	SSIM
NeRF	RGB	-	RGB	32.44	0.869
X-NeRF	RGB+MS	✗	RGB	31.33	0.862
X-NeRF	RGB+MS	✓	RGB	31.93	0.864
NeRF	MS	-	MS	33.53	0.917
X-NeRF	RGB+MS	✗	MS	31.96	0.897
X-NeRF	RGB+MS	✓	MS	33.87	0.918

Table 1: **Bimodal Cross-Spectral rendering quality – NeRF vs X-NeRF.** We report PSNR and SSIM averaged over the whole dataset.

utes yields stable poses, ready for training X-DVGO.

5. Experimental Results

We evaluate X-NeRF on two main tasks: novel view synthesis and cross-modal image alignment. In most tests, we highlight **best**, **second best** and **third best** methods according to average performance. Results on single scenes are reported in the **supplementary material**.

5.1. Novel View Synthesis

We start by evaluating the quality of novel views rendered by X-NeRF and X-DVGO for any modality. Specifically, given a single modality – MS, for instance – we render images from the viewpoints of that camera alone – e.g., the MS camera – and measure the quality over such modality – e.g., MS predictions by the MLP. To this aim, we report the PSNR and SSIM metrics [31] (the higher, the better), leaving out LPIPS – since meaningful for RGB images only.

Bimodal Cross-Spectral Radiance Field. As a first experiment, we train X-NeRF to deal with images belonging to two modalities, RGB and MS, and compare it with a vanilla NeRF trained on single modalities alone. In this case, a single relative pose between RGB and MS camera is learned during optimization. Tab. 1 collects the outcome of this evaluation. Each row corresponds to a specific model (NeRF or X-NeRF), the modalities used for training, optional use of NXDC space and the testing modality – which also bounds rendering resolution.

Considering RGB rendering, we can notice that the vanilla NeRF trained on RGB images alone achieves, overall, the best performance. This is not surprising, since the additional MS images processed by X-NeRF are at much lower resolution (about 100× smaller) and, of course, of different modality. However, the drop is moderate thanks to the MS bands partially overlapping the RGB ones. Moreover, we can appreciate how the drop is smaller when the NXDC space is used, showing how our proposal favors learning a joint representation of the two modalities by the MLP. By looking at MS rendered images, we can appreciate how X-NeRF outperforms vanilla NeRF with NXDC, rendering higher-quality images. The higher-resolution of RGB images and the partial overlap with MS ones allows

Configuration				Avg.	
Model	Train	NXDC	Test	PSNR	SSIM
NeRF	RGB	-	RGB	32.44	0.869
X-NeRF	RGB+MS+IR	✗	RGB	30.43	0.856
X-NeRF	RGB+MS+IR	✓	RGB	31.61	0.862
NeRF	MS	-	MS	33.53	0.917
X-NeRF	RGB+MS+IR	✗	MS	30.87	0.870
X-NeRF	RGB+MS+IR	✓	MS	33.53	0.914
NeRF	IR	-	IR	33.26	0.897
X-NeRF	RGB+MS+IR	✗	IR	31.60	0.869
X-NeRF	RGB+MS+IR	✓	IR	32.44	0.879

Table 2: **Trimodal Cross-Spectral rendering quality – NeRF vs X-NeRF.** We report PSNR and SSIM averaged over the whole dataset.

Configuration				Avg.	
Model	Train Time	NXDC	Test	PSNR	SSIM
X-NeRF	~2.5 hours	✓	RGB	31.61	0.862
X-DVGO	~22.5 mins	✓	RGB	31.77	0.887
X-NeRF	~2.5 hours	✓	MS	33.53	0.914
X-DVGO	~22.5 mins	✓	MS	33.22	0.922
X-NeRF	~2.5 hours	✓	IR	32.44	0.879
X-DVGO	~22.5 mins	✓	IR	31.60	0.908

Table 3: **Trimodal Cross-Spectral rendering quality – X-NeRF vs X-DVGO.** We report PSNR and SSIM averaged over the whole dataset.

for such improvement, while X-NeRF using NDC cannot exploit such advantage. Further experiments concerning NDC and NXDC are reported as **supplementary material**.

To summarize, X-NeRF achieve comparable performance with respect to NeRF when rendering RGB images – thanks to the NXDC space – while it can effectively exploit the learned cross-spectral representation of the scene to improve the quality of rendered MS images.

Trimodal Cross-Spectral Radiance Field. We now add a further modality to X-NeRF, training it to render jointly RGB, MS and IR images. With this setup we will conduct all the following experiments. Tab. 2 collects the outcome of this experiment. We can notice how, on any modality, the vanilla NeRF trained on the single modality alone achieves the best results on average. Again, we feel this trend to be not surprising, given the variety of content framed by the three modalities and the very different resolutions of each. However, we can observe once again how NXDC allows for the smallest drops. By looking at the individual modalities, on average X-NeRF achieves equivalent performance with respect to vanilla NeRF on MS images, while dropping on RGB and IR rendered images.

Learned Poses Analysis. We now inquire about how relative poses between RGB-MS and RGB-IR cameras are optimized by X-NeRF during training. Fig. 3 shows how MS and IR cameras centers move during training, according to the relative pose learned with respect to the RGB camera (green) after a certain number of epochs, encoded in blue

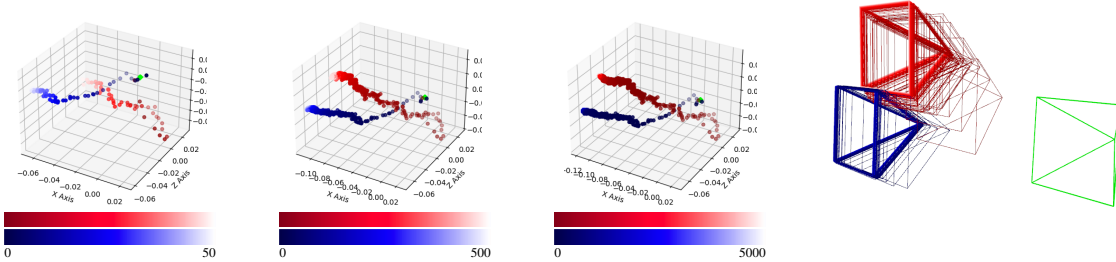


Figure 3: **Relative cameras positions during training.** Left: we show how camera centers translate in 3D space in 50, 500 and 5000 epochs. Right: we display camera frustums. After ~ 250 epochs (~ 7.5 minutes), cameras become stable.

Configuration		~ 16.5 mins			~ 22.5 mins			~ 30 mins			~ 1 hour		
Model	Test	Pose epochs	PSNR	SSIM	Pose epochs	PSNR	SSIM	Pose epochs	PSNR	SSIM	Pose epochs	PSNR	SSIM
X-NeRF	RGB	550	28.51	0.849	750	29.18	0.852	1000	29.67	0.854	2000	30.65	0.858
X-DVGO	RGB	50	31.55	0.887	250	31.77	0.887	500	31.83	0.888	1500	31.81	0.887
X-NeRF	MS	550	28.49	0.850	750	29.50	0.864	1000	30.38	0.877	2000	32.19	0.901
X-DVGO	MS	50	32.92	0.918	250	33.22	0.922	500	33.28	0.923	1500	33.37	0.923
X-NeRF	IR	550	28.55	0.838	750	29.55	0.849	1000	30.38	0.877	2000	31.30	0.869
X-DVGO	IR	50	31.40	0.906	250	31.60	0.908	500	31.57	0.908	1500	31.62	0.909

Table 4: **Trimodal Cross-Spectral rendering quality - fixed time budget.** We report PSNR and SSIM (dataset average) under different training schedules.

(MS) and red (IR) color intensities respectively. According to colors being normalized over different epoch ranges, specifically 50, 500 and 5000, we can notice how after roughly 250 epochs the relative poses get stable and very close to those obtained after an entire training cycle. We can notice how the camera visualized in Fig. 3 (right) are placed as in the real rig shown in Fig. 2.

Speeding-up Cross-Spectral Radiance Fields. We now evaluate the rendering performance of the X-NeRF accelerated variant, namely X-DVGO. To train X-DVGO on a single scene, we bootstrap camera poses by training for 250 epochs X-NeRF on the same scene – taking about 7.5 minutes. Then, we freeze relative poses and start training X-DVGO. Tab. 3 shows a comparison between X-NeRF and X-DVGO, trained on RGB, MS and IR modalities jointly and tested to render any of the three. In general, when rendering MS and IR images the two achieve very similar performance on average, with X-DVGO achieving slightly lower PSNR and higher SSIM scores, while when rendering RGB images X-DVGO outperforms X-NeRF on both metrics while training approximately $8\times$ times faster (i.e., 7.5 plus 15 minutes versus 2.5 hours).

We now study the impact of the bootstrapped poses on X-DVGO performance, aimed at assessing the advantages it yields in terms of time required for training. In Tab. 4 we report rendering performance by X-DVGO when trained with poses being optimized for different amount of epochs. We compare it to X-NeRF trained for an amount of time equal to the total time required by X-DVGO (i.e., bootstrapping plus actual 25K steps of training). We can notice how poses optimized for 50 epochs only already yields render quality not that far from those by X-NeRF trained for an entire cycle (5K epochs), with only 16.5 minutes of

Configuration			Avg.		
Model	Train Time	NXDC	RGB 1694×3434	MS 254×510	IR 181×363
X-NeRF	~ 2.5 hours	✗	0.214	0.234	0.277
X-NeRF	~ 2.5 hours	✓	0.668	0.667	0.672
X-DVGO	~ 22.5 mins	✓	0.632	0.630	0.639

Table 5: **Cross-Spectral alignment quality.** We report MI averaged over the whole dataset.

total training. With the very same time budget, X-NeRF achieves remarkably worse results. Some improvements are achieved by X-DVGO using poses optimized for 250 epochs, while elongating the poses initialization process for more epochs does not allow for further significant improvements. X-NeRF still results inferior in rendering quality when limiting the time budget up to one hour, confirming that X-DVGO achieves a better trade-off in terms of training time/rendering quality.

5.2. Cross-modal Alignment

To conclude, we assess how effective X-NeRF and X-DVGO are at rendering images aligned across spectra, i.e. so as to create a *virtual* Cross-Spectral camera. This evaluation is carried out by rendering images according to each of the three cameras viewpoints, and thus at the three different resolutions they are characterized by, which are then cropped to match the area common to the three – i.e., the one observed by the camera with the narrowest FoV, the MS camera in our case. This results in evaluating on 1694×3434 images when rendering from RGB camera viewpoints, 254×510 from MS viewpoints and 181×363 from the IR cameras. We compute pair-wise Mutual Information (MI, the higher the better) [52] across the three modality pairs, and then average the three scores we obtain.

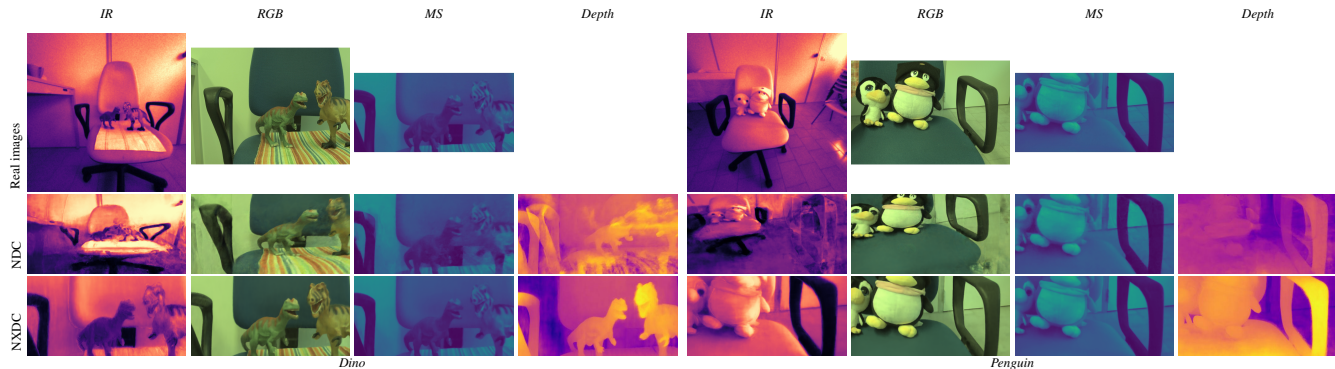


Figure 4: **Cross-spectral rendering, qualitative examples.** Top: real images collected with our rig, followed by images and depth maps rendered by X-NeRF using NDC (middle) or NXDC (bottom), both assuming the MS camera viewpoint.

Configuration	~16.5 mins			~22.5 mins			~30 mins			~1 hour		
	RGB	MS	IR	RGB	MS	IR	RGB	MS	IR	RGB	MS	IR
Model	1694×3434	254×510	181×363	1694×3434	254×510	181×363	1694×3434	254×510	181×363	1694×3434	254×510	181×363
X-NeRF	0.718	0.714	0.720	0.698	0.700	0.703	0.694	0.694	0.701	0.681	0.681	0.686
X-DVGO	0.577	0.573	0.586	0.632	0.630	0.639	0.639	0.636	0.646	0.650	0.649	0.656

Table 6: **Cross-Spectral alignment – fixed time budget.** We report MI (dataset average) under different training schedules.

Tab. 5 collects the outcome of this evaluation, involving X-NeRF – without and with NXDC – and X-DVGO. We can notice how the MI across modalities is very low when X-NeRF uses the classical NDC: indeed, the MLP learns to render the three different modalities by casting rays in very different regions of the 3D space, resulting in unaligned rendered images. On the contrary, NXDC allows for learning much better aligned representations, thus achieving much higher MI scores. We can observe this effect also qualitatively, by looking at images and depth maps rendered by X-NeRF. Fig. 4 reports two samples from the *Dino* and *Penguin* scenes of our collected dataset, followed by images rendered from MS viewpoint by X-NeRF, trained with NDC or NXDC convention. We can notice how the latter allows for rendering images that are aligned across spectra, and properly models the 3D space as we can notice by observing the rendered depth maps.

X-DVGO achieves results almost equivalent to X-NeRF, resulting in slightly lower MI scores. In Tab. 6, we show average MI scores at each resolution achieved when bootstrapping X-DVGO with poses initialized for different amounts of epochs, i.e. the same reported in Tab. 4. As we observed for rendering results, after 250 epochs the improvement almost saturates. By training X-NeRF with the same time budgets, alignment slightly reduces over time to favor the rendering quality over single modalities. The **supplementary material** provides more qualitative results.

5.3. Failure Cases and Limitations

Despite the high quality of both rendering and alignment yielded by X-NeRF, the task we are facing and the hypotheses under which we operate – partially known camera poses and very different sensors modalities, resolutions and FoVs – are particularly challenging, thus some failure cases oc-

cur. Specifically, for some scenes X-NeRF gets stuck into local minima and cannot align the three modalities at their best. We show in the **supplementary material** some examples of this occurrences, with two modalities being properly aligned and the third one resulting slightly drifted.

6. Conclusion

We proposed a novel approach based on Neural Radiance Field, to model scenes across different spectra. Thanks to NXDC, we learn an aligned representation across spectra and render images at the same arbitrary resolution from an arbitrary viewpoint, addressing several problems that do arise when attempting to learn a shared NeRF from multiple devices, such as misalignment between sensors and diversity in resolution. Moreover, by learning the relative poses between sensors, we can get rid of cumbersome cross-spectral calibration. We tested X-NeRF on images acquired by our multi spectral rig, showing the effectiveness of our approach in producing high quality registered images with different modalities. We believe that our work could be useful in several fascinating applications in multi-modal spectral understanding, which we aim at explore in the future. Moreover, our study is now limited to forward-facing scene. Future research will aim at extending X-NeRF also to 360° scenes, addressing the more challenging lightning conditions occurring and the increased aliasing due to huge difference of resolutions across modalities [2]. Finally, another intriguing direction would be to collect images with sensors with extremely different wavelength sensitivity such as X-rays, enabling world understanding at different 3D layers.

Acknowledgements. We gratefully acknowledge the funding support of Huawei Technologies Oy (Finland).

References

- [1] Alex Yu and Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks, 2021. [3](#), [5](#)
- [2] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5855–5864, October 2021. [2](#), [8](#)
- [3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P. A. Lensch. Nerf: Neural reflectance decomposition from image collections. In *ICCV*, 2021. [2](#)
- [4] Matthew Brown and Sabine Süsstrunk. Multi-spectral sift for scene category recognition. In *CVPR 2011*, pages 177–184. IEEE, 2011. [3](#)
- [5] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. Pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021. [2](#)
- [6] 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. [3](#)
- [7] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis from sparse views of novel scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021. [3](#)
- [8] Walon Wei-Chen Chiu, Ulf Blanke, and Mario Fritz. Improving the kinect by cross-modal stereo. In *BMVC*, 2011. [3](#)
- [9] Gyeongmin Choe, Seong-Heum Kim, Sunghoon Im, Joon-Young Lee, Srinivasa G Narasimhan, and In So Kweon. Ranus: Rgb and nir urban scene dataset for deep scene parsing. *IEEE Robotics and Automation Letters*, 3(3):1808–1815, 2018. [3](#)
- [10] Chen Feng, Shaojie Zhuo, Xiaopeng Zhang, Liang Shen, and Sabine Süsstrunk. Near-infrared guided color image dehazing. In *2013 IEEE international conference on image processing*, pages 2363–2367. IEEE, 2013. [3](#)
- [11] John Flynn, Michael Broxton, Paul E. Debevec, Matthew DuVall, Graham Fyffe, Ryan S. Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *CVPR*, 2019. [2](#)
- [12] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *CVPR*, 2021. [2](#)
- [13] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. [2](#)
- [14] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien P. C. Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, 2021. [2](#)
- [15] Tong He, John P. Collomosse, Hailin Jin, and Stefano Soatto. Deepvoxels++: Enhancing the fidelity of novel view synthesis from 3d voxel embeddings. In Hiroshi Ishikawa, Cheng-Lin Liu, Tomás Pajdla, and Jianbo Shi, editors, *ACCV*, 2020. [2](#)
- [16] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul E. Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, 2021. [2](#)
- [17] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1037–1045, 2015. [3](#)
- [18] Seungryong Kim, Dongbo Min, Bumsub Ham, Minh N Do, and Kwanghoon Sohn. Dasc: Robust dense descriptor for multi-modal and multi-spectral correspondence estimation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1712–1729, 2016. [3](#)
- [19] Seungryong Kim, Dongbo Min, Stephen Lin, and Kwanghoon Sohn. Deep self-correlation descriptor for dense cross-modal correspondence. In *European Conference on Computer Vision*, pages 679–695. Springer, 2016. [3](#)
- [20] Adam R. Kosior, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Sona Mokrá, and Danilo Jimenez Rezende. Nerf-vae: A geometry aware 3d scene generative model. In *ICML*, 2021. [2](#)
- [21] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. [2](#)
- [22] Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. Crowdsampling the plenoptic function. In *ECCV*, 2020. [2](#)
- [23] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. [2](#)
- [24] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. [2](#)
- [25] 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. *arxiv CS.CV 2107.13421*, 2021. [3](#)
- [26] Stephen Lombardi, Tomas Simon, Jason M. Saragih, Gabriel Schwartz, Andreas M. Lehrmann, and Yaser Sheikh. Neural volumes: learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 2019. [2](#)
- [27] 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. [2](#)
- [28] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. [3](#)
- [29] Max Mehlretter, Sebastian P Kleinschmidt, Bernardo Wagner, and Christian Heipke. Multimodal dense stereo matching. In *German Conference on Pattern Recognition*, pages 407–421. Springer, 2018. [3](#)

- [30] 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.*, 2019. 2
- [31] 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. 1, 2, 3, 4, 6
- [32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multi-resolution hash encoding. *arXiv:2201.05989*, Jan. 2022. 3, 5
- [33] Atsuhiko Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *ICCV*, 2021. 2
- [34] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. In *ICCV*, 2021. 2
- [35] 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. *arxiv CS.CV 2106.13228*, 2021. 2
- [36] Peter Pinggera¹², Toby Breckon, and Horst Bischof. On cross-spectral stereo matching using dense gradient features. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, volume 2, 2012. 3
- [37] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 2
- [38] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, 2021. 2
- [39] Leonid I Rudin and Stanley Osher. Total variation based image restoration with free local constraints. In *Proceedings of 1st international conference on image processing*, volume 1, pages 31–35. IEEE, 1994. 5
- [40] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4
- [41] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020. 2
- [42] Xiaoyong Shen, Li Xu, Qi Zhang, and Jiaya Jia. Multi-modal and multi-spectral registration for natural images. In *ECCV*, 2014. 3
- [43] Shreyas S Shivakumar, Neil Rodrigues, Alex Zhou, Ian D Miller, Vijay Kumar, and Camillo J Taylor. Pst900: Rgb-thermal calibration, dataset and segmentation network. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9441–9447. IEEE, 2020. 4
- [44] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *CVPR*, 2019. 2
- [45] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021. 2
- [46] Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *CVPR*, 2019. 2
- [47] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *arXiv preprint arXiv:2111.11215*, 2021. 3, 5
- [48] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, ICCV '98, page 839, USA, 1998. IEEE Computer Society. 5
- [49] Fabio Tosi, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Rgb-multispectral matching: Dataset, learning methodology, evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. CVPR. 2, 3
- [50] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a deforming scene from monocular video. In *ICCV*, 2021. 2
- [51] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2
- [52] Paul Viola and William M. Wells. Alignment by maximization of mutual information. *Int. J. Comput. Vision*, 24(2):137–154, sep 1997. 7
- [53] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 3
- [54] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2, 4, 5
- [55] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *CVPR*, 2021. 2
- [56] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *CVPR*, 2021. 2
- [57] Dan Xu, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5363–5371, 2017. 3
- [58] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 2
- [59] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 3

- [60] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2, 3
- [61] Xiaopeng Zhang, Terence Sim, and Xiaoping Miao. Enhancing photographs with near infra-red images. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 3
- [62] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *arxiv CS.CV 2106.01970*, 2021. 2
- [63] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 4
- [64] Tiancheng Zhi, Bernardo R Pires, Martial Hebert, and Srinivasa G Narasimhan. Deep material-aware cross-spectral stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1916–1925, 2018. 2, 3
- [65] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.*, 2018. 2