Hyper-NeRF: Hyperspectral Neural Radiance Fields with Continuous Radiance and Transparency Spectra

Anonymous ICCV submission

Paper ID ****

Abstract

In this work, we extend Neural Radiance Fields (NeRF) from RGB to hyperspectral data to compute hyperspectral 3D reconstructions of scenes from images taken with a hyperspectral camera and turntable. Hyperspectral imagery has been used in many applications to non-destructively determine the material and/or chemical compositions of samples, but is typically only used to create a single aggregate measurement of the subjects as opposed to a spatially varying description. Meanwhile, NeRFs have recently seen widespread success creating high quality 3D representations of scenes from images. Leveraging recent advances in NeRFs, we propose computing a hyperspectral 3D reconstruction in which every point in space and view direction is characterized by wavelength-dependent radiance and transparency spectra. We present a novel NeRF-based approach to predict continuous emittance and transmittance spectra instead of scalar volume density and 3-dimensional color intensity. To evaluate the approaches, a dataset containing 4 scenes with 48 hyperspectral images each was collected. We perform comparisons against traditional RGB NeRF baselines and apply ablation testing with discrete spectra representations. We show that NeRF naturally extends to hyperspectral data with minimal increase in computation, minimal decrease in accuracy, and enables several new potential applications and areas for future study.

1. Introduction

Hyperspectral imagery is a useful tool in many applications for non-destructively characterizing material and chemical compositions. For example, hyperspectral imagery is used in agriculture to assess plant health and nutrient content, in medicine to diagnose diseases, and in drilling to view otherwise invisible gasses like methane. In contrast to typical RGB images which have 3 color channels for each pixel, hyperspectral images consist of tens to hundreds of color channels (wavelengths) for each pixel and typically have little spectral overlap among channels. Because different materials and molecules have different reflectance, transmittance, and/or fluorescence properties at different wavelengths, hyperspectral data may be used to infer the composition of a sample. However, studying the *spatial* data in hyperspectral imagery is currently under-studied for a number of reasons, with many works only using the "image" part of hyperspectral imagery to select foreground pixels which are then averaged together.

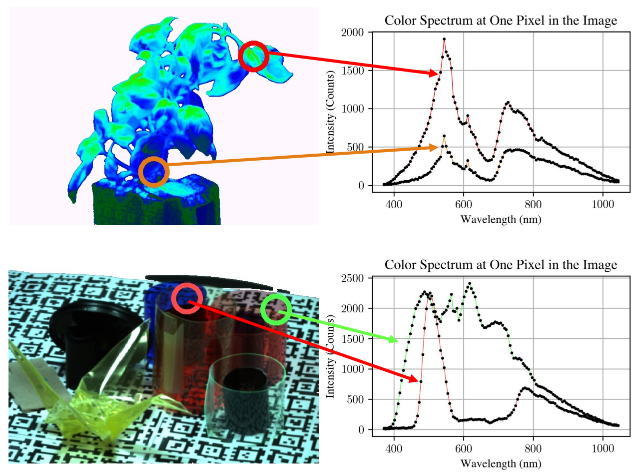

Figure 1. Instead of 3 color channels for each pixel, hyperspectral images have many color channels per pixel to measure the color spectrum for every pixel. In this work, we leverage recent advances in Neural Radiance Fields to create hyperspectral 3D scene representations.

We believe creating NeRF-based 3D reconstructions of hyperspectral data may help alleviate many issues associated with leveraging spatial hyperspectral information. Illumination angle dependence and low signal-to-noise ratio (SNR) would both be mitigated by *fusing* information from many images from different viewpoints. The radiance field representation also provides a continuous spatial interpolation, in contrast to the sparse point-cloud representations in traditional SfM or multi-view stereo (MVS) approaches. In our approach, we also show how we can use wavelength as a *continuous input* to the NeRF allowing interpolation of not only position and view angle, but also of wavelength. Finally, we believe NeRF-based approaches may be able to handle partial transparency and wavelength-dependent transparency better than SfM approaches.

Our contributions are as follows:

- Collect and share a **dataset** of hyperspectral images suitable for hyperspectral 3D reconstruction,
- Identify **special considerations** needed to accommodate hyperspectral camera limitations when computing NeRFs,
- Present novel **continuous representations for radiance and transmittance spectra** for use in **Hyper-NeRF** training and rendering, with ablations,
- **Demonstrate the feasibility** of creating hyperspectral 3D reconstructions using NeRFs, and
- Propose new **applications** of Hyper-NeRFs.

2. Related Works

Hyperspectral Imagery. We defer to the many high-quality review papers on detailed background and applications of hyperspectral imagery such as [6]. However, we briefly motivate the need for hyperspectral 3D reconstruction and discuss some practical considerations of hyperspectral cameras.

Challenges that have been identified in hyperspectral literature include the black-box nature of correlating spectra with sample properties [6], the low signal to noise ratio [12], and the high cost and inconvenience of high-resolution hyperspectral imaging. We believe that fusing multiple hyperspectral images into a 3D model can help scientists develop more mechanistic understandings of hyperspectral data and improve the signal to noise ratio. Further, we believe many recent advances surrounding NeRFs, such as NeRF in the Dark [15] and Deblur-NeRF [13], may also help extract more information with limited sensors.

Although we will not discuss the technical details, there are a few undesirable properties which are common to nearly all hyperspectral cameras. First, there is a trade-off between spatial, spectral, and temporal (exposure time) resolution such that obtaining a low-noise, high-resolution image with many wavelength bands will necessarily require a long (typically on the order of minutes) exposure time. Second, hyperspectral cameras have narrow fields of view and are incompatible with standard lenses due to the wavelength-dependent index of refraction of glass. Finally, aperture size is typically bounded due to interactions with order-blocking filters which correct diffraction side-effects. We discuss how we address these challenges for our dataset.

Hyperspectral 3D Reconstruction. Creating hyperspectral 3D reconstructions from hyperspectral images has been attempted in the past with point cloud-based methods. [25] creates separate point clouds for each wavelength channel then merges them to create a hyperspectral point cloud while [12] directly performs Structure-from-Motion on the hyperspectral data. Extending this, [14] designs custom hyperspectral keypoint feature descriptors for hyperspectral images to aid in 3D reconstruction, while several other works also address hyperspectral features for image classification [11]. However, Structure-from-Motion approaches often generate only sparse point clouds and hyperspectral imagery may often be too noisy and low resolution to obtain good multi-view stereo results. [20] takes a different approach and designs a hyperspectral structured light project device to measure 3D hyperspectral information. Somewhat similarly, [9] projects hyperspectral images onto existing 3D geometry models. However, these are not as flexible as a camera-only solution.

Neural Radiance Fields

Neural Radiance Fields (NeRFs) have *exploded* [2] in popularity since the original paper by Mildenhall et al. was published [15]. NeRFs present a deep-learning approach to obtaining a high quality 3D representation of a scene by learning a function mapping the location of a point in space and the direction from which it is being viewed to color radiance and volume density. To determine the color a pixel of an image should take, a rendering step queries the function along the pixel's corresponding image ray and composites the colors according to classical volume rendering [15]. A large body of works has since extended and improved upon the initial NeRF paper.

Although no NeRF works to our knowledge directly tackle the hyperspectral 3D reconstruction problem, we directly leverage several advancements such as the substantial efficiency improvements from Instant-NGP [16] and the open-source nerfstudio package and nerfacto implementation [18] which we build our implementation upon. We also draw inspiration from many related works. For example, several spatio-temporal [19, 3], deformable, and other NeRF works [5] append a scalar time variable to the 3D location input similar to an approach we compare against concatenating wavelength to location. Similarly, Zhi et al.'s semantic NeRF work using implicit scene representations for semantic super-resolution [24] inspires our continuous wavelength representation for hyperspectral super-resolution.

Several works could also complement our work well and we hope future research can incorporate their techniques for Hyper-NeRF. For example, RawNeRF [15] and NAN [17] both leverage NeRF's information fusing ability for low-light denoising which could help reduce the exposure time required. RawNeRF applies post-processing on the NeRF instead of the input photos, which could be applied to mitigate artifacts of hyperspectral cameras such as order-blocking filter interference. AR-NeRF [8] and Deblur-NeRF [13], which address depth of field/defocus and motion blur, respectively, could also be useful given the long exposure times and aperture limitations of hyperspectral cameras.

Hyperspectral Super-Resolution. Evidenced by numerous papers, datasets [1], and competitions [4], the hyperspectral super-resolution task has become increasingly popular. Hyperspectral super-resolution may refer to obtaining more wavelength resolution (*i.e.* use an RGB or multi-spectral image to predict a hyperspectral image), obtaining more spatial resolution (*i.e.* use a low-resolution hyperspectral image to predict a higher resolution one), or more commonly fusing together information from complementary sensors [7, 1]. Perhaps the most similar to this work is [22] which uses an implicit neural representation to predict a higher resolution image using a continuous function mapping pixel coordinate to color. We extend their work to 3D and put it in the context of NeRFs.

We are also proud to publish our 4-scene dataset; one plausible reason for the relatively greater popularity of hyperspectral super-resolution over hyperspectral 3D reconstruction is the lack of publicly available datasets for the latter.

In summary, we believe our work is highly complementary to existing works and supports a promising new direction of research in 3D hyperspectral reasoning.

3. Hyper-NeRF

We build our implementation of Hyper-NeRF upon nerfstudio's "nerfacto" implementation [18]. Figure 2 illustrates original and modified radiance + density prediction block of the network.

As compared to a $(H, W, 3)$ RGB image, a hyperspectral image can be represented by a (H, W, N) tensor, where H and W are the height and width of the image, and N is the number of channels/wavelengths.

Instead of directly predicting an N-dimensional color radiance, we choose to represent color radiance and transmittance both as continuous spectra: functions of wavelength. We do so by predicting latent vectors which represent parameters of learned spectral plots which can then be evaluated for a given wavelength.

In this section, we describe the math formulations and implementation details, though we also discuss and compare alternatives in Section 5.4.

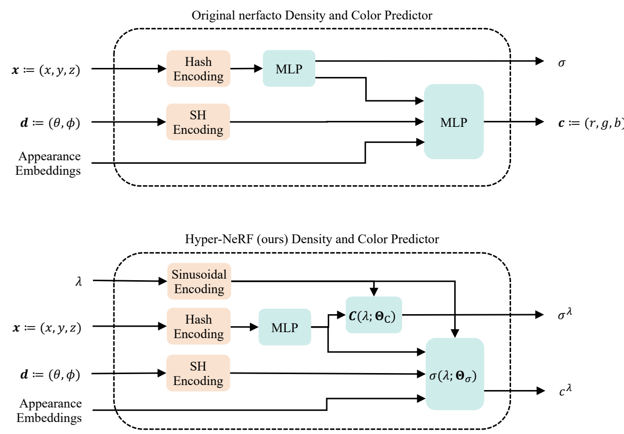

Figure 2. We base our implementation of Hyper-NeRF on nerfstudio's "nerfacto" implementation [18]. Since we do not change the overall architecture, shown above is just the differences between the networks that predict radiance and density given position, view direction, and optionally appearance embeddings and wavelength. The upper plot is the baseline nerfacto design while the bottom is our Hyper-NeRF design.

3.1. Color Radiance Spectrum Prediction

Whereas the original and nerfacto networks predict a 3-channel color vector, we choose to predict a continuous color radiance spectrum.

We predict the continuous radiance spectrum by first predicting a latent vector $\boldsymbol{\Theta}_C$ representing the parameters of a learned spectral plot. We then obtain the radiance c^λ for a given wavelength λ by passing the latent vector together with the a sinusoidal positionally encoded wavelength λ through a decoder $\boldsymbol{C}$. Formally, whereas the nerfacto (baseline) network outputs the color intensity on a ray as:

$$\boldsymbol{C}_0 : (\boldsymbol{x}, \boldsymbol{d}) \rightarrow \boldsymbol{c} := (r, g, b) \tag{1}$$

where $\boldsymbol{x} := (x, y, z)$ and $\boldsymbol{d} := (\theta, \phi)$ are the location and view direction of the ray, respectively, we predict the color radiance spectrum as:

$$\boldsymbol{C} : (\lambda; \Theta_C(\boldsymbol{x}, \boldsymbol{d})) \rightarrow c^\lambda \tag{2}$$

where $\Theta_c(\boldsymbol{x}, \boldsymbol{d}) : \mathbb{R}^5 \rightarrow \mathbb{R}^{n_\Theta}$ is a network that maps the ray's location and view direction to a latent vector $\boldsymbol{\Theta}$, and n_Θ is the dimensionality of the latent vector.

Implementation details are provided in 3.4.

3.2. Color Transmittance Spectrum Prediction

Similarly, the color transmittance spectrum describes a wavelength-dependent volume density. In other words, instead of using a scalar density field to describe the transparency of the scene, we investigate the possibility of using a wavelength-dependent density field.

Although wavelength-dependent transmittance can also be applied to RGB scenes and isn't strictly necessary for hyperspectral scenes, it is generally more interesting for hyperspectral imagery due to the fact that many materials are transparent in visible wavelengths but opaque in IR or vice-versa.

In the original and nerfacto NeRF implementations, the volume density is given by a scalar function $\sigma(\boldsymbol{x})$. Instead, we choose to model the volume density in much the same way as for color radiance: a network $\Theta_\sigma(\boldsymbol{x})$ predicts a latent vector $\boldsymbol{\Theta}_\sigma$ which is passed with the wavelength to another network

$$\sigma : (\lambda; \Theta_\sigma(\boldsymbol{x})) \rightarrow \sigma^\lambda \tag{3}$$

where σ^λ denotes the density at wavelength λ.

3.3. Wavelength-dependent Proposal Network

Finally, when choosing a wavelength-dependent volume density, it is also natural to make the sample proposal network (analagous to the "coarse" network) wavelength-dependent. However, in our ablations, we found that it wasn't necessary and caused more training instability. Nevertheless, we encourage others to try since it is likely dataset dependent.

3.4. Implementation

We find that using both $\boldsymbol{C}(\lambda; \boldsymbol{\Theta}_C)$ in place of $\boldsymbol{C}_0(\boldsymbol{x}, \boldsymbol{d})$ and $\sigma(\lambda; \boldsymbol{\Theta}_\sigma)$ in place of $\sigma(\boldsymbol{x})$ without any changes to the proposal network works well for our experiments and generalizes to arbitrary wavelength inputs, though we also present and test several other options in 5.4.

Thus, in our final implementation, to follow the typical chaining of networks, we apply the architecture shown in Figure 2. The latent vector predicted by the position MLP for the density spectrum network is also fed into the radiance spectrum network, but is concatenated with the view direction encoding and a per-image appearance embedding. The sinusoidal encoding for the wavelength uses 8 terms and the latent vector is 24 dimensional. The networks for $\boldsymbol{C}(\lambda; \boldsymbol{\Theta}_C)$ and $\sigma(\lambda; \boldsymbol{\Theta}_\sigma)$ are both identical 2-layer MLPs with 64 hidden dimensions, with the only difference being the input dimension to accomodate the radiance's additional view direction and appearance embedding inputs.

4. Dataset and Preprocessing

Before being able to train NeRF models on hyperspectral images, we first collect images using a hyperspectral camera and turntable, apply preprocessing, and obtain camera poses and intrinsics by running COLMAP on pseudo-RGB images.

Figure 3. The Surface Optics SOC710-VP camera is mounted on a tripod and the sample of interest is placed on a turnable in a Macbeth SpectraLight lightbooth. The camera is roughly 2 meters away from the scene due to its shallow depth of field and narrow field of view.

4.1. Data Collection Setup

In this work, we use the Surface Optics SOC710-VP camera, Macbeth SpectraLight lightbooth, and a custom turntable using a Dynamixel MX-106T with additional 58:9 gear reduction. The setup in shown in Figure 3. For each scene, we collect images every 15 degrees of the turntable from each of 2 camera elevations totaling 48 images per scene. We collect 4 scenes: two with plants and 2 with a collection of household objects and color filters.

4.2. Image Acquisition and Preprocessing

As mentioned in Section 2, hyperspectral cameras have inherent non-idealities that must be accounted for when collecting data. We focus our discussion on the Surface Optics SOC710-VP camera used in this work, which has high spatial (696×520 pixels) and spectral resolution ($N = 128$) at the expense of poor temporal resolution (long exposure time) and extends from 370nm to 1100nm causing some wavelength-dependent refractive index effects.

First, interactions with glass lenses and diffraction gratings necessitate careful choice of aperture size and lens. In short, the wavelength-dependent refractive index of glass (even for IR-corrected lenses) necessitates a small aperture to keep all wavelengths in focus while diffraction effects necessitate a large aperture to satisfy the criteria for the order-blocking filter commonly used in hyperspectral cameras. In response, we use a pre-calibrated 35mm lens with F5.6 aperture, and we place the camera around 2 meters from the scene to both increase the depth of field and accomodate the narrow field of view of the lens. We find that far-IR wavelengths are slightly out of focus and, although they are not particularly problematic in this work, techniques from [8, 13] may be used.

Second, the exposure time of the SOC710-VP in the

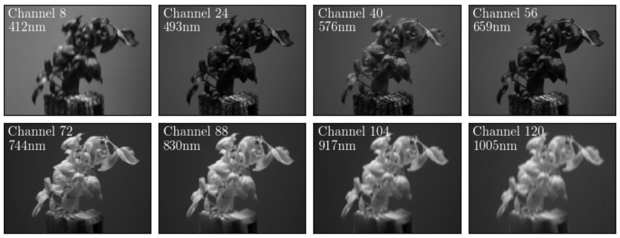

Figure 4. By visually inspecting the same image in several different wavelengths, it becomes obvious that the additional information afforded by hyperspectral imagery makes background removal significantly easier than in RGB images.

lightbooth is very long – about 2min20s per image – necessitating the use of a turntable. In turn, the image backgrounds do not rotate with the scene so they must be removed from the images. Fortunately, background removal is straightforward when leveraging hyperspectral data, as illustrated in Figure 4. We set the background color to pure white: 255 in all wavelength channels.

4.3. Computing Camera Poses and Scene Bounds

To compute the camera intrinsics and extrinsics necessary to train NeRF models, we create pseudo-RGB images to use in an off-the-shelf Structure-from-motion package. Although the turntable enforces the angular position of the camera in a "ring" around the scene, the distance, height, and orientation of the camera are unknown. We use COLMAP, a popular Structure-from-Motion package, to obtain camera poses and intrinsics. We generate pseudo-RGB images to feed into COLMAP by taking the 3 wavelength channels from the hyperspectral images which roughly correspond to red, green, and blue. Although a more accurate pseudo-RGB image could be generated, we find the narrower wavelength bands create more distinctive features and thus more reliable matching. Due in part to the narrow field of view and low resolution compared to *e.g.* smartphone cameras, we need to use an undistorted pinhole camera model (distortion parameters caused poor optimization results), have many high-quality features in the scene (which we achieve using AprilTags [10]), and apply a strict matching threshold (inlier ratio ≥ 0.70, # inliers ≥ 25).

Finally, as a byproduct of the narrow field of view, we also find it imperative to crop the ray sampler tightly to the scene to avoid sampling points that are only visible in a few cameras. Failing to do so results in "cheating" whereby the NeRF model synthesizes many 2D "screens" in front of each camera outside the field of view of the other cameras instead of a single consistent 3D object. To determine suitable ray sampling bounds, we canonicalize the camera poses according to Figure 5 and compute the "scene" bounding box, which describes the ray sampler's bounds, by projecting the cameras' fields of view onto the xz and yz planes.

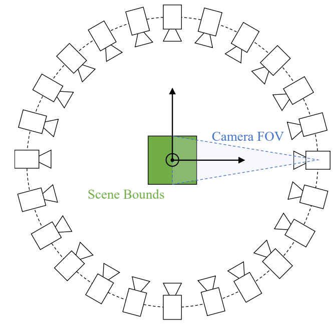

Figure 5. To tightly bound the scene to the objects of interest, we canonicalize the camera poses as shown and compute a bounding box centered at the origin whose size is determined by the camera's field of view.

4.4. Dataset Scenes

We collect a dataset of 4 scenes, with 2 of the scenes exhibiting intricate plant geometry ("Rosemary" and "Basil") and the other two exhibiting several objects with wavelength-dependent transparency and radiance/reflection ("Tools" and "Origami"). Given the hyperspectral camera's strength in measuring wavelength and comparative weakness at capturing spatial resolution, we expect the latter two scenes to be more challenging.

5. Experiments and Discussion

We train Hyper-NeRFs on the 4 scenes from our dataset and compare the results aoeu...

5.1. Evaluation Metrics

Validation Set. To form our validation set, we can sample from either images or wavelengths. Sampling from images is performed the standard way as in the NeRF literature: of the 48 images per image set, 5 are left out of the training set and used as ground truth against NeRF predictions. Sampling wavelengths is performed similarly: of the N wavelengths, we reserve $0.10 \cdot N$ for ground truth against NeRF's predictions of those wavelengths.

Metrics. As is standard in NeRF literature, we present PSNR, SSIM, and LPIPS metrics. Note that, for LPIPS, we use pseudo-RGB images extracted the same way as described in 4.3. In addition to quantitative metrics, we also provide a qualitative comparison of synthesized images.

5.2. RGB

We can first evaluate our hyperspectral approach on RGB images using stock nerfacto as a baseline. This is possible since a standard RGB image can be interpreted as an $N = 3$-channel hyperspectral image. However, because

Method	Rosemary			Basil			Tools			Origami		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
nerfacto	20.675	0.892	0.103	17.244	0.790	0.226	11.760	0.338	0.492	12.400	0.361	0.768
Ours-Cont	18.530	0.861	0.086	16.288	0.742	0.227	12.168	0.385	0.533	10.741	0.289	0.562
Ours-RGB	18.601	0.865	0.083	16.780	0.765	0.212	11.456	0.321	0.501	10.870	0.301	0.520
Hyper-NeRF (ours)	17.702	0.870	0.094	16.493	0.798	0.288	7.192	0.331	0.733	10.359	0.453	0.693

Table 1. Because our approach can handle arbitrary numbers of wavelengths, we can apply it to the RGB case (N=3 wavelengths) to compare with a baseline NeRF implementation (nerfacto). We observe that both our discrete and continuous approaches are comparable to the baseline for the easier Rosemary and Basil scenes, and outperform the baseline for the more challenging Tools and Origami scenes. "Ours-Cont" refers to the exact same implementation as Hyper-NeRF except we only train with 3 wavelengths. "Ours-RGB" refers to a slightly modified version of row 2 in the ablations (see Table 3) to output 3 discrete radiance and 3 discrete density channels. "Ours-Hyper" refers to our Hyper-NeRF implementation trained on a full, 128-channel hyperspectral image. Since our metrics are normalized to the number of wavelengths, we provide it as a reference to evidence that the hyperspectral performance on any given channel is comparable to the performance for RGB channels. We evaluate LPIPS on pseudo-RGB images extracted the same way as described in 5.2.

the wavelengths corresponding to (r, g, b) are very sparse as compared to hyperspectral images, the assumption that the color and density spectra are continuous is violated. Therefore, in addition to our approach using F_{2b}, we also present results using F_1 for both the emission and absorption networks.

As expected for the Rosemary and Basil scenes, which contain only opaque objects, our approach performs no better than the baseline. However, in the Tools and Origami scenes where there is significantly more wavelength dependent absorption and emission effects, we see that our approach slightly outperforms the baseline.

5.3. Hyper-NeRF Wavelength Generalization

Next, we evaluate our approach on hyperspectral data for our 4 image sets and withold both entire images and entire wavelengths from the training sets.

In particular, we seek to demonstrate 2 things: (1) that NeRF methods generalize well to hyperspectral data and (2) that we can suitably learn continuous representations for spectra. To do so, we train the same NeRF architecture 4 separate times: first using the full 128 wavelengths, then with only 64, 32, and 16 evenly sampled wavelengths. During evaluation, the networks must generalize to both unseen images *and* unseen wavelengths.

Table 2 and Figure 6 illustrate that the same network architecture can incorporate arbitrary wavelength supervision: increasing or decreasing the number of wavelengths used during training has a minimal effect on evaluation accuracy. From this we can deduce that *continuous* representations of radiance spectra can allow generalizing NeRF to arbitrary wavelengths.

Furthermore, the ability to interpolate between wavelengths reveals possible applications to the hyperspectral super-resolution task, very similar to [23].

5.4. Ablations

As mentioned in previous sections, there are some alternative options for how to achieve hyperspectral radiance and density predictions. We may:

1. simply output N-dimensional vectors instead of a scalar density or 3-channel color,
2. input the wavelength as another spatial dimension similar to the way time is handled in time-varying NeRFs [3, 19],
3. keep a grayscale density, and/or
4. augment the proposal networks (coarse networks) with the wavelength with the same options, or leave the proposal network as is.

We denote the options for the radiance spectrum as:

$$\begin{array}{rlll} \text{(ours)} & \boldsymbol{C} : & (\lambda\,;\,\boldsymbol{\Theta}_c(\boldsymbol{x},\boldsymbol{d})) & \rightarrow c^\lambda \\ \text{(nerfacto)} & \boldsymbol{C}_0 : & (\boldsymbol{x},\boldsymbol{d}) & \rightarrow \boldsymbol{c} := (r,g,b) \\ & \boldsymbol{C}_1 : & (\boldsymbol{x},\boldsymbol{d}) & \rightarrow (c^{\lambda_1},\dots,c^{\lambda_N}) \\ & \boldsymbol{C}_2 : & (\lambda,\boldsymbol{x},\boldsymbol{d}) & \rightarrow c^\lambda, \end{array}$$

where in $\boldsymbol{C}_2$, λ is concatenated with $\boldsymbol{x}$ before the hash encoding.

Similarly, we denote the options for the density spectrum as:

$$\begin{array}{rlll} \text{(ours)} & \sigma : & (\lambda\,;\,\boldsymbol{\Theta}_c(\boldsymbol{x})) & \rightarrow c^\lambda \\ \text{(nerfacto)} & \sigma_0 : & (\boldsymbol{x}) & \rightarrow \boldsymbol{c} := (r,g,b) \\ & \sigma_1 : & (\boldsymbol{x}) & \rightarrow (c^{\lambda_1},\dots,c^{\lambda_N}) \\ & \sigma_2 : & (\boldsymbol{x},\lambda) & \rightarrow c^\lambda. \end{array}$$

Finally, for the proposal network we only consider P_0, which denotes baseline nerfacto network, and P_λ, which denotes a proposal network augmented with the wavelength.

# of Wavelengths in Train Set	Train Set			Unseen Images			Unseen Wavelengths		Both Unseen	
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑
128	20.378	0.848	0.255	16.493	0.798	0.288	N/A	N/A	N/A	N/A
64	19.893	0.839	0.246	16.534	0.786	0.277	20.005	0.834	16.651	0.782
32	19.460	0.825	0.264	16.229	0.781	0.296	19.447	0.820	16.258	0.777
16	14.592	0.759	0.272	13.586	0.717	0.306	14.656	0.759	13.641	0.717

Table 2. Having 128 channels for each image allows us to withhold wavelengths from the training set and force the network to interpolate. The relatively small drop in performance when withholding even the vast majority of the wavelengths supports the claim that continuous radiance and transmission spectra are well suited for Hyper-NeRF.

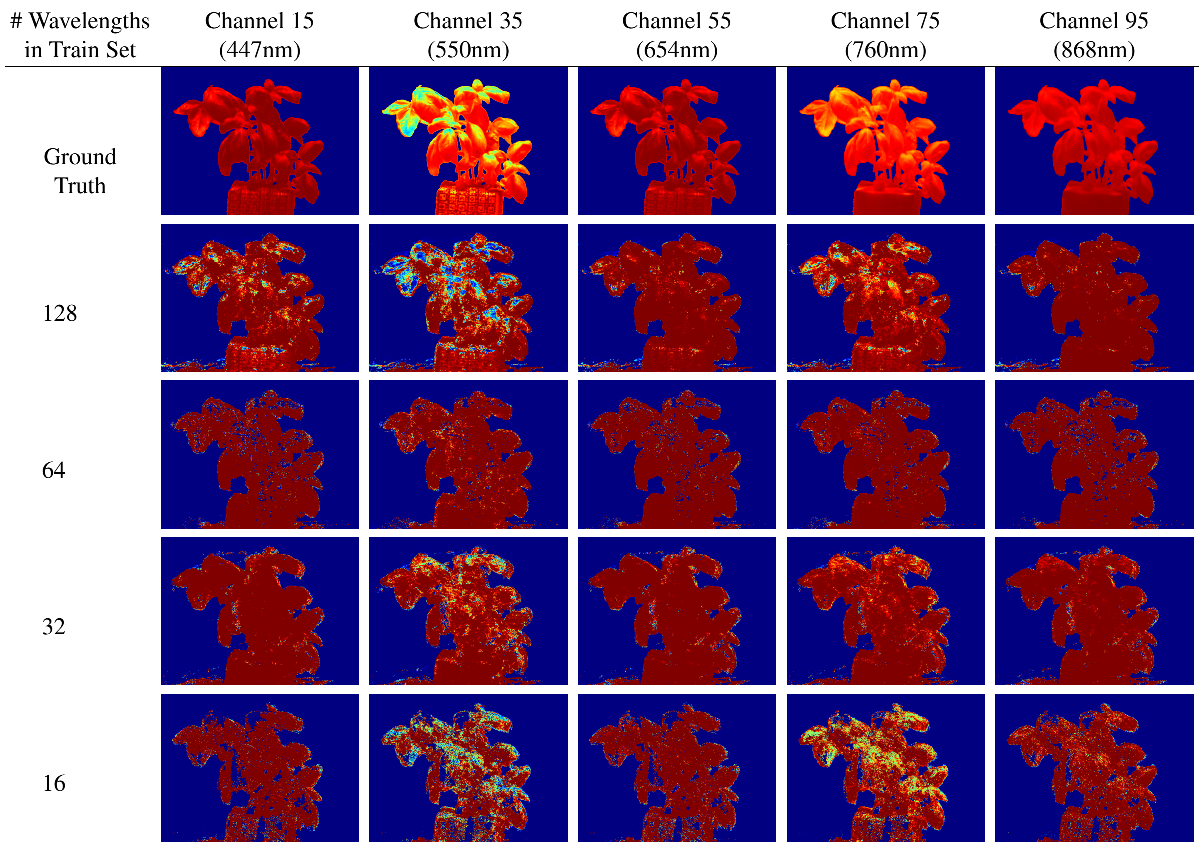

Figure 6. We visually observe the ability of Hyper-NeRF to interpolate wavelengths unseen in the training set. None of the wavelengths from this image were used in training (except 128 channel case), and none of the images used in training used these wavelengths (corresponds to Both Unseen in Table 2). We color the images using the "jet" colormap available in matplotlib and matlab for easier perception.

The ablation results are shown in Table 3 and a representative sample shown in Figure 7. All methods are comparable, with the exception of the last two rows on the Tools scene. This is believed to be a byproduct of imperfect COLMAP camera pose computation which resulted in a camera pose offset from the correct pose. Given that the LPIPS is not negatively affected and the SSIM is affected less than the PSNR, it is reasonable to expect that an image translation could produce such an error, especially given these are real datasets and not synthetic.

5.5. Additional Future Applications

The ability to represent a scene with a radiance field that is continuous not only in position and view direction but also in wavelength opens up a variety of applications which we very briefly demonstrate here.

Simulating Imaging Sensors. Typical imaging sensors in cameras (and human eyes as well) are sensitive not to a single wavelength, but to a finite band of wavelengths. The sensitivity vs wavelength plot has a significant impact on the color accuracy of the photos taken by the camera. Given a scene which has been captured using Hyper-NeRF, we can

Table 3. Ablations

Network Archictecure			Basil PSNR↑	Basil SSIM↑	Basil LPIPS↓	Tools PSNR↑	Tools SSIM↑	Tools LPIPS↓
$\boldsymbol{C}_1$	σ_0	P_0	16.493	0.819	0.317	15.335	0.588	0.553
$\boldsymbol{C}_1$	σ_1	P_0	16.358	0.809	0.332	14.375	0.431	0.709
$\boldsymbol{C}_2$	σ_2	P_0	16.219	0.777	0.303	15.708	0.642	0.568
$(ours)$ $\boldsymbol{C}$	σ	P_0	16.493	0.798	0.288	7.192	0.331	0.733
$\boldsymbol{C}$	σ	P_λ	14.637	0.710	0.283	13.221	0.399	0.663

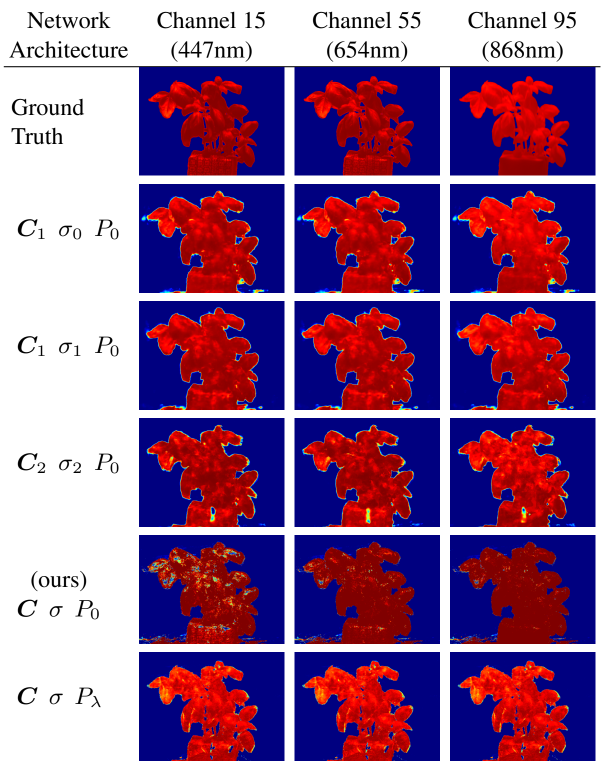

Figure 7. Comparing the different network architectures presented, we see that most methods perform similarly. Note that, unlike Figure 6, these wavelengths appear in the training sets but these images do not.

simulate the effect of different imaging sensors from different locations by integrating over both the ray *and* the wavelength during rendering.

Hyperspectral Super-Resolution. Hyperspectral super-resolution, in which we seek to (a) turn a *multi*spectral image (with fewer wavelengths than a hyperspectral image) into a hyperspectral image (with more wavelengths) or (b) turn a low-resolution hyperspectral image into a higher resolution, is an increasingly popular challenge in computer vision. Zhang et al. has already applied a similar continuous spectral network representation to 2D hyperspectral super-resolution [23], and leveraging multi-view consistency may further improve the performance of existing hyperspectral super-resolution approaches, and Section 5.3 has already demonstrated that interpolating wavelengths is possible with Hyper-NeRF.

Material and Chemical Composition Estimation. As mentioned in the introduction, hyperspectral cameras have been used to estimate the compositions of samples such as plant matter. Hyper-NeRF may enable estimating not only the composition percentages of various materials within a sample, but also their spatial distributions.

5.6. Limitations

We believe this work to be just the initial feasibility demonstration for further study regarding hyperspectral 3D reconstructions using NeRFs. As such, there are several limitations in this work ripe for future study. The relatively low resolution of hyperspectral cameras combined with the limited number of images taken (due to the long exposure time) result in a limited total number of rays with which to supervise NeRF training – incorporating recent developments from data-efficient [21], low-light [15], and motion-blurred [13] NeRF research may significantly improve results. Addressing de-focus [8] and tighter background removal are also important to obtain more accurate subject geometries.

6. Conclusions and Future Works

In this work, we showed that NeRFs can be naturally extended to hyperspectral imagery. We collected a dataset, described the special considerations needed to handle hyperspectral data, and presented a novel algorithm for creating Hyper-NeRFs that generalizes to arbitrary wavelength inputs.

We also posited on potential future applications of Hyper-NeRFs, including hyperspectral super-resolution, imaging sensor simulation, and structural material analysis.

References

[1] Naveed Akhtar, Faisal Shafait, and Ajmal Mian. Bayesian sparse representation for hyperspectral image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 3

[2] Frank Dellaert. NeRF explosion 2020, December 2020. 2

[3] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing, 2020. 2, 6

[4] Arad et al. Ntire 2018 challenge on spectral reconstruction from rgb images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1042–104209, 2018. 3

[5] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video, 2021. 2

[6] Marcin Grzybowski, Nuwan K. Wijewardane, Abbas Atefi, Yufeng Ge, and James C. Schnable. Hyperspectral reflectance-based phenotyping for quantitative genetics in crops: Progress and challenges. *Plant Communications*, 2(4):100209, 2021. Focus Issue on Crop Biology. 2

[7] Xian-Hua Han, Yinqiang Zheng, and Yen-Wei Chen. Hyperspectral image reconstruction using multi-scale fusion learning. *ACM Trans. Multimedia Comput. Commun. Appl.*, 18(1), jan 2022. 3

[8] Takuhiro Kaneko. AR-NeRF: Unsupervised learning of depth and defocus effects from natural images with aperture rendering neural radiance fields. In *CVPR*, 2022. 3, 4, 8

[9] Min H Kim, Todd Alan Harvey, David S Kittle, Holly Rushmeier, Julie Dorsey, Richard O Prum, and David J Brady. 3D imaging spectroscopy for measuring hyperspectral patterns on solid objects. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 2

[10] Maximilian Krogius, Acshi Haggenmiller, and Edwin Olson. Flexible layouts for fiducial tags. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2019. 5

[11] Brajesh Kumar, Onkar Dikshit, Ashwani Gupta, and Manoj Kumar Singh. Feature extraction for hyperspectral image classification: a review. *International Journal of Remote Sensing*, 41(16):6248–6287, 2020. 2

[12] Jie Liang, Ali Zia, Jun Zhou, and Xavier Sirault. 3d plant modelling via hyperspectral imaging. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 172–177, 2013. 2

[13] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V. Sander. Deblur-nerf: Neural radiance fields from blurry images. *arXiv preprint arXiv:2111.14292*, 2021. 2, 3, 4, 8

[14] Tengfei Ma, Yuxin Xing, Dawei Gong, Zijian Lin, Yuanpeng Li, Jiong Jiang, and Sailing He. A deep learning-based hyperspectral keypoint representation method and its application for 3d reconstruction. *IEEE Access*, 10:85266–85277, 2022. 2

[15] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. *CVPR*, 2022. 2, 8

[16] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 2

[17] Naama Pearl, Tali Treibitz, and Simon Korman. Nan: Noise-aware nerfs for burst-denoising. 2022. 2

[18] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023. 2, 3

[19] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video, 2020. 2, 6

[20] Yibo Xu, Anthony Giljum, and Kevin F Kelly. A hyperspectral projector for simultaneous 3d spatial and hyperspectral imaging via structured illumination. *Opt Express*, 28(20):29740–29755, Sep 2020. 2

[21] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 8

[22] Kaiwei Zhang. Implicit neural representation learning for hyperspectral image super-resolution, 2021. 3

[23] Kaiwei Zhang, Dandan Zhu, Xiongkuo Min, and Guangtao Zhai. Implicit neural representation learning for hyperspectral image super-resolution. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2022. 6, 8

[24] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[25] Ali Zia, Jie Liang, Jun Zhou, and Yongsheng Gao. 3d reconstruction from hyperspectral images. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 318–325, 2015. 2