Hyperspectral Neural Radiance Fields

Gerry Chen* Sunil Kumar Narayanan* Thomas Gautier Ottou* Benjamin Missaoui*
Harsh Muriki* Cédric Pradalier† Yongsheng Chen*

Abstract

Hyperspectral Imagery (HSI) has been used in many applications to non-destructively determine the material and/or chemical compositions of samples. There is growing interest in creating 3D hyperspectral reconstructions, which could provide both spatial and spectral information while also mitigating common HSI challenges such as non-Lambertian surfaces and translucent objects. However, traditional 3D reconstruction with HSI is difficult due to technological limitations of hyperspectral cameras. In recent years, Neural Radiance Fields (NeRFs) have seen widespread success in creating high quality volumetric 3D representations of scenes captured by a variety of camera models. 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 transmittance spectra. To evaluate our approach, a dataset containing nearly 2000 hyperspectral images across 7 scenes and 2 cameras was collected. We perform comparisons against traditional RGB NeRF baselines and apply ablation testing with alternative spectra representations. Finally, we demonstrate the potential of hyperspectral NeRFs for hyperspectral super-resolution and imaging sensor simulation. We show that our hyperspectral NeRF approach enables creating fast, accurate volumetric 3D hyperspectral scenes and enables several new 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, HSI 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 minimal 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.

Hyperspectral 3D reconstruction has recently been gaining interest given its unique and powerful applications [10, 15, 17, 18, 24, 27, 31]. In addition to potentially enabling estimation of material property distribution throughout a sample for *e.g.* contaminant detection and plant health monitoring, employing 3D method may also ameliorate signal-to-noise ratio (SNR) and illumination angle dependence by *fusing* information from many images. NeRF-based 3D reconstruction in particular addresses many challenges unique to hyperspectral data [13, 14, 17, 24]: better reconstructions and volumetric radiance field representation provides continuous spatial interpolation and translucency modeling, in contrast to the sparse point-cloud representations in traditional SfM or multi-view stereo approaches.

Building on prior works which represent spectral signatures with a discrete number of wavelengths, we propose using wavelength as a *continuous input* to the NeRF, allowing native interpolation of wavelength for improved flexibility and applications such as hyperspectral super-resolution.

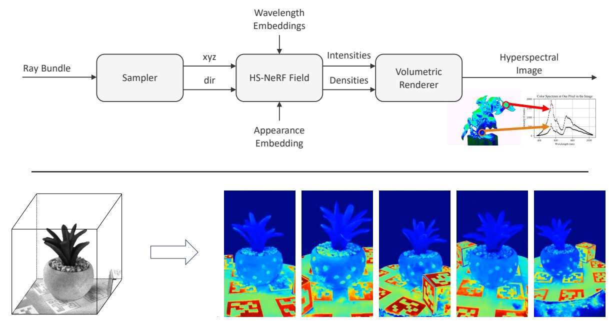

Figure 1. Instead of 3 color channels for each pixel, hyperspectral images have many color channels measuring the spectral signature for every pixel. In this work, we leverage recent advances in Neural Radiance Fields to create hyperspectral 3D scene representations. Top: Our HS-NeRF architecture with continuous wavelength representation. Bottom: Sample novel views rendered with our HS-NeRF model.

*Georgia Institute of Technology, Atlanta, GA, USA
†Georgia Tech Europe, Metz, France
This work was supported by the NSF (Award No. 2008302, ECCS-2025462, 2112533, 1936928); USDA (Award No. 2018-68011-28371); and NSF-USDA (Award No. 2020-67021-31526).
{gerry,cedric.pradalier}@gatech.edu
yongsheng.chen@ce.gatech.edu

Our contributions are as follows:

- Collect and share a **dataset** of ~2000 hyperspectral images suitable for hyperspectral 3D reconstruction,
- Identify **special considerations** needed for NeRFs to accommodate hyperspectral camera limitations,
- Introduce our **HS-NeRF model** for *H*yper*S*pectral 3D reconstruction with continuous wavelength representation,
- Validate HS-NeRF with **evaluations** and ablations, and
- Demonstrate potential **applications** of HS-NeRF.

2. Related Works

Hyperspectral Imagery.

We defer to the many high-quality review papers on HSI and its applications, such as [7]. 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 [7], the low signal to noise ratio [15], 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 [20] and Deblur-NeRF [16], may help extract more information with cheaper HSI sensors.

3D reconstruction is particularly difficult due to a few undesirable properties of HSI cameras. First, there is a tradeoff 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, lenses for hyperspectral cameras are limited in power due to the wavelength-dependent index of refraction of glass (even IR-corrected glass is not perfect), which creates more exaggerated chromatic aberration and increases the cost of optics. Many hyperspectral cameras have narrow fields of view as a result, while still suffering from inconsistent focus across wavelengths and narrow depths of field. Finally, aperture size is typically bounded due to interactions with order-blocking filters which correct diffraction side-effects, further limiting the ability to capture clear images in varying environments. We discuss how we address these challenges for our dataset and approach.

Hyperspectral 3D Reconstruction.

Creating hyperspectral 3D reconstructions from hyperspectral images has been attempted in the past with point cloud-based methods. [31] creates separate point clouds for each wavelength channel then merges them to create a hyperspectral point cloud while [15] directly performs Structure-from-Motion (SfM) on the hyperspectral data. Extending this, [18] 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 [12]. However, SfM approaches often generate only sparse point clouds and HSI may often be too noisy and low resolution to obtain good multi-view stereo results. Further, point clouds typically do not provide sufficient occupancy information to accurately compensate for shadows and lighting variations. [27] takes a different approach and designs a hyperspectral structured light project device to measure 3D hyperspectral information. Somewhat similarly, [10] projects hyperspectral images onto existing 3D geometry models. However, these are not as flexible or scalable as a camera-only solution.

Neural Radiance Fields.

Neural Radiance Fields (NeRFs) have *exploded* [4] in popularity since the original paper by Mildenhall et al. was published [20]. 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 [20].

We directly leverage several advancements in NeRF such as Instant-NGP [22] and the open-source *nerfstudio* package and nerfacto implementation [25], upon which we build our implementation. We also draw inspiration from many related works. For example, several spatio-temporal [5, 26], deformable, and other NeRF works [6] 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 [30] 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 HS-NeRF. For example, RawNeRF [20] and NAN [23] 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 [9] and Deblur-NeRF [16], 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.

NeRF-based Hyperspectral 3D Reconstruction

Recently, several works have studied NeRFs beyond RGB. [17] is most similar to our work, adapting the output of the NeRF "color" field from 3 channels to 34 to generate a hyperspectral NeRF for the application of data augmentation. [24] also uses separate output channels, but addresses aligning multiple different cameras (and modalities) to the same 3D scene via pose alignment. [13, 14] goes a step further and optimizes each camera's spectral sensitivity function (SSF) with spectral signatures parameterized as finite-dimensional vectors. We extend these works by using a *continuous* wavelength representation (rather than discrete wavelengths corresponding to vector indices, *e.g.* a 34-channel output parameterizes the spectral signature as the intensities at 34 discrete wavelengths).

Hyperspectral Super-Resolution.

Evidenced by numerous papers, datasets [1], and competitions [2], 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 [1, 8]. Perhaps the most similar to this work is [28] 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 dataset of almost 2000 hyperspectral images; 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 in 3D hyperspectral reasoning research.

3. HS-NeRF

Building on the "nerfacto" [25] implementation, we discuss 3 modifications for HS-NeRF to accomodate hyperspectral data: (1) color radiance prediction, (2) transmittance spectrum prediction, and (3) proposal network modification.

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.

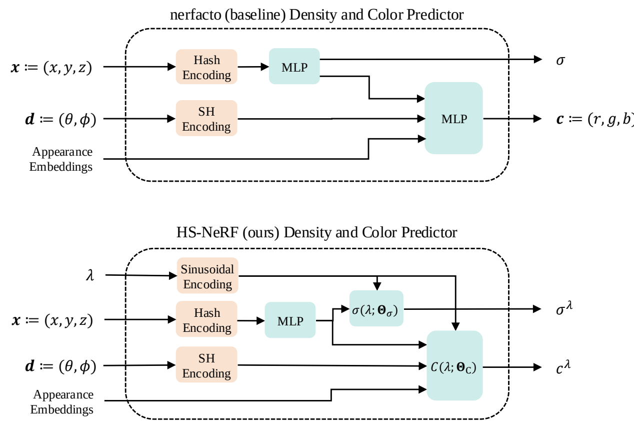

Figure 2. To handle hyperspectral data, we include a wavelength input to our network which predicts a scalar color intensity and a scalar transmittance. The network produces spectra for the color intensity and transmittance via the latent vectors $\boldsymbol{\Theta}_C$ and $\boldsymbol{\Theta}_\sigma$, respectively, and networks $\boldsymbol{C}(\lambda; \boldsymbol{\Theta}_C)$ and $\sigma(\lambda; \boldsymbol{\Theta}_\sigma)$ compute the value of the spectra at the queried wavelength.

We do so by predicting latent vectors which represent parameters of learned spectral plots which can then be evaluated for a given wavelength as shown in Fig. 2.

In this section, we describe the math formulations and some implementation details, though we also discuss and compare alternatives in Sec. 5.3. For additional details, please refer to the supplementary materials.

3.1. Color Radiance Spectrum Prediction

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:

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

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:

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

where $\Theta_c(\boldsymbol{x}, \boldsymbol{d})$ is a network that maps the ray's location and view direction to a latent vector $\boldsymbol{\Theta}_c$, which parameterizes a spectral signature decoded by the network $\boldsymbol{C}$.

3.2. Color Transmittance Spectrum Prediction

Similarly, the 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, 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 (*e.g*. glass, plastic bags).

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 $\boldsymbol{\Theta}_\sigma$ parameterizing a function of the wavelength decoded by 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 may also be natural to make the sample proposal network (analagous to the "coarse" network) wavelength-dependent. Including such a dependence may be especially useful in larger scenes with many partially transparent objects (such as plastic films). However, in ablations with our dataset, we found that doing so did not improve performance and caused more training instability. Furthermore, as we/nerfacto use the proposal loss from Mip-NERF 360 [3] which encourages the coarse loss to be an upper bound of the fine-level density network, the proposal network is penalized when it under-estimates *any* wavelength's density.

4. Dataset and Preprocessing

Before training 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.

4.1. Data Collection Setups

To demonstrate the generalizeability of our approach, we compose our dataset of scenes using two different hyperspectral cameras. The imaging setups are shown in Figs. 3 and 4, and full details on the cameras and data collection procedure are available in the supplementary materials.

4.2. Image Acquisition and Preprocessing

As mentioned in Sec. 2, hyperspectral cameras have inherent non-idealities that must be accounted for when collecting data. We focus our discussion on the two following hyperspectral cameras:

(a) The Surface Optics SOC710-VP has a 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 (blurry and misaligned in far-IR).

(b) The BaySpec GoldenEye camera also has a high spatial (640×512 pixels) and spectral ($N = 141$) resolution over the range 400 – 1100nm. Unlike the other camera, this one is of the "snapshot" type, meaning that an hypercube can be captured *almost* instantly, in around 3 seconds, but at the expense of significant grain and noise.

First, for both cameras, 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) necessi-

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 with a light gray background. The camera is roughly 2 meters away from the scene due to its shallow depth of field and narrow field of view.

Figure 4. The BaySpec GoldenEye camera is held with a laboratory clamp and the sample of interest (here, an *Anacampseros* plastic plant) is placed on a turnable under tungsten halogen lighting. The camera is roughly 20 centimeters away from the scene thanks to its wide field of view.

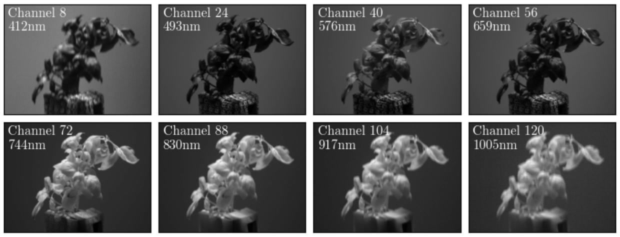

Figure 5. By visually inspecting the same image of the basil plant in several different wavelengths, it becomes obvious that the additional information afforded by HSI makes background removal significantly easier than in RGB images.

tates 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, the Surface Optics camera uses a pre-calibrated 35mm lens with F5.6 aperture, and we place it around 2 meters from the scene to both increase the depth of field and accommodate the narrow field of view of the lens. We find that far-IR wavelengths are slightly out of focus (*e.g*. Fig. 5) and, although they are not particularly problematic in this work, techniques from [9, 16] may be used. The BaySpec camera, on the other hand, has a pre-calibrated 8mm lens of 40° field of view that we use with a F16 aperture and place at just 20cm from the scene.

Second, 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 Fig. 5 where water in the paint is highly reflective in IR compared to the painted background. We set the background color to pure white: 255 (Surface Optics) or pure black: 0 (BaySpec) 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 grayscale images to use in COLMAP: an off-the-shelf SfM package. We generate the gray-scale images by selecting the channel with the greatest foreground intensity variance. Due in part to the narrow field of view (Surface Optics), low resolution, chromatic aberration (Surface Optics), and grainy noise (BaySpec) 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 [11]), and apply a strict matching threshold (inlier ratio $\geq$ 0.70, # inliers $\geq$ 25).

Finally, as a byproduct of the narrow field of view of the Surface Optics camera, 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 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 (see Fig. 9 in Supplementary).

4.4. Dataset Scenes

We collect the following datasets, depending on the camera:

(a) With the Surface Optics camera, we collect a dataset of 4 scenes of 48 images each, 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.

(b) With the BaySpec camera, we collect a dataset of 3 scenes of 433 images each, all exhibiting intricate plant geometry ("*Anacampseros*" and "*Caladium*" made of plastic, and a "Pinecone").

5. Experiments and Discussion

We train HS-NeRFs on the 7 scenes from our dataset and compare the results both quantitatively and qualitatively (see supplementary). We evaluate the reconstruction accuracy of the network compared to a nerfacto baseline and run an ablation. We also present sample applications of hyperspectral super-resolution and camera sensor simulation.

5.1. Evaluation Metrics

Validation Set

The validation set is formed the standard way as in the NeRF literature: 90% of the images in the dataset are used to train and the other 10% used for validation by comparing the actual image with the NeRF prediction.

Metrics

As is standard in NeRF literature, we present the Peak Signal to Noise Ratio (PSNR), the Structural Similarity Index Measure (SSIM), and the Learned Perceptual Image Patch Similarity (LPIPS) metrics. Note that, for LPIPS, we use pseudo-RGB images extracted as described in Sec. 6.2: as the integral of the product between RGB spectral sensitivity curves and the pixel's intensity spectrum. In addition to quantitative metrics, we also provide a qualitative comparison of synthesized images.

Table 1. RGB Results
Our HS-NeRF approach outperforms the NeRF baseline (nerfacto) for RGB images (N=3 wavelengths).

Surface Optics Datasets

Method	Rosemary			Basil			Tools			Origami		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
nerfacto	18.789	0.868	0.089	16.147	0.714	0.229	11.760	0.338	0.492	10.371	0.220	0.560
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
Ours-HS	18.327	0.886	0.083	16.548	0.664	0.172	15.591	0.575	0.489	10.359	0.453	0.693

BaySpec Datasets

Method	*Anacampseros*			*Caladium*			Pinecone		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
nerfacto	14.413	0.626	0.432	14.190	0.555	0.503	14.120	0.323	0.464
Ours-Cont	14.171	0.626	0.437	14.306	0.543	0.507	13.868	0.382	0.414
Ours-RGB	14.321	0.619	0.435	14.249	0.569	0.501	15.412	0.615	0.442
Ours-HS	20.315	0.726	0.297	19.084	0.705	0.530	20.066	0.580	0.885

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, and our approach can generalize to any number of channels. Instead of using standard RGB datasets, we test using pseudo-RGB images generated from our dataset as described in Sec. 6.2 to maintain the noise, aberration, and other effects present in the original hyperspectral images.

In addition to training our HS-NeRF network directly with 3 wavelengths ("Ours-Cont") and training a stock nerfacto model ("nerfacto"), we also consider two other comparisons. First, we train a modified version of the network in *Row 2* of the ablations to output 3 discrete radiance and 3 discrete density channels ("Ours-RGB"). Second, we provide the metrics for a HS-NeRF network trained and evaluated on the full, 128-channel hyperspectral image ("Ours-HS"), which serves to show that extending from RGB to hyperspectral does not significantly degrade performance. For Ours-HS, We evaluate LPIPS on pseudo-RGB images extracted as described in Sec. 6.2.

Our hyperspectral approach outperforms the baseline in almost every category. This is extremely pronounced for the BaySpec datasets (bottom) which have significant noise in the ground truth images. As a result, the model trained on the full hyperspectral data is more robust by virtue of the additional information given across different frames and wavelengths - similar to how camera calibration and 3D computer vision can often achieve sub-pixel accuracy.

5.3. Ablations

Instead of the HS-NeRF network described in Fig. 2, we also compare against simpler approaches to constructing hyperspectral NeRFs. We investigate the following simplifications (including one which is very similar to [17]):

1. The transmittance function is not wavelength dependent (scalar).
2. Instead of inputting the wavelength, the network simply outputs 128 channels (instead of 3 for RGB).
3. The wavelength is instead input as another spatial dimension similar to the way time is handled in time-varying NeRFs [5, 26].

We also investigate making the proposal networks wavelength-dependent as discussed in Sec. 3.3.

We denote the options for the radiance spectrum as:

$$\begin{aligned}
\text{(ours)}\quad & \boldsymbol{C} \;:(\lambda\,;\,\boldsymbol{\Theta}_c(\boldsymbol{x},\boldsymbol{d})) \rightarrow c^{\lambda} \\
& \boldsymbol{C}_1: \quad (\boldsymbol{x},\boldsymbol{d}) \quad \rightarrow (c^{\lambda_1},\dots,c^{\lambda_N}) \\
& \boldsymbol{C}_2: \quad (\lambda,\boldsymbol{x},\boldsymbol{d}) \quad \rightarrow c^{\lambda},
\end{aligned}$$

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

Similarly, we denote the options for density spectrum as:

$$\begin{aligned}
\text{(ours)}\quad & \sigma \;:(\lambda\,;\,\boldsymbol{\Theta}_\sigma(\boldsymbol{x})) \rightarrow \sigma^{\lambda} \\
& \sigma_0: \quad (\boldsymbol{x}) \quad \rightarrow \sigma \\
& \sigma_1: \quad (\boldsymbol{x}) \quad \rightarrow (\sigma^{\lambda_1},\dots,\sigma^{\lambda_N}) \\
& \sigma_2: \quad (\boldsymbol{x},\lambda) \quad \rightarrow \sigma^{\lambda}.
\end{aligned}$$

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.

The ablation results are shown in Tab. 2 and a representative sample shown in Fig. 6. We first observe that the continuous representations perform consistently well. Meanwhile, the discrete approaches excel at the noisy images from the BaySpec camera but struggle with the cleaner (but fewer) images from the Surface Optics scenes. Although not quantitatively represented, $\boldsymbol{C}_2, \sigma_2$ is significantly harder to train, frequently diverging and taking 3 times as long on average due to the fact that full passes through the network must be evaluated for every wavelength. We also observe that

Table 2. Ablations
Continuous wavelength representations perform similarly while enabling several additional applications

Surface Optics Datasets

Network Archictecure	Rosemary PSNR↑	Rosemary SSIM↑	Rosemary LPIPS↓	Basil PSNR↑	Basil SSIM↑	Basil LPIPS↓	Tools PSNR↑	Tools SSIM↑	Tools LPIPS↓
$\boldsymbol{C}_1$ σ_0 P_0	18.703	0.883	0.096	15.820	0.756	0.234	15.335	0.588	0.553
$\boldsymbol{C}_1$ σ_1 P_0	18.432	0.873	0.106	15.882	0.760	0.247	14.375	0.431	0.709
$(ours)$ $\boldsymbol{C}$ σ_0 P_0	18.327	0.886	0.083	16.548	0.664	0.172	15.591	0.575	0.489
$\boldsymbol{C}$ σ P_0	17.477	0.863	0.090	16.532	0.792	0.237	7.192	0.331	0.733
$\boldsymbol{C}_2$ σ_2 P_0	19.744	0.895	0.076	16.393	0.776	0.207	15.708	0.642	0.568
$\boldsymbol{C}$ σ P_λ	17.494	0.851	0.128	15.265	0.704	0.312	13.221	0.399	0.663

BaySpec Datasets

Network Archictecure	*Anacampseros* PSNR↑	*Anacampseros* SSIM↑	*Anacampseros* LPIPS↓	*Caladium* PSNR↑	*Caladium* SSIM↑	*Caladium* LPIPS↓	Pinecone PSNR↑	Pinecone SSIM↑	Pinecone LPIPS↓
$\boldsymbol{C}_1$ σ_0 P_0	20.279	0.734	0.295	19.084	0.705	0.530	20.102	0.729	0.399
$\boldsymbol{C}_1$ σ_1 P_0	20.128	0.729	0.305	19.933	0.678	0.495	19.857	0.726	0.372
$(ours)$ $\boldsymbol{C}$ σ_0 P_0	20.315	0.726	0.297	19.428	0.687	0.523	20.162	0.740	0.409
$\boldsymbol{C}$ σ P_0	20.095	0.705	0.320	18.942	0.653	0.514	19.596	0.729	0.409
$\boldsymbol{C}_2$ σ_2 P_0	21.259	0.711	0.319	18.989	0.595	0.568	19.732	0.561	0.434
$\boldsymbol{C}$ σ P_λ	15.031	0.667	0.351	15.930	0.626	0.556	15.700	0.485	0.484

the choice of transmittance function has little effect on the results, with Rows 1-2 matching well and Rows 3-4 matching well. Qualitatively (see supplementary), we can also observe that all methods except the wavelength-dependent proposal network perform well. Finally, we observe that our approach appears to have overall the best performance.

6. Example 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.

6.1. 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 image, 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 [29], and leveraging multi-view consistency may further improve the performance of existing hyperspectral super-resolution approaches.

We demonstrate hyperspectral super-resolution on our 8 image sets by witholding entire images and entire wavelengths from the training sets and evaluating the hyperspectral image predictions. To characterize the accuracy vs super-resolution factor, 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.

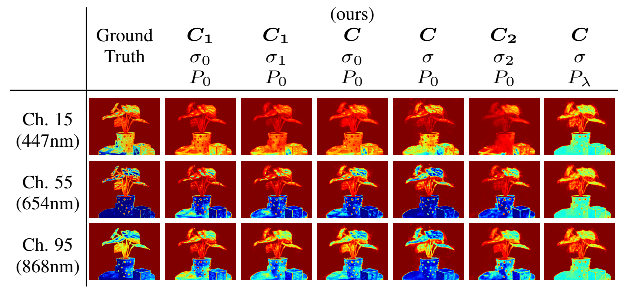

Figure 6. Comparing images (*Caladium* dataset, validation image) rendered by different networks, most methods perform similarly except the wavelength-dependent proposal network.

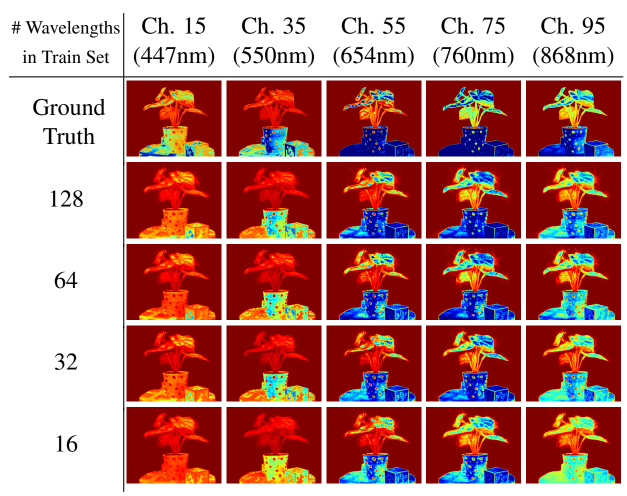

Figure 7. We visually observe the ability of HS-NeRF to interpolate wavelengths unseen in the training set. None of the wavelengths from this image were used in training (except 128 channel case), and none of the images used in training used these wavelengths (corresponds to Both Unseen in Tab. 3). We color images using the "jet" colormap from matplotlib for easier perception.

Table 3. Hyperspectral Super-Resolution

The relatively small drop in performance when withholding even the vast majority of the wavelengths from training supports the claim that continuous radiance and transmission spectra are well suited for HS-NeRF.

Basil Dataset

# 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

Anacampseros Dataset

# of Wavelengths in Train Set	Train Set			Unseen Images			Unseen Wavelengths		Both Unseen	
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑
128	20.550	0.730	0.294	20.315	0.726	0.297	N/A	N/A	N/A	N/A
64	20.947	0.735	0.461	20.701	0.732	0.459	20.979	0.736	20.732	0.733
32	20.760	0.734	0.469	20.512	0.731	0.468	20.773	0.734	20.527	0.732
16	19.868	0.727	0.477	19.705	0.725	0.474	19.854	0.728	19.690	0.726

Table 3 and Fig. 7 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 successfully allow generalizing NeRF to arbitrary wavelengths.

6.2. Simulating Imaging Sensors

HS-NeRF can also characterize *and* simulate different image sensors from a single photo of the imaged scene. Since camera image sensors each have a particular spectral response, the recorded RGB intensities are given by integrating (over all wavelengths) the product of the light spectrum reaching the sensor and the sensor's response curve. HS-NeRF can compute the light spectrum reaching the sensor at each pixel which enables both characterization and simulation.

To characterize the image sensor, we first localize the query photo's camera pose in the scene using COLMAP. We then render a hyperspectral image from the HS-NeRF. Finally, we solve for the spectral response by minimizing:

$$\left[\bar{r}(\lambda)\ \ \bar{g}(\lambda)\ \ \bar{b}(\lambda)\right] = \arg\min_{X} \sum_{i,j} (RGB_{ij} - X \cdot HS_{ij})^2 \tag{4}$$

where $X \in \mathbb{R}^{3\times128}$; $RGB_{ij} \in \mathbb{R}^3$ and $HS_{ij} \in \mathbb{R}^{128}$ denote the ij^{th} pixel of the photo and hyperspectral render, respectively; and $\bar{r}, \bar{g}, \bar{b}$ denote 128x1 vector parametrizations of the image sensor's spectral response, assuming each pixel has the same spectral response. To simulate an image sensor, we simply apply the response: $RGB_{simulated,ij} = \left[\bar{r}(\lambda)\ \ \bar{g}(\lambda)\ \ \bar{b}(\lambda)\right] \cdot HS_{ij}$. Figure 8 shows an example simulated photo alongside the real photo.

Figure 8. A demonstration of image sensor simulation (*Anacampseros* top, *Caladium* bottom). Left: A photo taken with a smartphone (a crop from Fig. 4). Right: The simulated photo computed from the HS-NeRF.

7. Conclusions and Future Works

In this work, we showed that NeRFs can naturally extend to hyperspectral imagery. We collected a dataset, described the special considerations needed to handle hyperspectral data, and presented and evaluated a novel algorithm for creating NeRFs with continuous wavelength representations.

We also demonstrated sample applications of hyperspectral NeRFs, including hyperspectral super-resolution and imaging sensor simulation. Future works include improved performance via architectural and process improvements and application of HS-NeRFs to non-destructively estimate material compositions.

References

[1] Akhtar, N., Shafait, F., Mian, A.: 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] Arad, B., et al.: NTIRE 2018 challenge on spectral reconstruction from rgb images. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1042–104209 (2018). https://doi.org/10.1109/CVPRW.2018.00138 3

[3] Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. CVPR (2022) 4

[4] Dellaert, F.: NeRF explosion 2020 (December 2020), https://dellaert.github.io/NeRF/ 2

[5] Du, Y., Zhang, Y., Yu, H.X., Tenenbaum, J.B., Wu, J.: Neural radiance flow for 4d view synthesis and video processing (2020). https://doi.org/10.48550/ARXIV.2012.09790, https://arxiv.org/abs/2012.09790 2, 6

[6] Gao, C., Saraf, A., Kopf, J., Huang, J.B.: Dynamic view synthesis from dynamic monocular video (2021). https://doi.org/10.48550/ARXIV.2105.06468, https://arxiv.org/abs/2105.06468 2

[7] Grzybowski, M., Wijewardane, N.K., Atefi, A., Ge, Y., Schnable, J.C.: Hyperspectral reflectance-based phenotyping for quantitative genetics in crops: Progress and challenges. Plant Communications **2**(4), 100209 (2021). https://doi.org/10.1016/j.xplc.2021.100209, focus Issue on Crop Biology 2

[8] Han, X.H., Zheng, Y., Chen, Y.W.: Hyperspectral image reconstruction using multi-scale fusion learning. ACM Trans. Multimedia Comput. Commun. Appl. **18**(1) (jan 2022). https://doi.org/10.1145/3477396, https://doi.org/10.1145/3477396 3

[9] Kaneko, T.: AR-NeRF: Unsupervised learning of depth and defocus effects from natural images with aperture rendering neural radiance fields. In: CVPR (2022) 2, 5

[10] Kim, M.H., Harvey, T.A., Kittle, D.S., Rushmeier, H., Dorsey, J., Prum, R.O., Brady, D.J.: 3D imaging spectroscopy for measuring hyperspectral patterns on solid objects. ACM Transactions on Graphics (TOG) **31**(4), 1–11 (2012) 1, 2

[11] Krogius, M., Haggenmiller, A., Olson, E.: Flexible layouts for fiducial tags. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (October 2019) 5

[12] Kumar, B., Dikshit, O., Gupta, A., Singh, M.K.: Feature extraction for hyperspectral image classification: a review. International Journal of Remote Sensing **41**(16), 6248–6287 (2020). https://doi.org/10.1080/01431161.2020.1736732 2

[13] Li, J., Li, Y., Sun, C., Wang, C., Xiang, J.: Spec-NeRF: Multi-spectral neural radiance fields. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2485–2489 (2024). https://doi.org/10.1109/ICASSP48485.2024.10446015 1, 3

[14] Li, R., Liu, J., Liu, G., Zhang, S., Zeng, B., Liu, S.: SpectralNeRF: Physically based spectral rendering with neural radiance field. Proceedings of the AAAI Conference on Artificial Intelligence **38**(4), 3154–3162 (Mar 2024). https://doi.org/10.1609/aaai.v38i4.28099 1, 3

[15] Liang, J., Zia, A., Zhou, J., Sirault, X.: 3d plant modelling via hyperspectral imaging. In: 2013 IEEE International Conference on Computer Vision Workshops. pp. 172–177 (2013). https://doi.org/10.1109/ICCVW.2013.29 1, 2

[16] Ma, L., Li, X., Liao, J., Zhang, Q., Wang, X., Wang, J., Sander, P.V.: Deblur-NeRF: Neural radiance fields from blurry images. arXiv preprint arXiv:2111.14292 (2021) 2, 5

[17] Ma, R., Ma, T., Guo, D., He, S.: Novel view synthesis and dataset augmentation for hyperspectral data using nerf. IEEE Access **12**, 45331–45341 (2024). https://doi.org/10.1109/ACCESS.2024.3381531 1, 3, 6

[18] Ma, T., Xing, Y., Gong, D., Lin, Z., Li, Y., Jiang, J., He, S.: A deep learning-based hyperspectral keypoint representation method and its application for 3d reconstruction. IEEE Access **10**, 85266–85277 (2022). https://doi.org/10.1109/ACCESS.2022.3197183 1, 2

[19] Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: CVPR (2021) 1

[20] Mildenhall, B., Hedman, P., Martin-Brualla, R., Srinivasan, P.P., Barron, J.T.: NeRF in the dark: High dynamic range view synthesis from noisy raw images. CVPR (2022) 2

[21] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 1

[22] Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15

(Jul 2022). https://doi.org/10.1145/3528223.3530127 2

[23] Pearl, N., Treibitz, T., Korman, S.: NAN: Noise-aware NeRFs for burst-denoising. In: CVPR (2022) 2

[24] Poggi, M., Zama Ramirez, P., Tosi, F., Salti, S., Di Stefano, L., Mattoccia, S.: Cross-spectral neural radiance fields. In: Proceedings of the International Conference on 3D Vision (2022), 3DV 1, 3

[25] Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. arXiv preprint arXiv:2302.04264 (2023) 2, 3

[26] Xian, W., Huang, J.B., Kopf, J., Kim, C.: Space-time neural irradiance fields for free-viewpoint video (2020). https://doi.org/10.48550/ARXIV.2011.12950, https://arxiv.org/abs/2011.12950 2, 6

[27] Xu, Y., Giljum, A., Kelly, K.F.: A hyperspectral projector for simultaneous 3d spatial and hyperspectral imaging via structured illumination. Opt Express **28**(20), 29740–29755 (Sep 2020). https://doi.org/10.1364/OE.402812 1, 2

[28] Zhang, K.: Implicit neural representation learning for hyperspectral image super-resolution (2021). https://doi.org/10.48550/ARXIV.2112.10541, https://arxiv.org/abs/2112.10541 3

[29] Zhang, K., Zhu, D., Min, X., Zhai, G.: Implicit neural representation learning for hyperspectral image super-resolution. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6 (2022). https://doi.org/10.1109/ICME52920.2022.9859739 7

[30] Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.: In-place scene labelling and understanding with implicit scene representation. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021) 2

[31] Zia, A., Liang, J., Zhou, J., Gao, Y.: 3d reconstruction from hyperspectral images. In: 2015 IEEE Winter Conference on Applications of Computer Vision. pp. 318–325 (2015). https://doi.org/10.1109/WACV.2015.49 1, 2

Hyperspectral Neural Radiance Fields

Supplementary Material

Qualitative Results Website

Please also refer to `https://hyperspectral-nerf.github.io/supplemental-results-webpage` for qualitative results.

A. Introduction to Supplementary Material

In this work, we demonstrated that Neural Radiance Fields (NeRFs) can be naturally extended to hyperspectral data and are a well-suited tool for hyperspectral 3D reconstruction. The implementation details provided in this supplemental document describe our simple approach to hyperspectral NeRF, but we anticipate future works by the community will improve upon our baseline implementation using our to-be-published dataset, future larger datasets, additional architecture and hyperparameter tuning, and recent advances in NeRFs.

Our full code and dataset will be made publicly available for the camera ready version.

B. Camera Pose Canonicalization

To tightly bound the scene to the objects of interest, we canonicalize the camera poses as shown in Fig. 9 and compute a bounding box centered at the origin whose size is determined by the camera's field of view. This improves training stability and performance.

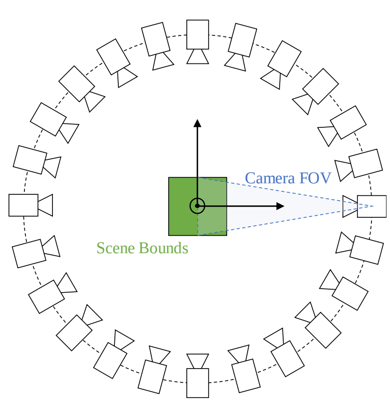

Figure 9. To tightly bound the scene to the objects of interest, we canonicalize the camera poses as shown and compute a bounding box centered at the origin whose size is determined by the camera's field of view.

C. Implementation Details

We build upon nerfstudio's nerfacto implementation, from commit ef9e00e. Our code will be made publicly available for the camera ready paper. The original nerfacto pipeline and field are shown in Figs. 10 and 11 respectively.

As briefly summarized in the main paper, we make relatively minimal modifications to the pipeline and field. Using the notation from Section 5.4: Ablations, $\boldsymbol{C}_1$ only changes the rightmost MLP in Fig. 11 to output 128 channels in the last layer instead of 3; $\boldsymbol{C}_2$ changes the positional hash encoding (ϕ in Fig. 11) to take 4 inputs instead of 3 (appending λ) and changes the rightmost MLP to only have 1 output for c instead of (r, g, b); and $\boldsymbol{C}$ is shown in Fig. 2 (bottom) of the main paper. For $\boldsymbol{C}$, the sinusoidal encoding for λ is taken to have 8 terms (tested 2, 4, 8, 16 terms, with 8 performing marginally better than 4 and 16, and 2 significantly worse). Also for $\boldsymbol{C}$, the component $\boldsymbol{C}(\lambda; \Theta_C)$ MLP from Fig. 2 of the main paper was taken to be identical to the rightmost MLP in Fig. 11 except with the appropriate additional number of inputs to accommodate concatenating the sinusoidally encoded wavelength, and with only 1 output for c instead of 3 for (r, g, b). The latent vector Θ_C was taken to be the same size as in the nerfacto implementation (15-dim), with increasing the size to 32 and 64 showing

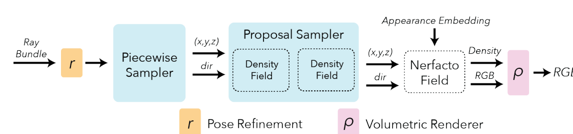

Figure 10. The original nerfacto pipeline (from nerfstudio docs) contains a proposal sampler, which is analagous to the "coarse" field from the original NeRF paper [21], and a "Nerfacto Field", which is analagous to the primary network from the original NeRF paper (F_Θ).

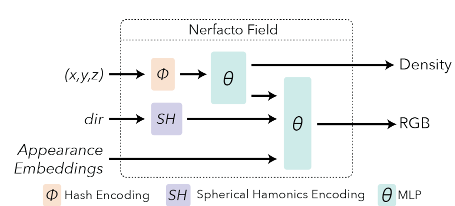

Figure 11. The original nerfacto field (from nerfstudio docs) is very similar to the original NeRF paper [21], but includes appearance embeddings [19] and uses slightly different encodings for the position and direction. This figure is reproduced in Fig. 2 of our main paper.

negligible performance improvement but increased training instability.

Similarly, σ_0 is the stock nerfacto field (scalar); σ_1 only changes the left MLP in Fig. 11 to have 128 outputs; σ_2 changes the positional hash encoding to take 4 inputs, and σ is as shown in Fig. 2 (bottom) of the main paper. The additional component $\sigma(\lambda; \Theta_C)$ MLP has 3 layers with 64-dim hidden layers and ReLU activations. The sinusoidally encoded λ is shared with $\boldsymbol{C}$ and the latent Θ_σ vector is shared with (identical to) the Θ_C vector.

Finally, P_0 is the stock nerfacto proposal network while P_λ augments the proposal network with the wavelength. For P_λ, the position is first run through a hash encoding and MLP as in P_0, except the MLP outputs a latent vector of dimension 7 instead of a scalar density. This latent vector is concatenated with a 2-term sinusoidally encoded wavelength and fed through a 2-layer network with 7-dim hidden layer to output a scalar density for inverse transform ray sampling. Like the original nerfacto pipeline, this sampling step occurs twice with identical architecture (but different weights) proposal networks.

Reiterating our implementation, our primary HS-NeRF implementation uses $\boldsymbol{C}(\lambda; \Theta_C)$, $\sigma_0(\lambda; \Theta_\sigma)$, and P_0, which we find to produce good results while also enabling wavelength interpolation.

C.1. RGB Implementations

Pseudo-RGB wavelengths. For the purposes of generating pseudo-RGB images, on the Surface Optics datasets we use the wavelengths 622nm, 555nm, and 503nm for R, G, and B channels respectively.

For the BaySpec datasets, we use a slightly more involved approach. We found that the BaySpec datasets were more sensitive to noise saturation and white balance, so we use an approach similar to that described in Section 6.2 of the main paper to generate pseudo-RGB images. Specifically, we first manually identify 5-10 point correspondences between a hyperspectral image and an iPhone photo of the same scene to represent pairs of colors that should be the same. Expressing the n points in the hyperspectral image as $X \in \mathbb{R}^{128\times n}$ and in the iPhone photo as $Y \in \mathbb{R}^{3\times n}$, we solve for a linear transformation $A \in \mathbb{R}^{3\times 128} = \arg\min_{A'} \|Y - A'X\|^2$ using the least squares solution. We then use this transformation to convert the hyperspectral image to pseudo-RGB. After using this initial approach to boot-strap certain components of the pipeline, we later apply the method described in Section 6.2 to generate pseudo-RGB renderings.

HS-NeRF RGB variation implementations. For the purposes of making a quantitative comparison to standard RGB NeRF, Section 5.2 and Table 1 of the main paper present variations of our approach applied to just 3-channel (RGB) images instead of the full 128-channel hyperspectral data. As described in the caption of Table 1, "Ours-Cont" refers to our HS-NeRF implementation but trained on only 3 wavelengths (so we maintain a continuous representation for radiance spectra, but have very weak supervision of only 3 channels), "Ours-RGB" refers to $\boldsymbol{C}_1, \sigma_1, P_0$ with 3 output channels for both $\boldsymbol{C}_1$ and σ_1, and "Ours-Hyper" refers to our HS-NeRF implementation trained on all 128 wavelengths. In the table for Ours-Hyper, PSNR and SSIM are evaluated over all 128 wavelengths while LPIPS is evaluated on the RGB images obtained using the Pseudo-RGB procedure.

D. Training Details

All networks were trained for 25000 steps, with 4096 train rays per step using the Adam optimizer. The proposal networks and field both used lr=1e-2, eps=1e-15, and an exponential decay lr schedule to 1e-4 after 20000 steps. Camera extrinsic and intrinsic optimization were both turned off, since evaluation metrics are skewed if camera parameters are modified. To accommodate imperfect camera poses, after COLMAP, stock nerfacto was run on Pseudo-RGB images for 100000 steps with camera optimization turned on and the resulting camera pose corrections were saved and used in subsequent tests. The Surface Optics datasets took roughly 20 minutes to train HS-NeRF while the BaySpec datasets took roughly 40 minutes to train on an RTX 3090 due to the need to re-cache a new set of 32 images every 50 steps (see next paragraph). Most architectures required similar training times, with the exception of the last two rows of the ablation: $\boldsymbol{C}_2\sigma_2P_0$ and $\boldsymbol{C}\sigma P_\lambda$ took at roughly three times as long.

For the Surface Optics datasets, of the 48 images per image set, 43 were used for training and 5 withheld for evaluation. Each step, the 4096 training rays were sampled randomly from all 43 training images, except for row 6 of the ablations where the training rays were sampled from only 10 of the 43 training images each step, with the choice of 10 images being re-sampled every 50 steps. The BaySpec datasets were too large to fit in VRAM so rays were sampled from 32 images every step, with the set of 32 images being re-sampled every 50 steps, with row 6 of the ablations being reduced to 12 images resampled every 50 steps.

In some approaches, not all wavelengths could be run for every step due to VRAM limits so a subset of wavelengths were sampled (randomly) for each step, but every sampled wavelength was run for every ray in the step. For rows 1 and 2 of the ablations, every wavelength could be run every step. For rows 3, 4 (HS-NeRF, ours), and 5, the number of wavelengths sampled per step were 8, 12, and 6, respectively.

For evaluation, every wavelength of every pixel of the 5 (Surface Optics) or 35 (BaySpec) evaluation images were

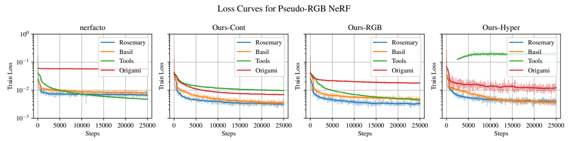

Figure 12. Loss curves for RGB NeRF correspond to the metrics from Table 1 in the main paper. Most scenes have converged by 25000 steps except the Tools scene which appears to have difficulty converging for all methods except "Ours-Cont"

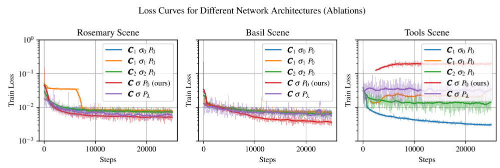

Figure 13. Loss curves for ablation testing (analagous to Table 3 in the main paper) shows that while the rosemary and basil scenes optimize well, the tools scene does not converge particularly well for any method, re-emphasizing the suspected pre-processing (COLMAP) inaccuracy.

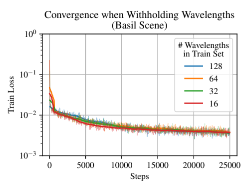

Figure 14. Loss curves for HS-NeRF trained with a subset of wavelengths (analogous to Table 2 in the main paper) shows that even training with only 1 out of every 8 wavelengths still has almost identical convergence rate w.r.t. number of steps.

evaluated and compared for each scene.

D.1. Commentary on the Tools Scene

The Tools scene experienced instabilities during training with several approaches including both HS-NeRF (ours) and nerfacto (RGB baseline). We anticipate that obtaining better camera intrinsics and extrinsics may correct this issue, since (a) every method had difficulty on this scene and (b) enabling camera pose optimization during NeRF training improved convergence for all methods. Better camera intrinsics could be obtained by initializing COLMAP with the intrinsics obtained from other scenes, and better camera extrinsics could be obtained through a combination of tuning COLMAP parameters, utilizing turntable priors, and a longer NeRF-based camera pose refinement as described in Appendix D. The poor convergence on the Tools scene for all methods is illustrated in both Fig. 12 (green curves) and Fig. 13.

D.2. Loss Curves

To demonstrate that all methods were fairly trained until convergence, the loss curves corresponding to some metrics given in the main paper are shown. As mentioned, the Tools scene appears to have difficulty converging for all methods including baseline nerfacto, suggesting possible pre-processing (COLMAP) inaccuracy. This is evident both in the green curves of Fig. 12 and in the rightmost plot of Fig. 13. Evidencing the hyperspectral super-resolution (spectral interpolation) application, Fig. 14 shows almost identical training loss for all subsets of wavelengths trained with.

E. Qualitative Example Results

A selection of example images and videos with brief explanations are provided at https://hyperspectral-nerf.github.io/supplemental-results-webpage to better gauge our results qualitatively.