SpectralSplatsViewer
An Interactive Web-Based Tool for Visualizing Cross-Spectral Gaussian Splats

Sinha, Saptarshi Neil; Kühn, Julius; Graf, Holger; Weinmann, Michael

DOI
10.1145/3665318.3677151

Publication date
2024

Document Version
Final published version

Published in
Proceedings - Web3D 2024

SpectralSplatsViewer: An Interactive Web-Based Tool for Visualizing Cross-Spectral Gaussian Splats

Saptarshi Neil Sinha
Fraunhofer IGD
Darmstadt, Hessen, Germany
saptarshi.neil.sinha@igd.fraunhofer.de

Julius Kühn
Fraunhofer IGD
Darmstadt, Hessen, Germany
julius.kuehn@igd.fraunhofer.de

Holger Graf
Fraunhofer IGD
Darmstadt, Hessen, Germany
holger.graf@igd.fraunhofer.de

Michael Weinmann
Delft University of Technology
Delft, Zuid-Holland, Netherlands
M.Weinmann@tudelft.nl

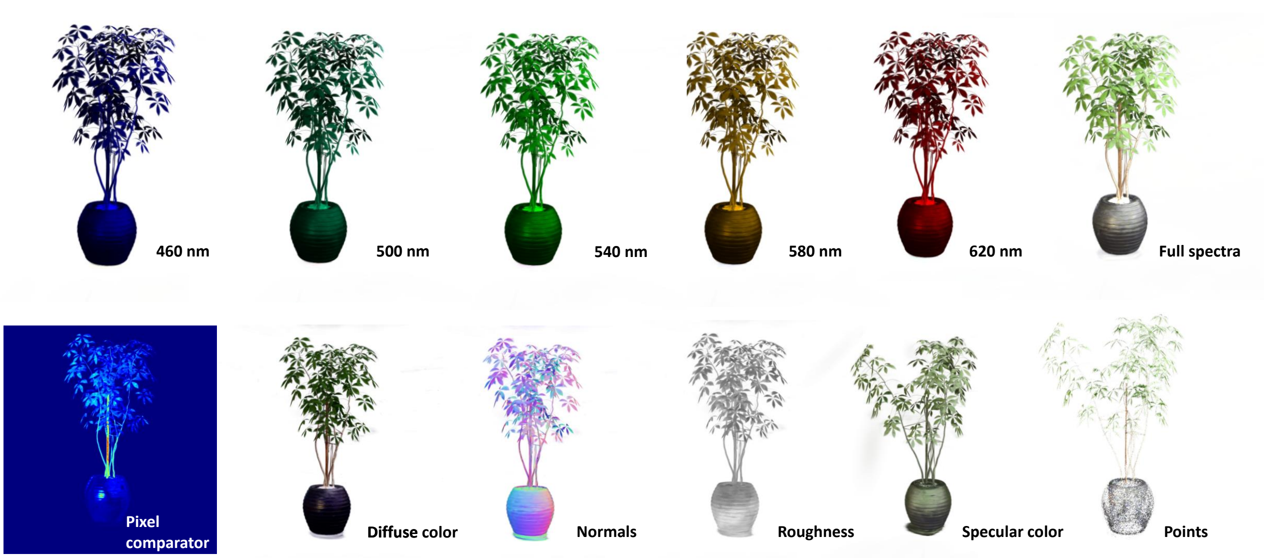


Figure 1: SpectralSpaltsViewer: Visualization of Spectral renderings, reflectance parameters and pixel comparator mode of the Ficus scene

ABSTRACT

Spectral rendering accurately simulates light-material interactions by considering the entire light spectrum, unlike traditional rendering methods that use limited color channels like RGB. This technique is particularly valuable in industries to assess visual quality before production. Moreover, Spectral imaging finds extensive applications in fields like agriculture for plant disease detection, cultural heritage for preservation, forensic science, environment monitoring and medical science among others. Advances in generating novel views from images have been achieved through methods like NERF and Gaussian splatting, which outperforms others in terms of quality. This paper introduces a web-based viewer built on the Viser framework for visualizing and comparing cross-spectral Gaussian splats from different views and during various training stages. This viewer supports real-time collaboration and comprehensive visual comparison, enhancing user experience in spectral data analysis. We conduct a user study and performance analysis to confirm its effectiveness and usability for different application scenarios, while also proposing potential enhancements for increased functionality.

WEB3D '24, September 25–27, 2024, Guimarães, Portugal

ACM ISBN 979-8-4007-0689-9/24/09
https://doi.org/10.1145/3665318.3677151

CCS CONCEPTS

• **Computing methodologies** → **Rendering**; **Artificial intelligence**; **Graphics systems and interfaces**; **Hyperspectral imaging**; • **Human-centered computing** → **Visualization systems and tools**.

KEYWORDS

Spectral Rendering, 3D graphics on the web, Gaussian splatting

ACM Reference Format:

Saptarshi Neil Sinha, Julius Kühn, Holger Graf, and Michael Weinmann. 2024. SpectralSplatsViewer: An Interactive Web-Based Tool for Visualizing Cross-Spectral Gaussian Splats. In *The 29th International ACM Conference on 3D Web Technology (WEB3D '24), September 25–27, 2024, Guimarães, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3665318.3677151

1 INTRODUCTION

The accurate visualization of 3D scenes is of great importance for numerous applications. Conventional tristimulus rendering computes light transport using only three color channels where the color space (e.g., RGB, HSV) is arbitrarily chosen based on application needs or desired outcome. However, this does not allow accurate color reproduction of the scene due to metamerism effects, where objects with different spectral reflectance properties can appear the same under specific lighting conditions, leading to color ambiguities. Overcoming this requires the consideration of physically accurate light exchange with materials and surfaces in the scene by considering the full spectrum of light. This is particularly relevant for domains like architecture, automotive industries, advertisement and design, where the virtual prototyping relies on predictive rendering - which involves the spectral simulation of light transport - to ensure a reliable assessment and evaluation of the visual quality of products prior to their physical production due to a color-correct scene reproduction. Furthermore, various scenarios rely on application-specific data captured based on different sensors, such as multi-spectral (MS) cameras [Micasense 4 24; Silios 4 24], infrared (IR) cameras [JENOPTIK 2024] or UV sensors [Lanteri and Pelosi 2021], and the respective processing, analysis and visualization of the respectively captured cross-spectral data. By analyzing correlations between spectra, a unified cross-spectral scene representation can be obtained, enabling comprehensive querying of information sensed across different spectra and leading to a deeper understanding of natural processes [Poggi et al. 2022]. Respective applications include smart/precision farming, where multi-spectral data facilitates early detection and monitoring of harmful algal bloom [Do Hyuck Kwon and Cho 2023] as well as the detection and classification of plant diseases [Moghadam et al. 2017]. Additionally, spectral imaging is utilized in automatic greenhouse surveillance [Yang et al. 2014]. Regarding the preservation of cultural heritage [Alfeld et al. 2018; Grillini et al. 2024; Landi and Maino 2011], multi-spectral information reveals important insights about materials used for various objects or paintings and, hence, regarding the production process and the available or preferred materials at the time of production as well as *restoring* eroded parts based on information in individual spectral bands that may exceed the visible range. In document analysis, the utilization of multispectral or hyperspectral information has been shown to be effective in unveiling hidden or altered features within documents [Qureshi et al. 2019]. Similarly, in the field of face recognition, the incorporation of spectral imaging techniques has been found to enhance the accuracy and robustness of facial recognition systems [Vetrekar et al. 2016]. Moreover, the application of spectral rendering holds significant importance in various domains such as medical sciences, forensic sciences, environment monitoring, and remote sensing [Zahra et al. 2024]. This is primarily due to its ability to enable more accurate analysis of different materials present in the scene. In this paper, we put our attention to efficient GPU-based spectral rendering. In contrast to previous spectral rendering frameworks like ART[The ART development team 2018], Mitsuba 2[Nimier-David et al. 2019], PBRT v4[Pharr 2020], Malia[Dufay et al. 2019] and Manuka[Fascione et al. 2018], that are limited to conventional scene representations in terms of point clouds or meshes, we propose a novel cross-spectral visualization framework based on a 3D Gaussian Splatting [Kerbl et al. 2023] representation derived from multi-view cross-spectral images or spectrum maps. Together with recent implicit scene representation in terms of Neural Radiance Fields (NERFs) [Mildenhall et al. 2020], explicit scene representation in terms of 3D Gaussian Splatting has been demonstrated to allow producing visually realistic 3D reconstructions from multi-view images. However, regarding NeRFs, the involvement of a scene representation in terms of a neural network with weights optimized according to the information in the given photographs and camera poses complicates the direct access to the scene and requires the combination of dense volumetric sampling and forward passes through the network to obtain local characteristics of the 3D scene. Circumventing the need for a neural network based scene representation with techniques like Gaussian splatting therefore improves the efficiency of scene representation and rendering, while additionally offering a scene representation that is more interpretable in terms where the scene is represented well or not and offering superior performance and quality compared to NERFs. While this motivates the use of a scene representation in terms of Gaussian splatting, the respective extension to multi-spectral scene representation and respective spectral visualization tools remains an open challenge. To visualize image based multi-spectral data in 3D, allowing enhanced material discrimination, 3D segmentation, and increased robustness to lighting conditions (i.e extract features that are less influenced by changes in illumination), this paper offers the following major contributions:

- We present a web-based viewer, inspired by the Viser framework [Nerfstudio Project 2023], that allows visualizing cross-spectral Gaussian splats. Our viewer supports the visualization of spectral splats at various stages of training, allowing for per-spectrum visualization and the comparison of splats. Thereby, our tool provides a convenient and intuitive interface for analyzing and comparing the results of spectral splatting.
- Our viewer supports the visualization of different texture maps utilized by the Gaussian splatting renderer for generating the final output. This includes texture maps for BRDF parameters, depth maps, and normal maps. Additionally, our viewer incorporates a version of Gaussian splatting [Kerbl et al. 2023] that enables the segmentation of the splats and, hence, allows the visualization of semantic object IDs. Furthermore, our viewer supports the visual comparison of these texture maps at different stages of the training process, providing comprehensive insights on how the rendering progresses over time.

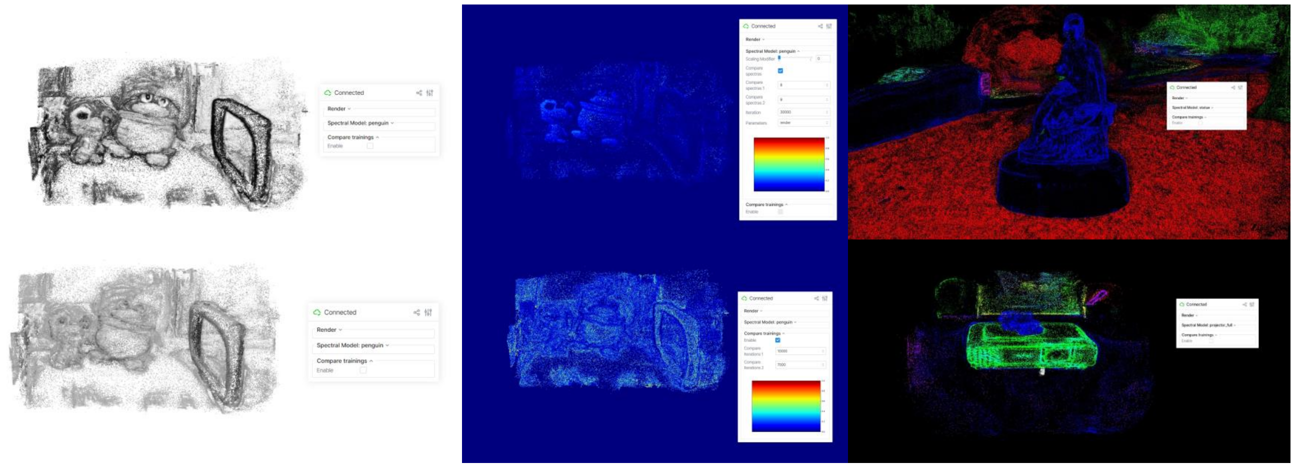


Figure 2: Point renderings of different spectra (left), pixel comparator (for different spectra and different iterations) (middle) and segmentation-maps (right)

- Our viewer allows users to engage in real-time collaboration by utilizing the spectral and output selection controls, enabling multiple users to collectively choose and visualize specific spectra, switch between different types of output visualizations, and compare them collaboratively.
- We provide an evaluation of our viewer in terms of a performance analysis as well as a user study of various aspects including ease of navigation, intuitiveness of the user interface, responsiveness, understandability, adequacy of information, future usability, likability, and overall experience. Furthermore, we involved domain experts to get insights on desirable features (e.g., Geometry exporter, material editing, collaborative view synchronization) that would have the highest impact.

2 RELATED WORK

2.1 Spectral renderers

CPU-based spectral rendering engines such as ART[The ART development team 2018], PBRT v3[Pharr et al. 2016], and Mitsuba[Jakob 2010] are commonly used by the scientific community and others like Manuka[Fascione et al. 2018] are also utilized in the industry, although they are not as prevalent. Most rendering engines that leverage the capabilities of graphical processing units (GPUs) are typically RGB-based renderers[Murray et al. 2021]. However, there is a growing number of academic spectral renderers that now take advantage of GPU acceleration, such as Mitsuba 2[Nimier-David et al. 2019], PBRT v4[Pharr 2020], and Malia[Dufay et al. 2019]. These GPU-based spectral renderers represent an emerging trend in the field and would play an important role in simulating real world spectral data.

2.2 Scene Representation based on NeRFs and Gaussian Splatting

In recent years, great improvements for generating photo-realistic novel views have been achieved based on novel learning-based scene representations in combination with volume rendering techniques. Neural Radiance Fields (NeRF)[Mildenhall et al. 2020] represent the scene based on a neural network, that predicts local density and view-dependent color for local points in the scene volume which, in turn, can be used to synthesize images of the scene based on volume rendering. By optimizing the predicted images to match the given input images under the respective view conditions, the network is trained to represent the scene. The great potential of this approach for novel view synthesis has led to numerous extensions, including the improvement of rendering quality by reducing liasing [Barron et al. 2021, 2022, 2023; Wang et al. 2022], accelerating the training of the underlying network [Chen et al. 2022a; Fridovich-Keil et al. 2022; Müller et al. 2022; Yariv et al. 2023] as well as handling more complex inputs in terms of unconstrained image collections [Chen et al. 2022b; Jun-Seong et al. 2022; Martin-Brualla et al. 2021].

Instead of relying on a neural-network-based scene representation, several works focused on representing scenes in terms of implicit surfaces [Ge et al. 2023; Wang et al. 2021, 2023] explicit representations based on points [Xu et al. 2022], meshes [Munkberg et al. 2022] or 3D Gaussians [Kerbl et al. 2023]. Point-based neural rendering techniques like Point-NeRF [Xu et al. 2022] combine accurate view synthesis based on NeRF with the rapid scene reconstruction capabilities of deep multi-view stereo methods, utilizing neural 3D point clouds for efficient rendering, thereby allowing to accelerate the training process. A recent approach [Zhang et al. 2023] additionally demonstrated that point-based methods are more suitable for scene editing. Recently, 3D Gaussian Splatting [Kerbl et al. 2023] has been demonstrated to surpasses existing implicit neural representation methods in terms of both quality and efficiency, thereby representing the current state-of-the-art in novel view synthesis. The approach relies on using a set anisotropic 3D Gaussians as an effective explicit scene representation and utilizing a fast tile-based differentiable rasterizer for image rendering.

Extensions of such novel scene representations to the spectral domain beyond the RGB channels remain an open challenge with only a few seminal works. Spectral variants of NERF, such as xNERF [Poggi et al. 2022] for cross-spectral spectrum-maps and SpectralNeRF [Li et al. 2022] for multi-spectral spectrum-maps, have demonstrated effectiveness in generating novel views across different spectral domains.

2.3 Neural rendering frameworks

Recent developments in neural rendering frameworks like NeRFFactory[Jeong et al. 2022], NerfAcc[Li et al. 2023], MultiNeRF[Mildenhall et al. 2022], and KaolinWisp[Takikawa et al. 2022] have notably enhanced the usability of NeRFs, each targeting different aspects such as benchmarking, modular design, or integrating multiple studies into one platform. Nerfstudio[Tancik et al. 2023] is a modular PyTorch framework, which supports real-time visualization, efficient data handling, and diverse export capabilities, all under an open-source Apache2 license, catering to both academia and industry. It surpasses other repositories for its modularity, real-time capabilities, and ease of use with user-captured data. Viser[Nerfstudio Project 2023], the web-viewer for Nerfstudio, has been developed using ThreeJS and ReactJS, and also offers Python-accessible API which can be used to customize viewer user interface. Viser is inspired from other open-source Python package that enables rapid development of demos or web applications for machine learning models like Gradio[Abid et al. 2019]. This is why Viser forms a perfect base framework for building our viewer application because then our application could be easily integrated to Nerfstudio and enhanced further with recent developments in this area. While scene representations in terms of both NeRFs and 3D Gaussian Splatting are supported, the extension to spectral scene representation remains as a challenge that we address with this paper.

3 SPECTRAL-SPLATS VIEWER

This section provides a comprehensive overview of the design and technical considerations that were taken into account during the development of our web-based framework for the visualization of multi/cross-spectral Gaussian splats.

We will first outline the overall architecture, which will be followed by an in-depth exploration of each individual component, detailing their functionalities and interactions within the system.

3.1 Architecture

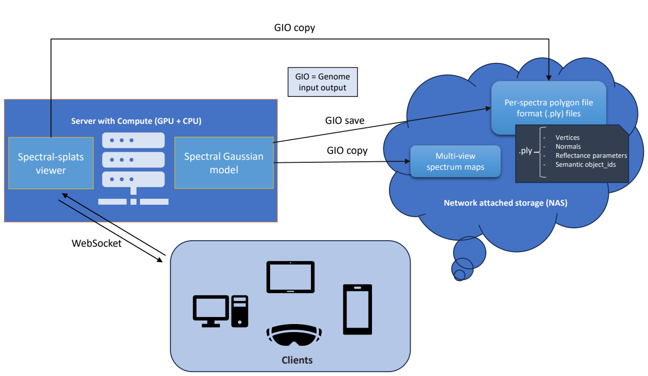


Figure 3: SpectralSplats Viewer: Architecture

The overall architecture of the viewer is illustrated in Figure 3. We used Viser[Tancik et al. 2023] as a base framework for our implementation. Viser is built on ReactJS and is accessible as a publicly hosted website. Viser is designed to accommodate users with both local and remote GPUs. It simplifies the process of utilizing remote computing resources by enabling port forwarding through SSH. As training starts, Viser dynamically displays a NeRF in real-time, allowing users to explore the scene by panning, zooming, and rotating during the optimization or while evaluating a trained model. Furthermore, the Gaussian splatting framework uses a similar training and visualization approach and, hence, makes Viser an ideal base framework for our purposes.

Server deployment: We utilize rootless Docker and Slurm[Jette and Wickberg 2023] workload manager to deploy and run both the Spectral splats viewer and the Spectral Gaussian model. Rootless Docker ensures that the deployment is secure and does not require root privileges [Matsumoto and Suda 2024], enhancing security and minimizing potential system vulnerabilities. Slurm, a workload manager, is employed to efficiently schedule and manage the computational tasks associated with the deployment of the viewer and the model, optimizing resource use and scaling according to demand.

Data transfer: We have extended Viser to interpret data generated in Polygon file format (.ply) by the Spectral Gaussian model from multi-view spectrum maps. In our implementation, data management is handled using GIO (Genome Input-Output), which utilizes secure shell (SSH) for secure data transfer within the docker containers and the network attached storage. This integration aligns seamlessly with the architecture of training and testing deep learning models, ensuring efficiency and security in data handling.

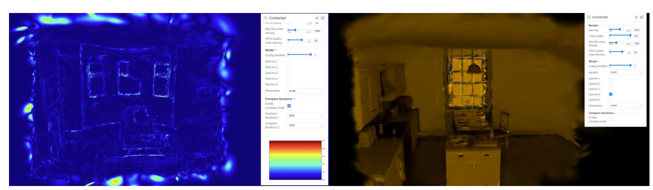


Figure 4: Spectral-Splats Viewer: *left:* Pixel-comparator (showing the difference between epochs (epoch 7000 and epoch 10000 in this case)) and *right:* spectral rendering (580nm) of the kitchen scene [Li et al. 2022]

Client renderer: Real-time visualization employs WebSockets and WebRTC to connect a session to the web client, eliminating the need for local GUI installations similar to Viser. Once the web viewer is opened, it establishes a WebSocket connection that displays rendered images. The viewer streams camera poses to the session, which then renders and sends back images via WebRTC. The viewer's UI and camera controls are built with ThreeJS, enabling the overlay of 3D assets like images on the renderings. This setup allows different types of client devices (as shown in Figure 3) to display and compare different spectral renderings both as splats (Figure 4) and points (Figure 2). We can use the Pixel comparator mode to compare different iterations at different spectra (see

Fig. 8) or vice-versa (see Fig.9). Different clients can also be used collaboratively using the viewer controls as shown in Fig. 5 and Fig. 6. However, the view is not synchronized between devices to support the possibility of individual scene exploration by different users. We need to enhance this basic feature by enabling controls, including views, that can be used collaboratively according to a specified configuration. Additionally, the session should be saved and capable of being resumed in the future to improve collaborative versioning of different tasks.

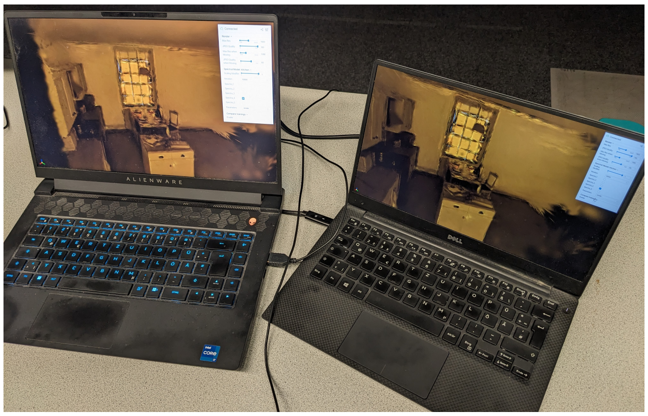


Figure 5: Collaborative mode: The selected spectra can be visualized in different devices collaboratively

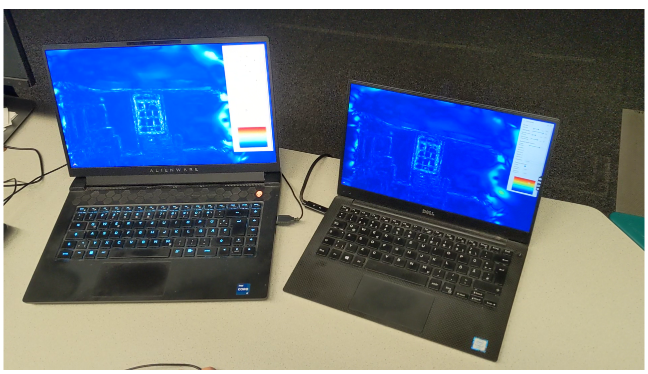


Figure 6: Collaborative mode: The selected iterations can be visualized and compared in different devices collaboratively

3.2 Implementation

The two main aspects of our implementation involve defining the data format of the PLY files and interpreting this data to be visualized by the viewer.

3.2.1 Data format: The spectral Gaussian model generates a PLY file per spectrum and per checkpoint (epochs in the training process where weights are saved) with vertex attributes covering the following characteristics:

- Position of the points (x,y,z)
- Opacities
- Scaling
- Rotations
- Normals
- Depth
- Reflectance parameters (or only color per point)
  - Specular
  - Specular Color
  - Diffuse
  - Diffuse Color
  - Roughness etc.
- Semantic object-ids (Distilled feature fields)

The reflectance parameters can optionally be replaced by a single color (r,g,b) that represents the color at a particular point. Additionally, the normals, depth, and semantic object-ids are optional parameters, and the viewer configuration determines what needs to be loaded.

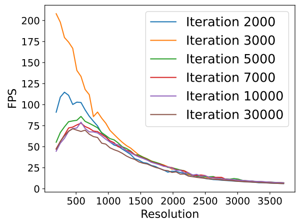
We used a PLY file per spectrum as opposed to a single PLY file as it has the following advantages:

- We can choose to optionally load only the iterations and spectra we need, which reduces the amount of data transferred. This improves both load balancing and performance.
- The other main advantage is that we can also visualize multiple spectra when the spectrum-maps are not completely co-registered. By *co-registered*, we mean that the pose of the cameras is the same for all the spectra in a particular view.

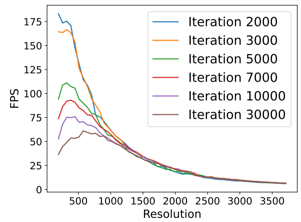
Defining the data format is this manner help us to be more flexible in regards to what we want to visualize.

3.2.2 Visualization: The visualization module is developed in Python and employs Viser, a tool built using ReactJS. We patched *the Viser framework* with some additional functionalities such as the inclusion of a legend with a specific colormap to better suit our specific requirements. The main features of the viewer include:

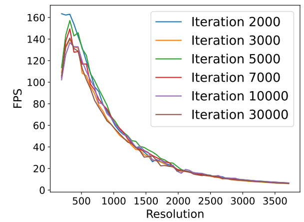
- The visualization of different spectra at various epochs (saved training iterations).
- An adjustable resolution for performance optimization, allowing users to reduce resolution when moving the camera or under other circumstances.
- A scaling factor to control the size of the splats.
- Input controls (Mouse or Keyboard) to navigate the scene.
- Pixel comparison, i.e. we provide capabilities where different renderings as well as images containing other attributes (reflectance parameters, depth, normals, semantic object-ids, etc.), can be compared at various iterations or spectra (see Fig. 8 and Fig. 9). The colormap used for comparisons can also be configured (jet, viridis, magma, etc.) . We used the jet colormap for our comparisons, as most relevant papers [Jiang et al. 2023] seem to use it for the comparison with ground truth references. This mode helps us in the following ways:
  - Comparing different iterations: This feature gives the user deeper insights into the training process, helping them decide which iteration to load based on performance and quality. It also aids in understanding the optimal training



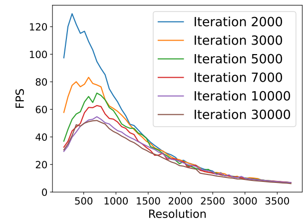
(a) Dragon doll



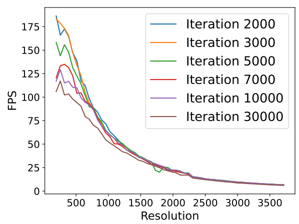
(b) Ficus



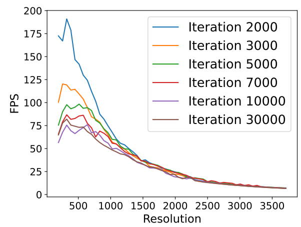
(c) Kitchen



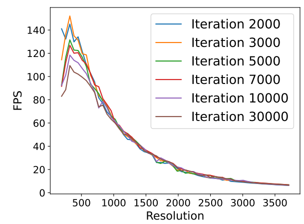
(d) Penguin



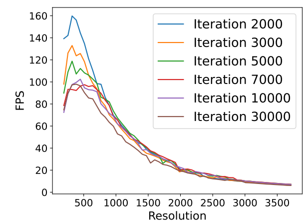
(e) Dino IR



(f) Dino MS



(g) Teapot



(h) Projector

Figure 7: Frames per seccond (FPS) of scenes at different resolutions and training iterations (i.e having different number of Gaussians).

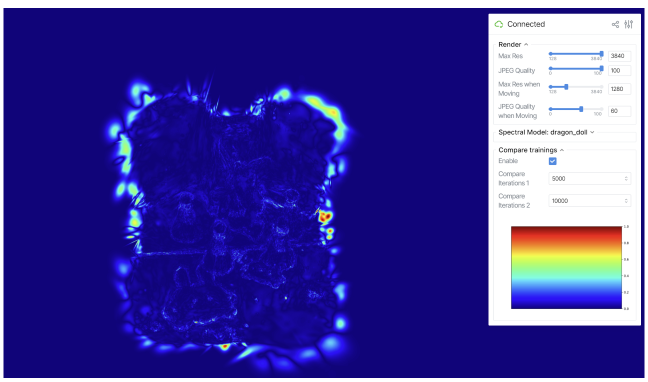


Figure 8: Iteration comparison mode (Dragon doll scene[Li et al. 2022]): Comparison of the pixel renderings at different iterations (shown for iterations 5000 and 10000)

duration for the Spectral Gaussian model on similar scenes in the future, depending on the needs. Additionally, for an expert user, it provides insights on how to improve the training process.

– Comparing spectral renderings: This mode can also be configured to compare different spectra as shown in Fig. 9. This is particularly important for understanding the differences between various spectral visualizations. Specifically, the comparison of semantic object-IDs could be useful for various smart farming applications such as plant disease detection[Moghadam et al. 2017], algal bloom detection [Do Hyuck Kwon and Cho 2023] or in cultural heritage for analysis and restoration of colored objects[Jones et al. 2020].

- Collaborative mode: The viewer enables collaborative visualization across multiple devices (Fig. 10). We need to make this mode more configurable from our point of view, which is why we conducted a user study with domain experts to gain insights on respective requirements and the practical relevance of this feature. The comments from these experts indicate that the collaborative mode in the pixel comparator setting (Fig. 6) was particularly interesting to them (as discussed in the evaluation).

4 EXPERIMENTAL EVALUATION

A performance analysis of the viewer was conducted to quantitatively assess the tool's efficiency. We conducted a user study to

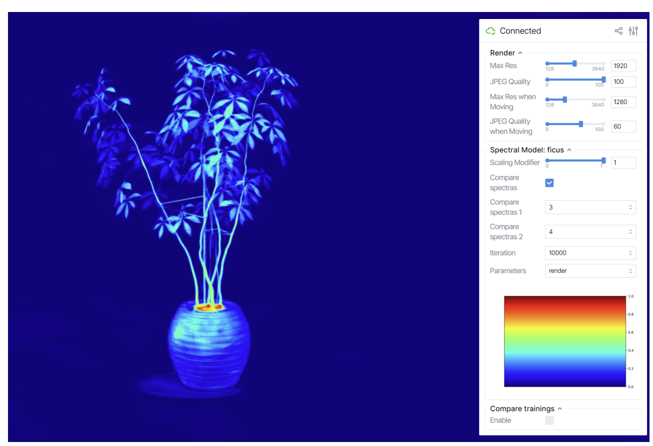


Figure 9: Spectra comparison mode (Ficus scene): Comparison of renderings at 580nm and 620nm

demonstrate the efficacy of our viewer through perceptual evaluation. The user study included assessments of visual quality, usability, comprehensibility, and performance.

4.1 Performance analysis

To assess the efficiency of our viewer in loading various spectral scenes, we measured the frame time (time to render each frame) at different resolutions ranging from 192x192 pixels to 3840x3840 pixels with a step size of 64 pixels along both axes. These measurements were taken for different types of cross-spectral scenes (see Fig. 11). We selected a diverse range of scenes, each with varying numbers of Gaussians, to ensure that our performance analysis accurately reflects the viewer's efficiency across different scenarios. The frame time measurements were conducted at various training iterations, specifically at iterations 2000, 3000, 5000, 7000, 10000, and 30000, which corresponded to different numbers of Gaussians generated in the training process. To report the results, we calculated the inverse of the frame time, which is commonly known as Frames per Second (FPS) and the corresponding plots are presented in Fig. 7. The performance analysis was done on a PC equipped with an RTX 3090 GPU which hosted the GPU and was connected to two clients during that time.

The graphical analysis reveals that in order to achieve a frame rate exceeding 30 FPS, which is essential for smooth navigation on various client devices such as mobiles, tablets, and PCs, it is crucial to maintain a resolution below 1536 x 1536 pixels. Furthermore, the data indicates that at very high resolutions, the number of Gaussians does not have significant impact in the frame rate across different training iterations. However, there is a significant influence at resolutions below 1500 pixels. By selecting a lower number of training iteration at these lower resolutions, provided that the rendered output maintains acceptable quality in lower training iteration, we can optimize performance. For most scenes, we observed that 10,000 training iterations and a resolution of 1280 x 1280 pixels delivered good perceptual quality, with a frame rate of approximately 45-50 FPS. The performance level achieved is robust enough to ensure seamless navigation on a wide range of client browsers and devices, including tablets, mobile phones, laptops, and PCs.

The average FPS across all resolutions for a specific training iteration is detailed in Table 1. The data from the table reveals that the average FPS typically ranges between 35-40, which facilitates smooth navigation through the scenes. Additionally, performance can be further enhanced by opting for a lower resolution during camera movement.

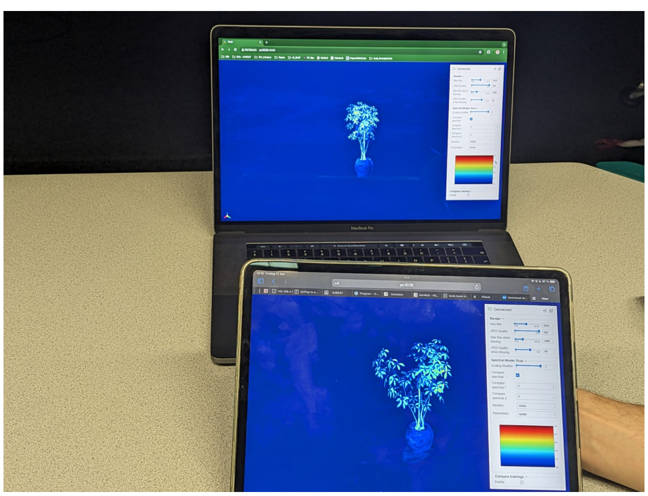


Figure 10: Collaborative mode (Ficus scene): The figure shows that the one can use the tool to collaborate on the same scene using different devices

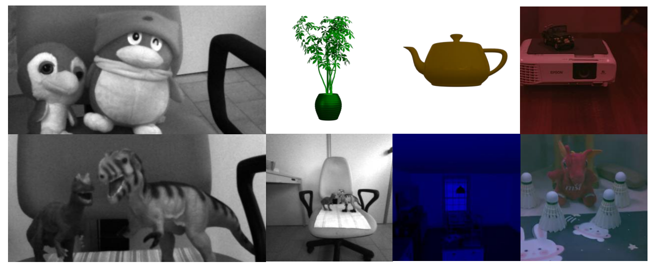


Figure 11: Different scenes considered for performance analysis: Ficus dataset [Mildenhall et al. 2020], Teapot dataset [Jiang et al. 2023], Dino and Penguin datasets [Poggi et al. 2022], as well as Projector, Dragon-doll and Kitchen datasets [Li et al. 2022]

4.2 User-study

We conducted two user studies: one for non-expert users and another for domain experts who also participated in the first user study.

4.2.1 Participants: As participants for the user study, we mainly selected domain experts (15 of 20 participants), i.e. people that were familiar with spectral data or have knowledge in the fields of computer graphics and computer vision. We chose such a constraint as we wanted to understand how our viewer can help this community using spectral data by analyzing spectral Gaussian splats. Moreover, the participants had different backgrounds, including researchers, doctors, professors, designers, software developers and also persons without any academic background. All participants were uninformed about the experiment's aims, gave informed consent and reported either normal or corrected-to-normal visual and auditory acuity.

4.2.2 Design of experiment:

User-study 1 (Evaluation for the current functionalities): In order to evaluate the visualization of cross-spectral renderings of the Gaussian splats, the participants had to navigate through three scenes involving reconstructions form both multi-spectral [Poggi et al. 2022] (Dino and penguin) and infrared [Poggi et al. 2022] (Dino) spectrum maps. Hence, the study considers cross-spectral data for evaluation. The users were presented with a set of eight questions and had to provide ratings on a 7-point Likert scale [Jebb

Table 1: Average FPS (considering rendering resolution from 192 to 3840 pixels) for 8 spectral datasets at different points of training

Training iteration	Teapot (No. of Spectra: 5)		Penguin (No. of Spectra: 10)		Kitchen (No. of Spectra: 5)		Ficus (No. of Spectra: 5)	
	No. of Gaussians	Avg. FPS	No. of Gaussians	Avg. FPS	No. of Gaussians	Avg. FPS	No. of Gaussians	Avg. FPS
2000	75348	40.770	87523	41.014	39596	43.195	61259	43.663
3000	98707	41.282	222036	33.631	71772	38.434	113224	43.827
5000	97503	39.563	376899	30.113	102215	43.366	136372	35.209
7000	91551	40.210	468916	27.448	121652	40.497	158656	33.883
10000	74914	38.211	561636	25.943	141039	40.119	244292	30.627
30000	78527	36.859	561636	24.391	141039	38.171	363099	27.440
	Dragon doll (No. of Spectra: 5)		Dino MS (No. of Spectra: 10)		Dino IR ((No. of Spectra: 1)		Projector (No. of Spectra: 5)	
	No. of Gaussians	Avg. FPS	No. of Gaussians	Avg. FPS	No. of Gaussians	Avg. FPS	No. of Gaussians	Avg. FPS
2000	67027	36.610	88141	48.080	41208	45.264	35945	44.311
3000	128071	49.535	233337	38.766	114950	44.945	88362	39.884
5000	178328	34.630	368680	37.474	186800	42.707	110969	40.540
7000	206430	32.747	454009	34.599	239264	39.753	138239	38.512
10000	227095	31.971	530290	30.840	287433	39.243	168844	37.262
30000	227095	28.906	530290	30.806	350396	34.650	168844	33.611

et al. 2021] after they had finished interacting with the viewer performing the necessary tasks. The tasks involved:

- Interacting with the different user-interface elements and scene navigation: The users selected different spectra, scaling factor, BRDF parameters, etc.) and navigated through the scenes.
- Using the pixel comparator:
  - Compare iterations: The user compared the different outputs rendered (BRDF parameters, depth, semantic segmentation, render output, etc.) at different iterations, i.e. at different points of the training for a particular spectrum.
  - Compare spectra: The users had to perform a similar operation while comparing different spectra at a particular iteration.

The questions considered different aspects such as ease of use and the intuitiveness of the user-interface, etc., and the results of which are described in detail in Sec. 4.2.3.

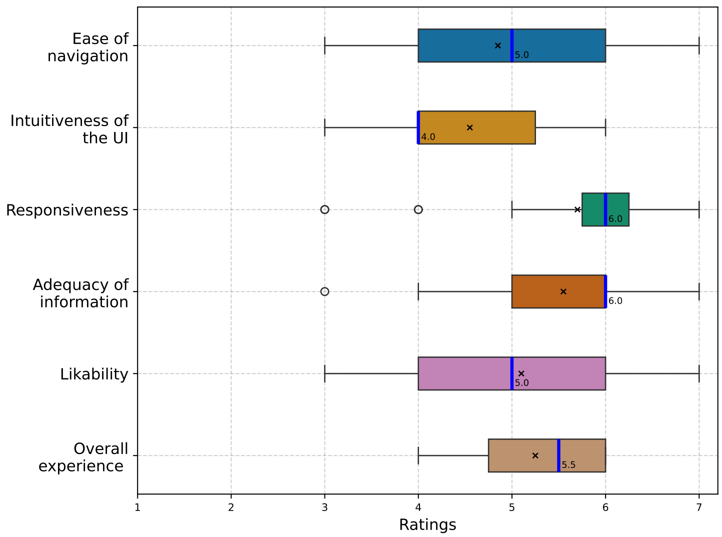
User-study 2 (Evaluation for future enhancement): In the second user study, we specifically targeted expert users with background knowledge in working with spectral data. The aim was to understand which feature enhancements would make the viewer more attractive to these users. The study was conducted after the participants had completed user-study 1 and were familiarized with the collaborative mode of our viewer. They were asked three questions regarding future feature improvements related to collaborative mode, geometry exporter, and material editing. The results of this study are presented in Section 4.2.3.

The spectral-splats viewer was hosted in a PC having graphics card RTX 3090. The users accessed the web page using their own laptops or PC's in Google chrome.

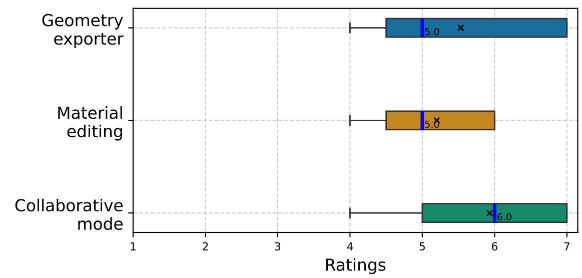
4.2.3 Results and analysis (User-studies). The statistical distribution of our user studies are presented using a box plot in Fig. 12

Analysis of User-study 1: The *Ease of Navigation* aspect assessed how participants managed the scene controllers and viewed the scene from various perspectives, with most navigating through the viewer quite fluently. Under *Intuitiveness of the UI*, the focus was on the ease of understanding user-interface controls and user familiarization. The results indicated that intuitiveness was just above average, primarily because non-experts required additional time to become acquainted with the viewer, compounded by the absence of meta-data information for our scenes. The *Responsiveness* aspect evaluated performance, crucial since the viewer is web-based and not hosted on local devices. The viewer showed excellent responsiveness even under multiple simultaneous users. Regarding *Adequacy of Information*, users generally found the provided data sufficient for comparison modes and most domain experts appreciated this feature. For *Likability*, participants' likelihood to recommend the viewer varied, likely due to the mixed user base of general users and domain experts, though overall likability was good. In the scope of additional comments the domain users expressed strong enthusiasm for the pixel comparator mode, as it significantly enhanced their understanding of scenes in terms of different spectra and training iterations. Lastly, the *Overall Experience* received positive feedback, indicating that users, including those without expert backgrounds, could effectively navigate and utilize the viewer's features.

Analysis of User-study 2: In this segment of the study, the focus was on enhancing the viewer for optimal future usability. In addition to the ratings, we also asked the users to provide any comments or notes along with their responses. These comments are valuable for us as they provide context and insights into the ratings, particularly from users who have domain expertise in this field. Expert users expressed significant interest in the *collaborative mode*, suggesting that further development of this feature could substantially enhance the viewer's usability across diverse scenarios. Domain experts also highlighted the potential industrial applications, particularly in environments where individuals need to annotate various regions in a spectral scene for collaboration. Additionally, the *geometry exporter* feature was noted as beneficial for users who work with other 3D editing tools, as it facilitates modifications of 3D data for various uses. Some experts thought *material editor* could help changing the light in the scene which could be necessary for certain scenarios. These observations highlight the critical role of these



(a) User-study 1 (Evaluation for the current functionalities)



(b) User-study 2 (Evaluation for future enhancement)

Figure 12: In the above figure, the statistical distribution of the ratings of the user-studies are presented. The inter-quartile range (IQR) is depicted as a colored box, with the median represented by a blue line inside the box and the mean indicated by an × symbol. The whiskers of the box plot extend to 1.5 times the IQR. Any data points lying outside this range are shown as individual points, denoting them as outliers.

features in broadening the viewer's practical utility and applicability in professional environments.

5 LIMITATIONS

From user feedback and the study, limitations of the viewer were identified, emphasizing aspects for further enhancement. Users provided feedback on several areas for improvement, including tool-tips for better usability, more intuitive user-interface elements and refined keyboard controls. To optimize the collaborative mode, it is crucial to enhance annotation capabilities, session syncing, and implement a geometry exporter. Additionally, addressing the need for more compute power, load balancing capabilities, and higher resolution results will transform our tool into a fully-fledged product with improved functionality, user experience, and professional applicability.

6 CONCLUSION AND FUTURE WORK

Our web-based viewer is the only tool available for visualizing cross-spectral splats using the Gaussian splatting framework. It enables intuitive analysis and comparison of training model-generated splats, providing valuable insights for enhancing training and understanding the process. The viewer has diverse applications in smart farming, cultural heritage, and other fields where spectral data is utilized. User-studies and performance analysis confirm its efficiency, with potential usability enhancements through proposed features.

For future work, we envisage the possibility of integrating a geometry and texture exporter into the viewer, supporting various formats to enable content visualization across different applications and platforms. Another potential extension could involve modifications to lighting and reflectance parameters, which would allow us to render scenes with varied materials and lighting in different spectral ranges. Currently the viewer also supports simple collaborative functions which can be enhanced in the future to serve wide range of industrial use-cases.

ACKNOWLEDGMENTS

The work presented in this paper has been funded by the European Commission during the project PERCEIVE under grant agreement 101061157.

REFERENCES

Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. 2019. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. *arXiv preprint arXiv:1906.02569* (2019).

Matthias Alfeld, Maud Mulliez, Jonathan Devogelaere, Laurence de Viguerie, Philippe Jockey, and Philippe Walter. 2018. MA-XRF and hyperspectral reflectance imaging for visualizing traces of antique polychromy on the Frieze of the Siphnian Treasury. *Microchemical Journal* 141 (2018), 395–403. https://doi.org/10.1016/j.microc.2018.05.050

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. arXiv:2103.13415 [cs.CV]

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5460–5469.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 19697–19705.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022a. TensoRF: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision.* 333–350.

Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. 2022b. Hallucinated neural radiance fields in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 12943–12952.

Ather Abbas Sanghyun Park Gibeom Nam Jae-Hyun Yoo Kyunghyun Kim Hong Tae Kim JongCheol Pyo Do Hyuck Kwon, Seok Min Hong and Kyung Hwa Cho. 2023. Deep learning-based super-resolution for harmful algal bloom monitoring of inland water. *GIScience & Remote Sensing* 60, 1 (2023), 2249753. https://doi.org/10.1080/15481603.2023.2249753

Antonin Dufay, David Murray, Romain Pacanowski, et al. 2019. The Malia rendering framework. https://pacanows.gitlabpages.inria.fr/MRF.

Luca Fascione, Johannes Hanika, Marco Leone, Marc Droske, Jan Schwarzhaupt, Tomislav Davidovic, Alexander Weidlich, and Johannes Meng. 2018. Manuka: A batch-shading architecture for spectral path tracing in movie production. *ACM Transactions on Graphics* 37, 3 (2018), 31:1–31:18. https://doi.org/10.1145/3182161

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5491–5500.

Wenhang Ge, Tao Hu, Haoyu Zhao, Shu Liu, and Ying-Cong Chen. 2023. Ref-NeuS: Ambiguity-Reduced Neural Implicit Surface Learning for Multi-View Reconstruction with Reflection. *arXiv preprint arXiv:2303.10840* (2023).

Federico Grillini, Lavinia de Ferri, George Alexis Pantos, Sony George, and Margunn Veseth. 2024. Reflectance imaging spectroscopy for the study of archaeological pre-Columbian textiles. *Microchemical Journal* 200 (2024), 110168. https://doi.org/10.1016/j.microc.2024.110168

Wenzel Jakob. 2010. Mitsuba 2: Physically based renderer. http://www.mitsuba-renderer.org.

Alexander T Jebb, Vivian Ng, and Louis Tay. 2021. A Review of Key Likert Scale Development Advances: 1995-2019. *Frontiers in Psychology* 12 (2021), 637547. https://doi.org/10.3389/fpsyg.2021.637547

JENOPTIK. Accessed: 31-05-2024. EVIDIR alpha thermal imaging camera and infrared modules – one size for all variants. https://www.jenoptik.com/products/cameras-and-imaging-modules/thermographic-camera/thermal-imaging-camera.

Yoonwoo Jeong, Seungjoo Shin, and Kibaek Park. 2022. NeRF-Factory: An awesome PyTorch NeRF collection. https://github.com/kakaobrain/NeRF-Factory/.

Morris A. Jette and Tim Wickberg. 2023. Architecture of Slurm Workload Manager. In *Job Scheduling Strategies for Parallel Processing: 26th Workshop, JSSPP 2023, St. Petersburg, FL, USA, May 19, 2023, Revised Selected Papers* (St. Petersburg, FL, USA). Springer-Verlag, Berlin, Heidelberg, 3–23. https://doi.org/10.1007/978-3-031-43943-8_1

Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. 2023. GaussianShader: 3D Gaussian Splatting with Shading Functions for Reflective Surfaces. *arXiv preprint arXiv:2311.17977* (2023).

Cerys Jones, Christina Duffy, Adam Gibson, and Melissa Terras. 2020. Understanding multispectral imaging of cultural heritage: Determining best practice in MSI analysis of historical artefacts. *Journal of Cultural Heritage* 45 (2020), 339–350. https://doi.org/10.1016/j.culher.2020.03.004

Kim Jun-Seong, Kim Yu-Ji, Moon Ye-Bin, and Tae-Hyun Oh. 2022. HDR-Plenoxels: Self-calibrating high dynamic range radiance fields. In *Proceedings of the European Conference on Computer Vision*. 384–401.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Marco Landi and Giuseppe Maino. 2011. Multispectral Imaging and Digital Restoration for Paintings Documentation. In *Image Analysis and Processing – ICIAP 2011*, Giuseppe Maino and Gian Luca Foresti (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 464–474.

L. Lanteri and C. Pelosi. 2021. 2D and 3D ultraviolet fluorescence applications on cultural heritage paintings and objects through a low-cost approach for diagnostics and documentation. In *Optics for Arts, Architecture, and Archaeology VIII*, Haida Liang and Roger Groves (Eds.), Vol. 11784. International Society for Optics and Photonics, SPIE, 1178417. https://doi.org/10.1117/12.2593691

Boyi Li, Junting Xie, Yifu Zhang, Lingjie Zhang, Weiming Wang, and Hongdong Li. 2022. SpectralNeRF: Physically Based Spectral Rendering with Neural Radiance Fields. *ACM Transactions on Graphics (TOG)* 41, 4, Article 109 (2022).

Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. 2023. NerfAcc: Efficient Sampling Accelerates NeRFs. *arXiv preprint arXiv:2305.04966* (2023).

Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7206–7215.

Naoki Matsumoto and Akihiro Suda. 2024. bypass4netns: Accelerating TCP/IP Communications in Rootless Containers. arXiv:2402.00365 [cs.NI]

Micasense. Accessed: 2024-04-24. Micasense RedEdge-MX DUAL. https://drones.measurusa.com/products/micasense-rededge-mx-dual.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman, Ricardo Martin-Brualla, and Jonathan T. Barron. 2022. MultiNeRF: A Code Release for Mip-NeRF 360, Ref-NeRF, and RawNeRF. https://github.com/google-research/multinerf

Peyman Moghadam, Daniel Ward, Ethan Goan, Srimal Jayawardena, Pavan Sikka, and Emili Hernandez. 2017. Plant Disease Detection Using Hyperspectral Imaging. In *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. 1–8. https://doi.org/10.1109/DICTA.2017.8227476

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

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. 2022. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8270–8280.

David Murray, Alban Fichet, and Romain Pacanowski. 2021. *Efficient Spectral Rendering on the GPU for Predictive Rendering*. Springer, 673–698. https://doi.org/10.1007/978-1-4842-7185-8_42

Nerfstudio Project. 2023. Viser: A library for interactive 3D visualization in Python. https://github.com/nerfstudio-project/viser.

Marion Nimier-David, Daniele Vicini, Thomas Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 38, 6 (December 2019), 203:1–203:17. https://doi.org/10.1145/3355089.3356498

Matt Pharr. 2020. PBRT version 4. https://github.com/mmp/pbrt-v4.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation* (3rd ed.). Morgan Kaufmann.

Matteo Poggi, Pierluigi Zama Ramirez, Fabio Tosi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. 2022. Cross-Spectral Neural Radiance Fields. In *Proceedings of the International Conference on 3D Vision*. 3DV.

Rizwan Qureshi, Muhammad Uzair, Khurram Khurshid, and Hong Yan. 2019. Hyperspectral document image processing: Applications, challenges and future prospects. *Pattern Recognition* 90 (2019), 12–22. https://doi.org/10.1016/j.patcog.2019.01.026

Silios. Accessed: 2024-04-24. Off-the-shelf Snapshot Multispectral Cameras. https://www.silios.com/multispectral-imaging.

Towaki Takikawa, Or Perel, Clement Fuji Tsang, Charles Loop, Joey Litalien, Jonathan Tremblay, Sanja Fidler, and Maria Shugrina. 2022. Kaolin Wisp: A PyTorch Library and Engine for Neural Fields Research. https://github.com/NVIDIAGameWorks/kaolin-wisp.

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. 2023. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*.

The ART development team. 2018. The Advanced Rendering Toolkit. https://cgg.mff.cuni.cz/ART.

N.T. Vetrekar, R. Raghavendra, and R.S. Gad. 2016. Low-cost multi-spectral face imaging for robust face recognition. In *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*. 324–329. https://doi.org/10.1109/IST.2016.7738245

Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. 2022. NeRF-SR: High quality neural radiance fields using supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia*. 6445–6454.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Adv. Neural Inf. Proc. Syst.* 354 (2021), 27171–27183.

Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. 2023. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3272–3283.

Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. 2022. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5438–5448.

I.-Chang Yang, Kuang-Wen Hsieh, Chao-Yin Tsai, Yu-I. Huang, Yu-Liang Chen, and Suming Chen. 2014. Development of an automation system for greenhouse seedling production management using radio-frequency-identification and local remote sensing techniques. *Engineering in Agriculture, Environment and Food* 7, 1 (2014), 52–58. https://doi.org/10.1016/j.eaef.2013.12.009

Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P. Srinivasan, Richard Szeliski, Jonathan T. Barron, and Ben Mildenhall. 2023. BakedSDF: Meshing neural SDFs for real-time view synthesis. In *Proceedings of the ACM SIGGRAPH 2023 Conference*, Erik Brunvand, Alla Sheffer, and Michael Wimmer (Eds.). 46:1–46:9.

Anam Zahra, Rizwan Qureshi, Muhammad Sajjad, Ferhat Sadak, Mehmood Nawaz, Haris Ahmad Khan, and Muhammad Uzair. 2024. Current advances in imaging spectroscopy and its state-of-the-art applications. *Expert Systems with Applications* 238 (2024), 122172. https://doi.org/10.1016/j.eswa.2023.122172

Yi Zhang, Xiaoyang Huang, Bingbing Ni, Teng Li, and Wenjun Zhang. 2023. Frequency-Modulated Point Cloud Rendering with Easy Editing. *arXiv preprint arXiv:2303.07596* (2023).