Simplifying NeRF: Creating an Intuitive Web-Based 3D Scene Interface

Eduard von Briesen

March 31, 2023
Version: My First Draft

Department of Mathematics,
Informatics and Statistics,
Institute of Informatics

Munich Film School,
Chair for AI

Masters Thesis

Simplifying NeRF: Creating an Intuitive Web-Based 3D Scene Interface

Eduard von Briesen

Supervisors Prof. Dr. Sylvia Rothe and Cristoph Weber

March 31, 2023

Eduard von Briesen
Simplifying NeRF: Creating an Intuitive Web-Based 3D Scene Interface
Masters Thesis, March 31, 2023
Supervisors: Prof. Dr. Sylvia Rothe and Cristoph Weber

LMU Munich
Department of Mathematics, Informatics and Statistics
Institute of Informatics
Artificial Intelligence and Machine Learning (AIML)
Akademiestraße 7
80799 Munich

Abstract

Write abstract

Abstract (German)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

1 Introduction

Neural Radiance Fields (NeRF) have emerged as a transformative technology in 3D scene modeling and rendering, offering unprecedented realism and detail. This advancement has significantly impacted various applications, from virtual reality to cultural heritage preservation.

Two prominent NeRF frameworks with user interfaces, namely Instant NGP [22] and Nerfstudio [34], have emerged as leaders in enabling users to explore and manipulate 3D scenes. These frameworks offer features such as real-time scene rendering, adjustable training parameters, and the creation of camera trajectories for video rendering.

Additionally, several innovative projects have expanded the NeRF landscape. Notably, CLIP-NeRF [40], Instruct-NeRF2NeRF [13], Text2LIVE [3], and SINE [2] have introduced text-based editing approaches, broadening the possibilities for manipulating NeRF models. PaletteNeRF [41] focuses on color editing, while NeRF-Editing [43] enables mesh editing.

Despite these advancements, the technical complexity of these frameworks frequently acts as a barrier to broader accessibility, indicating a need for improvements in user experience. These interfaces usually require a high degree of technical knowledge, as they are intended to supplement, rather than substitute, command-line interfaces. For activities like video data preprocessing, model training, and output rendering, users are required to navigate through terminal-based processes.

This complexity not only limits the potential user base to individuals with technical expertise but also hinders the creative and innovative application of NeRF technology across broader fields. Consequently, there exists a critical need to enhance the user experience and develop solutions that simplify the interaction with NeRF frameworks, making them more approachable and usable for a diverse range of users beyond the realm of technical specialists.

Research Objectives

The research objectives of this study are as follows:

1. **Exploration of NeRF Interaction Capabilities**: This study aims to explore the existing interaction capabilities within NeRF frameworks comprehensively. It involves an analysis of the current state of NeRF interfaces and an investigation into user engagement, visualizations, and manipulation of NeRF scenes.
2. **Development of a Web-Based User Interface**: Building on insights gained from the exploration phase, the primary objective is to design and implement a user-friendly web-based interface for NeRF.
3. **Streamlined NeRF Creation and Manipulation**: The central goal is to simplify the process of NeRF creation and manipulation, eliminating the need for users to deal with complex command-line interfaces or extensive local setup. The web-based interface will provide an intuitive and efficient user experience.
4. **Integration of Diverse Editing Plugins**: To enhance the creative potential of NeRF, various editing plugins will be integrated into the web-based interface. The objective is to expand the functionality and versatility of the NeRF framework.

The research aims to advance NeRF frameworks' capabilities and accessibility, making them accessible to a broader audience and fostering innovation in 3D scene modeling and rendering.

Research Question

This research is guided by the following questions:

1. **Enhancing NeRF Frameworks:** How can a web-based interface improve the user experience and accessibility of NeRF frameworks, and what impact will these enhancements have on user-friendly NeRF creation and manipulation?
2. **Overcoming Technical Challenges:** What technical challenges and limitations are associated with current NeRF frameworks and interfaces, and how can innovative design and technology choices in a web-based interface overcome these challenges?
3. **Innovative Editing Integration:** How can novel editing approaches be seamlessly integrated into a web-based NeRF interface to enhance creativity and usability, and how do these methods compare with traditional NeRF editing techniques?

Scope of the Study

This research focuses on the development and evaluation of a web-based interface for NeRF, aiming to improve its accessibility and usability. The study will concentrate on interface design, user interaction, and the integration of editing functionalities, without delving into the underlying algorithms of NeRF technology itself. It is delimited by its emphasis on interface design over algorithmic advancements in NeRF processing.

Significance of the Study

By addressing the usability challenges of current NeRF frameworks, this research aims to make 3D scene modeling more accessible, fostering innovation and broadening the application of this technology across various fields. The development of a web-based interface could significantly lower the entry barrier to NeRF, enabling artists, designers, and educators to leverage this technology without requiring deep technical expertise.

add structure of the thesis

Background

2

2.1 View Synthesis

View synthesis is a process used in computer graphics and computer vision that involves creating new, synthetic images of a scene from viewpoints that were not originally captured by a camera. This technique leverages existing images and often incorporates geometric information about the scene, enabling the generation of perspective-correct views from desired locations. The goal of view synthesis is to produce realistic and accurate representations of a three-dimensional scene from novel viewpoints, enhancing applications such as virtual reality (VR), augmented reality (AR), 3D television, and film production. The following sections provide an overview of traditional view synthesis techniques.

Image-Based Rendering Techniques Image-based rendering (IBR) techniques focus on synthesizing new views of a scene using pre-captured images, relying minimally on geometric models. The core premise of IBR is to directly utilize the radiance information captured in these images, manipulating it to generate new viewpoints without the need for detailed 3D reconstruction. This approach is particularly advantageous in scenarios where fast rendering is necessary or when accurate geometric data is unavailable. IBR methods are characterized by their ability to deliver photorealistic results, as they capture lighting, shadows, and reflections true to the original scene. These methods, such as those using light fields or lumigraphs, excel in applications where visual realism is critical, efficiently simulating the complexity of real-world lighting and textures [5, 7, 10, 12, 17].

Volumetric Methods Volumetric rendering techniques construct a three-dimensional volume of the scene, often represented as a grid of voxels. Each voxel contains data such as color and opacity, which contribute to the final image through a process akin to 3D texturing. Unlike surface-based modeling, which requires explicit surfaces, volumetric methods fill the entire data space, allowing for the handling of complex phenomena like fog, clouds, and fire, which do not have clear boundaries. This approach is advantageous when the scene involves intricate details and volumetric phenomena that traditional polygon-based rendering might struggle to capture accurately. Volumetric methods often employ techniques like voxel coloring to integrate multiple images into a cohesive 3D model [8, 31].

2.2 Neural Approaches to View Synthesis

Neural approaches to view synthesis have significantly advanced the capabilities of generating realistic views from sparse data. These methods integrate deep learning techniques to enhance the synthesis process, offering substantial improvements over traditional geometric and image-based methods.

Deep Learning for View Synthesis Deep learning models, particularly Convolutional Neural Networks (CNNs), are employed to predict depth and color information from sparse sets of images. These models facilitate the synthesis of intermediate views by interpolating between captured viewpoints, efficiently handling scenarios with incomplete data where traditional methods would struggle [15, 19, 26].

Learning Volumetric Representations Recent advances in neural view synthesis have explored volumetric representations to effectively encode three-dimensional scene information. This approach leverages deep learning to construct volumetric grids or embeddings that capture both color and spatial data, allowing for dynamic and realistic rendering of new views. Such volumetric techniques represent a significant departure from traditional modeling by providing a framework where 3D features are embedded directly into the network, enabling sophisticated scene understanding and rendering without explicit geometric reconstruction [18, 32].

Multiplane Images and Scene Representation Multiplane Images (MPIs) represent a neural approach to view synthesis where scenes are decomposed into layers at different depths. Neural networks learn to create these layered representations from a limited number of views. The learned MPIs can then be efficiently re-rendered from new perspectives using traditional graphics techniques. This method provides a robust solution for generating photorealistic and geometrically coherent views by accommodating variations in scene complexity and handling occlusions effectively [20, 25, 28, 35].

End-to-End Learning End-to-end learning with deep neural networks has significantly simplified the view synthesis process by enabling a single network to handle multiple synthesis tasks concurrently. This approach reduces the complexity and potential failure points associated with multi-stage synthesis methods. By training on large datasets of posed imagery, these networks learn complex mappings from input images to synthesized views, efficiently handling challenging scenarios [6, 11].

2.3 Neural Radiance Fields for View Synthesis

Neural Radiance Fields (NeRF) offer a novel approach to view synthesis that significantly advances previous techniques. Utilizing a fully connected deep neural network, NeRF optimizes a continuous volumetric scene function from a sparse set of input views [21]. This method diverges from traditional discrete representations like voxel grids or mesh geometries, employing a more fluid and detailed scene representation.

Implementation NeRF synthesizes views by querying a 5D coordinate spacecomprising spatial locations (x, y, z) and viewing directions (θ, ϕ) along camera rays (see Fig. 2.1 a). Classic volume rendering techniques are then applied to convert the output densities and colors into images. At the heart of this process is a Multilayer Perceptron (MLP), a type of neural network characterized by layers of interconnected nodes which process input data sequentially from input to output. The MLP in NeRF takes sampled 3D points and corresponding viewing directions as input and outputs the color and volume density (see Fig. 2.1 b). This output is then composited into a 2D image using volume rendering techniques (see Fig. 2.1 c). Because this rendering function is differentiable, NeRF uses gradient descent to minimize the difference between synthesized images and observed ground truth, optimizing the neural representation of the scene (see Fig. 2.1 d).

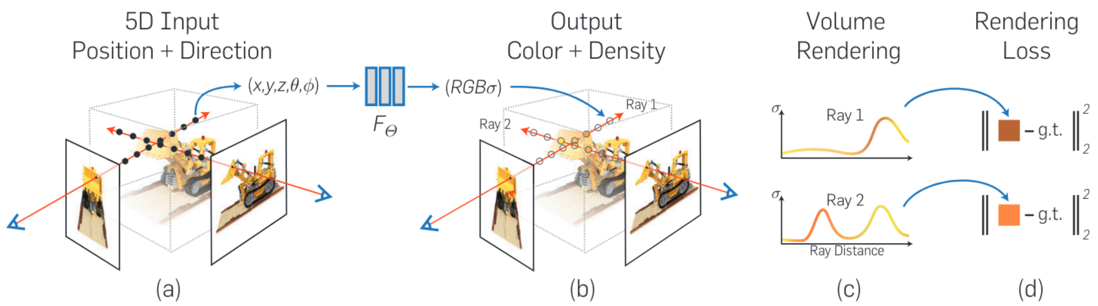


Fig. 2.1.: Overview of the NeRF scene representation and rendering procedure, illustrating the stages of sampling, neural processing, and image composition (from [21]).

Advantages of NeRF Compared to the methodologies discussed above, NeRF offers several advantages. It surpasses the capabilities of mesh and voxel-based methods in rendering high-resolution details and handling intricate geometries and material properties. The continuous volumetric representation is not only capable of producing more photorealistic images but also remains highly efficient in memory usage. This efficiency facilitates handling complex real-world scenes without the prohibitive storage and computation costs associated with traditional 3D representations.

Extensions and Applications of NeRF The landscape of NeRF has been enriched by several innovative extensions that broaden the utility and interactivity of NeRF models. NeRF frameworks with user interfaces have emerged, enabling users to explore and manipulate 3D scenes [22, 34]. These frameworks offer features such as real-time scene rendering, adjustable training parameters, and the creation of camera trajectories for video rendering. Recent advancements have introduced text-based manipulation, allowing users to edit and transform scenes through natural language descriptions [2, 3, 13, 14, 40]. Another area of development is in color and appearance editing, where new methods enable adjustments of scene colors, consistent across varying views [41]. Additionally, some tools have integrated mesh editing capabilities, enhancing the geometric manipulation within NeRF-generated scenes [43]. These enhancements not only extend the functional range of NeRF but also significantly improve their application in creative industries.

In conclusion, NeRF's innovative use of neural networks for scene representation sets a new standard for photorealistic view synthesis, delivering high-quality results that are both computationally efficient and visually impressive.

3 Related Work

3.1 Nerfstudio

Nerfstudio [34] represents a significant advancement in making Neural Radiance Fields accessible to non-technical users. Its design focuses on modularity, ease of use, and integration capabilities, which are crucial for practical applications and academic research.

Modularity Nerfstudio is built on a modular framework that allows users to easily customize and extend their NeRF implementations. This modularity supports various input data formats, making it versatile for different real-world scenarios. A wide range of existing methods are already well integrated into Nerfstudio, including Instant-GPT [22], their own Nerfacto [@23] method that combines various existing techniques, and several of the above mentioned extensions [13, 14].

Real-Time Web Viewer One of the standout features of Nerfstudio is its real-time web viewer, which enables visualization of NeRF training and outputs directly through a web browser. This eliminates the need for high-end local GPU setups, broadening the tool's accessibility [24].

Fig. 3.1.: Nerfstudio's real-time web viewer. (from [34])

Flexibility of Data Handling Nerfstudio simplifies the process of importing and exporting data, supporting a wide range of formats to support a variety of use-cases. Users can easily import images and videos, including data from mobile capture apps such as Polycam [@27] and Record3D [@30]. Additionally, the framework supports exporting results in various formats such as videos, point clouds, and meshes. This flexibility allows users to integrate NeRF outputs into diverse creative and technical applications.

Community and Open-Source Contribution Being open-source, Nerfstudio encourages community-driven development and continuous improvement, facilitating updates that keep pace with the latest research and technological advances. This openness also allows users to adapt the tool to their specific needs.

Limitiations Despite advancements, Nerfstudio's technical complexity and reliance on command-line interfaces for key operations like model training and data preprocessing remain significant barriers. These aspects limit its accessibility to those with specific technical skills and deter broader creative uses. The user experience still requires technical knowledge, underscoring the need for more intuitive interfaces that simplify interaction and expand its user base beyond technical specialists.

3.2 Luma AI

Luma AI [@1] is making Neural Radiance Fields accessible to non-technical users in a commercial space. This platform leverages augmented reality (AR) to guide users through the capture process, greatly simplifying the creation of NeRFs from everyday smartphones.

Guided Capture Process Luma AI utilizes AR to assist users in capturing images from optimal angles and distances, ensuring that the collected data is suitable for NeRF generation. This guided process reduces the complexities involved in capturing the necessary footage for effective NeRF modeling.

Cloud-Based NeRF Generation Once the footage is captured, it is automatically processed in Luma AIs cloud-based system to generate a NeRF, requiring no user input for configuration. This automation not only simplifies the user experience but also makes powerful 3D reconstruction technology readily accessible to a broad audience.

Viewing and Editing Created scenes can be viewed directly within the app or through a web browser (see Fig. 3.2). While the editing capabilities are limited, users can make basic adjustments, reshoot parts of the scene, and interact with the generated NeRF in an intuitive manner. The features here are similar to what is offered in Nerfstudio's viewer (see Fig. 3.1), but with a more modern user-interface.

Fig. 3.2.: Luma AI Web Viewer.

Export Capabilities Luma AI offers various export formats for the generated scenes, allowing users to utilize these outputs in different applications or platforms, further enhancing the utility of the captured NeRFs. They also provide a social platform for sharing and viewing NeRFs, fostering a community around the technology.

Limitations Despite its innovative approach, Luma AI's main limitation lies in the lack of user control over the NeRF training process. The automated system, while user-friendly, does not allow for adjustments in the training parameters or the refinement of the final model. This lack of control can result in less than optimal NeRF outputs for users who may require more precise or customized 3D representations.

4 Methodology

This research was organized into three sequential phases: initial user research, prototype development, and user testing. Each phase was designed to inform and refine the subsequent stages, ensuring a systematic approach to developing a user-friendly NeRF interface. This iterative process aimed to align closely with user needs and feedback, creating a design that is both intuitive and functional.

Initial User Research The foundational stage of this research involved conducting a series of in-depth interviews to gather insights into the user experience of NeRF technology. The primary objective is to understand the varied challenges, needs, and preferences of users, ranging from novices to experts in NeRF model creation, particularly those with ties to the film industry. This exploratory phase is crucial for identifying the key features and improvements necessary for a more accessible and efficient NeRF interface. (see Chapter 5)

Design and Development Process The transition from initial user research findings to a functional prototype was a multi-step process focused on capturing user needs and translating them into a tangible design. This phase involved the creation of a user flow diagram, site map, wireframes, and a working prototype, each building upon the insights gained from the previous stage. (see Section 6.1)

User Study and Evaluation To evaluate the usability and overall utility of the developed prototype, a comprehensive user study was conducted. The primary aim of this study was to collect feedback on the prototype's user experience, identify any challenges participants encountered, and gauge their levels of satisfaction with the interface. Employing a mixed-methods approach allowed for a blend of quantitative and qualitative data collection and analysis, offering a multifaceted view of the prototype's performance in real-world tasks. (see Chapter 8)

5 User Research

5.1 Participant Selection Criteria

Participants were carefully chosen based on their prior experience with NeRF technology and their connection to the film industry, resulting in a group of four experts. This selection ensured a diversity of perspectives, encompassing a broad spectrum of technical proficiency and practical applications of NeRF. By including individuals who have utilized NeRF in various capacities, the study aimed to uncover both the shared challenges faced by all users and the unique requirements of distinct user groups within the film industry.

5.2 Interview Methodology

The interviews were designed as semi-structured conversations, following a core set of prepared questions (see Appendix A.1) while also allowing for spontaneous discussions and additional queries. Conducted one-on-one (3 online, 1 in person), these interviews facilitated a personalized dialogue with each participant, offering insights into individual experiences and perspectives. Although the interviews were prepared in English, all conversation were held in German, to ensure comfort and clarity for participants, potentially leading to more candid and informative discussions.

The structured flow of questions began with learning about the participants' backgrounds and experiences with NeRF technology, gradually moving towards more detailed questions about their specific needs, challenges, and desired improvements in NeRF interfaces. Participants were also invited to propose features or functionalities they believed would enhance the usability and effectiveness of a NeRF interface for their professional or academic projects.

To ensure a comprehensive analysis, interviews were recorded and transcribed with participants' consent, allowing for a detailed review and coding of the responses. This process enabled the identification of recurring themes, challenges, and preferences across the participant group, providing a solid foundation for the subsequent phases of prototype development and user testing. The insights gained from this initial research phase were instrumental in shaping the direction and focus of the interface design, ensuring it would effectively address the real-world needs of NeRF users.

5.3 Key Findings

NeRF in the Film Industry NeRF technology is being explored for various applications in the film industry, such as visual effects, virtual production, and pre-production location scouting. Despite its potential to simplify the creation of 3D scenes, current limitations in model quality, lack of editable models, and insufficient detail hinder its professional use. However, its capability for quick 3D scene captures offers significant benefits for pre-visualization and planning in the pre-production phase, although concerns about model scale accuracy for export remain. [**P2**, **P4**]

> *Also in der Planungsphase kam [NeRF] glaube ich ziemlich gut an, Setbegehung, Planung vorne vom eigentlichen Dreh, aber die Qualität war halt noch nicht so überzeugend.*
>
> — **Participant 4**

Optimizing Parameters and Workflow Creating NeRFs typically involves preprocessing input data, training models, and exporting outputs. Technical users emphasize the importance of parameter optimization in improving NeRF quality, with iterative training and results analysis being crucial parts of their workflow. Tools like TensorBoard [36] are utilized for quantifying variations in training outcomes. [**P1**, **P3**]

> *Probleme? Optimisierung, also Datensätze. Beim Training würde ich sagen, eine große Sache ist die Optimierung und die Parameterisierung, vor allem in Nerfstudio.*
>
> — **Participant 1**

User Interface and Accessibility A consensus among users highlights the need for an intuitive, all-encompassing user interface that minimizes reliance on console commands. Features that allow users to visually navigate and control the NeRF creation pipeline, including real-time progress feedback and the ability to pause and adjust processes at any stage, are highly valued. [**P1**, **P2**, **P3**]

> *Das Wichtigste wäre für mich, dass ich halt nichts in der Konsole machen muss. Also, dass ich einfach das Programm starte und dann alles in der UI machen kann, hochladen und dann auch irgendwie durchgeführt werde.*
>
> — **Participant 2**

Comprehensive Error Handling and Visualization Effective error feedback and clear, informative visualization tools are critical for user satisfaction. Users express frustration with vague error messages and cumbersome command-line interactions for troubleshooting and adjustments. [**P2**, **P3**]

> *Also wenn eine Fehlermeldung ist, dann wäre es halt cool, wenn die einem das irgendwie genauer sagen würden, was der Fehler ist, und dann nicht einfach so irgendein Log.*
>
> — **Participant 2**

File Management and Project Structure Efficient file and project management, with clear distinctions between different stages (preprocessing, training, rendering) and support for various input formats, is essential. Users discuss challenges with current tools regarding data organization, suggesting improvements for handling input data and managing projects. [**P1**, **P3**]

Integration and Export Options Strong integration capabilities with popular 3D and VFX software and flexible export options are desired. Users discuss the importance of being able to easily import NeRF-generated scenes into tools like Unreal Engine [@38] or Blender [@4] for further processing and use in production-quality projects. [**P1**, **P2**, **P4**]

Multi-Mode Operation The necessity for multi-mode operation in NeRF tool interfaces emerges as a significant insight, underlining the importance of accommodating a broad spectrum of users, from novices to experts. A simplified mode is envisioned to cater to beginners, offering an intuitive and streamlined workflow, whereas an advanced mode is tailored for experienced users requiring detailed control over the NeRF creation process. [**P1**, **P2**, **P3**, **P4**]

> *Es muss halt verständlich genug sein, dass Filmstudenten davon keine Angst haben. Aber trotzdem, [das Maß] wie sehr du deine Parameter ansetzen kannst [nicht] zu limitieren.*
>
> — **Participant 1**

Summary

These findings highlight the demand for a NeRF tool interface that is user-friendly, versatile, and capable of supporting a wide range of workflows and user expertise levels. The ideal tool would combine intuitive project management and visualization features with powerful customization options, robust error handling and feedback mechanisms, and effective performance management capabilities.

Application Design

6

6.1 Design Process

The transition from initial user research findings to a functional prototype was a multi-step process focused on capturing user needs and translating them into a tangible design. This section outlines the journey from abstract requirements to the creation of the prototype, emphasizing the methodologies employed at each step.

From User Research to User Flow Diagram

Following the completion of initial user research, the first step involved modeling the key findings into a user flow diagram (Figure 6.1). This diagram served as a visual representation of the user's journey through the prototype, highlighting the main actions, decisions, and interactions users would have with the system.

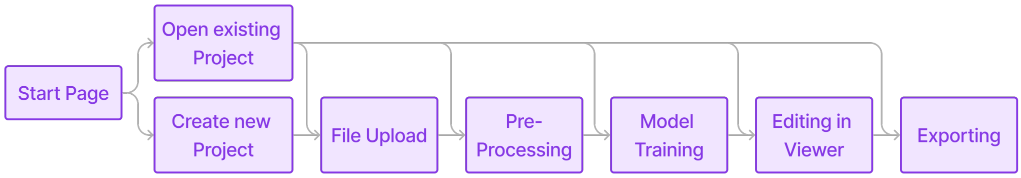


Fig. 6.1.: User Flow Diagram

The purpose of the user flow diagram was not only to map out the envisioned user experience but also to identify any potential bottlenecks or usability issues early in the design process.

Developing the Site Map

Building on the foundation laid by the user flow diagram, the next step was to expand this outline into a detailed site map. The site map provided a more comprehensive view of the prototype's structure, detailing the relationships between different pages and features. This excerpt from the site map (Figure 6.2) illustrates the level of detail and complexity involved in mapping out the prototype's user interactions.

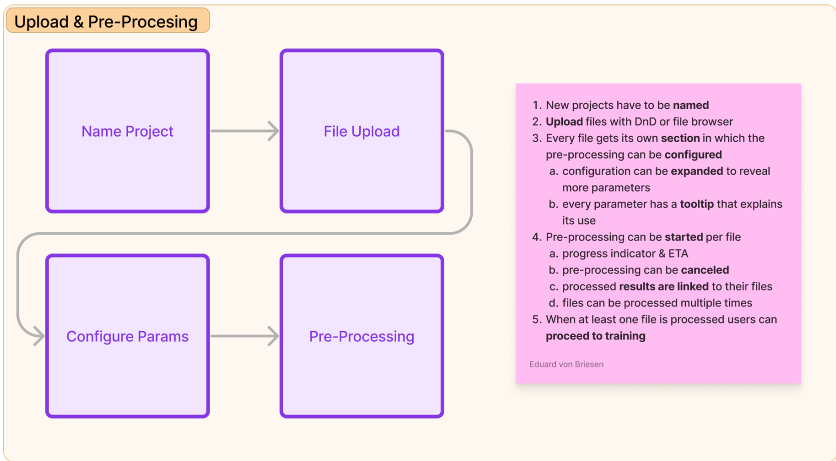


Fig. 6.2.: Excerpt of a View from the Flow Diagram with Detailed Interactions

This expanded view was instrumental in ensuring that the user flow remained intuitive across the broader system, facilitating easy navigation and a cohesive user experience.

Wireframing

With a solid understanding of the user flow and site structure, the focus shifted to wireframing. Initial wireframes were created to model the overall layout of the interface, providing a skeletal framework for the visual design. These wireframes were kept intentionally simple to prioritize structural and functional decisions over aesthetic considerations (see Fig. 6.3). At this stage, emphasis was placed on the placement of key elements, usability, and adherence to the user flow and site map.

Refinement Through Development

The transition from wireframes to a working prototype involved iterative refinement during the development phase. As the prototype took shape, initial designs were continuously evaluated and adjusted based on practical considerations and technical constraints. This phase allowed for a deeper exploration of interactions, animations, and the overall look and feel of the interface. It was during this time that the wireframes evolved into a more

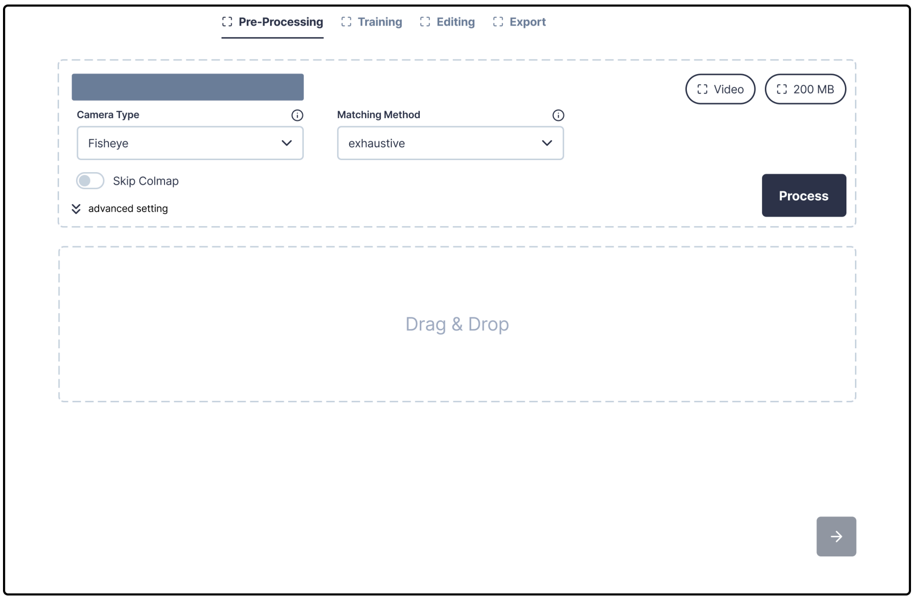


Fig. 6.3.: Wireframe of the Pre-Processing Section

detailed and user-friendly interface, with adjustments made as necessary to enhance usability and ensure a seamless user experience.

6.2 User Interface Design

The user interface was designed to be as simple as possible while still providing all necessary functionality. The design of the prototype can be broken down into two main parts: a dashboard that gives an overview of all projects and a project section that provides users with the tools to create and edit NeRF models.

Dashboard

The dashboard is the first view that users see when opening the application. It shows all previously created projects and allows users to create new ones. Projects are represented as cards, showing the project name, a preview of the provided input images (if present), and tags that indicate the current status of the project (see Fig. 6.4 1). An additional card is present through which users can create a new project, by providing a name (see Fig. 6.4 2). Projects can be opened by clicking a button on the respective card, when creating a new project, users are redirected to the project section.

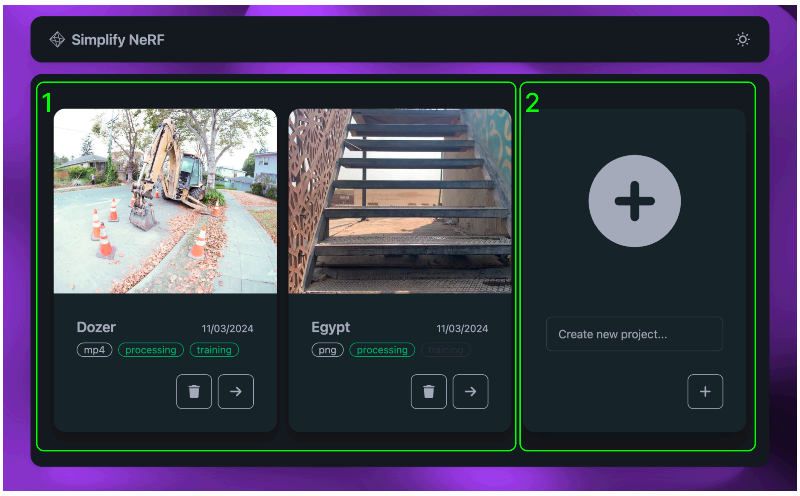


Fig. 6.4.: Screen Capture of the Dashboard

Project Section

The project section is the core of the application, through which users can create and edit NeRF models. The section is divided into three parts: the input section, the training section, and the rendering section. Across all sections, users can track their progress through a progress bar at the top of the screen, that also enables easy navigation between the different sections (see Fig. 6.5 1).

Input Section

The input section combines the first few interactions, as mapped out in the user flow diagram. First, users are prompted to upload their input data, which can be done by dragging and dropping files into the browser window or by clicking a button to open a file dialog (see Fig. 6.5 2). Files can be either a set of images or a video, and there are some guardrails in place to ensure that the input data is valid. Once the input data is uploaded, it has to be processed before it can be used for training.

The pre-processing can be configured by the user, this includes parameters such as the lens type, or matching method (see Fig. 6.5 3). Parameter input fields vary based on the type of input data and are only shown when relevant. Every parameter has a tooltip that explains its purpose, and advanced settings can be accessed by hovering over the question mark next to the parameter name. For advanced users, the settings can be further customized by activating the advanced settings using the toggle in the top right (see Fig. 6.6). Once the user is satisfied with the settings, they can start the pre-processing. Feedback on the progress of the pre-processing is given through a console that shows the output of the

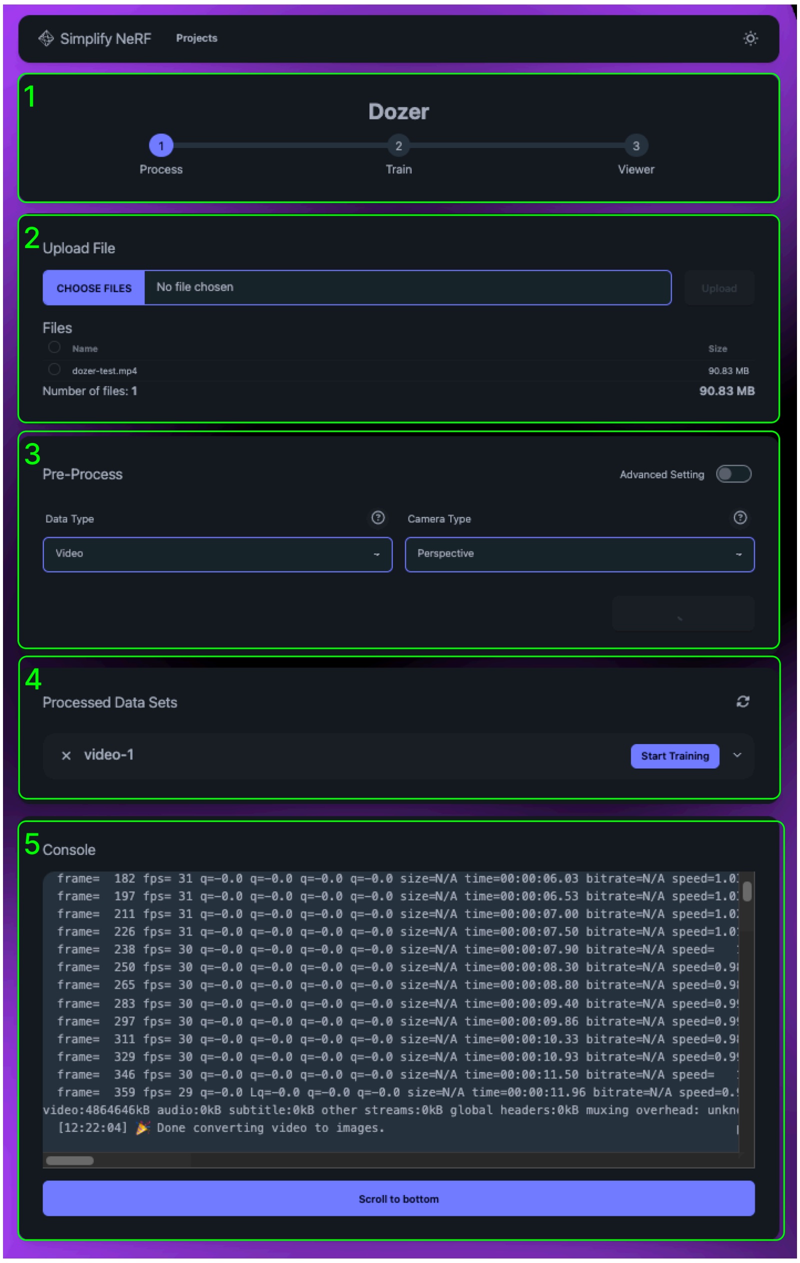


Fig. 6.5.: Processing Input Data

process running on the server (see Fig. 6.5 5). When the pre-processing is finished, the user can move on to the training section.

In case the data was already pre-processed, a list is visible that shows all available pre-processed data, and the user can select one to use for training (see Fig. 6.5 4). Users can also inspect the configuration with which the data was pre-processed, and delete it if necessary.

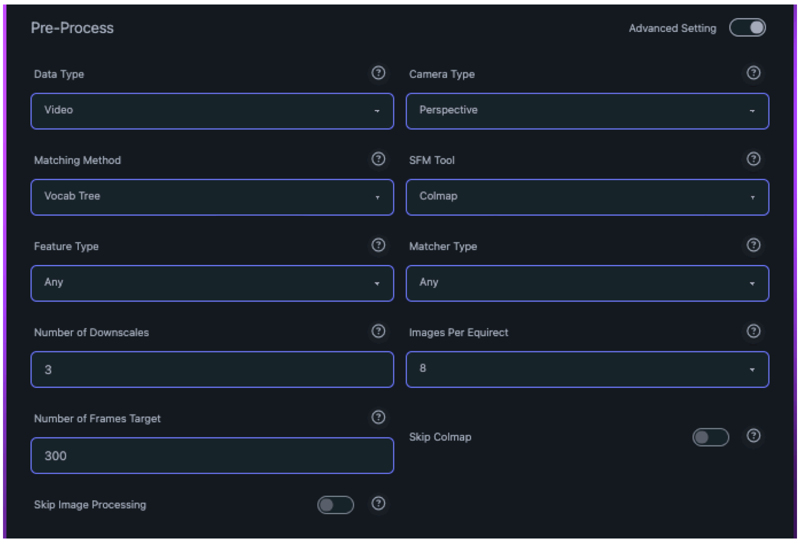


Fig. 6.6.: Advanced Settings

Training Section

The training section is structured similarly to the input section, with a form that allows users to configure the training process (see Fig. 6.7 1), and a console that shows the output of the training process running on the server. When the user is satisfied with the configuration, they can start the training process.

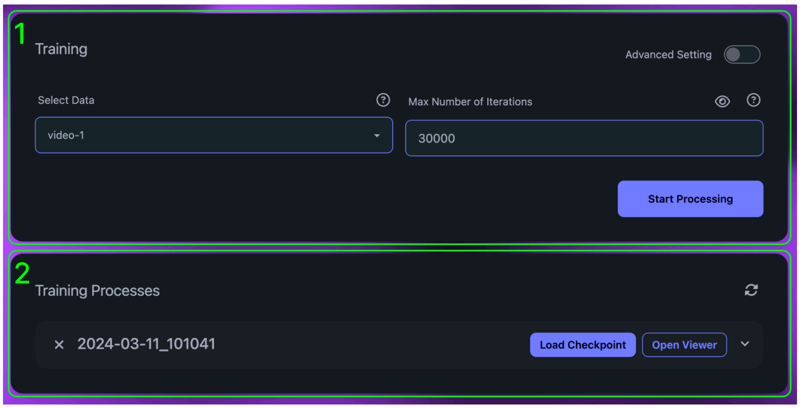


Fig. 6.7.: Training Section

Previous training runs are listed, and users can inspect the configuration with which the training was run, and delete it if necessary (see Fig. 6.5 2). The viewer can be opened to inspect the results of the training run, or an existing checkpoint can be selected to continue training from that point.

Viewer Section

The viewer section is the final step in the process, and allows users to inspect their NeRF model, while it is still training or after the training has finished (see Fig. 6.8 1). At its core is the Nerfstudio Viewer that is integrated into the application. It provides users with all the functionality available in the standalone version, with a few integrations that simplify the render process. Instead of providing commands that have to be executed in the terminal, the rendering is started by clicking a button.

The renders are listed below the viewer, and once they finished processing, they can be downloaded to the user's machine (see Fig. 6.8 2). Due to the narrow layout of the page, the viewer can easily be opened in a new tab to provide a better viewing experience.

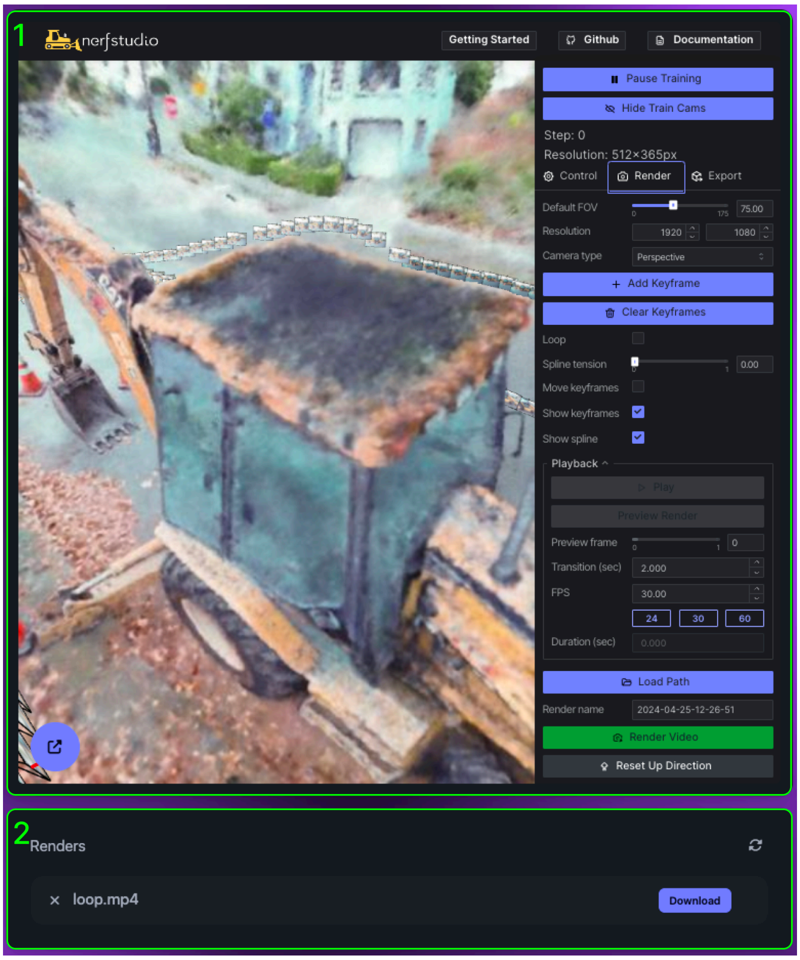


Fig. 6.8.: Viewer Section

7 Technical Implementation

7.1 System Architecture

The architecture follows a standard client-server pattern, with the server functioning as a wrapper for the nerfstudio CLI, and the client as a web application. The server is responsible for handling incoming requests from the client and translating them into commands that the nerfstudio CLI could understand. Requests from the client are sent to the server using HTTP requests, and, in the case of an asynchronous operation, the server would update the client using WebSockets.

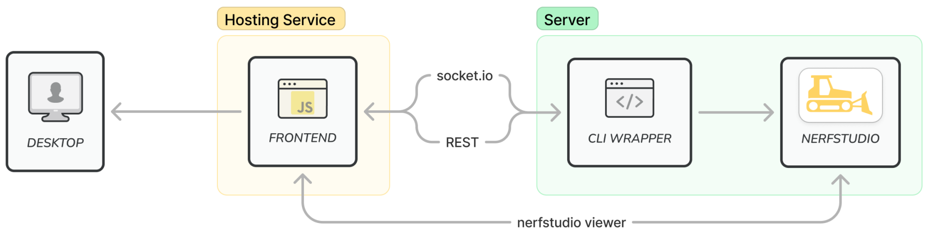


Fig. 7.1.: System Architecture

For the prototype, all components were composed into a single Docker container and deployed as a single unit. This approach leveraged the pre-configured container provided by nerfstudio and allowed for a quick and easy deployment of the prototype.

7.2 Frontend Development

The frontend is built with React [@29], chosen for its popularity and strong support in web application development. Vite [@39] serves as the build tool, offering a fast and efficient development experience. For styling, Tailwind CSS [@33], a utility-first CSS framework, provides a set of predefined classes to style components efficiently. Additionally, the daisyUI [@9] component library supplies pre-styled components, facilitating rapid UI construction.

Extensibility

Extensibility was a key consideration during the development of the frontend, as the underlying nerfstudio CLI is in itself extendable. All parameters for processing and training a NeRF model are configurable using JSON or strongly typed TypeScript objects.

```
const stepsPerSave: NumberInput = {
  name: "stepsPerSave",
  label: "Steps Per Save",
  tooltip: "Number of steps between each save of the model.",
  inputType: "number",
  defaultValue: 1000,
};
```

Listing 7.1: Parameter Option configuration

Currently, the supported input types are number, select, and boolean, but it is easy to add new types by extending the configuration object. It is also possible to define dependent parameters that are only shown when a certain condition is met. Images to illustrate the effect of a parameter can be added as well, to provide additional context to the user.

Filters and presets are also configurable using arrays of the names of the parameters that should be included in the filter or preset. The full configuration can be found in the `frontend/src/config` folder in the codebase.

Nerfstudio Viewer Integration

The nerfstudio viewer is built using Viser [24], a library for building 3D visualizations using python. This posed some limitations in integrating it into the frontend, as it is not easily possible to directly embed a python application into a web application. To work around this, the viewer was hosted by nerfstudio and embedded into the frontend using an iframe.

Some modifications could be made to the viewer to make it more suitable for embedding. This included some simple styling changes to make the viewer fit better into the frontend. More complex changes were also made to improve the user experience. The viewer contained several interactions where it was necessary for the user to copy console commands, to be used in the CLI. These interactions were replaced with buttons that send a request to the server to execute the command instead.

The main shortcoming of this approach is that the frontend is unaware of the state of the viewer, and cannot update the UI based on actions triggered in the viewer. It has to rely on polling the server to get the current state of rendering processes to give feedback to the user.

These modifications were done at build-time of the container by applying a patch to the nerfstudio source code of the base image.

7.3 Backend Development

The backend was built using tRPC [@37], a framework for building type-safe APIs in TypeScript. This type-safety was useful in building the API, as nerfstudio endpoints require a specific set of parameters of various types, that could be easily defined using TypeScript and reused in the frontend.

A major concern for the backend was the handling of asynchronous operations. Processes can often take several minutes to complete, and the user should be able to see the progress of these operations. tRPC implements a feature called subscriptions, which allows the client to subscribe to a certain event and receive updates when that event occurs. This is used for any long-running operations, such as pre-processing or training a NeRF model.

Some additional endpoints were implemented using Express. This includes the file upload endpoint, which is used to upload images and videos to the server. As well as the render endpoint, called through the viewer, as there does not exist a tRPC client for Python.

```
export const nerfstudioRouter = router({
  process: publicProcedure
    .input(
      z.object({
        project: z.string(),
        dataType: z.enum(["images", "video"]),
        ...
      }),
    )
    .subscription(({ input }) => {
      return observable<{message: string}>((emit) => {
```

```
    const args = ...
    const process = spawn("ns-process-data", args, {
     cwd: path.join(WORKSPACE, input.project),
    })
    process.stdout.on("data", (data: any) => {
     emit.next({message: data.toString()});
    });
   });
  })
});
```
Listing 7.2: Example tRPC endpoint for Pre-Processing

All project related data is stored in a workspace directory, which is mounted as a volume in the docker container. This allows for the data to persist between container restarts, and for the user to access the data outside of the container.

7.4 Challenges and Solutions

Limitations of tRPC

Although tRPC is a powerful tool for building APIs, with tight integration into the frontend, it is not without its limitations.

It lacks support for `multipart/form-data` file uploads, which are necessary for uploading images and videos. This was solved by implementing a separate endpoint using Express, which is used to upload files to the server.

The way tRPC handles subscriptions is also not ideal for long-running processes. If the client disconnects, the subscription is lost, and the client will lose the context for incoming events. A better solution could use a standard WebSocket connection, with the events containing the state necessary for the client to properly update the UI.

As mentioned above, tRPC does not have a client for Python, which is necessary for the viewer. This required the implementation of another separate endpoint using Express, which is called by the viewer to render the NeRF model.

All these limitations could be solved by using a more general-purpose framework, such as Express.

Working with the nerfstudio CLI

Building the prototype against the nerfstudio CLI offered a useful layer of abstraction. This enabled rapid development, without having to worry about the underlying implementation details of a huge codebase. However, this abstraction comes at the cost of flexibility and overall complexity of the system.

The CLI is built using tyro [42], a tool for building command line interfaces in Python using configuration objects. Efforts were made to translate the configuration objects into TypeScript objects, that could be used throughout the project. This would allow for a more seamless integration of the CLI. Sadly, this was not possible, as the configuration objects are not easily serializable, and would require a lot of additional work to implement.

Thus the implemented solution has to rely on manually constructing the commands based on documentation, which is error-prone and not very maintainable.

7.5 Future Directions

Improved CLI Integration

As outlined above, the approach of wrapping the nerfstudio CLI with a custom API has some limitations. A more integrated and robust solution would be built within the nerfstudio codebase itself.

A possible solution might be able to reuse the existing configuration object, used for CLI, to build a REST API. This would allow for a more seamless integration of the CLI into the frontend, and would allow for more flexibility in the future. It would also reduce the overall complexity of the system, as it would not require a separate server to handle requests.

Improved Viewer Integration

The current approach of embedding the viewer in an iframe is not ideal. The nerfstudio viewer in version 0.3.4 and earlier, was built using React with a direct integration of Viser. With the first major release of nerfstudio, the viewer was rewritten in Python, and the React integration was removed.

Re-building this projects frontend with a framework such as Viser, would be very limiting in terms of features and flexibility. A better solution would be to integrate Viser into the frontend, akin to the original nerfstudio viewer. This integration would allow for a more seamless interaction between the frontend and the viewer, and improve the overall user experience. UI elements could be re-used to ensure a consistent look and feel, and the frontend could react to events triggered in the viewer.

8 User Study and Evaluation

To evaluate the usability and overall utility of the developed NeRF interface prototype, a comprehensive user study was conducted. The primary aim of this study was to collect feedback on the prototype's user experience, identify any usability challenges participants encountered, and understand their satisfaction levels with the interface. Employing a mixed-methods approach allowed for a blend of quantitative and qualitative data collection and analysis, offering a multifaceted view of the prototype's performance in real-world tasks.

Participants were given a set of tasks to complete within the prototype, followed by a User Experience Questionnaire (UEQ) and a follow-up interview to gather detailed feedback on their experiences.

8.1 Participant Selection Criteria

Participants were selected in a similar fashion to the initial user research phase, with a focus on individuals working in the film industry and possessing varying levels of experience with NeRF technology, including none at all.

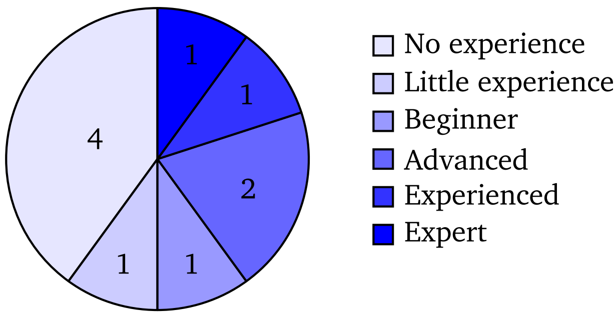


Fig. 8.1.: Participant Experience Levels

8.2 Tasks Based Usability Test

The usability test was conducted in a controlled environment, with participants being asked to complete a two tasks with the prototype. The tasks were designed to cover a range of functionalities and features of the prototype, and represent a typical workflow when creating NeRF models.

1. Task: Create a new project.
2. Task: Upload a prepared video file.
3. Task: Pre-process the uploaded file to prepare it for training.
4. Task: Switch to an existing project that already pre-processed data.
5. Task: Start a NeRF training.
6. Task: Create a camera path in the viewer.
7. Task: Export a video.

To keep an appropriate time frame, none of the tasks required completion of a training process, and pre-processed data and pre-trained models were provided to the participants. On average, participants took 30 minutes to complete the tasks.

Participant were passively observed while working on their tasks, to identify any problems or operation errors they encountered and to determine their overall performance. In addition the screen was recorded to capture the participants interactions with the prototype, and to allow for a more detailed analysis of their behavior later on.

8.3 User Experience Questionnaire

After completing their tasks, users were asked to fill out the User Experience Questionnaire (UEQ) [16], a standardized questionnaire for the assessment of user experience. The placement of the UEQ after the tasks was chosen to capture the immediate impressions of the participants, while the experience was still fresh in their minds and without being influenced by the follow-up interview. It measure user experience in six dimensions:

- **Attractiveness** - the overall impression of the product
- **Perspicuity** - the clarity and understandability of the product
- **Efficiency** - the perceived effort required to use the product
- **Dependability** - the perceived reliability and trustworthiness of the product
- **Novelty** - the perceived originality and innovation of the product

- **Stimulation** - the perceived level of excitement and engagement with the product

This covers both classical usability goals (Efficiency, Perspicuity, Dependability) and user experience qualities (Novelty, Stimulation). Attractiveness is purely a valence dimension, and is not directly related to usability or user experience.

In total the questionnaire consists of 26 items, each represented by two terms of opposite meaning. The order of the terms is randomized for each item, to avoid bias. Participants are asked to rate each item on a 7-point scale, from -3 to +3, with 0 representing a neutral response. An example of the scale can be seen below:

boring o o o o o o o exciting

The format of the questionnaire gives participants a clear and simple way to quickly express their feelings and thoughts about the prototype, without much effort.

In this study the questionnaire was filled out by participants in digital form, using a web-based survey tool. The survey included additional questions to gather demographic information and to capture prior experience with NeRF and other 3D modeling tools. This allowed for a more efficient data collection and analysis, across in-person and remote participants. source

8.4 Follow-up Interview

After completing the usability test, participants were engaged in a short follow-up interview, to gather more detailed feedback on their experience with the prototype. Similar to the initial user interviews, these interviews were semi-structured, following a predefined set of question, with room for participants to share their own thoughts and suggestions. The questions can be categorized into general usability, tasks specific feedback and suggestions for improvement. The interview template can be found in the appendix A.2.

8.5 Data Analysis

Both the recordings of the usability test and the follow-up interviews were analyzed to identify common themes and patterns in the feedback of participants. The video recordings were coded to highlight any usability issues or challenges that participants encountered during the tasks. The audio recordings of the interviews were transcribed and coded.

The data was then categorized and analyzed to identify common themes and patterns across the participants.

Analysis of the UEQ data was done using the standard procedure for the questionnaire. The UEQ provides a data analysis tool in form of spreadsheet, that calculates all necessary values and visualizes the results.

In summary, this user study and evaluation was pivotal in validating the effectiveness of the NeRF interface prototype, uncovering valuable insights into its usability, and identifying opportunities for further refinement. The mixed-methods approach ensured a balanced assessment, capturing both the tangible aspects of interface interaction and the subjective experiences of the users, providing a solid foundation for the next stages of development.

Results

9

This section presents the results of the user study conducted to evaluate the usability of the application. First the results of the quantitative user experience questionnaire are presented, to gauge the overall user experience. Following this, the results of the qualitative user testing are analyzed, to provide more detailed insights into the usability of the application. Finally, the results are integrated and discussed.

9.1 User Experience Questionnaire

Even with the relatively small sample size of 10 participants, the results of the User Experience Questionnaire (UEQ) provide a good overview of the overall user experience. The scores for the different scales of the UEQ are shown in Figure 9.1.

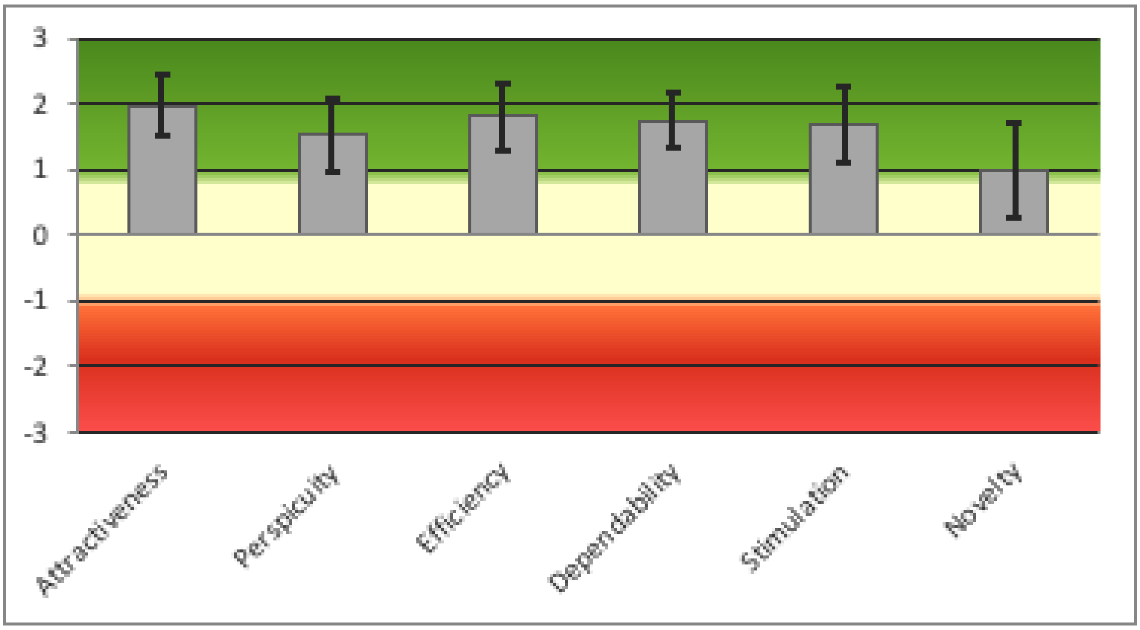


Fig. 9.1.: Results of the User Experience Questionnaire

The *Novelty* scale scores the lowest with a value just below 1 and the *Attractiveness* scale has the highest score with 1.98. The other scales score between 1.5 and 1.8, indicating a generally positive user experience.

To put the results into perspective, the scores of the UEQ can be benchmarked against the results of other studies. The UEQ provides a benchmark containing the results of 452 other studies. The benchmarking results of the UEQ are shown in Figure 9.2. *Attractiveness* and *Dependability* classify as *Excellent*, placing them in the top 10% of all studies. *Stimulation* and *Efficiency* both are classified as *Good*, while *Novelty* and *Perspicuity* only classify as *Above Average*.

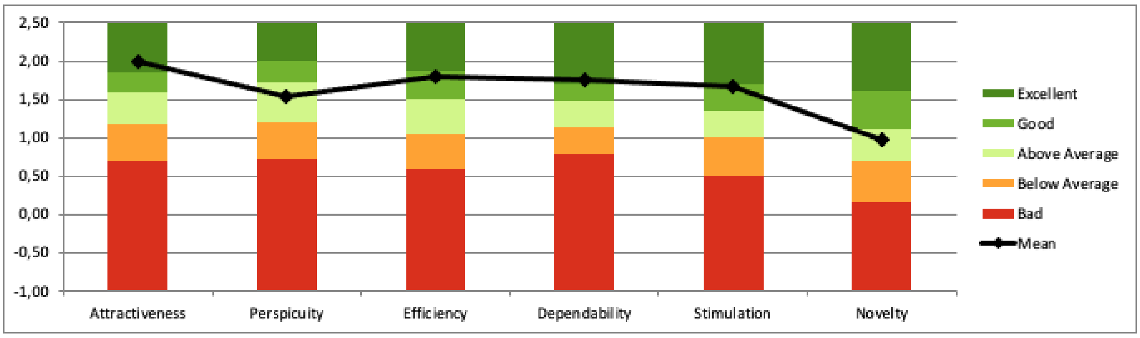


Fig. 9.2.: Benchmarking Results of the User Experience Questionnaire

specify relation between variance and confidence interval

The variance of the scores is relatively high, due to the small sample size. Across the first five scales it is between 0.49 and 0.92, only exceeding 1 for the *Novelty* scale with a value of 1.31.

Table 9.1 provides a summary of the results of the UEQ, besides the mean values and confidence intervals, shown in Figure 9.1, it also includes the standard deviation. The standard deviation can be interpreted as a measure of agreement between the participants, with lower values indicating higher agreement. Any value below 0.83 is considered *high agreement*, while values between 0.83 and 1.01 are considered *medium agreement* and values above 1.01 are considered *low agreement*. Of the six scales, three have a high agreement, two have a medium agreement and one has a low agreement.

Scale	Mean	Conf.	Conf. Int.	Std. Dev.	Agreement
Attractiveness	1,983	0,445	1,538 - 2,428	0,718	high
Perspicuity	1,525	0,573	0,952 - 2,098	0,924	medium
Efficiency	1,800	0,500	1,300 - 2,300	0,806	high
Dependability	1,750	0,432	1,318 - 2,182	0,691	high
Stimulation	1,675	0,594	1,081 - 2,269	0,958	medium
Novelty	0,975	0,710	0,265 - 1,685	1,145	low

Tab. 9.1.: Summary of the User Experience Questionnaire Results

Summary

The analysis of the UEQ results gives some insights regarding the application's user experience. Higher scores in the *Attractiveness* and *Efficiency* scales, accompanied by high agreement among participants, indicate that these aspects of the application are both effective and appealing. The consistency in these scores suggests that such attributes can be quickly gaged by users, even within the limited timeframe of the user study, leading to a uniformly positive perception.

On the other hand, the lower score and agreement on the *Novelty* scale indicate varied perceptions of the applications innovativeness. This could imply that while the application may not introduce new features or functionalities, it effectively repackages existing ones within an intuitive user interface. The moderate novelty score might reflect a use of familiar concepts and interactions, which can reduce the learning curve by leveraging well-understood mechanisms.

However, it is important to note that these findings are based on a relatively small sample size of 10 participants. While the results provide valuable insights, they should not be overvalued or considered definitive. Caution should be exercised in generalizing these findings without further validation from a larger, more diverse sample.

9.2 Findings from Qualitative User Testing

This section presents the findings from the user study conducted to evaluate the user interface of the application. The results are organized into categories reflecting overall impressions, learning curve and accessibility, identified issues, and recommendations for improvement, which now include feature requests.

Overall Impressions

Users generally find the interface user-friendly, effective, and aesthetically pleasing, enhancing the user experience across various functionalities.

> *Also es ist auf jeden Fall sehr schönes Tool, also angenehm zum Nutzen.*
>
> — **Participant 2**

User-Friendly Participants highlighted the intuitive layout and ease of navigation within the interface, appreciating how straightforward it was to perform tasks without prior training or extensive help. [**P1**, **P2**, **P4**, **P5**, **P7**, **P8**, **P10**]

Effective The effectiveness of the interface was noted in terms of its responsiveness and reliability. Users were satisfied with the speed at which the interface responded to commands and the consistency of its performance during tasks, which helped in building trust and reducing frustration during interactions. [**P2**, **P4**, **P7**, **P8**, **P10**]

Aesthetically Pleasing Many users commented on the visual appeal of the interface, mentioning the modern and clean aesthetic that made the experience more engaging. [**P7**, **P8**, **P10**]

These aspects collectively contribute to a positive user experience, making the application not only a tool that meets functional needs but also a pleasure to use, thereby encouraging repeated and prolonged engagement.

Learning Curve and Accessibility

Accessibility and ease of use are the main concerns of the application, driven to reduce the technical knowledge required to enable a broader user base. This section elaborates on the specific feedback received regarding the short comings and successes of the application in this regard.

Ease of Use As mentioned above, the overall user-friendliness was well received by participants across the board. Feedback from users indicates that the layout and workflow of the application facilitate quick learning and ease of use. [**P4**, **P5**, **P7**, **P8**, **P10**] The progress indicator as shown in Figure 6.7 was particularly appreciated, as it helped users understand the current state of the application and what steps were required to complete a task. [**P5**, **P7**, **P8**, **P9**]

Overwhelming Aspects Despite the general ease of use, some users expressed concerns about overwhelming aspects of the interface. The parameter settings often require a deep understanding of NeRF technology to understand there effects, but this did not hinder the overall usability of the application. [**P5**, **P7**] The viewer component, while powerful, presents a steep learning curve due to its complexity and the dense presentation of information and controls. New users, unfamiliar with NeRF and in particular Nerfstudio, might find this part of the application challenging to navigate initially. [**P2**]

Technical Complexity Reducing the technical complexity inherent from NeRF applications is crucial in making the application more accessible to a broader audience. Based on the feedback received from users, that were previously unfamiliar with NeRF, the application requires users to have some level of prior knowledge. The guided experience and reduced amount of options to configure, was appreciated by certain participant. It enabled them to quickly get started with the application without feeling overwhelmed. Most users felt confident, that after a short learning phase, they would be able to use the application effectively. [**P2**, **P3**, **P4**, **P5**, **P6**, **P7**, **P8**] However the feedback indicates a need for an onboarding process or tutorials to help new users overcome initial hurdles and gain confidence in using advanced features. In contrast, advanced users appreciated the depth of control and customization available. [**P9**, **P10**] The ability to fine-tune settings through the advanced options was seen as a valuable feature, allowing them to optimize the processes for their specific needs.

Project Management Novice users appreciated the similarities to existing project management of a already familiar tool from the film industry. [**P5**] For experienced users, the abstraction of tedious project management tasks, like file uploads and data organization, was well received. [**P1**] This allowed them to focus on the core tasks of training and rendering, without getting bogged down by administrative overhead.

Technical Language One particularly insightful point of concern, was the unfamiliar terminology used, as there exist differences between some terms in the context of NeRF and the film industry. [**P5**] This can lead to confusion and hinder the learning process, as users struggle to understand the meaning and implications of certain terms. Adjusting the language used in the application to specifically target the intended audience, can help bridge this gap and further reduce the learning curve.

In summary, the application succeeds in providing new users with access to advanced NeRF tools, while also catering to the needs of experienced users by offering advanced options for customization.

Identified Issues

The following issues were identified during the user testing sessions, based on feedback from participants. Most of these issues became apparent while participants were performing tasks, and then were further discussed during the post-task interviews.

Unclear Navigation Users experienced confusion about navigating through tasks and understanding their progression within the application. This was particularly evident when having to switch to a different project. The intended way to navigate to the dashboard was to click on the logo, which was not immediately clear to all users. Some users succeeded in finding the dashboard after some exploration, while others used browser navigation to return to the dashboard. [**P1**, **P4**, **P5**, **P6**, **P8**, **P10**] Some participant also mentioned uncertainties about navigating between the different sections of a project. The progress indicator can be used to navigate between the different sections, but some users took some time to discover this feature. [**P9**]

Inconsistent Wording An issue that almost many participants stumbled upon was the inconsistent wording used specifically on the button to start the training process. In the version of the application used during the user study, the button was labeled "*Start Processing*", which was confusing to most users. [**P2**, **P3**, **P6**, **P7**, **P8**]

Project Creation Similar to the wording issue, the process of creating a new project was not as intuitive as intended. Many participants encountered a small annoyance when they tried to click on the big plus icon placed in the center of the card, which was not clickable. (see Fig. 6.4) Although everyone quickly discovered the intended way to create a new project, this issue was mentioned by almost all participants. [**P3**, **P4**, **P5**, **P6**, **P7**, **P8**]

File Upload A few participant experienced issues with the file upload process. [**P1**, **P4**, **P6**] The system model is quite complex, as it involves multiple steps and the feedback provided was lacking in some cases. Before pre-processing can start, the user has to select the input data to use, upload it to the server, wait for the upload to complete before continuing. Users were not always aware of the progress, and would sometimes continue to the next step before the upload was complete.

Viewer and Screen Layout Feedback indicates a need for a more flexible UI that adjusts to different screen sizes and supports fullscreen modes. [**P10**]

add more details

Awerness of Progress In some cases, users were not aware in what stage of the creation process they were in. This was somewhat caused by the inconsistent wording (see 9.2), but also by the high degree of similarity between the pre-processing and training steps. [**P2**]

Console Output The console output was perceived differently by users. Some users appreciated the detailed feedback and insights provided, even with no technical background. [**P5**] Others that could not interpret the output, found it useless and distracting. [**P3**, **P4**]

Recommendations for Improvement

This section outlines some of the recommendations for improving the application based on the feedback received from participants. In contrast to the identified issues, these recommendations are more general and focus on improving the overall user experience.

Enhance Interface Usability To address the need for a more personalized and less distracting user interface, the animated background should be able to be disabled. Additionally, implementing a 'back-to-top' button would streamline navigation, enabling users to quickly return to the top of the page without manual scrolling. [**P10**]

Viewer Customization and Usability Enhancements Several enhancements to the viewer are recommended to improve its functionality and usability:

- **Multiple Viewing Modes**: Incorporate the ability to view the cameras perspective, as well as the scene from different angles, providing users with a more comprehensive understanding of the scene. [**P5**]
- **Advanced Camera Controls**: Introduce adjustable controls for camera speed, motion blur, and the ability to set up speed ramps, to provide users with greater creative control. [**P5**]
- **Measurement Tools**: Add tools that enable users to take precise measurements within the viewer, useful for detailed scene planning and analysis. [**P7**]

The participants requesting these features were exclusively users with a background in film production, who would include these features in their workflow.

Expand Advanced Settings and Benchmarks User with an academic background in NeRF requested the following features, when evaluating the application for their workflow:

- **Support for different NeRF implementations**: Allow users to select different NeRF implementations, providing flexibility and customization options based on their requirements. [**P1**]
- **Advanced-Advanced Settings**: Provide deeper customization options for experienced users who require fine control over parameters, possibly allowing them to inject custom settings or scripts. [**P9**]
- **Integration of Benchmarking Tools**: Integrate tools like TensorBoard or WandDB to allow users to perform benchmarks, providing insights into model performance. [**P10**]

Camera and Shot Management Develop a more structured approach to camera and shot management within the application to support complex productions. Allow users to label and organize different camera shots and paths, making it easier to manage multiple views or scenes within a single project. [**P5**]

Tighter Integration of Viewer One participant suggested a tighter integration of the viewer with the rest of the application. Building the viewer controls directly into the interface, instead of inserting the viewer as a separate component, to provide a more seamless experience. [**P9**]

General Feedback

Lack of Modification Options Creatives would like to see more options for modifying the scene, such as adding objects or changing the lighting. [**P3**]

add more details

Use Cases Some participants saw no potential use cases for the application or NeRF in general in their current workflow. [**P3**] Others cited the limitation to static scenes as a major drawback, as they require dynamic scenes for their projects. [**P4**, **P5**] But the possibility for stylized and 'impossible' camera movements was inspiring application in advertisement and music videos. [**P4**] Pre-visualization was also mentioned as a potential use case, as it allows for quick and easy scene creation that can be used for planning and storyboarding. [**P4**]

9.3 Integration and Findings

Add integration and findings

10 Discussion

Write discussion

10.1 Interpretation of Results

10.2 Implications for the Film and VFX Industry

10.3 Integration of User Feedback

Based on the issues identified during user testing, several adjustments were made to enhance the usability and intuitiveness of the application. These changes are outlined below:

Improved Navigation To address the confusion in navigation to the dashboard (**??**), a dedicated button was added to the navigation bar. This feature has should improved the clarity of navigation to users significantly. (see Fig. 10.1)

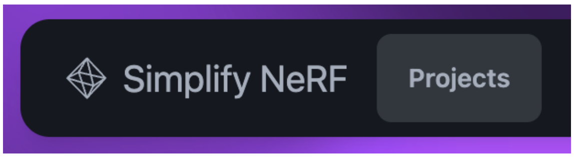


Fig. 10.1.: Dedicated Button for Dashboard Navigation

Clarified Wording The wording on the button to start the training process was changed from "*Start Processing*" to "*Start Training*", which aligns better with user expectations and reduces confusion. (see Fig. 10.2)

Enhanced Project Creation The project creation process was moved into a modal dialog, which not only eliminates a point of confusion but also clarified the need to name projects before creation. (see Fig. 10.3)

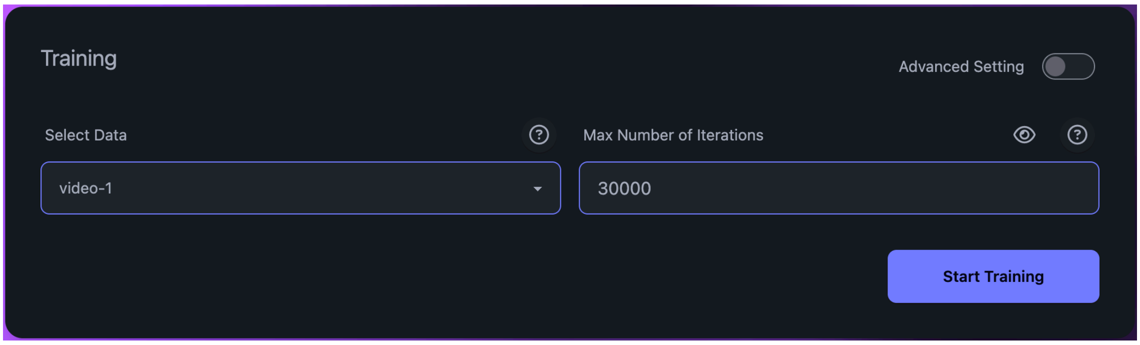


Fig. 10.2.: Consistent Wording for Training Button

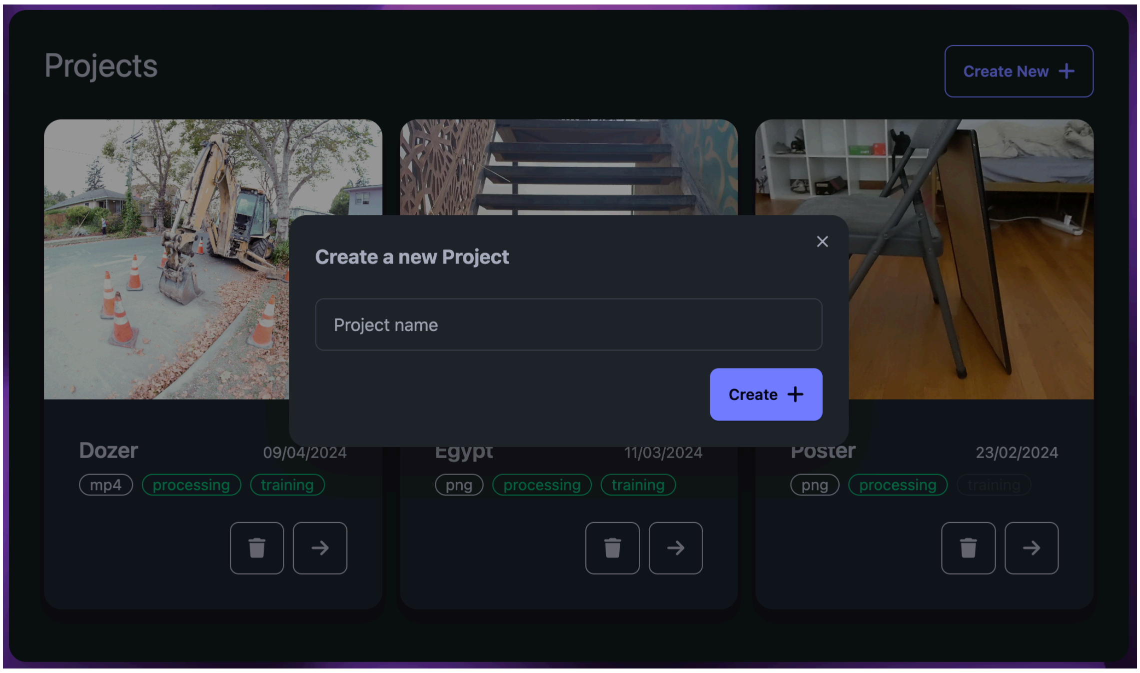
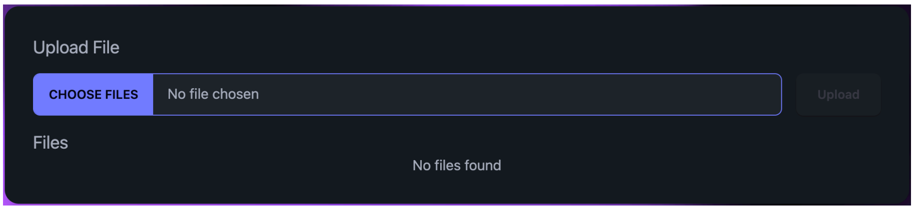


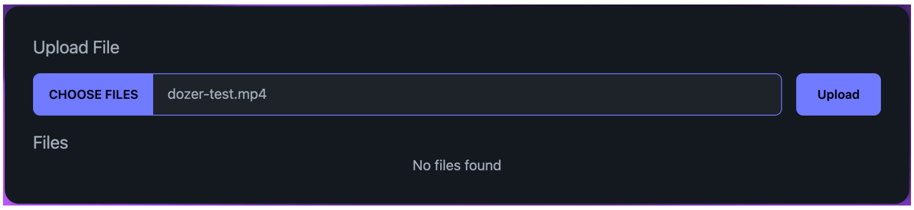
Fig. 10.3.: Modal Dialog for Project Creation

File Upload Improvements The file upload process was improved by adding some guardrails, to ensure user would not accidentally skip a step. The upload button starts out disabled, so that the only interactive element is the file-input field. Once a file is selected, the button becomes active, indicating to the user that they can proceed to upload their selected file. Only one the file is uploaded, the UI elements related to pre-processing appear, guiding the user through the next steps. This solution is likely to prevent many of the issues users encountered when uploading files during testing. (see Fig. 10.4)

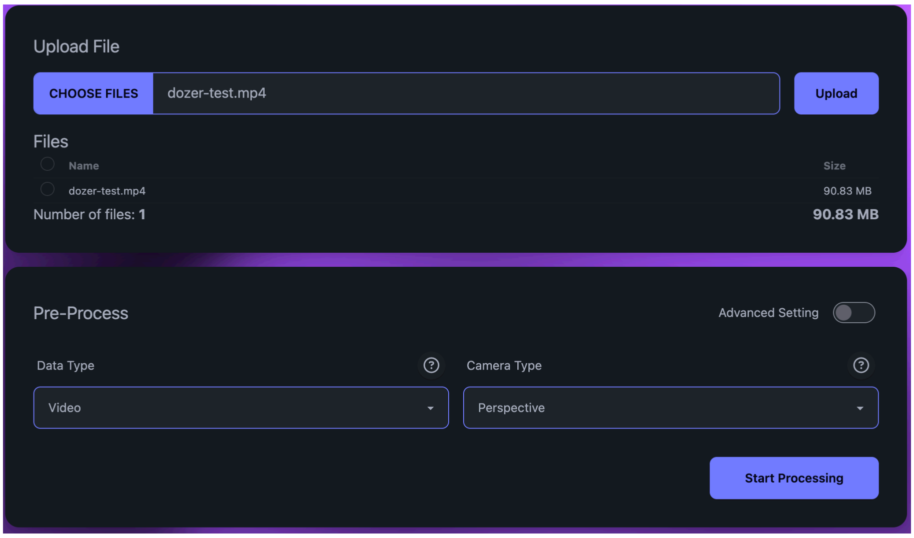
10.4 Limitations



(a) Initial State



(b) File Selected



(c) File Uploaded

Fig. 10.4.: Improved File Upload Process

11 Conclusion

Write conclusion

11.1 Key Findings

11.2 Contributions to the Field

11.3 Future Work

Appendix

A

bring both ques-tion-naires into the same format

A.1 Interview Questions

Introduction

- Thank the interviewee for their participation.
- Briefly explain the purpose of the interview, which is to gather insights for the development of a NeRF interface.
- Assure the interviewee that their responses will be kept confidential.

Background

- Can you briefly describe your experience with NeRF or 3D modeling in general?

Needs and Challenges

- What specific tasks or objectives do you typically aim to achieve when working with NeRF models or 3D scenes?
- What are the main challenges or pain points you encounter when using current NeRF frameworks or interfaces?

Usability and Features

- In your opinion, what features or functionalities would make a NeRF interface most useful for your work or research?
- Are there any specific editing or manipulation tools you find lacking in current NeRF interfaces?
- How important is real-time visualization and interactivity in a NeRF interface for your needs?

Ease of Use

- How do you envision the ideal user interface for NeRF? What elements would make it easy to use, even for those with limited technical expertise?

- What level of technical knowledge or familiarity with 3D modeling should the ideal NeRF interface require from its users?

Integration and Compatibility

- Are there any other software tools or workflows you typically use alongside NeRF, and how important is it for a NeRF interface to integrate with these tools?
- Do you have any preferences regarding the file formats or data compatibility that the NeRF interface should support?

Feedback and Suggestions

- Are there any additional thoughts, suggestions, or requirements you would like to share regarding the development of a NeRF interface?
- Is there anything else you believe is essential for us to understand about your needs and expectations?

Closing

- Thank the interviewee for their time and valuable input.
- Provide contact information for any follow-up questions or clarifications.

A.2 User Study Questionnaire

Usability Experience

- Can you share your overall impressions of using the Simplify NeRF application?
- Were there any specific features or functionalities of the application that stood out to you positively? If so, why?
- On the other hand, were there any aspects of the application that you found challenging or frustrating? Please elaborate.

Task-Specific Feedback

- Were there any particular steps or actions within the tasks that you found confusing or unclear? If yes, could you describe them?
- Were there any features or functionalities you expected to find in the application that were missing? If so, what were they?

Suggestions for Improvement

- Based on your experience using the Simplify NeRF application, do you have any suggestions for improving its usability or functionality?
- Are there any specific changes or enhancements you would like to see in future versions of the application?
- How do you think the application could better meet your needs or expectations as a user?

Closing

- Is there anything else you would like to add or share about your experience with the Simplify NeRF application?
- Thank the participant for their time and valuable feedback. Offer contact information for any follow-up questions or clarifications.

A.3 User Testing Results

A.4 Prototype Documentation

Bibliography

[2] Chong Bao, Yinda Zhang, Bangbang Yang, et al. "SINE: Semantic-driven Image-based NeRF Editing with Prior-guided Editing Field". In: *The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*. 2023 (cit. on pp. 2, 8).

[3] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. *Text2LIVE: Text-Driven Layered Image and Video Editing*. _eprint: 2204.02491. 2022 (cit. on pp. 2, 8).

[5] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. "Unstructured lumigraph rendering". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, Aug. 1, 2001, pp. 425–432 (cit. on p. 5).

[6] Qifeng Chen and Vladlen Koltun. "Photographic Image Synthesis with Cascaded Refinement Networks". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. ISSN: 2380-7504. Oct. 2017, pp. 1520–1529 (cit. on p. 7).

[7] Shenchang Eric Chen and Lance Williams. "View interpolation for image synthesis". In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '93. New York, NY, USA: Association for Computing Machinery, Sept. 1, 1993, pp. 279–288 (cit. on p. 5).

[8] Brian Curless and Marc Levoy. "A volumetric method for building complex models from range images". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, Aug. 1, 1996, pp. 303–312 (cit. on p. 6).

[10] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. "Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, Aug. 1, 1996, pp. 11–20 (cit. on p. 5).

[11] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. "Deep Stereo: Learning to Predict New Views from the World's Imagery". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). ISSN: 1063-6919. June 2016, pp. 5515–5524 (cit. on p. 7).

[12] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. "The lumigraph". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, Aug. 1, 1996, pp. 43–54 (cit. on p. 5).

[13] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. "Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023 (cit. on pp. 2, 8, 9).

[14] Jan-Niklas Dihlmann, Andreas Engelhardt, and Hendrik P. A. Lensch. "SIGNeRF: Scene Integrated Generation for Neural Radiance Fields". In: *arXiv.org* (2024) (cit. on pp. 8, 9).

[15] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. "Learning-based view synthesis for light field cameras". In: *ACM Transactions on Graphics* 35.6 (Nov. 11, 2016), pp. 1–10 (cit. on p. 6).

[16] Bettina Laugwitz, Theo Held, and Martin Schrepp. "Construction and Evaluation of a User Experience Questionnaire". In: USAB 2008. Vol. 5298. Nov. 20, 2008, pp. 63–76 (cit. on p. 31).

[17] Marc Levoy and Pat Hanrahan. "Light field rendering". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, Aug. 1, 1996, pp. 31–42 (cit. on p. 5).

[18] Stephen Lombardi, Tomas Simon, Jason Saragih, et al. "Neural volumes: learning dynamic renderable volumes from images". In: *ACM Transactions on Graphics* 38.4 (Aug. 31, 2019), pp. 1–14 (cit. on p. 6).

[19] Maxim Tatarchenko, Maxim Tatarchenko, Alexey Dosovitskiy, et al. "Single-view to Multi-view: Reconstructing Unseen Views with a Convolutional Network". In: *arXiv: Computer Vision and Pattern Recognition* (2015) (cit. on p. 6).

[20] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, et al. "Local light field fusion: practical view synthesis with prescriptive sampling guidelines". In: *ACM Transactions on Graphics* 38.4 (Aug. 31, 2019), pp. 1–14 (cit. on p. 6).

[21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, et al. "NeRF: representing scenes as neural radiance fields for view synthesis". In: *Communications of the ACM* 65.1 (Dec. 17, 2021), pp. 99–106 (cit. on p. 7).

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding". In: *ACM Trans. Graph.* 41.4 (July 2022). Place: New York, NY, USA Publisher: ACM, 102:1–102:15 (cit. on pp. 2, 8, 9).

[24] *nerfstudio-project/viser*. original-date: 2022-11-16T00:13:51Z. Mar. 25, 2024 (cit. on pp. 9, 25).

[25] Eric Penner and Li Zhang. "Soft 3D reconstruction for view synthesis". In: *ACM Transactions on Graphics* 36.6 (Dec. 31, 2017), pp. 1–11 (cit. on p. 6).

[26] Peter Hedman, Peter Hedman, Julien Philip, et al. "Deep blending for free-viewpoint image-based rendering". In: *ACM Transactions on Graphics* (2019) (cit. on p. 6).

[28] Pratul P. Srinivasan, Pratul P. Srinivasan, Richard Tucker, et al. "Pushing the Boundaries of View Extrapolation With Multiplane Images". In: (2019) (cit. on p. 6).

[31] Steven M. Seitz and Charles R. Dyer. "Photorealistic Scene Reconstruction by Voxel Coloring". In: *International Journal of Computer Vision* 35.2 (Nov. 1, 1999), pp. 151–173 (cit. on p. 6).

[32] Vincent Sitzmann, Justus Thies, Felix Heide, et al. "DeepVoxels: Learning Persistent 3D Feature Embeddings". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). ISSN: 2575-7075. June 2019, pp. 2432–2441 (cit. on p. 6).

[34]Matthew Tancik, Ethan Weber, Evonne Ng, et al. “Nerfstudio: A Modular Framework for Neural Radiance Field Development”. In: *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH ’23. 2023 (cit. on pp. 2, 8, 9).

[35]Tao Zhou, Tinghui Zhou, Richard Tucker, et al. “Stereo Magnification: Learning View Synthesis using Multiplane Images”. In: *arXiv: Computer Vision and Pattern Recognition* (2018) (cit. on p. 6).

[36]*tensorflow/tensorboard*. original-date: 2017-05-15T20:08:07Z. Apr. 24, 2024 (cit. on p. 14).

[40]Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. “CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 3835–3844 (cit. on pp. 2, 8).

[41]Qiling Wu, Jianchao Tan, and Kun Xu. *PaletteNeRF: Palette-based Color Editing for NeRFs*. _eprint: 2212.12871. 2022 (cit. on pp. 2, 8).

[42]Brent Yi. *brentyi/tyro*. original-date: 2021-10-05T08:54:08Z. Mar. 26, 2024 (cit. on p. 28).

[43]Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, et al. “NeRF-Editing: Geometry Editing of Neural Radiance Fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 18353–18364 (cit. on pp. 2, 8).

Webpages

[@1]Luma AI. *Luma AI*. Luma AI. URL: `https://lumalabs.ai/` (visited on Apr. 25, 2024) (cit. on p. 10).

[@4]*Blender - Free and Open 3D Creation Software*. URL: `https://www.blender.org/` (visited on Apr. 25, 2024) (cit. on p. 15).

[@9]*daisyUI Tailwind CSS Components (version 4 update is here)*. URL: `https://daisyui.com/` (visited on Mar. 25, 2024) (cit. on p. 25).

[@23]*Nerfacto*. URL: `http://docs.nerf.studio/nerfology/methods/nerfacto.html` (visited on Apr. 25, 2024) (cit. on p. 9).

[@27]*Polycam - LiDAR & 3D Scanner for iPhone & Android*. URL: `https://poly.cam/` (visited on Apr. 23, 2024) (cit. on p. 10).

[@29]*React*. URL: `https://react.dev/` (visited on Mar. 25, 2024) (cit. on p. 24).

[@30]*Record3D 3D Videos and Point Cloud (RGBD) Streaming for iOS*. URL: `https://record3d.app/` (visited on Apr. 23, 2024) (cit. on p. 10).

[@33]*Tailwind CSS - Rapidly build modern websites without ever leaving your HTML*. Nov. 15, 2020. URL: `https://tailwindcss.com/` (visited on Mar. 25, 2024) (cit. on p. 24).

[@37]*tRPC - Move Fast and Break Nothing. End-to-end typesafe APIs made easy. | tRPC*. URL: `https://trpc.io/` (visited on Mar. 25, 2024) (cit. on p. 26).

[@38]*Unreal Engine*. Unreal Engine. URL: `https://www.unrealengine.com/en-US/home` (visited on Apr. 25, 2024) (cit. on p. 15).

[@39]*Vite*. URL: `https://vitejs.dev` (visited on Mar. 25, 2024) (cit. on p. 24).

List of Figures

List of Tables