Dreamcrafter: Imagining Future Immersive Radiance Field Editors with Generative AI

Cyrus Vachha

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-124
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-124.html

May 17, 2024

Acknowledgement

I would like to thank my advisor Professor Bjoern Hartmann for his mentorship, support, and advice on my research at Berkeley. I would also like to thank Bala Kumaravel for his mentorship and introducing me to VR research at Berkeley. Thank you to members of the BiD lab for assisting me in my project and giving their feedback and support. Thank you to the Nerfstudio team and Professor Angjoo Kanazawa for their support and introducing me to NeRF research at Berkeley. I'm grateful for the support from my co-authors, colleagues, and mentors during my undergraduate and graduate projects at Berkeley and Microsoft. I acknowledge my other co-authors for Dreamcrafter: Yixiao Kang, Zach Dive, Ashwat Chidambaram, Anik Gupta, and Eunice Jun. I would also like to thank my family for their support.

Dreamcrafter: Imagining Future Immersive Radiance Field Editors with Generative AI

by

Cyrus Vachha

A thesis submitted in partial satisfaction of the

requirements for the degree of

Masters of Science

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Björn Hartmann, Chair
Professor Angjoo Kanazawa

Spring 2024

Dreamcrafter: Imagining Future Immersive Radiance Field Editors with Generative AI

Dreamcrafter: Imagining Future Immersive Radiance Field Editors with Generative AI

Cyrus Vachha

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee

Björn Hartmann
Research Advisor

May 10, 2024
(Date)

⋆ ⋆ ⋆ ⋆ ⋆ ⋆ ⋆

Angjoo Kanazawa
Second Reader

May 14, 2024
(Date)

Abstract

Dreamcrafter: Imagining Future Immersive Radiance Field Editors with Generative AI

by

Cyrus Vachha

Masters of Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Björn Hartmann, Chair

Authoring 3D scenes is a central task for spatial computing applications. Two competing visions for lowering existing barriers are (1) focus on immersive, direct manipulation of 3D content; or (2) leverage recent techniques that capture real scenes (3D Radiance Fields such as NeRFs, 3D Gaussian Splatting) and modify them at a higher level of abstraction, at the cost of high latency. We unify the complementary strengths of these approaches and investigate how to integrate generative AI advances into real-time, immersive 3D Radiance Field editing. We introduce Dreamcrafter, a VR-based 3D scene editing system that: (1) provides a modular architecture to integrate generative AI systems; (2) combines different levels of control for creating objects, including natural language and direct manipulation; and (3) introduces proxy representations that support interaction during high-latency operations. We also contribute empirical findings about how and when people prefer different controls when working with radiance fields in a user study. Avenues for future work for developing interactions and features for multi-modal 3D editors leveraging generative models and radiance fields are also discussed.

To my family

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor Professor Björn Hartmann for his mentorship, support, and advice on this project and my research at Berkeley. I would also like to thank Bala Kumaravel for his mentorship and introducing me to VR research at Berkeley. Thank you to the members of the BiD lab for assisting me in my project and giving their feedback and support. Thank you to the Nerfstudio team and Professor Angjoo Kanazawa for their support and introducing me to NeRF research at Berkeley. I'm grateful for the support from my co-authors, colleagues, and mentors during my undergraduate and graduate projects at Berkeley and Microsoft. I would also like to acknowledge my other co-authors for Dreamcrafter: Yixiao Kang, Zach Dive, Ashwat Chidambaram, Anik Gupta, and Eunice Jun. I would also like to thank my family for their support.

Chapter 1

Introduction

1.1 Overview

Recent advances in photo-realistic novel 3D representations like NeRFs and 3D Gaussian Splatting offer exciting new avenues for 3D world creation and reconstruction for immersive worlds. Neural Radiance Fields (NeRFs) [30] ushered a wave of exploration in the development and applications of similar novel 3D representations. Concurrently, there has been a wave of research in generative AI models, most notably text-to-image diffusion models and large language models. These advances have lowered the barrier to entry for artistic creation of 3D scenes and objects while also enabling new forms of interfaces and interactions through text-based prompting and multi-modal input.

Immersive systems like Virtual Reality and Augmented Reality present an ideal medium for viewing and embodying these 3D radiance field representations due to offering a photorealistic dreamlike experience beyond traditional 3D representations. Displaying radiance fields in VR as demonstrated in Welcome to Light Fields [33] and Immersive Light Field Video [4] show the potential in viewing photo-realistic scenes in VR. NeRFs and 3DGS show potential as new forms of 3D representations due to their high fidelity and are easier to capture than traditional 3D scanning or modeling methods. However these representations lack controllability in editable capabilities which justifies further exploration into how to leverage these representations in traditional or new creative workflows.

Despite the increased interest in these 3D representations and a significant effort in improving quality and adding capabilities in graphics and vision research, there has been a lack of exploration in novel user interfaces and systems that leverage radiance fields in new ways. Photo-realistic representations like radiance fields as well as generative AI systems offer new interactions, and require a re-imagining of existing 3D toolsets and systems to leverage these new capabilities these representations have to offer. Recent commercial and industry interest is growing to develop ways to effectively leverage generative AI tools in creative systems and workflows. Understanding how users interact with these new mediums through user studies could help develop better ways to create effective systems and design a new wave of creative tools.

In this thesis, I explore components of a future 3D editor for interfacing with and generating 3D radiance field scenes. In the following chapter, we introduce Dreamcrafter [47], an early exploration and implementation of a system that investigates ideas discussed. In our paper, we perform user studies to evaluate how users interact with the scene editing and creation tools presented at different levels of control as enabled by generative AI tools.

The following chapter discusses methods for editing these novel 3D representations by mentioning related work, modalities for editing through natural language or from traditional workflows, and stylizing scenes. The final chapter discusses potential future work and research topics to explore this vision of methods for capturing, editing, and viewing/interacting with these 3D representations and generative models fully within VR. This chapter is based on ideas based on an early proposal of this project [45]. Despite this discussion primarily focusing on VR interfaces due to additional affordances from spatial interactions, these systems and ideas are generalizable to 2D mediums and interfaces as well.

Chapter 2

Dreamcrafter

The following chapter is a research project (Dreamcrafter: Immersive Editing of 3D Radiance Fields Through Flexible, Generative Inputs and Outputs [47]) that is co-written with my collaborators Yixiao Kang, Zach Dive, Ashwat Chidambaram, Anik Gupta, Eunice Jun, and Björn Hartmann.

2.1 Abstract

Authoring 3D scenes is a central task for spatial computing applications. Two competing visions for lowering existing barriers are (1) focus on immersive, direct manipulation of 3D content; or (2) leverage AI techniques that capture real scenes (3D Radiance Fields such as NeRFs, 3D Gaussian Splatting) and modify them at a higher level of abstraction, at the cost of high latency. We unify the complementary strengths of these approaches and investigate how to integrate generative AI advances into real-time, immersive 3D Radiance Field editing. We introduce Dreamcrafter, a VR-based 3D scene editing system that: (1) provides a modular architecture to integrate generative AI algorithms; (2) combines different levels of control for creating objects, including natural language and direct manipulation; and (3) introduces proxy representations that support interaction during high-latency operations. We also contribute empirical findings about how and when people prefer different controls when working with Radiance Fields in a user study.

2.2 Introduction

Spatial computing applications such as Augmented and Virtual Reality rely on 3D content and scenes. Thus, creating appropriate tools for authoring and editing 3D content has been a long-standing key challenge for HCI researchers.

Traditionally, mesh-and-texture-based approaches have been used to author 3D content. Various research efforts to introduce better editing techniques notwithstanding (e.g., [20, 1]), the expertise hurdle to create and modify 3D content in this way has been high, generally leaving such authoring to a small number of expert users.

One avenue to lower the authoring barrier has been to embrace authoring in VR (e.g. Google Tiltbrush [5]), where direct 3D input is possible through VR controllers (or gestures) in an immersive environment. This approach decreases the gulf of execution [19] inherent in prior approaches to modeling 3D content using 2D input devices.

More recently, two additional developments hold the promise of reducing authoring burdens. First, novel approaches for representing 3D scenes based on radiance fields (e.g., NeRFs [30] and 3D Gaussian Splatting [24]) allow for straightforward capture of photorealistic environments from real scenes using common cameras, instead of having to model objects from scratch. Second, generative AI developments have introduced novel ways of editing radiance field scenes at higher levels of abstraction, e.g. through text instructions (as in Instruct-NeRF2NeRF [17]). While offering the ability to edit at a semantic level rather than a lower geometry level, such techniques also tend to be compute-intensive and not yet amenable to run in realtime.

The different approaches—rapid direct manipulation on one hand and high-level instruction-based editing on the other hand—recall long-standing arguments in the HCI community on the benefits of direct control vs. delegation [41]. In this paper, we investigate if it is possible to unify the complementary strengths of real-time, immersive editing on the one hand, and generative AI-based approaches to high-level scene editing (with high latency) on the other hand under a common interaction framework.

We introduce Dreamcrafter, a Virtual Reality 3D content generation and editing system assisted by generative AI. The core idea behind Dreamcrafter is to use direct manipulation for spatial positioning and layout; and leverage generative AI for editing style and appearance of objects. Because generative AI edits are unlikely to run in real-time, Dreamcrafter introduces rapid proxy representations, e.g. using a 2D diffusion model to create a stand-in image for a longer-running 3D generative task. Dreamcrafter enables both 2D (image) and 3D output.

Dreamcrafter makes three core contributions: 1) it provides a modular architecture to integrate different generative AI algorithms into editing interactions; 2) it combines different levels of control for creating objects, including direct manipulation and natural language; and 3) it introduces proxy representations that support ongoing interactions during high-latency operations on 3D content.

We investigate how users decide between different levels of control over a scene and how they use proxy representations through a first-use study with seven participants. We find that participants use different levels of control across tasks even though they create more objects using AI only. Yet, participants report feeling more in control when they can sculpt object shapes and then use generative AI to stylize those objects rather than create objects using generative AI alone. Participants also find the proxy representations useful for reasoning about scene composition.

2.3 Background

We give a brief overview of Radiance Fields (NeRFs and Gaussian Splatting) and Stable Diffusion.

Radiance Fields. Recent years have seen a move from traditional 3D graphics using meshes and geometries to more photorealistic rendering techniques, such as Neural Radiance Fields (NeRFs) [30] and Gaussian Splatting [24]. Radiance fields are 3D representations of scenes or objects, as a function of radiance given position and view direction, that can exhibit photorealistic view-dependent effects. NeRFs are 3D representations that optimize a volumetric 3D scene as a radiance field using a neural network trained on a set of images. 3D Gaussian Splatting (3DGS) is visually akin to NeRFs. 3D Gaussian Splatting represents scenes with 3D Gaussians to support faster training and rendering via differentiable rasterization for high-quality real-time visualizations. These techniques have been shown to be highly effective at modeling details with realistic lighting, shadowing, and surfaces for real-world captures. And, with the increase in applications requiring 3D content, these models can be effectively used to quickly capture and create assets.

Stable Diffusion. Stable Diffusion [40] is a deep learning model for synthesizing, or generating, images from text inputs using a diffusion model. ControlNet [54] is a network architecture enhancement to text-to-image models, like Stable Diffusion to condition the model on an input image, generating stylized outputs. The conditioning could be filters such as depth or edges that preserve those features of the original input image in the generation.

2.4 Related Work

The most related prior work falls into two areas: 1) novel 3D scene representations and tools for using them and 2) creation systems in VR. We review each area in turn.

Generating and editing novel 3D representations using NeRFs and 3DGS

Several recent rendering techniques build upon NeRFs [30] and 3D Gaussian Splatting (3DGS) [24]. For example, LERF [25], or Language Embedded Radiance Fields use CLIP embeddings [37] to allow users to query a NeRF using natural language to determine regions of interest. ConceptGraphs [15] uses a similar technique with CLIP embeddings but processes a more traditional 3D representation of point clouds rather than NeRFs. These developments have indicated the importance of object-centric labeling and editing in the systems that are built and have guided our design of editing components of 3D scenes in VR. By focusing on radiance fields, we hope to lower the barriers to a wide range of representations, tasks, and applications in the future.

While effective, these 3D representations are difficult for end users to create, manipulate, and use. Recent efforts to make NeRFs more approachable have included consumer facing systems such as Luma AI [28], and research friendly APIs such as Nerfstudio [43] and Instant-NGP [32]. Instruct-NeRF2NeRF [17] allow users to provide as input a text prompt and an existing NeRF and output a new NeRF stylized according to the text prompt, relying on 2D text and image conditioned diffusion models such as InstructPix2Pix [2]. We interface with Instruct-NeRF2NeRF to allow users to edit their 3D scenes and objects. 3D Generative models such as DreamFusion [35] or Shap-E [23] also allow users to build a NeRF or mesh from a text prompt.

More recently, concurrent works such as GaussianEditor [12] and SigNeRF [9] demonstrate systems that allow users to edit existing 3DGS or NeRFs or composite/generate new ones in a scene. However, these systems don't provide real-time feedback and offer a limited toolset for scene editing.

Creation Systems in VR

There is a long history of creation systems in VR. 3DM [6] laid the groundwork by presenting a 3D modeling system operated via a 6-DoF mouse, offering a novel way to interact with digital objects in three-dimensional space. Building on this, ISAAC [31] introduced scene editing within Virtual Environments, allowing for a more intuitive and immersive design process. Coninx et al. investigated hybrid 2D and 3D editing [7]. CaveCAD [34], a system for freeform virtual sculpting of organic shapes, enables artists and designers to conceptualize and iterate on their creations in an intuitive manner that closely mirrors the physical sculpting process. Furthermore, Google's TiltBrush [5] allows creators to paint with virtual light and textures, extending the canvas beyond the limits of traditional media. Similarly, VR games like Dreams [10], Figmin XR [51], and Horizon Worlds [49] have provided valuable insights into user interaction models, offering a glimpse into how VR can facilitate complex design tasks while maintaining user-friendly interfaces. Han et al. demonstrate the next steps in HCI design and interaction with virtual environments by increasing accuracy and range of physical gesture recognition, an approach that lends itself to more natural and user-friendly interaction with the surrounding virtual environment [16].

More recently, researchers have begun to explore the incorporation of generative AI in virtual environments. For example, the Large Language Model for Mixed Reality (LLMR) framework [8] leverages Large Language Models (LLM) and the Unity game engine for real-time creation and modification of interactive Mixed Reality experiences, showcasing the potential of LLMs to facilitate intuitive and iterative design in mixed reality applications. Style2Fab[13] also demonstrates the ability of generative models in personalized 3D model generation. The Dynamics-Aware Interactive Gaussian Splatting System [22] also enables the creation of animated and interactive experiences within virtual reality settings.

Our system leverages generative AI and natural language to assist in 3D scene editing in virtual environments, but prior and concurrent works don't aim to create creativity tools leveraging radiance fields.

2.5 Design Goals

Based on our review of related work, we identified a lack of research into interaction techniques for working with emerging radiance field techniques and generative AI in VR. Therefore, we formulated the following design goals:

- **Focus on creating and editing radiance field objects in VR.** We want to support users in populating 3D scenes with radiance field objects. This may involve updating objects already in the scene or creating completely new objects.
- **Enable both direct manipulation and instruction-based editing.** Users may prefer different levels of control for various scene editing tasks. For example, users may want to directly manipulate objects for detailed edits while preferring natural language instructions for larger scale edits. Users should have access to both.
- **Offer modular architecture to allow integration of future generative AI advances.** An important aim of Dreamcrafter is to provide users with state-of-the-art 3D object editing and generation technologies for environment design in VR, so a modular framework is necessary.
- **Preserve real-time interaction regardless of the latency of editing operations.** For real-time scene editing, users should not be hindered by the system's latency. In the event that a process cannot be performed online, users should have access to previews of the edits they have made to the VR environment.

2.6 System Design and Implementation

Dreamcrafter provides an interface to edit and generate radiance field objects using generative AI-enabled tools. Dreamcrafter supports different levels of user control and gives real-time proxy representations to preview time-consuming edits and introduces new workflows leveraging image diffusion models (i.e., Stable Diffusion).

System Details and Interface

Users can select fixed regions in space or existing objects in the scene to apply spatial annotations. Existing or pre-captured radiance field objects can be added to the scene via an object menu. Generations and edits can be re-done or deleted. Each type of edit and module is designed in the framework to be interchangeable and modular allowing new types of interactions to be added in the future, or replace existing ones. Spatial annotations are added to objects or spaces that are assigned edits with corresponding proxy representations based on edit instructions. Figure 2.1 shows spatial annotations applied in a scene.

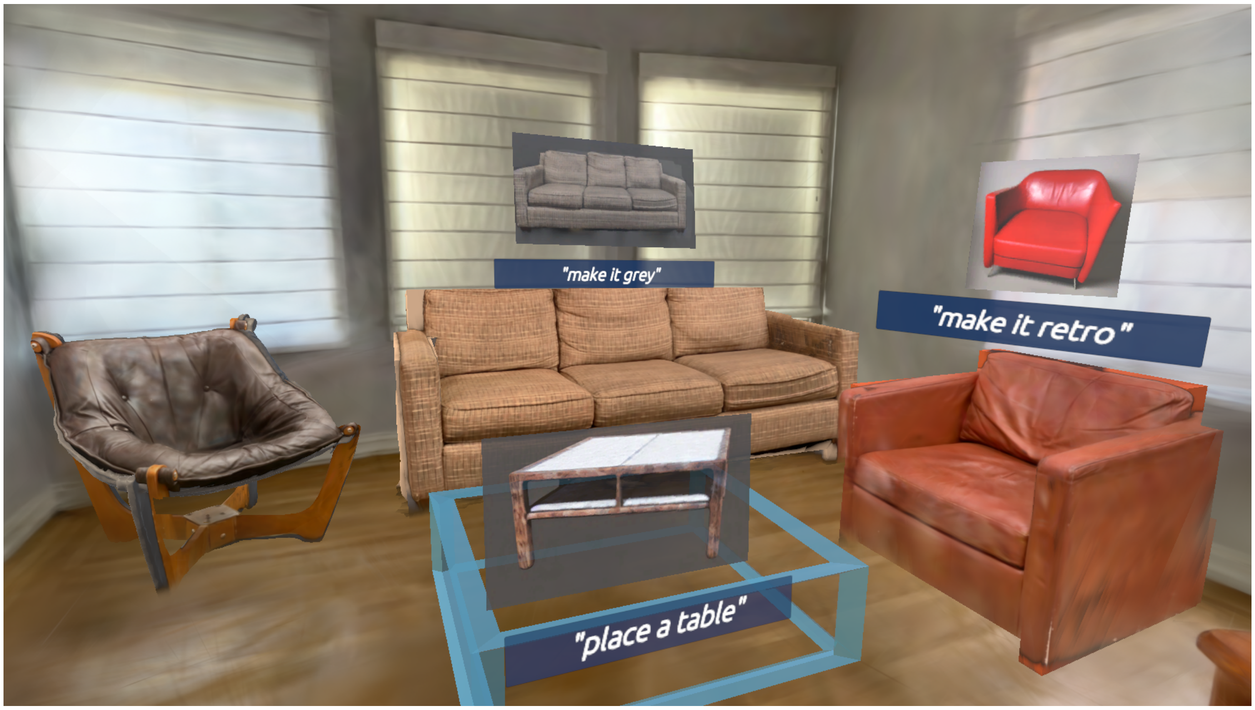


Figure 2.1: **Dreamcrafter edit interface with spatial annotations.** View of edits and spatial annotations made in scene. Spatial annotations are placed over objects or spaces that are assigned edits. Based on the type of edit a 2D or 3D proxy edit preview representation is also shown.

Key interactions

Dreamcrafter supports four main interactions for moving, editing, and generating new radiance field objects.

Move objects

Users can move objects (generated or radiance field based) with spatial manipulations with hand movements and VR controls. Objects can be positioned, rotated, or scaled within the scene. Physics can be applied to help align the objects or stack generated objects. Figure 2.2 illustrates this interaction.

Edit radiance field objects via prompting

Radiance field objects can be given stylistic or basic structural edits by pointing at an object and speaking an instruction, e.g. "Make this chair chrome and futuristic." See Figure 2.3. A render of the object is given to the Instruct-Pix2Pix module, which applies the instruction to

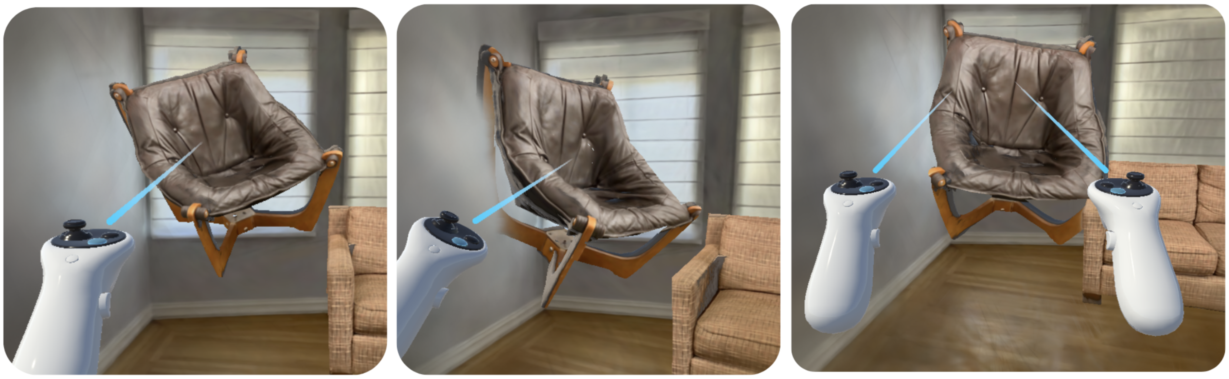


Figure 2.2: **Object transformations and direct manipulations** (Left) Positioning object in the scene (Center) Rotating object. (Right) Scaling object

show as a 2D preview of the edit. We chose to use Instruct-Pix2Pix to preview this edit since it is a 2D equivalent of the 3D edit modules we use. Users can select from three edit variants, which will be applied for the final 3D object edit. Users can re-prompt edit instructions to quickly iterate and preview before running a time consuming full 3D edit.

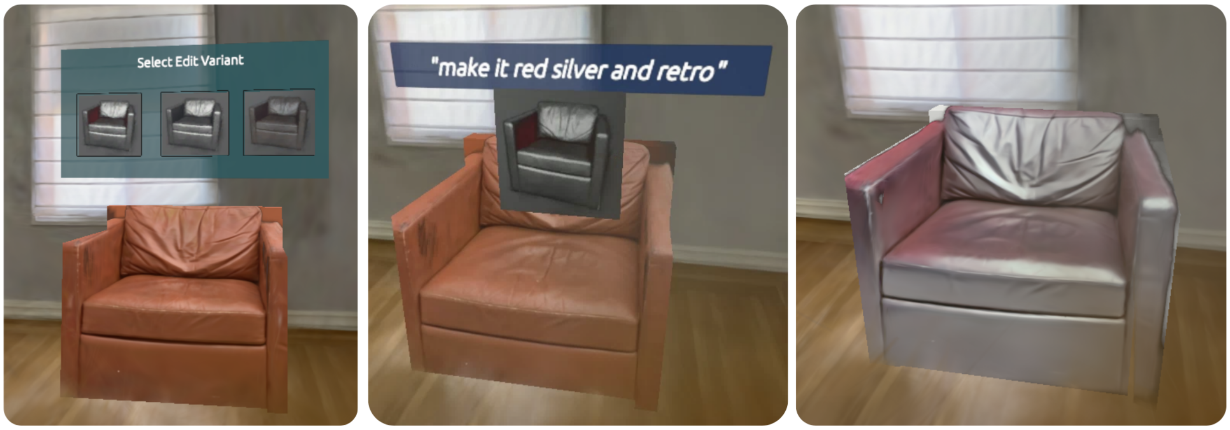


Figure 2.3: **Radiance Field Object Editing with preview** (Left) Edit variants are presented to a user. (Center) Displaying selected edit preview as a spatial annotation. (Right) Fully processed 3D edit replaces the original

Generate objects via prompting

Users can generate objects by pointing at the ground and speaking a prompt of the object they want to create (Figure 2.4). This sends an API call to the 3D generative module that includes Shap-E [23], which generates a low fidelity mesh and render, and the render is stylized using depth conditioned ControlNet [54] with the initial prompt. Optionally, the object generation and image stylization module can be themed to the scene through in-painting and masking methods. The object stylization can help to generate objects that thematically and stylistically fit in the environment which could match closer user's expectations of the generation results rather than a generic or random generation. The user can select from three stylized 2D image variants of the object. This allows the system to reduce ambiguity during 3D object generation and give the user further control over the offline time consuming generation. During an offline process, the full fidelity 3D objects are generated and placed in the scene.

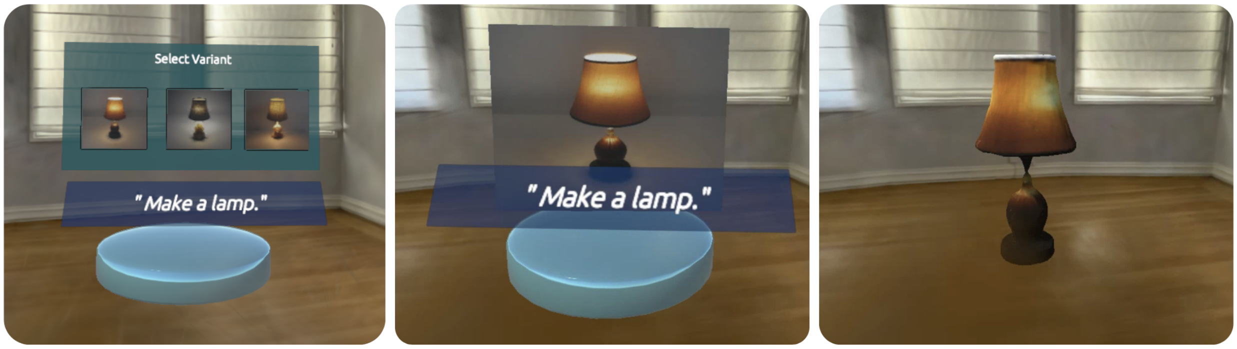


Figure 2.4: **Object Generation via Prompting** (Left) Object generation variations from speech input. (Center) Displaying selected generation preview as a spatial annotation. (Right) Fully processed 3D generation in the scene.

Sculpt then stylize objects

Alternatively, users can generate objects by creating an arrangement of basic 3D primitives (i.e., spheres, cubes, and cylinders) (Figure 2.5). The system takes a snapshot of a render of this arrangement, and then stylizes it with ControlNet based on a user-given prompt. Once the user confirms the stylized and sculpted generation, the object can be placed in the scene. The proxy arrangement of the object is added to the scene with a render of the stylized version. For the offline process, the image preview is converted into a 3D mesh or gaussian splat and placed in the scene.

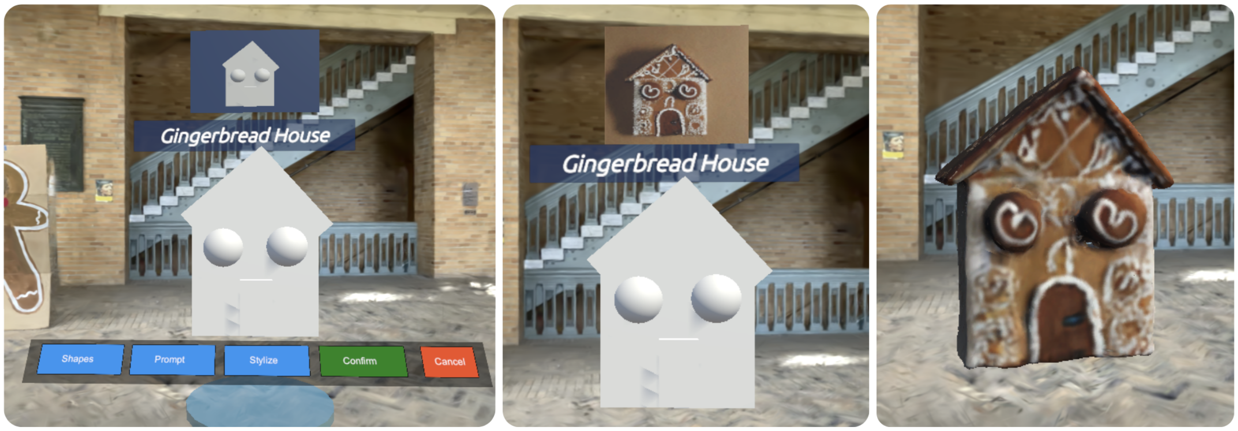


Figure 2.5: **Object Generation via Sculpting** (Left) Sculpting toolkit to create primitive shape arrangement (Center) Displaying stylized sculpted object preview as a spatial annotation. (Right) Fully processed 3D generation in the scene.

Proxy representations: Labels and Previews

Proxy representations are intended to help users see the impact of their editing operations in real time. There are two types of proxy representations: labels and image previews. Figure 2.3 (center) and Figure 2.4 (center) show the labels and image previews. The labels show the prompts users have spoken aloud as commands to the generative AI modules (e.g., "make the sofa blue"). The image previews show 2D versions of the anticipated generation. These image previews are generated using Instruct-Pix2Pix which is the underlying 2D image editing system used for the 3D radiance field editing system, Instruct-NeRF2NeRF or Instruct-GS2GS.

Both the labels and image previews are associated with radiance field objects in the scene. This is done through a spatial annotation framework we developed. The framework logs each object's positions, object type, generative AI prompt, and image preview to a JSON file used for 3D generation and replacement, which we discuss next. Optionally, instead of an image preview, the system can run the text-to-image module to show the generated 3D object in the scene online in close to real-time. This 3D generation may be of a lower fidelity than the final offline generated 3D model, but can help the iterative process of arranging and re-editing objects.

Stable Diffusion-Assisted Creation: Magic Camera

Users can position a virtual camera to visualize a stylized 2D render of the scene. This feature, called the magic camera, stylizes a snapshot of the view of the scene through the ControlNet module given a prompt. The resulting stylization gives a generally coherent and realistic composition of the scene based on the arrangement of objects. Using ControlNet conditioned on canny edges allows the 2D place holders to appear as integrated objects in the scene. This feature can be analogous to traditional 3D editing systems for rendering a final 2D image render output (generally a higher fidelity output given a proxy/textureless 3D representation in the editor). When composition radiance field objects into a scene, the lighting is baked per scene, and therefore for each object and room, which may be from separate scenes, causing the lighting to differ. If ControlNet is run over a snapshot of the composite of these objects in the scene, the output render appears to re-stylize the objects and environment which matches the lighting, shadows, and slight positioning of the objects to appear more realistic or stylized into a different scenario based on a text prompt. A benefit of using stable diffusion and ControlNet is that it adds additional realistic detail to the objects. If an object is generated from text-to-3D or the NeRF/3DGS capture is not very clear, a stable diffusion render generally fixes these issues and makes it appear more realistic. The magic camera can be used as another way to control 2D image generation. Additionally, it could be possible to potentially re-stylize the 3D scene with a 3D output based on the stylized scene preview from the magic camera, through converting the 2D image of the scene into a 3D scene through depth in-painting methods, multi-view diffusion like CAT3D [14], or other systems such as RealmDreamer [42]. This could create an iterative design process where a user could create a general layout of the objects and positions in the scene, and can use the magic camera to stylize it, and then iteratively edit the 3D scene.

Generative AI Modules and Offline Processing

Using the JSON log output from the spatial annotation system, Dreamcrafter makes instruction and tool specific API calls for each generative AI module as shown in Figure 2.6. A Python broker server receives a server message from the Unity project and forwards instruction parameters (e.g., instruction type, text prompt, image input) to the specified module. Figure 2.7 shows an overview of the system architecture.

Object generation uses a 3D generative module Shap-E, and a 2D image stylization module ControlNet and Stable Diffusion. The full object 3D generations use 2D-image to 3D-model models such as GRM [52] or text-to-3D based system. The final 3D object edits are done using Instruct-NeRF2NeRF for NeRFs, or Instruct-GS2GS [46] for Gaussian Splatting objects. The modules are exchangeable and can be implemented to use updated AI models. After the edited objects are added to the scene, users can repeat the process and edit the scene again, creating an iterative design process.

Additional Implementation Details

The system was implemented in Unity and the XR Toolkit for VR integration and uses a gaussian splatting visualizer plugin [36]. A python flask server is used to interface with the generative modules which are run using the HuggingFace python inference API. When a user initiates a preview edit, a C# script sends a message to the flask server to request the

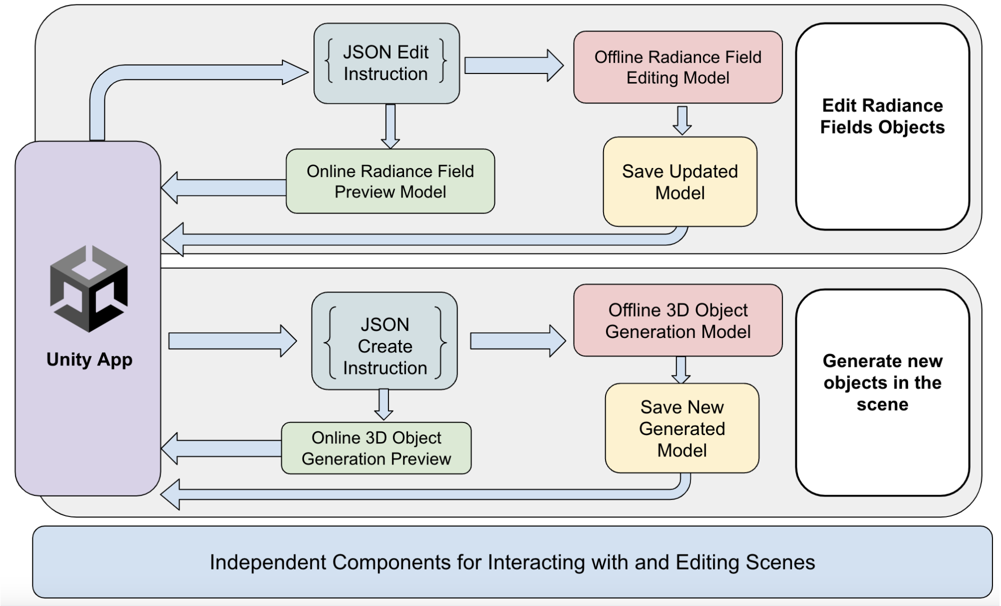

JSON Edit Instruction
Offline Radiance Field Editing Model
Edit Radiance Fields Objects
Online Radiance Field Preview Model
Save Updated Model
Unity App
JSON Create Instruction
Offline 3D Object Generation Model
Generate new objects in the scene
Save New Generated Model
Online 3D Object Generation Preview
Independent Components for Interacting with and Editing Scenes

Figure 2.6: **Dreamcrafter system architecture.** Pipeline overview of system including three distinct modules to edit scenes: edit radiance field objects, create new radiance field or mesh objects, and create 2D stable diffusion renders of scenes. Input files generated from Unity are in green, generative model components are in yellow, and saved results are in red.

specified module to get called, which performs the generation given the input parameters (images and or text prompt) and generates an output. Some of these modules are optimized for speed and to return additional outputs.

2.7 Stable Diffusion Scene Editor

Another system exploration during the project's development was using Stable Diffusion to pre-visualize stylized scenes based on the construction of a scene of primitive objects, created and arranged within the VR interface, as shown in Figure 2.8. This system uses the magic camera as described earlier, a virtual camera that renders the scene and produces a stylized 2D render from ControlNet, but instead of applying over a radiance field scene, this system applies it to a scene of primitive objects. Users could specify a prompt and apply ControlNet, conditioned on depth, to view the stylized scene in a floating window which updates in close to real-time.

We observed that ControlNet was able to understand what the arrangements of primitive objects were given the prompt. For instance, if we had a few cubes roughly creating the outline of a couch and two cylinders creating the shape of a lamp, without additional contextual information about what the objects were, beyond a global text instruction ”realistic living

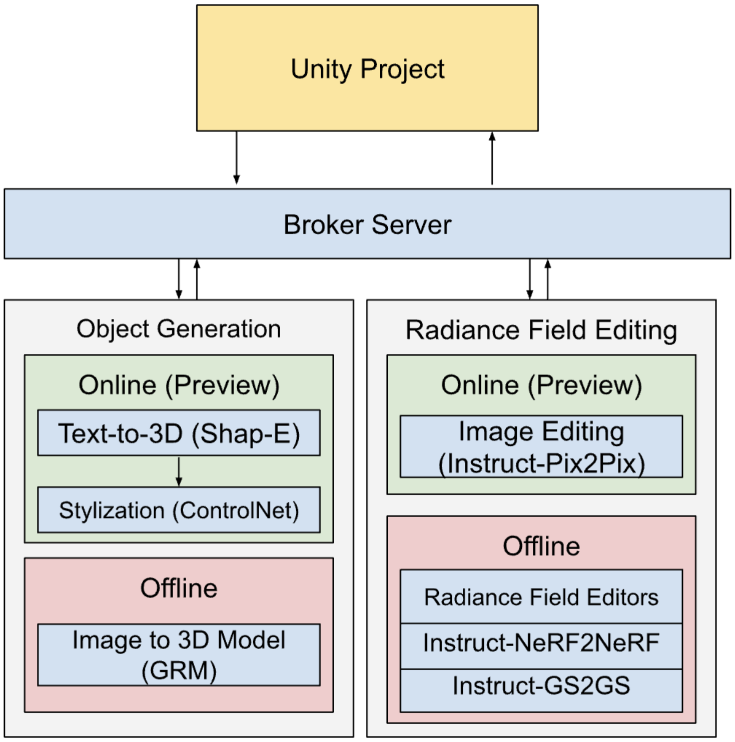


Figure 2.7: **Dreamcrafter modules system overview.** Modules processing pipeline: The Unity project sends API calls to the broker server to run instructions from specific generation modules and their outputs get sent back to the unity project. Online modules are run for previewing generations, and offline modules are run after editing is complete.

room", the stylized render showed a high fidelity render of the living room with the couch and lamp stylized. Figure 2.9 shows multiple prompts applied to the same arrangement of primitives.

For a potential further addition, objects could be spatially tagged with a label text prompt and be individually stylized, and then be composited into a stylized render of the scene through masking and inpainting. Each stylized 2D render could be converted to 3D and placed in a stylized 3D scene. Potentially, the final 2D render can be converted to a 3D scene using an image to 3D scene system like RealmDreamer [42]. This 3D scene could be given to the primary Dreamcrafter system for additional editing after pre-processing and segmentation.

2.8 Evaluation

Two research questions motivated the evaluation:

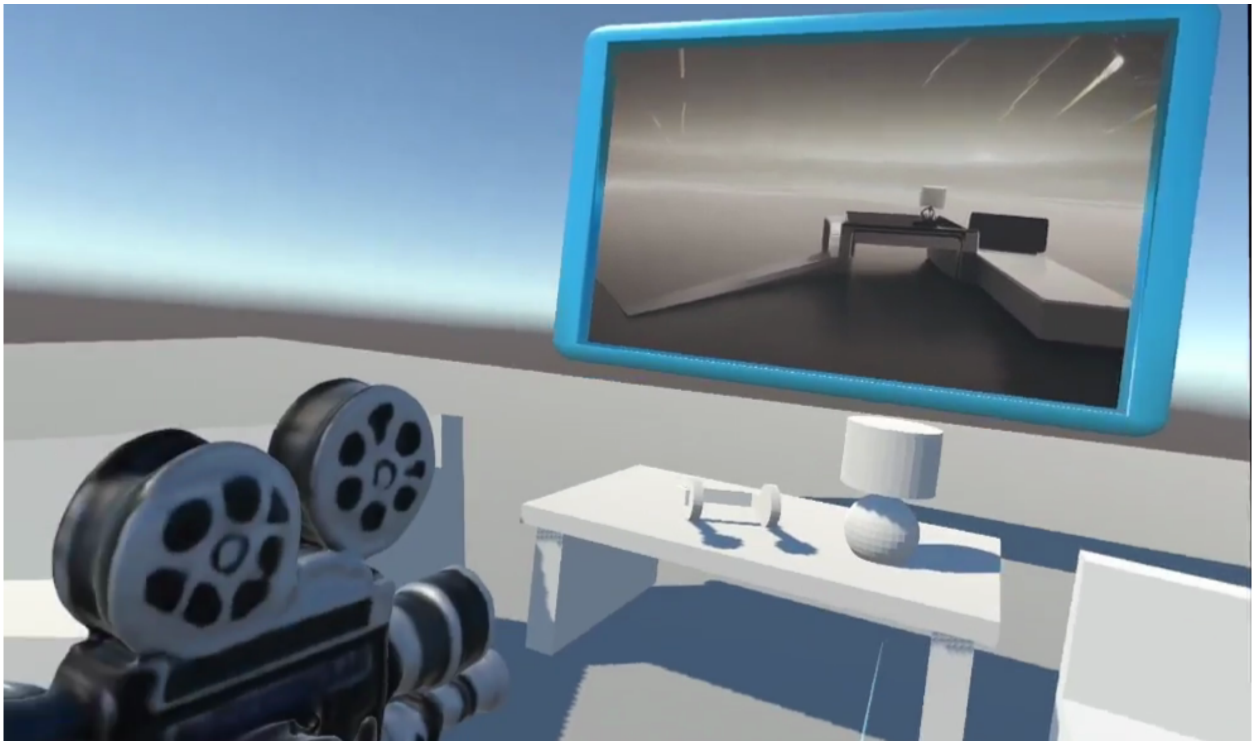


Figure 2.8: **Stable Diffusion Scene Editor** A user can arrange primitive objects in a scene and position a virtual camera to stylize a 2D render of the arrangement as shown from a preview on a floating window.

1. **RQ1 - Levels of control.** How do users want control over scene edits? Specifically, when do they choose to generate objects via prompting or sculpting? Why?
2. **RQ2 - Proxy representations.** What are users' reactions to the proxy representations? Are they sufficient for envisioning final scene edits?

Study Design and Procedure

After participants gave informed consent, the researchers walked participants through a tutorial introducing the interactions for editing and creating objects in Dreamcrafter. The tutorial took approximately 30 minutes. Once participants practiced and expressed feeling comfortable performing the interactions, they were presented with the scenario of designing a 3D environment for a winter holiday party. They were asked to complete the following tasks:

- **Dining area for six.** Participants set up a dining area for six people. The 3D environment was already populated with a couple of tables and a chair that participants

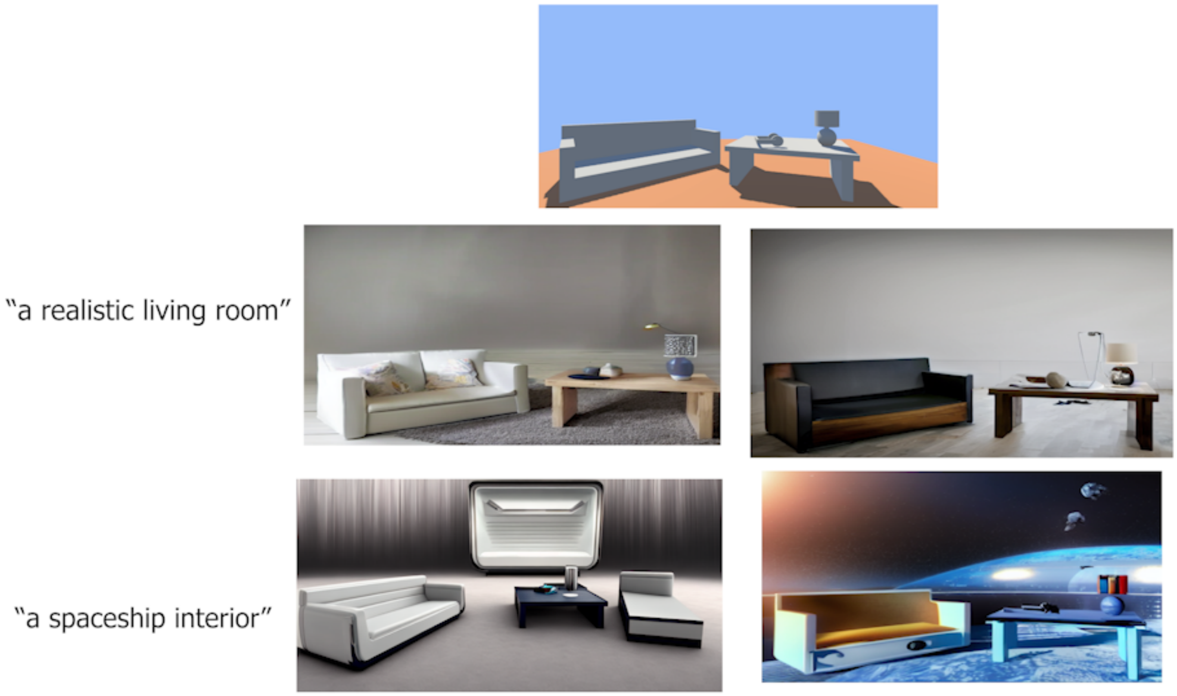


Figure 2.9: **Stable Diffusion Scene Editor** Multiple generations given a scene of primitive objects and text prompts

could duplicate or edit.

- **Photo area for party guests.** Participants decorated an area for taking pictures. The task was to create a North Pole scene by considering snowmen, elves, or trees.
- **Gingerbread house.** Participants created a gingerbread house with two windows and one door.
- **Unstructured editing.** At the very end, participants were given five minutes for free-form editing where they could revisit any of the tasks above as they edited the scene to their liking.

We designed the tasks such that they required a range of editing and creating operations, where different modalities would likely shine. The dining area task was the most scaffolded, with relevant objects populating the scene already. We anticipated that this would encourage participants to edit the existing objects using Stable Diffusion. The photo area was more open-ended with opportunities to edit existing objects and create new ones via prompting or sculpting. The gingerbread house was the most specific task, likely requiring a significant amount of control. For all the tasks, participants were encouraged to use any interactions as they saw fit.

Upon completing the tasks, participants completed an exit survey and interview. In total, the study lasted approximately 90 minutes.

Participants

Participants were recruited via word-of-mouth through VR-related Slack channels, newsletters, and mailing lists. Participants self-reported having relatively little experience in VR (median=2/5). Four of the seven participants had prior experience with 3D tools (Unity or Blender), and two participants had prior experience with creative generative AI tools. Participants were compensated $35 for their time.

Measures and Analysis

For each task, we recorded and analyzed videos for how participants manipulated objects (i.e., editing vs. creating; prompting vs. sculpting) and why. We also thematically analyzed their open-ended survey questions and interview responses.

Results

Overall, participants reported that Dreamcrafter helped them edit the scene as they wished [P2, P4, P6, P7]. P5 expressed how the scene they created using Dreamcrafter was "not what [they] thought but more interesting."

RQ1: Levels of control

Overall, participants rated their success in achieving their desired edits highly (Dining area: median=5/7, Picture area: median=5/7, Gingerbread house: median=4/7). For all tasks, participants more frequently generated objects using prompting instead of sculpting. Four out of seven participants used a mixture of prompting and sculpting across the study tasks (Table 2.1). Three even used both prompting and sculpting within the same task. For example, P1 created most of the gingerbread house via sculpting but then wanted to augment it with prompt-generated windows.

When asked why they chose to create objects via prompting, participants explained that prompting was easier to use [P2, P3, P4, P5, P7]. Prompting helped them "save time" [P1], required less active user involvement [P2], and resulted in "more polished" results [P3]. P4, explained, "*The prompting tool did make it extremely easy to take what I am thinking and make a relatively accurate depiction.*"

Participants had mixed opinions on how well prompting served their goals when they had specific details in mind. P1 and P6 explained that they preferred prompting over sculpting depending on "*typically how complicated I expected the object to be*" [P6]. At the same time,

Table 2.1: **Evaluation: Different levels of control used.**
The number of objects created using each approach are in parentheses. Participants used a combination of editing existing objects, creating objects via prompting, and creating objects through sculpting then stylizing throughout the tasks. Four out of seven participants used a combination of prompting and sculpting throughout the study, including sometimes for the same task. While the majority of participants created the majority of objects via prompting alone, participants reported gravitating towards sculpting to control generation.

ID	Dining	Picture	Gingerbread
P1	Edit (2)	Prompt (1)	Prompt (1), Sculpt (1)
P2	Prompt (2)	Prompt (3)	Prompt (1)
P3	Edit (2)	Prompt (3)	Prompt (3)
P4	Edit (1)	Prompt (3), Sculpt (1)	Sculpt (1)
P5	Prompt (4)	Prompt (6)	Prompt (1), Sculpt (6)
P6	Edit (2)	Prompt (3)	Sculpt (1)
P7	Edit (2), Prompt (1)	Prompt (4)	Prompt (1)

P4 reported "*[the generated 2D proxy representation] sometimes fell short in some minor details of what was described in the prompt.*"

In contrast to prompting, participants reported feeling they had more control when sculpting then stylizing objects [P1, P4, P5]. P4 explained, "*if I had an idea in my head that I know how I wanted it to look like...it kind of had a little more restriction what the AI used to create versus the prompting*". When asked when they chose to sculpt, P1 and P5 explained that they preferred sculpting large-scale objects, such as the gingerbread house. At the same time, most participants, including P7 who did not use sculpting, wanted to have access to more shapes [P4, P5, P6, P7] and finer grained object manipulation [P2, P4, P6, P7], suggesting that sculpting may ultimately be more desirable than we saw in our study.

RQ2: Proxy representations

Six out of seven participants primarily relied on the image previews to get a sense of the scene's overall composition [P1, P2, P3, P4, P6, P7]. Participants also reported that the image previews helped them visualize individual objects [P4, P5, P6].

System Limitations and Strengths

A primary limitation was the scene's physics. For six of the seven participants, rotating and arranging objects in the scene were difficult [P2, P3, P4, P5, P6, P7]. For example, when editing the dining area, P2 expressed "*When chairs would fall over, it was very hard to put them back up. Also, if I wanted to rotate or move the chairs they would tend to change size, so by the end most of the chairs were all different sizes.*"

Another important limitation was inaccurate speech recognition, which became a major burden for users relying on prompting [P1, P2, P3]. Despite this, most participants relied on prompting for setting up the picture area and gingerbread house, so we would expect that improved speech recognition would lead to more reliance on prompting. Related, because the system had a five second speech detection window, P5 expressed wanting the system to give them more time to express all the details they had in mind.

Other technical challenges that participants reported were feedback time while waiting for Stable Diffusion results [P1, P5], awkward VR controller mappings [P6, P7], discomfort in VR [P2, P6].

Despite challenges with object manipulation and speech recognition, all participants expressed wanting to use Dreamcrafter in the future for a myriad of reasons: interior design [P1, P3, P6, P7], "my creative side" [P1], CAD in engineering [P4], and video game design [P5]. P2 preferred to use a non-VR version. For P5, P6, and P7, generating objects via prompting was the best part of the system. This suggests that even with user experience issues, providing multiple forms of user control, proxy representations, and access to generative AI modules were desirable for diverse spatial computing applications and users.

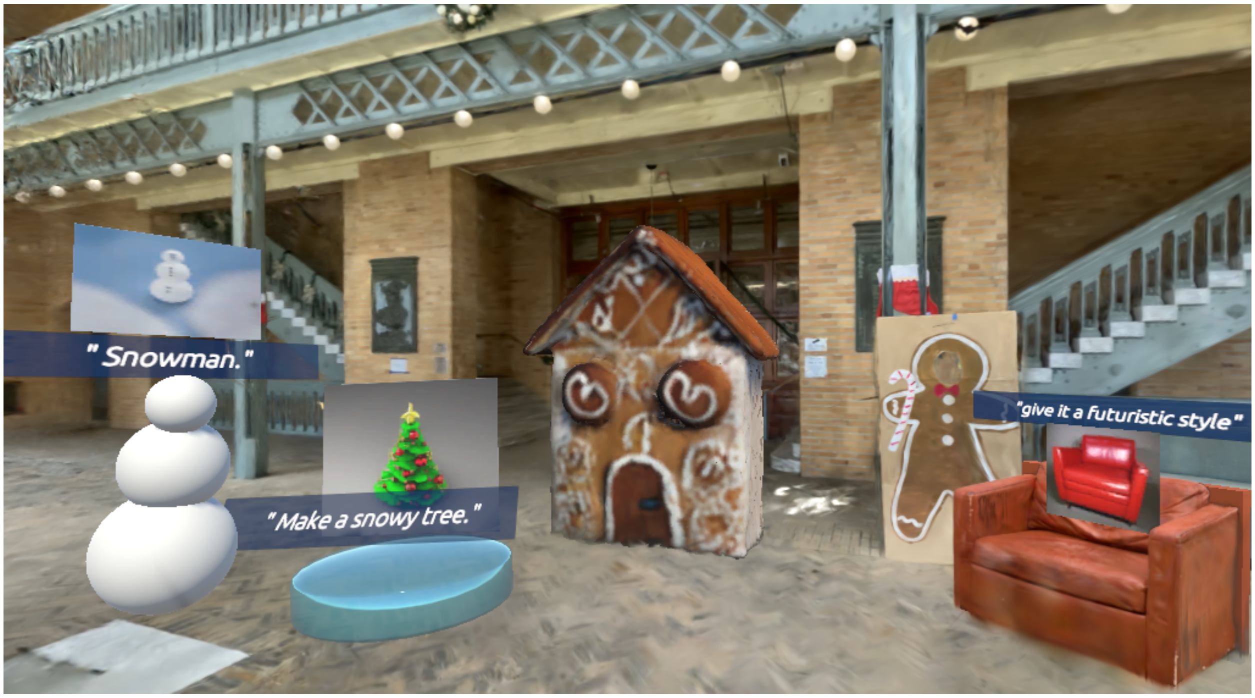


Figure 2.10: Spatial annotation tags are placed over the radiance field objects and generated objects with given instructions and preview generations.

2.9 Discussion

We investigate how to incorporate the benefits of real-time, immersive editing and the advantages of high-level scene editing using generative AI. We develop and evaluate the Dreamcrafter system, which provides a modular architecture for generative AI algorithms, offers different levels of interactive control, and leverages proxy representations to show previews of high-latency edits to radiance field objects.

Through a first-use study, we find that users, including those without VR or scene editing experience, find the direct manipulation (sculpting) and natural-language based (prompting) interactions useful for editing and creating objects. Most use a mixture of both interactions. Sculpting objects and then stylizing them with generative AI helps participants feel they have more control over the generation process. Yet, participants create more objects using only natural language prompts. This is not surprising given the relative speed with which generative AI models can create object proxies (previews). Interestingly, despite the control direct manipulation affords them, participants preferred generative AI-based object creation over sculpting when they had very specific details for what they wanted objects to look like. These findings suggest that sculpting may be useful for giving the general shape of an object while prompting is useful for its specifics. Both sculpting and prompting appear to serve different purposes in users' design processes, so supporting both forms of control is necessary for scene editing tools to support a diversity of creative paths and styles [39].

Furthermore, participants found Dreamcrafter's 2D proxy representations of high-latency 3D object editing and creating operations useful for editing 3D scenes. This suggests the importance of realtime feedback for spatial computing tasks. This also suggests that leveraging 2D generation for 3D scenes may be a promising path forward for providing realtime feedback. Additionally, providing both text and image proxy representations may be especially important for future semantic, generative AI-based scene editing systems.

Overall, in Dreamcrafter, we explore not only the feasibility but also the benefits of providing both rapid direct manipulation and high-level instruction-based editing support in 3D scene editing. Through varying levels of control and proxy representations, Dreamcrafter is a step towards continuing to lower the barriers to 3D scene editing, especially for emerging graphical representations such as NeRFs and Gaussian Splats.

2.10 Limitations and Future Work

There are three limitations to this work that offer opportunities for future work.

Global scene editing. Dreamcrafter supports editing and creating radiance field objects within an environment. However, users may want to edit aspects of the underlying environment as they design their scenes. One way we have begun to explore this possibility is through developing functionality that allows users to take a snapshot of an environment from a fixed perspective and then stylize that snapshot, in a manner similar to how sculpted objects are stylized in Dreamcrafter currently. The resulting generation suggests a possible way to stylize the scene and all objects contained within it together. Ideally, users should be able to define the perspectives they take snapshots from and how they stylize the scene, perhaps even controlling which objects receive the global style treatment.

Additional levels of control. To further enhance user interaction, a key area of focus should be the development of more intuitive ways to place and interact with radiance field objects. For instance, rather than rely solely on voice commands, what if Dreamcrafter could provide text or gesture input for expressing generative AI prompts? Yet another alternative could be generating sculpted objects in addition to styles so that users could rapidly iterate on the form in addition to texture of generated objects.

It's also worth noting that the modularity of our content generation and spatial annotation system is a cornerstone of Dreamcrafter's design. This approach not only facilitates future enhancements and integrations of different AI technologies but also serves as a robust framework for the development of AI-assisted interfaces in various applications.

Even more rapid proxies. While Dreamcrafter currently supports prompt labels and image previews, what might alternative proxies or intermediate proxies between 2D and 3D objects look like? For example, would users find 3D wireframes just as useful as the 2D image previews? Additionally, if users were to make global scene changes to all the objects, what would an appropriate proxy representation for those changes look like? Additionally, recent work in image to 3D systems run in close to realtime and may be able to show 3D proxy previews at a reasonable fidelity.

Automatic Segmentation. Dreamcrafter currently takes in input of full 3DGS and objects, however it currently is unable to edit objects that are fixed in the scene. To enable editing and placement of objects baked in existing scenes, having automatic semantic segmentation could be used to streamline the editing workflow, making it more efficient for users, without requiring manual segmentation.

2.11 Conclusion

The core idea behind Dreamcrafter is to use direct manipulation for spatial positioning and layout; and leverage generative AI for editing style and appearance of objects. Because generative AI edits are unlikely to run in real-time, Dreamcrafter introduces rapid proxy representations, e.g. using a 2D diffusion model to create a stand-in image for a longer-running 3D generative task. Dreamcrafter enables both 2D (image) and 3D output. In a first-use study, participants report feeling more in control of AI generation when they first sculpt objects before stylizing them with generative AI. Participants also report finding proxy representations useful for scene editing.

Chapter 3

Radiance Field Editing

3.1 Introduction

Radiance fields like NeRFs, 3DGS require new interactions and methods not found in traditional 3D tools due to their different representations from meshes or surfaces. Current 3D modeling toolkits like Blender and Autodesk Maya don't support radiance fields yet. With advances in diffusion models and generative 2D and 3D models, recent work has explored using natural language and other methods to edit the implicit representation of NeRFs, and have applied some of these methods to other radiance field representations like 3DGS. These kinds of interactions with those 3D representations allow more natural interactions and encourage further exploration in interfaces for 3D editing.

In this chapter, I discuss methods and an interface for compositing with NeRFs, semantically editing radiance fields with natural language, and generating stylized 3D objects.

3.2 Compositing Radiance Fields with Visual Effects

Integration of radiance fields like NeRFs in 3D editors for compositing and creating scenes composed of NeRFs and meshes was very limited until recent developments. I created a pipeline for integrating NeRFs into traditional compositing VFX pipelines using Nerfstudio [44]. Since NeRFs are rendered with alpha compositing and volume rendering, to composite NeRFs, they can be layered over each other, and traditional techniques of motion control for camera tracking from filmmaking can be applied to enable NeRFs or 3DGS to be composited and to assemble scenes.

This approach involves using Blender, a widely used open-source 3D creation software, to align camera paths and composite NeRF or 3DGS renders with meshes and other NeRFs, allowing for seamless integration of NeRFs into traditional VFX pipelines as shown in Figure 3.1. This allows for more controlled camera trajectories of photorealistic scenes, compositing meshes and other environmental effects with NeRFs, and compositing multiple NeRFs in a single scene. This approach is also generalizable and can be adapted to other

3D tool sets and workflows. One of the initial goals was to enable a more seamless integration of NeRFs into visual effects and film production. Documentation can be found here: `https://docs.nerf.studio/extensions/blender_addon.html`

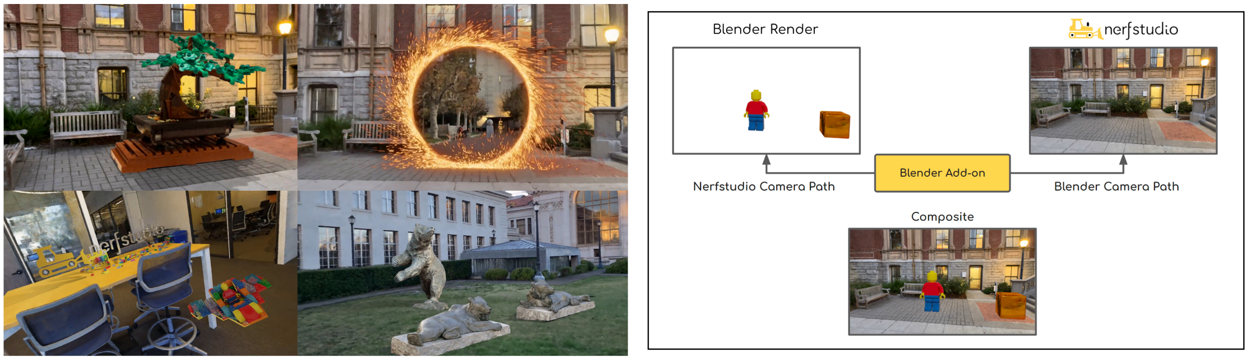


Figure 3.1: **Nerfstudio VFX Add-on Renders and Pipeline** (Left) Results from NeRF composite renders using the Blender VFX Add-on. (Right) Pipeline for compositing renders from Blender and Nerfstudio.

Method

A proxy NeRF (point cloud or mesh) representation is placed in the scene. Camera aligned renders are created from transforming the Blender virtual camera path coordinate system to be relative to the origin of the NeRF representation in the Blender scene for each frame in the render. This technique also allows for compositing multiple NeRF objects and environments in a single scene by rendering an accumulation render for each of the cropped NeRF objects individually to act as an alpha mask. Additional lighting effects like shadows and reflections can be rendered from the NeRF representation and composited. This approach can be used to composite NeRFs with real camera footage, meshes, or other radiance fields. Composited objects like meshes can be relighted by a NeRF scene by rendering 360 HDR environment maps, and since the scene is volumetric, these maps can be rendered along a path. This approach is also useful for assigning a scale to a NeRF scene which helps for rendering VR180 and omnidirectional-stereo (VR360) renders. This method is generalizable to additional toolkits beyond Blender since it is primarily transforming camera transform coordinates from the 3D editor camera to the nerfstudio camera space. The main limitation of this approach is that it doesn't run in real-time, and therefore is not interactable. More recently Unreal Engine integrations for NeRFs and 3DGS have been created which allow for real-time interactions [28, 48].

3.3 Edits with Natural Language

Instruct-GS2GS

Editing radiance fields based on semantic natural language instructions was proposed by Instruct-NeRF2NeRF [17] which applies global scene edits by iteratively updating the dataset with Instruct-Pix2Pix applied over training images and the text prompt. My co-author, Ayaan Haque, and I extended this method to Gaussian Splatting in Instruct-GS2GS [46] to enable these natural language edit capabilities with 3DGS scenes as shown in Figure 3.2. This approach adopts a similar pipeline for iterative dataset update, but instead of updating an individual training image per iteration, we perform a full dataset update and apply Instruct-Pix2Pix to all images and then train for 2.5k iterations. This process is repeated for 2-3 full dataset updates (training for 5k-7.5k iterations) which compounds the edits performed by the previous dataset update.

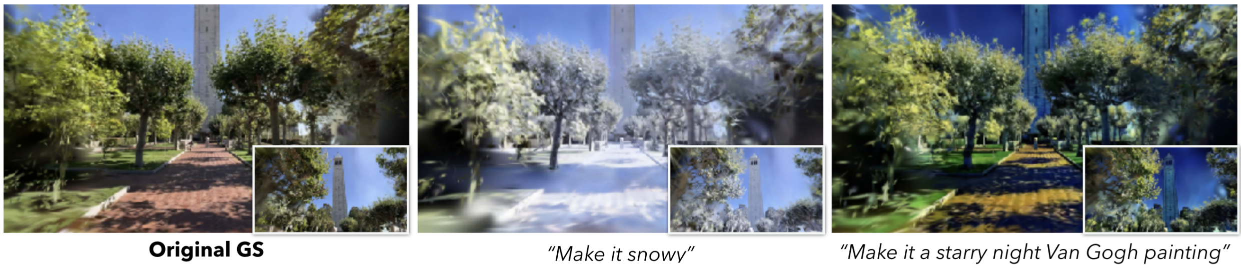


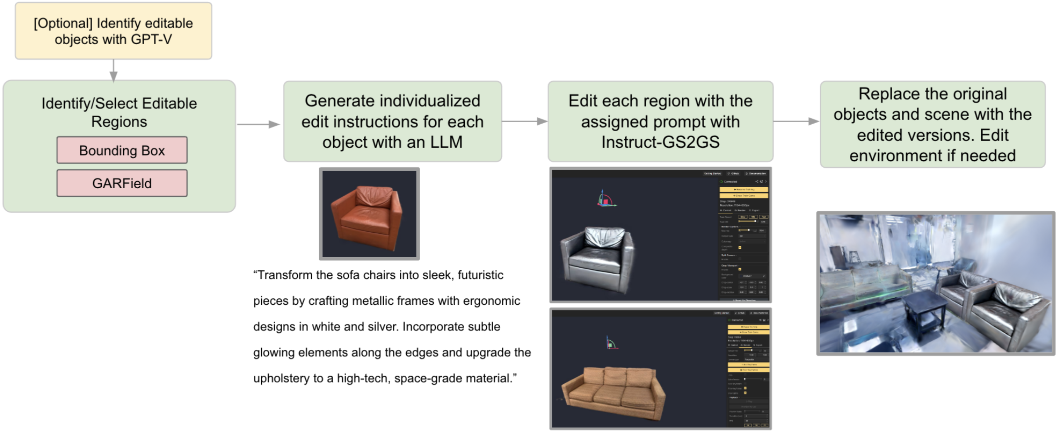
Figure 3.2: **Instruct-GS2GS Semantic Edits** Semantic edits from text instructions over 3DGS scenes for global stylistic changes. (Figure from Instruct-GS2GS[46])

The results show similar edit capabilities as Instruct-NeRF2NeRF which enable some targeted edits to certain regions, but occasionally has limitations by performing unwanted edits and blurry results due to inconsistent views especially when editing over large scenes of multiple objects.

Language Assisted Environment Editing

To improve controllability and constraints over the individual regions in the scene and prevent unwanted edits, editing regions could be specified through semantic masking or user bounding box selections. Since Instruct-GS2GS applies global text instructions, there is ambiguity in which object the edit could be referring to, such as a prompt with "make this chair red". I propose assigning specified edit regions each with an individual text prompt, which could be further customized to be object specific using an LLM. If a general instruction is given for a scene, "make the room look like a futuristic spaceship", Instruct-GS2GS is unable to perform significant view-consistent edits to all objects within the scene since Instruct-

Pix2Pix is applied on every image giving different results due to a broad prompt, and more detailed prompts helps stronger generations. Figure 3.3 shows this pipeline of assigning and editing regions of the 3D scene from an input scene and instruction.


[Optional] Identify editable objects with GPT-V
Identify/Select Editable Regions
Bounding Box
GARField
Generate individualized edit instructions for each object with an LLM
Edit each region with the assigned prompt with Instruct-GS2GS
Replace the original objects and scene with the edited versions. Edit environment if needed

"Transform the sofa chairs into sleek, futuristic pieces by crafting metallic frames with ergonomic designs in white and silver. Incorporate subtle glowing elements along the edges and upgrade the upholstery to a high-tech, space-grade material."

Figure 3.3: **LLM Assisted Environment Editing Pipeline** Pipeline to edit scenes using an LLM to specify edit instructions and a VLM to identify editable regions.

This new approach could first identify relevant major objects in a scene, possibly through a VLM, segment those objects, and use an LLM to generate individualized object specific and descriptive edit instructions for each furniture object (for a chair "make it metallic, futuristic, style with metallic flat legs" instead of the global text prompt). Each object could be edited with this instruction and replace the original object in the scene.

Chapter 4

Future Work for 3D Editing Systems

This chapter will discuss potential future work and features in an immersive 3D editor building off of Dreamcrafter. To facilitate intuitive 3D editor features that allow the user to choose their own level of specificity of editing instructions (supporting high to low level editing interactions which may require additional minimal input and additional effort from the user), radiance fields would need to be pre-processed and semantically analyzed. This would enable intuitive speech commands for object placement and generation, but also give the option for users to have more levels of control as well. Users should be able to create or capture the environment, enter into the scene in VR, and fill and edit the reconstruction with additional generations.

4.1 Editor Components

Environment Aware Actions from Language Instructions

Specific instructions from natural language can be used to edit the environment at a higher abstraction level than traditional 3D modeling toolkits. This system could leverage the understanding of the environment to identify the user's intent regarding the object or space the instruction applies to and the desired action, such as placement or generation. For instance, if a user points or looks at a certain region of the room and says "place a door over there", the system should be able to generate a 3D representation of the door in the same stylistic attributes of the environment and in the correct location on the wall, not intersecting with other geometry. The system should support the ability to make implicit references in the scene such as "change the color of *that* wall to green". Recent work, What's the Game then [21], proposes using LLMs with semantic scene graphs to reference and instruct objects in scenes through speech and gestures.

Positioning Generated Objects from Language Instructions

Determining the optimal location of the generation based on a vague text instruction and possibly region of the user's view could be approached in a few ways, including using vision language models. OCTO+ [53] describes a method for placing objects by using a vision model like GPT-4V which is given the users' view and the text prompt and told to find the region where the object should be added by identifying objects in the scene and using an LLM to place it in a logical position relative to relevant objects. Semantic Placement [38] proposes a method to identify an image mask for the relevant regions given an input image and text prompt. This approach also uses an LLM and VLM to analyze the image and generate a heatmap for the semantic placement masks. For instructions that reference objects in the scene such as "place a lamp next to that chair" or "move the laptop to the left side of the desk", similar methods can be applied as well. GALA3D [56] proposes a method to generate compositional scenes of multiple objects based on a text prompt by using LLMs to generate layouts and optimize composited gaussians, as well as to perform natural language instructions for object placement and movement. 3D scene based understanding as shown in 3D-LLM [18] would be useful in scene generation and help facilitate natural language instructions that refer to objects in the scene. While these methods show effective results, further work should explore larger scale scenes and more precise object positioning for 3D data, since most systems mentioned here use 2D vision models.

Generate Coherent Stylized 3D Generations

The generation should have similar materials, color, or artistic style matching the visual style of other furniture rather than a generic or random generation that would result naively calling a text-to-3D system. Having a multi-modal input of text with an image of the environment to inpaint a 2D representation into, or getting keywords of the environment from a vision language model could help guide the generation to be in the correct style. Input modalities beyond text would be required for better generations. A VLM could also be used to assign the appropriate scale of the generated object relative to the other objects in the scene. When objects are stylized and placed in the scene, their lighting may not be accurate and match with the environment, therefore it may be possible to train dataset of layer/relighting images over the object where each image is possibly run through ControlNet, IC-Light [55], or an AI upscaler system which implicitly relights the composited object into the scene by adding shadows and reflections. A similar method could be used to add additional artificial detail in these scenes by using a ControlNet diffusion based upscaler over training images and adding additional images to the dataset from rendered views close to surfaces, running these through the upscalers, and retraining over the existing scene using a method like iterative dataset update.

VR Interactions for Manipulation

Using an immersive editor enables additional interactions for environmental editing compared to desktop editors, such as using hand and eye tracking for object placement and moving. When editing radiance field objects, editing could be achieved with direct spatial manipulation which is more appropriate for a 3D spatial setting. Using a minimal amount of menus and other tools could be optional if the natural language instructions for object placement, editing, and manipulation are intuitive enough. Simple gestures could be used to toggle across varying levels of specificity for edits (scene to individual object). User motion controls or gestures could be used for implicit references to movement and transformation for objects. It may be possible to use an LLM to parse user speech input into a set of instructions that the system can understand for these tasks. Possibly vague or previously unseen or undefined controls could be interpreted by vision and language models as new forms of input. However, certain users with artistic experience and other tasks may require more fine-grain control, such as sculpting, as found in traditional 3D tools, and therefore this system should support those tools as well and adapt to the user's preferences.

4.2 AR/VR HMD Assisted NeRF Capture

Capturing 3D scenes for traditional photogrammetry, NeRFs, and 3D Gaussian Splatting is generally done by taking a series of images of a scene with a camera of known intrinsics and structure from motion algorithms can determine the extrinsics or camera poses of the images in the scene. The camera images and poses (translation, rotation, scale) are used to train the NeRF. Currently, capturing a NeRF requires stabilized slow camera movement/orbiting around an object at a high resolution from multiple viewpoints and angles to capture the lighting all around an object/environment. Capturing radiance field environments can sometimes be challenging if casually captured due to limitations in SfM methods like COLMAP and the necessity to capture surfaces at most angles for a good reconstruction. Generally, these may be improved with a capture strategy like orbiting around an object at different elevations for scanning an object, or circling around the borders of a room to capture a larger scene. Until sparse and casually captured images can provide reasonable reconstruction of a scene, capturing NeRFs and 3DGS are challenging especially to new users.

I propose an alternative method to capturing 3D scenes, using cameras from VR/AR HMDs, which could capture the scene from the user's perspective as they walk through a scene. Recent work using Project Aria glasses demonstrates using a camera embedded in a pair of glasses to capture a scene and generate a NeRF dataset[11]. However, in these approaches, there is a lack of visual feedback component to guide the user where to look at or identify un-captured regions. Smartphone based methods such as LabelAR [27] help capture different regions of an object for computer vision datasets, and Luma AI uses ARKit to guide a user over a specified object region to assist a user to capture an object and different angles.

I propose a concept to use an AR/passthrough VR HMD system which could capture the scene with on-device cameras including RGB, wide angle cameras used for tracking, or optionally other sensors like depth cameras or LiDAR sensors to create a 3D capture. Since the VR headset is continuously running SLAM, those camera poses can be used as the training dataset camera poses to create the final NeRF dataset. Hardware components required are present in state of the art mixed reality headsets such as Apple Vision Pro and Meta Quest 3. Cameras on the headset could be equipped for high frame rate and wide angles which would be optimal for clean dataset collection. Ideally a 360 camera could be mounted to capture more of the scene. If there are regions of the scene too hard to capture (like from the bottom angles), a diffusion model can add unseen detail and views there using an object completion system, such as demonstrated in Reconfusion [50], or a video diffusion model could help complete gaps in unseen regions. In addition to taking in frames from the camera input, pre-processing could be applied to identify relevant images in the video sequence (which could be long) of the scene capture to ensure the dataset isn't unnecessarily long and covers all regions in the room/scene given the depth scan of the scene and prevent repetitive/unusable frames. When capturing a dataset in a large scene it is common for certain objects to be observed multiple times, causing the dataset to contain redundant frames. Possibly, it could render/sample frames after the scene has been trained to find the optimal dataset with the fewest number of frames. Ideally, a dataset assistance/cleanup system like this should be able to allow a user to briefly look around and walk in an environment, and given this limited data, reconstruct the scene properly. If cleaner datasets are collected through these methods, they could be eventually used as training data for a 3D foundation model.

In this system, as depicted in Figure 4.1, a user could enter a space and specify their capture volume/region of interest through an AR visualization, which could be of a room scale environment or around an object. Next, the system could create a trajectory for capture and guide the user where to look and navigate to capture all parts of the environment. To quantify the quality of the scan in progress, the system could keep note of what angles a surface was observed as, or run a LiDAR reconstruction of the scene and identify which parts are more sparse. Recent systems have demonstrated using real-time 3DGS reconstruction [29] by streaming camera poses and images to train in real-time.

4.3 Radiance Field Pre-Processing

Radiance field scenes as input can be pre-processed and have individual interactable/editable objects segmented before being given as input into the system using editing tools, or given semantic prompts such as in LERF. Such a system could use GPT-4V to analyze relevant editable objects in the scene and pre-segment those objects in the scene to make them as editable objects in the system. Generally radiance field scenes are captured with many existing objects in the environment, which may occlude each other (a capture of a room may have furniture items against the wall or have many objects on a table). A pre-process

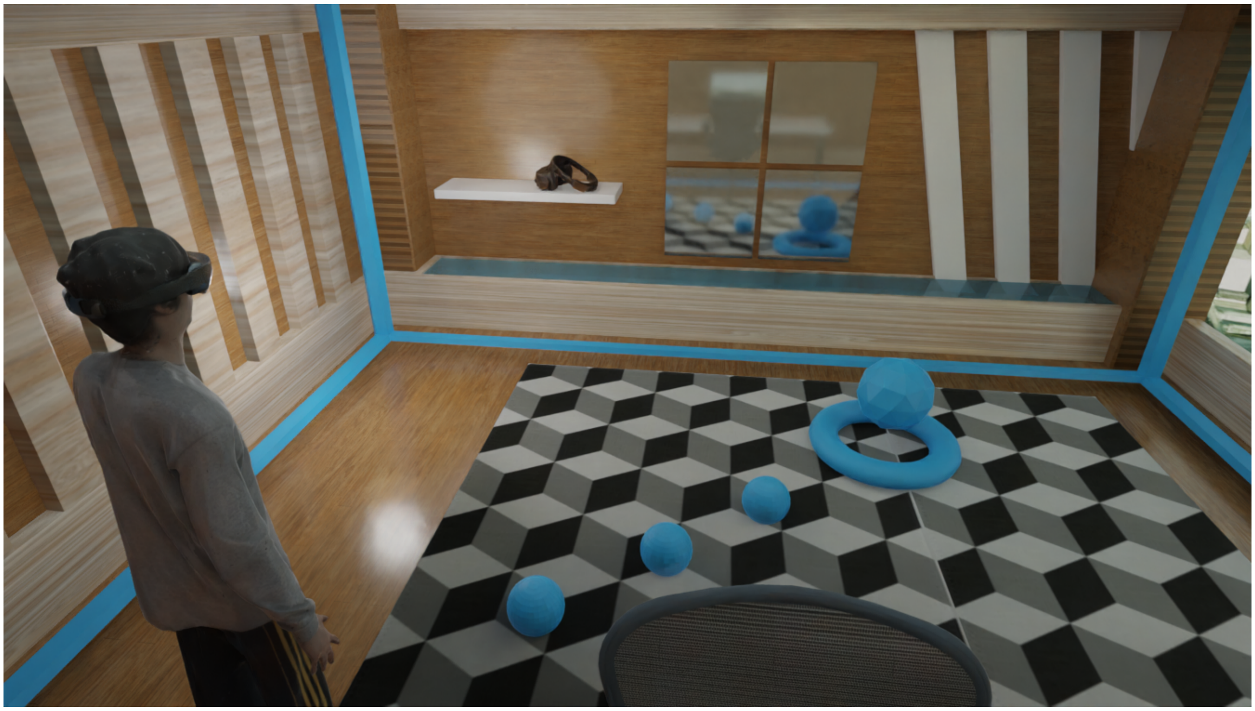


Figure 4.1: **VR/AR HMD Assisted 3D Capture** An interface for guiding a user to capture a NeRF dataset using head-mounted cameras and headset poses from a VR/AR headset

system, should identify and segment the possible objects in the scene that are interactable such as furniture or small items, and could ignore parts like the room or vents which are fixed to the environment. Next, with the individually segmented objects, they should be "completed" or "repaired" to make it complete for the missing regions of the object that were occluded. For instance, if there is a table with a lot of objects on it, the system should first segment out the other objects, and then inpaint all the regions where those objects were, and eventually to output a clear, clean table ready to be repositioned or edited in the system. In the instance that a user is only editing an object and in more detail, the semantic grouping could have a lower level of abstraction among the individual components of a single object, which could be identified with GARField [26], allowing edits towards specific parts like the arm rests of a chair or wheels on a car. In an ideal scenario this would run in real time as the user is editing in the system since any number of objects could be selected and the level of abstraction of component selection could vary.

4.4 Lower-Fidelity Representations as Input to Generate High Fidelity Scenes

In addition to incorporating captured radiance fields into this system, other methods of generating scenes could be used to add or stylize existing 3D scenes. Scenes and objects could be designed at a higher abstraction level through primitive objects, as discussed in the Dreamcrafter stable diffusion scene editor. These lower fidelity representations are much easier to design and iterate, and can offer a variety of different higher fidelity generations from the given arrangement of primitives using methods from stable diffusion and ControlNet generalized to 3D objects and scenes. These lower fidelity proxy representations, optionally paired with semantic information like text prompts, could help add controllability in 3D scene generations. Once converted to a radiance field, it can be edited in methods discussed earlier. Arrangements of proxy representations could also be sourced from other mediums such as images or videos of arrangements of physical objects or gestures/motion from users. In the case of virtual production and pre-visualization, methods discussed above could be used to create a system that enables users to create low fidelity approximations of scenes, movement of objects, and camera movement as input modalities to generate a stylized high fidelity output from a video diffusion model. As described in Sora's technical report [3], video diffusion models may have the potential to generate large scale 3D scenes. These could be also edited through methods discussed above or used to complete or extend 3D scenes. These methods should leverage all capabilities of the editing and generation systems.

4.5 Conclusion

Recent advances in radiance fields and generative AI systems have enabled new forms of interactions and systems for creative tasks. When imagining the functionality of future of 3D editors, we can leverage novel 3D representations like NeRFs and 3DGS for high fidelity reconstructions and 3D content paired with generative models, including LLMs and diffusion models, to design adaptable novel, intuitive, and multi-modal interactions.

Bibliography

[1] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. "ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models". In: *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*. UIST '08. Monterey, CA, USA: Association for Computing Machinery, 2008, pp. 151–160. ISBN: 9781595939753. DOI: `10.1145/1449715.1449740`. URL: `https://doi.org/10.1145/1449715.1449740`.

[2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. "InstructPix2Pix: Learning to Follow Image Editing Instructions". In: *CVPR*. 2023.

[3] Tim Brooks et al. *Sora: Video generation models as world simulators*. 2024. URL: `https://openai.com/index/video-generation-models-as-world-simulators/`.

[4] Michael Broxton et al. "Immersive Light Field Video with a Layered Mesh Representation". In: 39.4 (2020), 86:1–86:15.

[5] Tilt Brush. *https://www.tiltbrush.com/*. 2016.

[6] Jeffrey A. Butterworth et al. "3DM: a three dimensional modeler using a head-mounted display". In: *ACM Symposium on Interactive 3D Graphics and Games*. 1992. URL: `https://api.semanticscholar.org/CorpusID:9197179`.

[7] Karin Coninx, Frank Van Reeth, and Eddy Flerackers. "A hybrid 2D/3D user interface for immersive object modeling". In: *Proceedings Computer Graphics International*. IEEE. 1997, pp. 47–55.

[8] Fernanda De La Torre et al. "Llmr: Real-time prompting of interactive worlds using large language models". In: *arXiv preprint arXiv:2309.12276* (2023).

[9] Jan-Niklas Dihlmann, Andreas Engelhardt, and Hendrik P.A. Lensch. *SIGNeRF: Scene Integrated Generation for Neural Radiance Fields*. 2024.

[10] Dreams. *https://www.playstation.com/en-us/games/dreams/*. 2020.

[11] Jakob Engel et al. *Project Aria: A New Tool for Egocentric Multi-Modal AI Research*. 2023. arXiv: `2308.13561 [cs.HC]`.

[12] Jiemin Fang et al. "GaussianEditor: Editing 3D Gaussians Delicately with Text Instructions". In: *CVPR*. 2024.

[13] Faraz Faruqi et al. “Style2Fab: Functionality-Aware Segmentation for Fabricating Personalized 3D Models with Generative AI”. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 2023, pp. 1–13.

[14] Ruiqi Gao* et al. “CAT3D: Create Anything in 3D with Multi-View Diffusion Models”. In: *arXiv* (2024).

[15] Qiao Gu et al. “ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning”. In: *arXiv* (2023).

[16] Sujuan Han, Shuo Liu, and Lili Ren. “Application of human-computer interaction virtual reality technology in urban cultural creative design”. en. In: *Sci. Rep.* 13.1 (Sept. 2023), p. 14352.

[17] Ayaan Haque et al. “Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

[18] Yining Hong et al. *3D-LLM: Injecting the 3D World into Large Language Models*. 2023. arXiv: `2307.12981 [cs.CV]`.

[19] Edwin L Hutchins, James D Hollan, and Donald A Norman. “Direct manipulation interfaces”. In: *Human–computer interaction* 1.4 (1985), pp. 311–338.

[20] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. “Teddy: a sketching interface for 3D freeform design”. In: *ACM SIGGRAPH 2006 Courses*. SIGGRAPH ’06. Boston, Massachusetts: Association for Computing Machinery, 2006, 11–es. ISBN: 1595933646. DOI: `10.1145/1185657.1185772`. URL: `https://doi.org/10.1145/1185657.1185772`.

[21] Nicholas Jennings et al. “What’s the Game, then? Opportunities and Challenges for Runtime Behavior Generation”. In: (2023).

[22] Ying Jiang et al. “VR-GS: A Physical Dynamics-Aware Interactive Gaussian Splatting System in Virtual Reality”. In: *arXiv preprint arXiv:2401.16663* (2024).

[23] Heewoo Jun and Alex Nichol. “Shap-e: Generating conditional 3d implicit functions”. In: *arXiv preprint arXiv:2305.02463* (2023).

[24] Bernhard Kerbl et al. “3D Gaussian Splatting for Real-Time Radiance Field Rendering”. In: *ACM Transactions on Graphics* 42.4 (July 2023). URL: `https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/`.

[25] Justin Kerr et al. “LERF: Language Embedded Radiance Fields”. In: *International Conference on Computer Vision (ICCV)*. 2023.

[26] Chung Min Kim et al. *GARField: Group Anything with Radiance Fields*. 2024. arXiv: `2401.09419 [cs.CV]`.

[27] Michael Laielli et al. "LabelAR: A Spatial Guidance Interface for Fast Computer Vision Image Collection". In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, pp. 987–998. ISBN: 9781450368162. DOI: `10.1145/3332165.3347927`. URL: `https://doi.org/10.1145/3332165.3347927`.

[28] *Luma Labs AI*. 2023. URL: `lumalabs.ai`.

[29] Hidenobu Matsuki et al. "Gaussian Splatting SLAM". In: (2024).

[30] Ben Mildenhall et al. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *ECCV*. 2020.

[31] Mark Mine. "ISAAC : A Virtual Environment Tool for the Interactive Construction of Virtual Worlds". In: (June 1995).

[32] Thomas Müller et al. "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding". In: *ACM Trans. Graph.* 41.4 (July 2022), 102:1–102:15. DOI: `10.1145/3528223.3530127`. URL: `https://doi.org/10.1145/3528223.3530127`.

[33] Ryan S. Overbeck et al. "A system for acquiring, processing, and rendering panoramic light field stills for virtual reality". In: *ACM Trans. Graph.* 37.6 (Dec. 2018). ISSN: 0730-0301. DOI: `10.1145/3272127.3275031`. URL: `https://doi.org/10.1145/3272127.3275031`.

[34] Kevin Ponto et al. "SculptUp: A rapid, immersive 3D modeling environment". In: *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. 2013, pp. 199–200. DOI: `10.1109/3DUI.2013.6550247`.

[35] Ben Poole et al. "DreamFusion: Text-to-3D using 2D Diffusion". In: *arXiv* (2022).

[36] Aras Pranckevičius. *Gaussian Splatting playground in Unity*. 2023. URL: `https://github.com/aras-p/UnityGaussianSplatting`.

[37] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: `2103.00020 [cs.CV]`.

[38] Ram Ramrakhya et al. *Seeing the Unseen: Visual Common Sense for Semantic Placement*. 2024. arXiv: `2401.07770 [cs.CV]`.

[39] Mitchel Resnick et al. *Design principles for tools to support creative thinking.(2005)*. 2005.

[40] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: `2112.10752 [cs.CV]`.

[41] Ben Shneiderman and Pattie Maes. "Direct manipulation vs. interface agents". In: *interactions* 4.6 (1997), pp. 42–61.

[42] Jaidev Shriram et al. *RealmDreamer: Text-Driven 3D Scene Generation with Inpainting and Depth Diffusion*. 2024. arXiv: `2404.07199 [cs.CV]`.

[43] Matthew Tancik et al. "Nerfstudio: A Modular Framework for Neural Radiance Field Development". In: *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*. ACM, July 2023. DOI: `10.1145/3588432.3591516`. URL: `https://doi.org/10.1145%2F3588432.3591516`.

[44] Cyrus Vachha. *Creating Visual Effects with Neural Radiance Fields*. 2023. arXiv: `2401.08633 [cs.CV]`.

[45] Cyrus Vachha. *NeRF Environment Creation System for VR*. 2022. URL: `https://cvachha.github.io/nerfenvironmentcreation.html`.

[46] Cyrus Vachha and Ayaan Haque. *Instruct-GS2GS: Editing 3D Gaussian Splats with Instructions*. 2024. URL: `https://instruct-gs2gs.github.io/`.

[47] Cyrus Vachha et al. "Dreamcrafter: Immersive Editing of 3D Radiance Fields Through Flexible, Generative Inputs and Outputs". In: (2024).

[48] *Volinga AI*. 2023. URL: `volinga.ai`.

[49] Meta Horizon Worlds. *http://www.oculus.com/facebookhorizon*. 2020.

[50] Rundi Wu et al. "ReconFusion: 3D Reconstruction with Diffusion Priors". In: *arXiv* (2023).

[51] Figmin XR. *https://overlaymr.com/*. 2022.

[52] Yinghao Xu et al. *GRM: Large Gaussian Reconstruction Model for Efficient 3D Reconstruction and Generation*. 2024. arXiv: `2403.14621 [cs.CV]`.

[53] Luke Yoffe, Aditya Sharma, and Tobias Höllerer. *OCTOPUS: Open-vocabulary Content Tracking and Object Placement Using Semantic Understanding in Mixed Reality*. 2023. arXiv: `2312.12815 [cs.CV]`.

[54] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. *Adding Conditional Control to Text-to-Image Diffusion Models*. 2023. arXiv: `2302.05543 [cs.CV]`.

[55] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. *IC-Light GitHub Page*. 2024.

[56] Xiaoyu Zhou et al. *GALA3D: Towards Text-to-3D Complex Scene Generation via Layout-guided Generative Gaussian Splatting*. 2024. arXiv: `2402.07207 [cs.CV]`.