Dreamcrafter: Immersive Editing of 3D Radiance Fields Through Flexible, Generative Inputs and Outputs

Cyrus Vachha[1], Yixiao Kang[1], Zachary Dive[1], Ashwat Chidambaram[1], Anik Gupta[1],
Eunice Jun[2], Björn Hartmann[1]
[1]University of California, Berkeley
[2]University of California, Los Angeles
{cvachha, yixiao_kang, zach_dive, ashwatc, anik.gupta, bjoern}@berkeley.edu, emjun@cs.ucla.edu

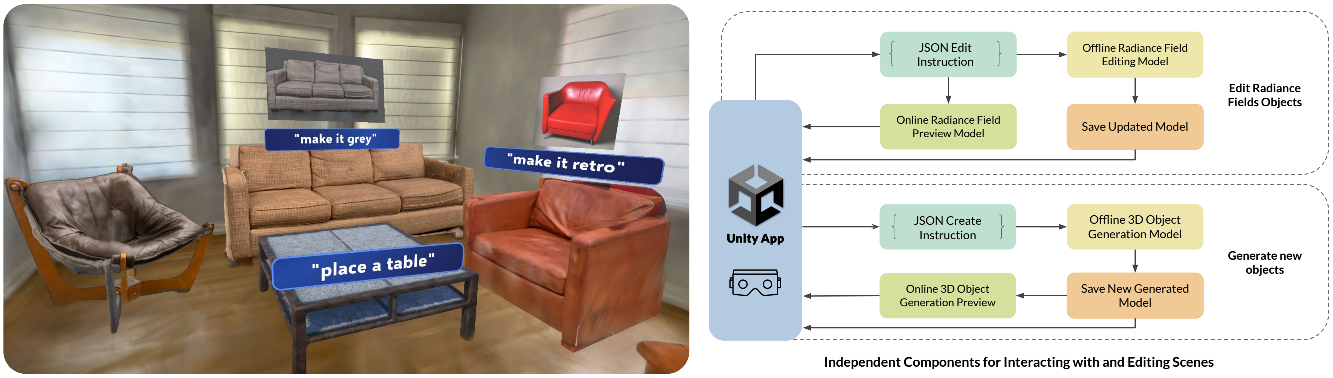
"make it grey"
"make it retro"
"place a table"
JSON Edit Instruction
Offline Radiance Field Editing Model
Online Radiance Field Preview Model
Save Updated Model
Edit Radiance Fields Objects
Unity App
JSON Create Instruction
Offline 3D Object Generation Model
Online 3D Object Generation Preview
Save New Generated Model
Generate new objects
Independent Components for Interacting with and Editing Scenes

Figure 1: System Overview: (Left) View of edits and spatial annotations made in scene. (Right) Pipeline overview of system including three distinct modules to edit scenes: edit NeRF models, create new NeRF models, and create 2D stable diffusion renders of scenes.

ABSTRACT

Authoring 3D scenes is a central task for spatial computing applications. 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 contribute empirical findings on control preferences and discuss how generative AI interfaces beyond text input enhance creativity in scene editing.

1 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., [2, 19]), 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 [18] 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 [24] and 3D Gaussian Splatting [22]) 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 [16]). 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 the 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 [33]. 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 technical contributions: (1) Photorealistic scene representation. We use Gaussian splatting and neural radiance fields (NeRF) instead of traditional mesh-based representations. (2) Modular architecture. This enables the system to continuously integrate state of art generative AI models and leverage both 2D and 3D proxy representations. (3) Flexibility in scene editing. A combination of voice prompts and detailed sculpting using primitives gives both general and advanced users extensive flexibility.

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 eight participants. Using Dreamcrafter, participants could either (i) generate entire objects using AI or (ii) first construct 3D objects using pre-defined shapes (i.e., spheres, cubes, etc.) and then stylize the construtions using generative AI. While participants created more objects using the former interaction, they felt more in control with the latter interaction. Regardless of generation approach, participants find the proxy previews useful for scene composition.

2 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) [24] and Gaussian Splatting [22]. 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 is akin to NeRFs. The main difference is that Gaussian Splatting uses 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 [32] is a deep learning model for synthesizing, or generating, images from text inputs using a diffusion model. ControlNet [39] is a network architecture enhancement to text-to-image models to condition the model on an input image, generating stylized outputs.

3 RELATED WORK

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

3.1 Generating and editing novel 3D representations using NeRFs

Several recent rendering techniques build upon NeRFs [24] and 3D Gaussian Splatting (3DGS) [22]. For example, LERFs [23], or Language Embedded Radiance Fields use CLIP embeddings [30] to allow users to query a NeRF using natural language to determine regions of interest. ConceptGraphs [14] 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 [1], and research friendly APIs such as Nerfstudio [34] and Instant-NGP [26]. Instruct-NeRF2NeRF [16] 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 [3]. We interface with Instruct-NeRF2NeRF to allow users to edit their 3D scenes and objects. DreamFusion [28] also allows users to build a NeRF or mesh from a text prompt.

Approaches to interactive editing of radiance fields are emerging. NeRFShop [26] allows selecting, transforming, or warping NeRF objects in a single scene with real-time feedback; however it lacks generation capabilities. GaussianEditor [7] presents a web based interface for 3DGS including object generation, but crucially does not offer real-time proxies, making edits visible in 10-15 minutes. Neither offer immersive interfaces.

3.2 Generative Scene Building Systems

Previous work has explored how to build editing interfaces for generative AI models for 2D images or 3D meshes. WorldSmith [9] generates 2D scenes composed of multiple text prompts and blends across the generated images tiles. Similar to Dreamcrafter, it allows users different abstractions of editing such as to generate images via prompting or sketching but offers neither immersive interactions nor 3D results. Text2Room[17] generates a fixed 3D room given a single text prompt. Dreamcrafter allows editing of existing 3D scenes and generation of individual objects.

3.3 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 [25] introduced scene editing within Virtual Environments, allowing for a more intuitive and immersive design process. Coninx et al. investigated hybrid 2D and 3D editing [8]. CaveCAD [27], 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 [11], Figmin XR [37], and Horizon Worlds [36] 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 [15]. Several projects explore immersive scene editing for related domains. Flowmatic explores arranging interactive elements. 360proto enables prototyping VR and AR interfaces through paper mockups. 360proto can help visually arrange scenes through layering but has limited editing capabilities. Neither focuses on interactions for generative AI.

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 [10] 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[12] also demonstrates the ability of generative model in personalized 3D model generation. The Dynamics-Aware Interactive Gaussian Splatting System [20] 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.

4 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 provides 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.

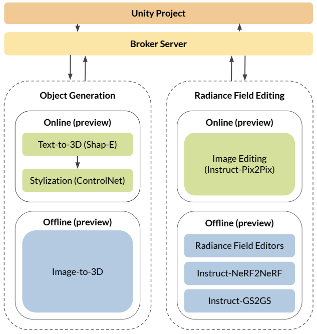

Figure 2: Dreamcrafter system overview. Modules processing pipeline: The Unity project sends API calls to the broker server to run instructions from specific generation modules and their outputs get sent back to the Unity project. Online modules are run for previewing generations, and offline modules are run after editing is complete.

5 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). 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 1 shows spatial annotations applied in a scene.

5.1 Key interactions

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

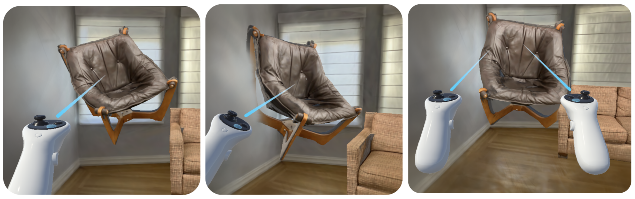

Figure 3: Object transformations and direct manipulations (Left) Positioning object in the scene (Center) Rotating object. (Right) Scaling object

5.1.1 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 3 illustrates this interaction.

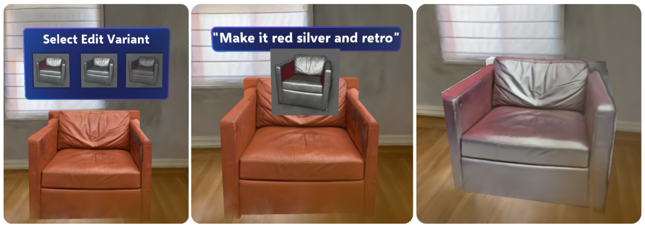

Figure 4: Radiance Field Object Editing with preview (Left) Edit variants are presented to a user. (Center) Displaying selected edit preview as a spatial annotation. (Right) Fully processed 3D edit replaces the original

5.1.2 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 4. A render of the object is given to the Instruct-Pix2Pix module, which applies the instruction to 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. Edits take approximately 10 seconds to generate previews. Objects can be duplicated, re-edited, or deleted.

5.1.3 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 5). This sends an API call to the 3D generative module that includes Shap-E [21], which generates a low fidelity mesh and render, and the render is stylized using depth conditioned ControlNet [39] 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 user can select from three stylized 2D image variants of the object. Generations take approximately 15 seconds to generate previews. During an offline process, the full fidelity 3D objects are generated and placed in the scene.

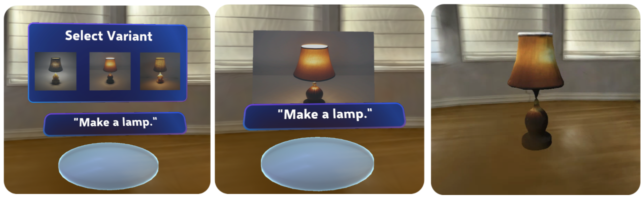

Figure 5: Object Generation via Prompting (Left) Object generation variations from speech input. (Center) Displaying selected generation preview as a spatial annotation. (Right) Fully processed 3D generation in the scene.

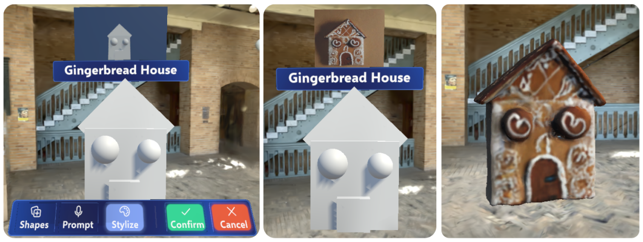

Figure 6: Object Generation via Sculpting (Left) Sculpting toolkit to create primitive shape arrangement (Center) Displaying stylized sculpted object preview as a spatial annotation. (Right) Fully processed 3D generation in the scene.

5.1.4 Sculpt then stylize objects. Alternatively, users can generate objects by creating an arrangement of basic 3D primitives (i.e., spheres, cubes, and cylinders) (Figure 6). A limited set of tools are provided to position, rotate, uniformly scale the shapes. Users then take a snapshot of this arrangement, which is stylized 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.

5.2 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 4 (center) and Figure 5 (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.

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.

Good proxies should be fast to generate and accurate in previewing the final object. Dreamcrafter uses 2D image proxies because existing 3D object editing and generation pipelines use 2D images under the hood. For example, Instruct-GS2GS uses Instruct-Pix2Pix to first generate a 2D image from a natural language prompt and then transform the 3D scene into its edited version guided by the 2D images. By accessing the generated Instruct-Pix2Pix 2D image as the proxy in seconds, Dreamcrafter is able to show a preview quickly and, critically, by design, ensure that the 2D image is an accurate proxy of the 3D object.

In other words, Dreamcrafter's approach to generating proxy representations contributes a generalizable template for leveraging intermediate representations of high-latency 3D operations as proxies.

5.3 Modular System Design using Generative AI modules

Dreamcrafter's Unity client offers a modular interface to multiple plug-and-play modules for real-time interactive and offline processing tasks. The system is designed to easily update to newer iterations of these generative AI models, which are commonly developed due to rapid interest in this field.

5.3.1 Online Processing Modules. A set of generative modules are used to create rapid previews visible in the VR scene. Radiance field object editing tasks use Instruct-Pix2Pix, an intermediate model used for the full 3D edit which runs in 15 seconds. the object generation via prompting instruction use an text to 3D module, Shap-E, to generate a low fidelity mesh and NeRF render. This render is then stylized with ControlNet conditioned on edges and the same text prompt to create three 2D preview variant generations. We use Shap-E since it creates a render of a single object and an object centric generation than a regular text to image model, and provides a close approximation of using a more detailed text to 3D system. Object generation via sculpting displays a 2D preview generated via ControlNet conditioned on depth and a snapshot of the arranged 3D primitives. The sculpted arrangement acts as a 3D proxy representation.

5.3.2 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. 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 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 [38] or text-to-3D based system. The final 3D object edits are done using Instruct-NeRF2NeRF for NeRFs, or Instruct-GS2GS [35] 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.

5.4 Scene Outputs

5.4.1 3D Scenes. After offline processing, fully edited scenes can be viewed as a 3D Unity scene composed of radiance field objects and meshes. Optionally, training images can be captured of the scene to create a radiance field of the entire scene.

5.4.2 Magic Camera. Users can position a virtual camera, we call the Magic Camera, which stylizes a snapshot of a view of the scene given a prompt through the ControlNet module. The resulting stylization gives a coherent and realistic composition of the scene based on the content and arrangement of objects, analogous to rendering a frame in a traditional 3D editor. This feature can be extended to act as a method of controlability in AI generated images or video by using This 2D image output as input to an image-to-video model. The magic camera output image could potentially be used as input into to an image-to-3D scene system [13] which would generate a 3DGS scene, editable in Dreamcrafter. 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.

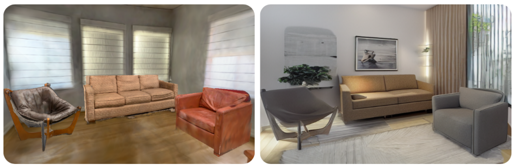

Figure 7: Magic Camera (Left) Scene input from virtual Unity camera (Right) Stylized image output from Magic Camera given the prompt: "realistic apartment living room"

5.5 Additional implementation details

Dreamcrafter is implemented in Unity using the Unity XR toolkit and MRTK plugins for VR support. Gaussian Splats were rendered using the open source Unity Gaussian Splatting viewer [29], and example splats were trained using Nerfstudio and Luma AI. The Unity app interfaces with the online generative modules using a sending calls from a C# server to a python flask server which makes API calls to separate generative with specified parameters.

6 EVALUATION

Two research questions motivated the evaluation:

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?

6.1 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 could duplicate or edit.
- **Photo area for party guests.** Participants decorated an open 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 and showcase the flexibility of tools supported. The dining area task was the most scaffolded, with relevant objects populating the scene already and a small object library of radiance field objects was given. We anticipated that this would encourage participants to edit the existing radiance field objects or add relevant objects from the object library. The photo area was more open-ended with opportunities to place objects and generate new ones via prompting or sculpting in an open area of the room. We use this task to examine when users decide to prompt or sculpt and the benefits of the proxy representations for scene composition. 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 interaction as they saw fit.

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

6.2 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.

6.3 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.

6.4 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." due to the sometimes unexpected results from the generative models.

6.4.1 *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 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.

Table 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 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 area	Photo area	Gingerbread house
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)

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, 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.

6.4.2 *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]. For example, P1 described how the previews were "helpful to put stuff around and see how it works for each other." Similarly, P3 remarked how each preview "helps for arrangement in the space."

Participants also reported that the image previews helped them visualize individual objects [P4, P5, P6]. For instance, P6 said "It was easy to create an object that was somewhat close to what I was envisioning based on the preview it generated."

Despite reporting that the were previews helpful for scene composition and object styling, when asked how sure they were about how the final scene would look, P1, P2, and P6 reported feeling unsure, rating their certainty at a 1 or 2 on the five-point scale. The median score across all participants was a 3 out of 5. P5, who found previews helpful for envisioning individual objects but not the entire scene, pointed out a key limitation of the previews was that size information was lost: "Some preview of the size an object would take would be useful for just the prompting / not sculpting part." Therefore, proxy representations, while helpful for drafting scenes and objects, are incomplete for fine-grained scene layout and detailed editing.

Participants first envisioned and then described to the researchers a scene based on the task instruction, then generated objects, and finally positioned them. Some participants had a particular style in mind (P1, P5) and tended to generate/edit objects to achieve this style. Four participants (P1, P4, P5, P6) chose to use sculpting for the gingerbread house construction task to control generated details (e.g., placing two spherical windows above a rectangular door). During the dining scene, most participants opted to use the existing radiance field objects for tables and chairs. Four participants further stylized the existing objects to be consistent with their desired theme (e.g., Game of Thrones-esque).

6.4.3 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.*"

A noticeable limitation during the tasks was that the sculpting tool was sometimes difficult to use effectively. It took some time for the users to create the desired arrangement of shapes and users wanted additional familiar functionality present in most other systems (duplication, grouping, deletion). This difficulty may have influenced their experience and affected the accuracy of the comparison with prompting, which was much easier to use.

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 give them more time to express all the details they had in mind. In addition, the text-to-image models sometimes provided unexpected generations which required users to re-prompt the system multiple times.

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.

6.5 Revisions to System

Based on our preliminary user study, we updated the system to address user concerns and improve existing features. Based on feedback regarding the 2D proxies (specifically from P5's comments on scene composition), we implemented 3D proxy representations for object generation. This method imports the intermediate low-fidelity mesh generated by Shap-e which can then be placed and scaled in the scene and give the users a better sense of the object placement, as well as work better with the Magic Camera by providing a reference for an object. We show a comparison between the original 2D and new 3D proxy in Figure 8.

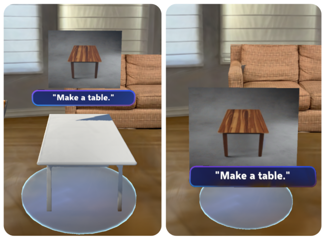

Figure 8: 3D and 2D Proxy Representations (Left) New 3D proxy showing a low fidelity mesh preview (Right) Original 2D proxy representation with image preview

To prevent the need for users to over-explain a prompt to get a detailed stylistic generation, we also experimented with the concept of generating more detailed prompts with additional scene specific context by appending specific keywords to the object generation prompt to make the generation stylistically consistent with the scene objects. We use GPT-4o with vision and prompt "*Act as an AI world building assistant and given this is the view from my vr headset, I want to use speech to generate a new object in this space and given the prompt, a ML model creates a 3D object out of it. Given a prompt and image, I want you to make sure the object appears stylistically similar to the scene shown in the image and other objects by adding additional keywords to the prompt to describe the color, material, and other structural details. I will tell you a prompt and give an image and you will give the slightly longer version of the prompt with the detail to make it stylistically consistent.*" Given an images of the scene and a very short prompt of the object, it adds descriptions of materials and colors so the generated object matches with other objects in the scene. We experimented with this pipeline independently, but have not integrated it in the full system yet.

7 DISCUSSION

We investigate how to incorporate the benefits of real-time, immersive editing and the advantages of high-level scene editing using

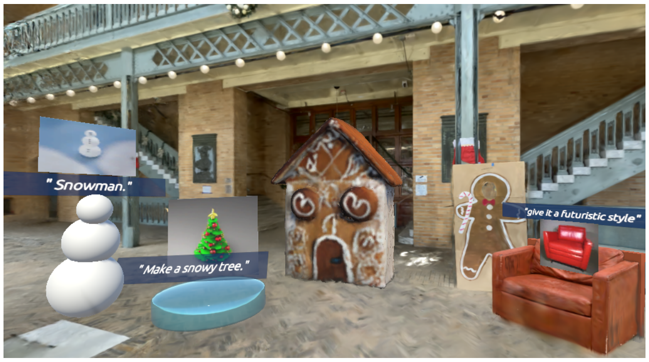

Figure 9: Spatial annotation tags are placed over the radiance field objects and generated objects with given instructions and preview generations.

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 [31].

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.

Applicability to future user interfaces for generative models. We believe that Dreamcrafter could also act as a world creation or staging tool for other generative AI design systems for 2D or video output, we call spatial prompting. There is a desire for more visual interfaces to image/video generative models in consumer applications. A system we explored during the project's development was using the Magic Camera to pre-visualize stylized scenes through ControlNet and Stable Diffusion based on the construction of a scene of only primitive objects, created and arranged within the VR interface. Even with minimal object detail (e.g., cubes as a couch), the system produced highly stylized, recognizable scenes and objects based on a single global scene prompt. Future improvements could involve tagging objects for individual stylization and converting 2D renders into 3D scenes. Dreamcrafter could serve as an early exploration into spatial prompting systems that offer more control for 2D/3D/video scene generation systems beyond limited text prompting interfaces which are currently in SOTA consumer applications. Scenes and objects could be designed at a higher abstraction level through primitive objects. 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. 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, potentially using an LLM to interpret vague instructions. 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 [4], video diffusion models may have the potential to generate large scale 3D scenes and virtual worlds. These could be also edited through methods in Dreamcrafter discussed above or used to complete or extend 3D scenes. These methods could leverage all capabilities of the editing and generation systems presented in Dreamcrafter.

8 LIMITATIONS AND FUTURE WORK

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

Global scene editing. Dreamcrafter supports editing and creating individual 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 ways to control generation. A key focus of future work should be the development of more intuitive ways to generate radiance field objects. For instance, rather than rely solely on voice commands, what if users could Dreamcrafter with text or 2D/3D sketches as input, which then get translated into or serve as generative AI prompts? Incorporating voice commands for positioning like "place the table next to the blue chair" would make the system more user-friendly without having to manually place objects.

We anticipate that Dreamcrafter's modular design will help explore new interaction techniques. Dreamcrafter has separate modules for object generation, for using AI to create new objects, and spatial annotation, for placing objects in the scene. By separating these concerns, Dreamcrafter has the potential to evolve with not only new AI technologies but also new 3D representations (i.e., whatever may replace radiance fields for photorealistic rendering in the future).

Even more rapid proxies. While Dreamcrafter currently supports speech-to-text 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 wireframe outlines just as useful as the 2D image previews? Furthermore, if users could stylize entire scenes, what would the appropriate proxy for the entire scene be? A sketch of the new alongside the old?

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.

We believe that these avenues of future work can apply to future 3D editors.

9 CONCLUSION

The idea behind Dreamcrafter is to use direct manipulation for spatial positioning and layout; and leverage generative AI for editing style and appearance of photorealistic 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.

REFERENCES

[1] Luma Labs AI. 2023. lumalabs.ai.

[2] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2008. 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* (Monterey, CA, USA) *(UIST '08)*. Association for Computing Machinery, New York, NY, USA, 151–160. https://doi.org/10.1145/1449715.1449740

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

[4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. 2024. Sora: Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/

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

[6] Jeffrey A. Butterworth, Andrew Davidson, Stephen Hench, and Marc Olano. 1992. 3DM: a three dimensional modeler using a head-mounted display. In *ACM Symposium on Interactive 3D Graphics and Games*. https://api.semanticscholar.org/CorpusID:9197179

[7] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. 2023. GaussianEditor: Swift and Controllable 3D Editing with Gaussian Splatting. arXiv:2311.14521 [cs.CV] https://arxiv.org/abs/2311.14521

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

[9] Hai Dang, Frederik Brudy, George Fitzmaurice, and Fraser Anderson. 2023. WorldSmith: Iterative and Expressive Prompting for World Building with a Generative AI. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 63, 17 pages. https://doi.org/10.1145/3586183.3606772

[10] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2023. Llmr: Real-time prompting of interactive worlds using large language models. *arXiv preprint arXiv:2309.12276* (2023).

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

[12] Faraz Faruqi, Ahmed Katary, Tarik Hasic, Amira Abdel-Rahman, Nayeemur Rahman, Leandra Tejedor, Mackenzie Leake, Megan Hofmann, and Stefanie Mueller. 2023. 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*. 1–13.

[13] Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. 2024. CAT3D: Create Anything in 3D with Multi-View Diffusion Models. *arXiv* (2024).

[14] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. 2023. ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning. *arXiv* (2023).

[15] Sujuan Han, Shuo Liu, and Lili Ren. 2023. Application of human-computer interaction virtual reality technology in urban cultural creative design. *Sci. Rep.* 13, 1 (Sept. 2023), 14352.

[16] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. 2023. Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

[17] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. 2023. Text2Room: Extracting Textured 3D Meshes from 2D Text-to-Image Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 7909–7920.

[18] Edwin L Hutchins, James D Hollan, and Donald A Norman. 1985. Direct manipulation interfaces. *Human–computer interaction* 1, 4 (1985), 311–338.

[19] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. 2006. Teddy: a sketching interface for 3D freeform design. In *ACM SIGGRAPH 2006 Courses* (Boston, Massachusetts) *(SIGGRAPH '06)*. Association for Computing Machinery, New York, NY, USA, 11–es. https://doi.org/10.1145/1185657.1185772

[20] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. 2024. VR-GS: A Physical Dynamics-Aware Interactive Gaussian Splatting System in Virtual Reality. *arXiv preprint arXiv:2401.16663* (2024).

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

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

[23] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. 2023. LERF: Language Embedded Radiance Fields. In *International Conference on Computer Vision (ICCV)*.

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

[25] Mark Mine. 1995. ISAAC : A Virtual Environment Tool for the Interactive Construction of Virtual Worlds. (06 1995).

[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127

[27] Kevin Ponto, Ross Tredinnick, Aaron Bartholomew, Carrie Roy, Dan Szafir, Daniel Greenheck, and Joe Kohlmann. 2013. SculptUp: A rapid, immersive 3D modeling environment. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. 199–200. https://doi.org/10.1109/3DUI.2013.6550247

[28] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv* (2022).

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

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV]

[31] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. Design principles for tools to support creative thinking.(2005).

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs.CV]

[33] Ben Shneiderman and Pattie Maes. 1997. Direct manipulation vs. interface agents. *interactions* 4, 6 (1997), 42–61.

[34] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. 2023. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*. ACM. https://doi.org/10.1145/3588432.3591516

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

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

[37] Figmin XR. 2022. https://overlaymr.com/.

[38] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. 2024. GRM: Large Gaussian Reconstruction Model for Efficient 3D Reconstruction and Generation. arXiv:2403.14621 [cs.CV]

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