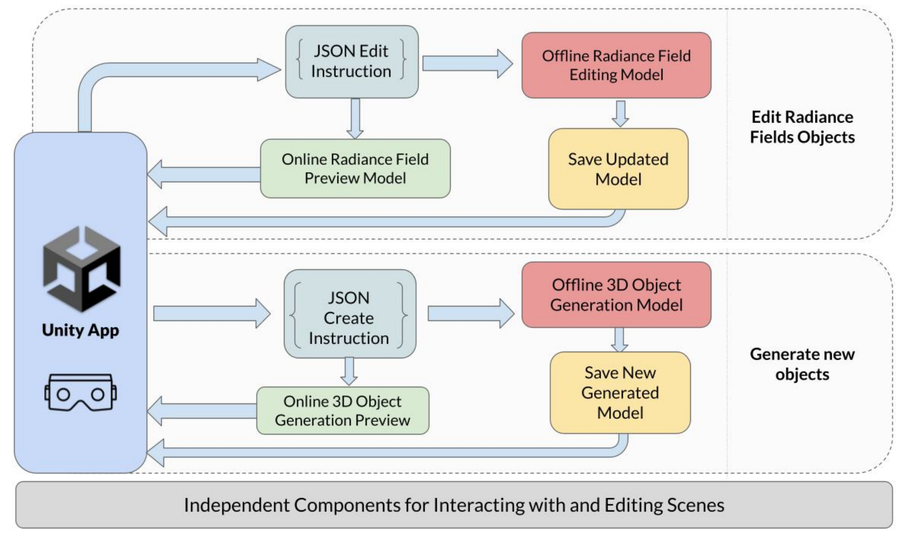
DreamCrafter

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]

{cvachha, yixiao_kang, zach_dive, ashwatc, anik.gupta, bjoern}@berkeley.edu, emjun@cs.ucla.edu

[1]University of California, Berkeley [2]University of California, Los Angeles



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. Dreamcrafter uses direct manipulation for spatial positioning and layout; and leverages 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. 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).

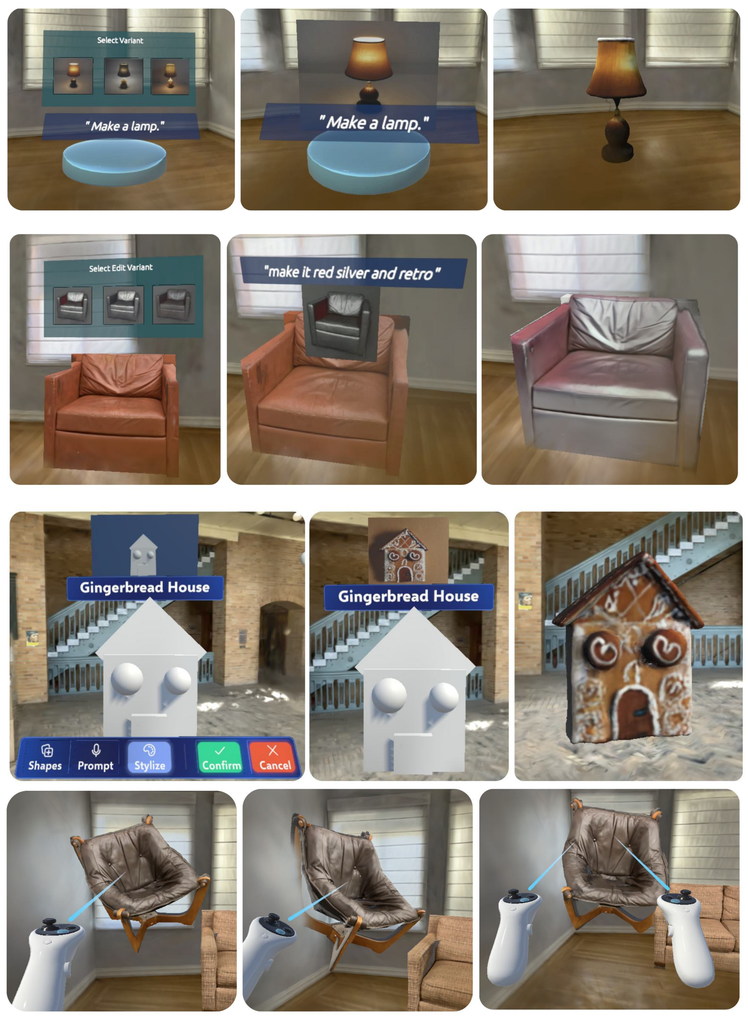
Key Interactions

Generate
Generate Objects Via Prompting

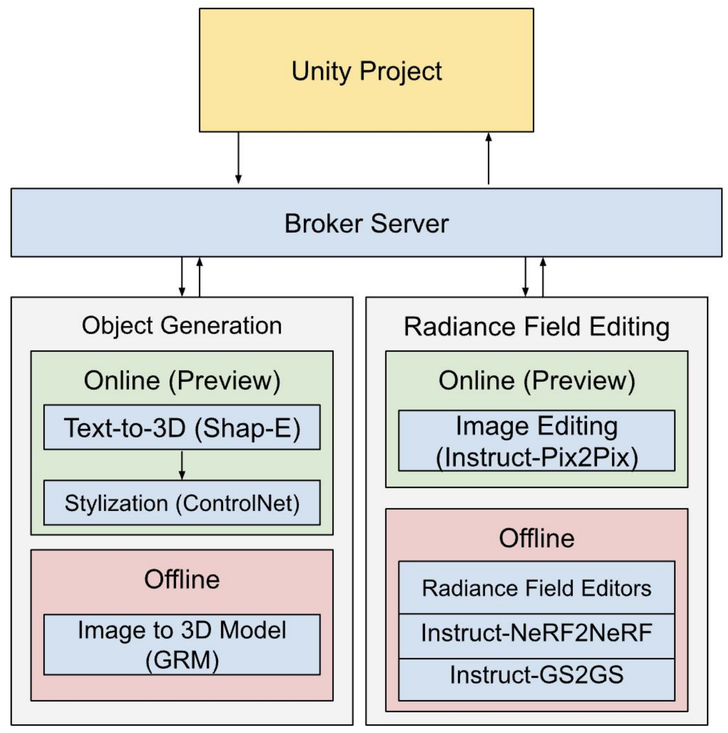
Edit
Edit Radiance Field Objects Via Prompting

Sculpt
Sculpt Then Stylize Objects

Position
Move, Rotate, Scale Objects



3D Generative Modules use 2D Proxy Representations



Generative 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. A Python broker server receives a server message from the Unity project and forwards instruction parameters to the specified module. After the edited objects are added to the scene, users can repeat the process and edit the scene again, creating an iterative design process. The modules are exchangeable and can be implemented to use updated AI models.

Berkeley UNIVERSITY OF CALIFORNIA · JACOBS INSTITUTE FOR DESIGN INNOVATION, COLLEGE OF ENGINEERING, UC BERKELEY · UCLA