GaussianShopVR: Facilitating Immersive 3D Authoring Using Gaussian Splatting in VR

Yulin Shen
Computational Media and Arts
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
yshen654@connect.hkust-gz.edu.cn

Boyu Li
Computational Media and Arts
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
blibr@connect.hkust-gz.edu.cn

Jiayang Huang
Computational Media and Arts
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
jhuang130@connect.hkust-gz.edu.cn

David Yip
Computational Media and Arts
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
daveyip@hkust-gz.edu.cn

Zeyu Wang*
Computational Media and Arts
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China
zeyuwang@ust.hk

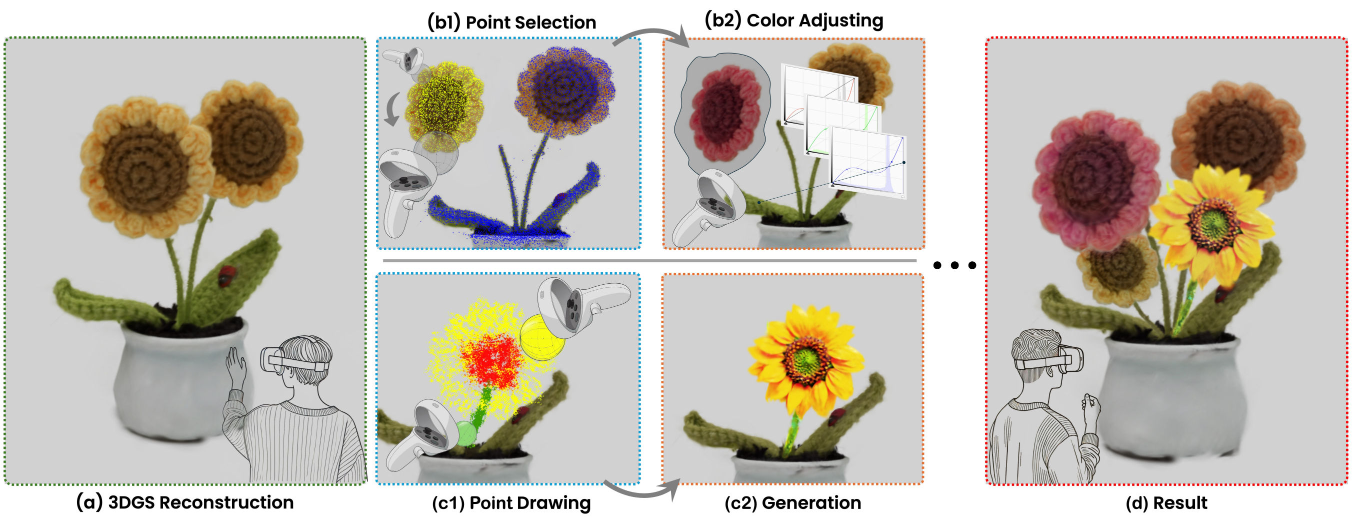

Figure 1: Illustration of the GaussianShopVR system. (a) Users can quickly obtain photorealistic 3D assets via 3D Gaussian Splatting reconstruction as the basis for 3D authoring. (b1) Users can precisely select points to support various subsequent editing tasks, such as object splitting at any level of detail, content removal, inpainting, and (b2) color adjusting. (c1) Users can intuitively draw point clouds in VR to guide and control AI optimization to achieve complex tasks like (c2) generation and 3D inpainting. Integrating intuitive VR interactions, versatile editing functionalities, photorealistic representations, and accessible digital assets, GaussianShopVR introduces a novel approach to immersive 3D content authoring.

*Corresponding author.

Abstract

We present GaussianShopVR, a VR-based authoring system for controllable and fine-grained editing of 3D Gaussian Splatting

UIST '25, Busan, Republic of Korea

ACM ISBN 979-8-4007-2037-6/25/09
https://doi.org/10.1145/3746059.3747803

(3DGS). Although 3DGS has gained popularity for its capability to quickly create digital replicas of real-life scenes and its compatibility with existing rendering pipelines, current methods often struggle with detailed 3D editing due to cumbersome target area selection and inadequate spatial control on AI-driven editing. GaussianShopVR addresses these challenges by harnessing intuitive VR interactions for efficient point selection and drawing. We further enhance editing methods to support interactive creative tasks, including precise object splitting, real-time color adjustment, and controllable object generation from drawn points. We evaluate GaussianShopVR through three user studies focusing on point selection efficiency (N=18), controllable 3D generation (N=20), and overall usability (N=10). The findings suggest that GaussianShopVR provides an immersive, flexible, and controllable approach to 3DGS-based content creation in VR. Our code is available at https://github.com/CISLab-HKUST/GaussianShopVR.

CCS Concepts

• **Human-centered computing → Interactive systems and tools**; • **Computing methodologies → Virtual reality**; *Point-based models.*

Keywords

3D authoring, VR authoring system, Gaussian Splatting editing

ACM Reference Format:
Yulin Shen, Boyu Li, Jiayang Huang, David Yip, and Zeyu Wang. 2025. GaussianShopVR: Facilitating Immersive 3D Authoring Using Gaussian Splatting in VR. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28–October 01, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3746059.3747803

1 Introduction

3D digital content has been ubiquitous in spatial design, animation, games, and virtual reality (VR) applications. However, traditional mesh-based 3D asset creation remains time-consuming and labor-intensive, often requiring geometry modeling, sculpting, UV mapping, and various postprocessing steps. This calls for more efficient and intuitive 3D authoring approaches that can significantly improve the efficiency and quality of 3D asset creation.

3D Gaussian Splatting (3DGS) [27] is regarded as a potential representation for efficient 3D asset creation [62]. It has emerged as a representation suitable for reconstructing 3D content from a collection of images using differentiable rendering techniques. Compared with using digital creation tools [5, 54] for mesh-based modeling, 3DGS can create photorealistic digital replicas of real-life scenes in minutes, which significantly reduces modeling time. Furthermore, it is compatible with existing graphics pipelines and supports real-time rendering, which has significant advantages over other representations like NeRF [38] and DeepSDF [41]. 3DGS is differentiable and can be easily optimized by neural networks, supporting various AI-driven editing tasks, including style transfer [8] and object generation [29, 52]. These advantages offer 3DGS great potential for a new and efficient 3D authoring representation.

Despite these advantages of 3DGS, there is a lack of support for fine-grained and controllable editing of 3DGS (shown in Table 1). In a typical creative editing process, creators often need to accurately select target areas and intuitively modify the content. This poses two key challenges of 3D authoring using the 3DGS representation: difficulties in fine-grained selection and inadequate control during AI-driven editing.

First, existing methods for 3DGS editing [8, 10, 23, 33, 39, 43, 57, 65] struggle with efficiently selecting regions, as 3DGS is a special form of point cloud representation (see Figure 2). Some systems [5, 43, 54] use 2D user interfaces to select points from screen space via raycasting techniques. However, each raycasting operation selects the entire frustum behind the screen. Consequently, users must frequently change their viewpoints and repeatedly select or deselect point clouds, making the process both cumbersome and imprecise. Other 3DGS editing methods [8, 10, 23, 33, 39, 65] lift multiview 2D masks to indicate 3D target areas. The multiview masks can be obtained by manual annotation with image editing software, which is highly time-consuming. AI-based segmentation models [28] can automatically generate masks under different views, but can only work at an object level and may encounter 3D inconsistency caused by occlusions and the performance of AI models.

Second, existing AI-driven editing methods [8, 52, 53, 66] based on text and image prompts fail to have enough control over the editing process, especially in shape control. Text or image prompts used in prior work [8, 52] are inadequate for extracting the necessary 3D information to guide optimization. Unlike 2D editing, which benefits from shape constraints like sketching [70] or semantic maps [42], 3D editing lacks efficient prompts to input 3D priors into neural networks, leading to issues like 3D inconsistency [44, 52] and depth misestimation [50].

To tackle these two challenges, we propose GaussianShopVR, a system that leverages VR user interfaces for 3DGS authoring. Prior research [15, 31, 60, 71] has demonstrated the intuitive nature of 3D interaction in VR for point selection and manipulation. We hypothesize that VR interfaces can significantly help identify editing areas by selecting 3DGS points and drawing new points. Furthermore, selected or drawn 3D content can provide spatial information and constraints as input for generative AI models. To support general editing tasks, we implemented a hierarchical object structure in GaussianShopVR tailored for 3DGS while preserving the differentiability of the entire scene for AI-based optimization. Based on efficient target area identification, we enhance some editing tools and integrate them into GaussianShopVR to support a wide range of creative tasks, including object splitting, object manipulation, object generation, and color adjustment.

We conducted user studies to evaluate the efficiency of selecting 3DGS points using GaussianShopVR, as well as the controllability of AI methods enabled by the identification of 3D target areas. Another user study was conducted to further explore how GaussianShopVR facilitates 3D authoring using 3DGS as a creative medium. The study results show that GaussianShopVR is an effective 3D authoring tool and can support a wide range of content creation and spatial design.

In summary, this paper makes the following contributions:

- We developed GaussianShopVR, a 3D authoring system in VR for controllable and fine-grained 3DGS editing. To the best of our knowledge, GaussianShopVR is the first VR system

Table 1: Comparison of 3DGS editing systems. GaussianShopVR supports point-level editing of 3DGS through both manual and AI-driven methods. It requires no cumbersome preprocessing and uniquely provides immersive spatial control, enabling precise and flexible AI-based operations.

System	w/o Preprocessing	Editable	Manual Editing	AI Editing	Immersive	Spatial Control for AI
GaussianEditor [8]	✓	Object	✗	✓	✗	✗
VR-GS [23]	✗	Object	✓	✗	✓	✗
Dreamcrafter [55]	✓	Object	✓	✓	✓	✗
SuperSplat [43]	✓	Point	✓	✗	✗	✗
Ours	✓	Point	✓	✓	✓	✓

to enable detailed, point-level editing on 3DGS, benefiting from the intuitive spatial interactions in VR.

- We enhanced multiple tools tailored for GaussianShopVR to facilitate interactive 3D authoring. These editing tools include both manual and AI-driven approaches, such as efficient object splitting, controllable object generation, 3D inpainting, color adjustment, and scene composition.
- We conducted three user studies to evaluate the efficiency of 3DGS selection, the controllability of AI generation, and how GaussianShopVR facilitates 3D authoring. The results show that GaussianShopVR offers an intuitive, controllable, and versatile experience of 3DGS-based content creation.

2 Related Work

2.1 VR/AR Interfaces for 3D Authoring

Developing creative tools for immersive content creation is a vibrant area of research. Devices such as head-mounted displays (HMDs), smart glasses, and LiDAR-equipped tablets offer distinct advantages for 3D content creation, enabling direct six-degree-of-freedom interactions for highly arbitrary editing operations. VR painting can provide users with an immersive feeling to stimulate the desire to create and can be regarded as a new art medium [24, 34]. CASSIE [68] leverages freehand mid-air sketching and devises a novel 3D optimization framework to create connected curve network armatures. Yu et al. [67] proposed a method to transform unstructured 3D sketches into piecewise smooth surfaces that preserve the geometric features of sketches and thus convert 3D sketching to 3D objects. Some other works leverage VR/AR devices to facilitate environmental design [16, 58] and character animation [30]. SceneCtrl [69] allows the user to interactively edit the real scene sensed by HoloLens, such that the reality can be adapted to suit virtuality. PointShopAR [59] uses tablets to capture point clouds in augmented reality to support environmental design prototyping. Dreamcrater [55] enables object-level generation and editing of 3DGS, and allows users to directly manipulate 3DGS objects within a VR environment for 3D scene authoring. VR-GS and VR-DoH [23, 36] add physics-based simulations to interact with 3DGS objects in VR and thus are capable of modifying their posture. These works show great potential for using VR/AR interfaces to facilitate 3D authoring.

2.2 Generative AI for 3D Authoring

The difficulty in acquiring 3D assets troubles content creators, and some research tends to leverage generative AI methods to quickly obtain 3D assets. There are two major frameworks: a mesh-based framework and a differentiable representation-based framework.

Mesh-based methods apply AI models to different traditional sections of computer graphics, such as geometry [37, 47, 56], texture [13, 19, 46, 63, 70], and image-based rendering [40, 49].

Differentiable representation-based tends to use implicit representations optimized by deep learning. DeepSDF [41], Neural Radiance Fields (NeRF) [38], 3DGS [27]. show great potentials on object generation [14, 29, 52], scene generation [11, 50] and dynamic content creation [35, 61, 64].

Although recent research has focused on leveraging generative AI to improve 3D generation quality via text or image prompts, few works have focused on effective control over the generation process to obtain specific, desired 3D assets. DreamSketch [32] utilizes sketches as guidance but often misestimates object positions and depth, necessitating manual adjustments. Coin3D [14] introduces geometric primitives as constraints for generative AI models; however, this approach primarily controls coarse-level shapes and lacks the ability to manage fine-grained details. Consequently, achieving controllable, precise, and fine-grained 3D editing and generation remains an open challenge.

2.3 3D Gaussian Splatting Editing

Previous works, such as PointShop3D [73], have focused on editing scanned point data represented as surfels, which are inherently constrained to surfaces. In contrast, 3DGS employs volumetric point representations that extend beyond surfaces into the volume. Due to its volumetric, point cloud-like nature, explicitly editing 3DGS points via traditional 2D interfaces is challenging, significantly limiting their practical application in 3D authoring workflows.

Some methods [8, 23, 28, 33, 39, 57, 72] focus on using 2D images to guide the editing of 3DGS. These 2D images can be obtained via generative AI or manual creation. GaussianEditor [8] uses multiview semantic masks generated by SAM [28] to trace editing areas, while SAM generates 3D inconsistent masks and thus leads to inaccurate 3D editing areas. TIP-Editor [72] uses user-placed 3D bounding boxes, text prompts, and image prompts to edit 3DGS. VR-GS [23], InFusion [33], RefFusion [39], and GScream [57] also rely on user-provided multiview masks to remove objects and design models for inpainting. All of these methods need to use 3D

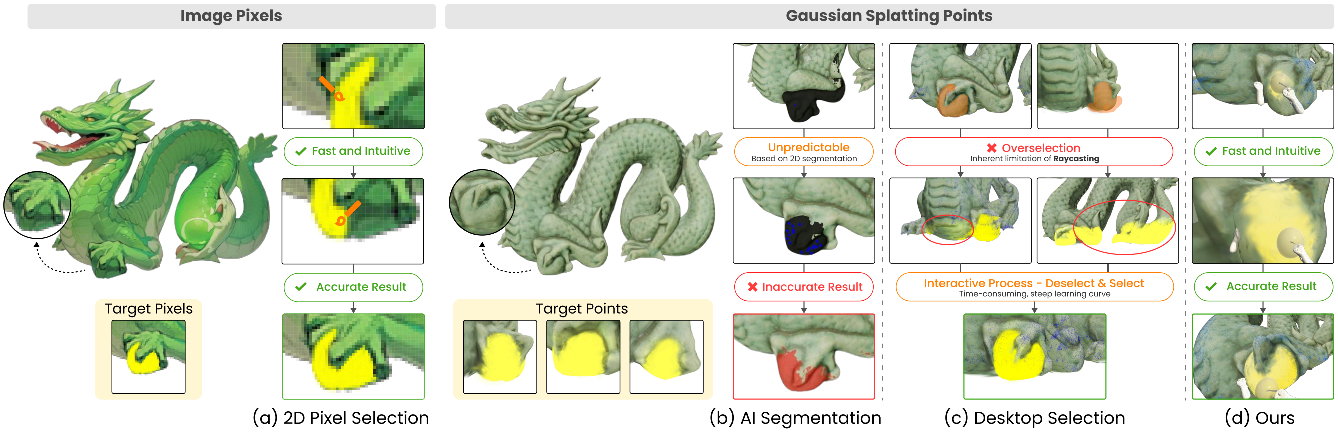

Figure 2: Approaches to 3DGS point selecting. (a) In 2D image editing, designers effortlessly select target regions using an intuitive brush-based pixel selection tool. However, accurately selecting target splats in 3D space for 3D Gaussian Splatting editing is quite challenging. (b) Automated segmentation based on SAM is common for 3DGS editing tools (e.g., GaussianEditor [8]) but often struggles with complex scenes. (c) Desktop software (e.g., SuperSplat [43]) leverages raycasting-based selection that frequently leads to misselection due to depth ambiguity. (d) With our approach, users can interactively "touch" and select points in 3D space via a VR controller, which is more efficient and precise.

information given outside the system, and it is hard to edit arbitrary areas. Moreover, these methods relying on multiview training images only support one-time editing, where once 3DGS scenes are modified, the training images can not be used as guidance for optimization anymore.

Other approaches [8, 66] utilize Score Distillation Sampling (SDS) to enable editing on 3DGS. These methods leverage 3D priors distilled from pretrained 2D diffusion models to guide the optimization of 3DGS models. Despite their promising results, SDS-based approaches typically encounter several limitations, such as prolonged training durations, inconsistencies in generated 3D geometry, and overly high contrast or unnatural appearances.

Recent work [21, 23, 36] has explored interactive modification of 3DGS with spatial input. VR-GS [23] employs object segmentation and image-based inpainting on training images to reconstruct scenes with multiple objects. It visualizes the scenes in VR and incorporates physics-based simulations to enhance interactivity. VR-DoH [36] facilitates direct interaction with 3DGS points in a VR environment, treating them akin to virtual clay. GSDeformer [21] leverages deformation cages to manipulate the pose of 3DGS objects. Although these methods offer intuitive ways to interact with 3DGS, they primarily support modifications at the object level, lacking the ability to precisely edit fine details and intricate structures.

3 System Design

3.1 Preliminaries

3D Gaussian splatting. Gaussian Splatting [27] employs a collection of point-like anisotropic 3D splats. Specifically, each Gaussian is defined by a center $\boldsymbol{x} \in \mathbb{R}^3$, a scaling factor $\boldsymbol{s} \in \mathbb{R}^3$, and a rotation quaternion $\boldsymbol{q} \in \mathbb{R}^4$. Additionally, an opacity value $\alpha \in \mathbb{R}$ and a color feature $\boldsymbol{c} \in \mathbb{R}^C$ for spherical harmonics (SH) coefficients are maintained for rendering, where spherical harmonics can be used to model view-dependent effects. These parameters can be collectively denoted by Θ, with $\Theta_i = \{\boldsymbol{x}_i, \boldsymbol{s}_i, \boldsymbol{q}_i, \alpha_i, \boldsymbol{c}_i\}$ representing the parameters for the i-th Gaussian. Rendering of the 3D Gaussians involves projecting them onto the image plane as 2D Gaussians and performing alpha composition for each pixel in front-to-back depth order, thereby determining the final color. Each pixel in the rasterized image can be calculated by the following formula:

$$C = \sum_{i \in \{1..N\}} \hat{\boldsymbol{c}}_i \sigma_i \prod_{i-1}^{j=1} \left(1 - \sigma_j\right), \tag{1}$$

$$\sigma_i = \alpha_i e^{-\frac{1}{2}(\Delta \boldsymbol{x})^T \Sigma_i^{-1} (\Delta \boldsymbol{x})}, \tag{2}$$

$$\Sigma_i = J W R_i S_i S_i^T R_i^T W^T J^T, \tag{3}$$

where $\hat{\boldsymbol{c}}_i$ is the color obtained from projected 2D Gaussian, j is the index of the Gaussian points in front of i according to their distances to the optical center in ascending order, N is the number of Gaussians, $\Delta \boldsymbol{x}$ is the position offset to the center, Σ_i is the projected covariance to image space made by a viewing transformation denoted by W and the Jacobian J of the affine approximation of the projective transformation.

3DGS editing with diffusion models. Diffusion models are proposed to generate images with text input from users. Recent research [20, 44, 53] has seen the potential of applying 2D diffusion processes to the 3D realm, as the diffusion models are trained with massive data and thus become 3D-aware. There are two main ways to leverage prior in 2D diffusion models to edit 3DGS.

The first is to construct a collection of multiview edited images $\hat{I}$ generated by diffusion models based on the original renderings I. Then, these images and their corresponding rendering camera parameters are used to optimize the 3D model Θ.

The second is to utilize score distillation sampling (SDS) [44] to optimize a radiance field by distilling the priors from a Text-to-Image (T2I) diffusion model for 3D generation. The pre-trained diffusion model Φ is used to predict the added noise given a noised image $\hat{I}_t$ and its text condition y.

$$\nabla_\Theta L_{SDS}(\Phi, \hat{I} = f(\Theta)) = \mathbb{E}_{\epsilon,t}\left[w(t)\left(\epsilon_\Phi\left(\hat{I}_t; y, t\right) - \epsilon\right)\frac{\partial \hat{I}}{\partial \Theta}\right], \quad (4)$$

where ϵ is the added noise, $f(\cdot)$ is the differentiable image formation process, and $w(t)$ is a predefined weighting function derived at timestep t.

3.2 Design Considerations

To achieve flexible and controllable editing with 3DGS, it is crucial to address two primary interaction challenges regarding the representation and optimization of 3DGS. First, users must be able to efficiently identify the specific areas they wish to edit. Second, they should have effective control over the AI-driven editing process.

Efficient 3D target area specification. 3DGS presents unique challenges for target area selection due to its point cloud-like representation. As illustrated in Figure 2, traditional 2D interfaces rely on raycasting, projecting cylindrical selections from screen space into the 3D scene, which requires users to repeatedly adjust viewpoints and refine selections, resulting in a tedious and time-consuming process. AI-based segmentation models [7, 10, 65], which project 2D masks onto 3D areas, offer an automated solution but face significant limitations. These models are either restricted to object-level segmentation, lacking the granularity required for fine-grained edits, or fail to handle intricate geometric structures, producing unreliable and inconsistent results. This lack of fine-grained control limits users' ability to edit with precision and flexibility, hindering their capacity to realize diverse creative ideas.

Controllable AI-driven editing. Most current methods rely on text or image prompts to control the editing of 3D Gaussian Splatting. However, such inputs fail to convey the spatial information necessary for precise control. Unlike 2D editing, which benefits from a variety of constraints such as sketches [70] or semantic maps [42], 3D editing lacks similar prompt types that can incorporate 3D priors into neural networks to control the generation or editing process to get desired results. For instance, in 2D, a user can sketch the outline of a hat to generate the desired shape, while there is no equivalent prompt in the 3D domain to guide AI-driven editing due to the lack of a spatial user interface.

4 The GaussianShopVR System

Our system requires two components: a PC-powered VR device as the frontend and a GPU server as the backend. The frontend PC does not require significant computing power, as it is primarily used for basic manipulation and rendering. The backend server needs to handle complex editing functions. In our implementation, the frontend uses a Meta Quest 3 headset and a PC with an NVIDIA 4090 graphics card, and the backend uses a server equipped with an NVIDIA 4090 graphics card and an Intel Xeon Platinum 8370C CPU. We use Unity [54] to create the VR environment and user interfaces. The VR interface for rendering Gaussian Splatting is built upon an open-source Unity project [12], which serves as the foundation for our implementation. All operations in VR are synchronized with the server, ensuring that the scene remains editable by AI models. Rendering speed is mainly influenced by the number of 3DGS points within the view frustum. At a resolution of 2064 × 2208 per eye, our system runs at approximately 70 FPS with 0.8 million points and 55 FPS with 2.2 million points.

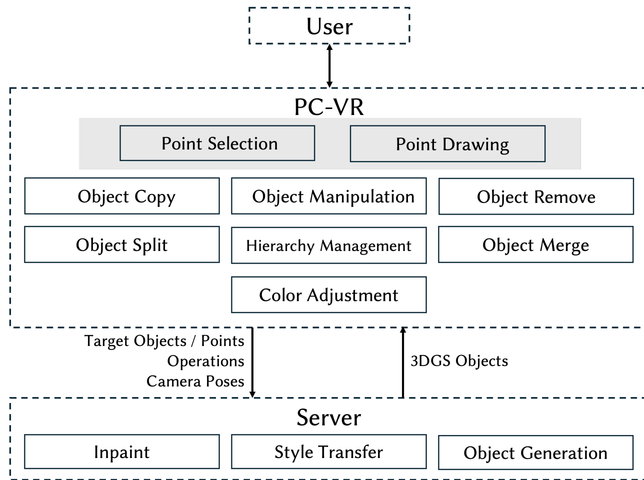

Figure 3: Illustration of our system architecture. User interactions in VR are transmitted to the backend server, where they are synchronized and processed for AI-driven editing.

4.1 Efficient 3D Target Area Specification

Using VR controllers in GaussianShopVR, users can easily select 3DGS points and draw point clouds, which are challenging tasks in conventional 2D interfaces.

Point cloud selection. GaussianShopVR facilitates efficient selection of 3DGS points through an intuitive, VR-based interaction method utilizing a semi-transparent spherical cursor attached to the VR controller. Gaussian splat centers are distinctly visualized as discrete points, and points encompassed by the sphere are visually highlighted to indicate their selection. To support precise selection of areas of any size, we implemented manual radius adjustment for the spherical cursor. This allows users to interactively change the selection granularity, providing intuitive control tailored to their editing needs. Additionally, users can smoothly extend or refine selections by moving the VR controller through the 3D space, enabling efficient targeting of specific regions.

Point cloud drawing. Creating point clouds, a task traditionally cumbersome and unintuitive with 2D interfaces, becomes significantly more natural and immersive within a 3D virtual environment using VR controllers. In GaussianShopVR, users intuitively create point clouds by directly drawing in three-dimensional space using a spherical brush attached to the VR controller. This spherical brush can be dynamically scaled, allowing users to adjust the brush size to precisely match the desired level of detail. Furthermore, GaussianShopVR provides functionality to assign custom colors to the drawn point clouds, enabling users to visually distinguish

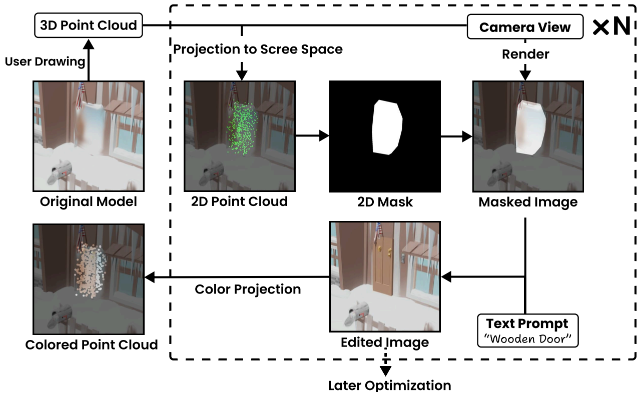

Figure 4: Illustration of obtaining 3D-consistent images. A 2D mask is generated corresponding to the target area and a specific viewpoint. Renderings from various viewpoints, along with their corresponding masks, are input into AI models to get edited images. Users can get a collection of multiview edited images as guidance for later 3DGS optimization.

and categorize different objects or regions intuitively during the authoring process.

4.2 Controllable AI-Driven Editing

Users interact with the VR interface to select or draw points that define the areas to be edited. The selected or drawn point clouds can serve as constraints that guide AI models during the editing process. GaussianShopVR can leverage specified areas to obtain 3D-consistent edited 2D renderings for later optimization, and can also leverage specified areas for regularization during optimization.

Specified points for 3D-consistent image editing. Existing methods [8, 23, 33, 39, 57] process training images in advance to guide 3DGS optimization, which can not be used for iterative editing. We design a method illustrated in Figure 4 to enhance the methods leveraging AI-edited 2D renderings to optimize 3DGS. Our method can generate 3D-consistent masks for 2D image editing rather than let users create segmentation masks manually. Users first use VR user interfaces to select or draw some points. Then, users should determine viewpoints to obtain 2D renderings. Under a given viewpoint, the selected points are projected into screen space to create a mask. Renderings from various viewpoints, along with their corresponding masks, are then input into AI models, such as MVInpainter [6] and Wan2.1-VACE [25], to generate edited images. Users can obtain guided results from multiple perspectives. Once users select their preferred images, the corresponding colors are projected back onto the specified points. The set of selected images can further be used to optimize the chosen 3DGS points.

Specified points as constraints for optimization. Users specify the editing area by drawing and selecting points, which are then copied into $\hat{P}$. These points form a coarse representation of the desired shape and can naturally become the initial constraint. However, during the optimization, 3DGS point clouds will be updated and densified due to guidance, which can result in the shape and appearance of 3DGS objects being far from what users expect. To ensure the optimized points adhere to the original shape and appearance, we introduce multiple regularization terms:

$$\begin{aligned} L_{\text{shape}} &= \frac{\lambda_{s1}}{|P|} \sum_{p_i \in P} \exp\Big(\lambda_{s2} \left\| \boldsymbol{x}_{p_i} - \boldsymbol{x}_{\hat{p}_i} \right\|\Big), \\ L_{\text{color}} &= \frac{\lambda_c}{|P|} \sum_{p_i \in P} \left\| \boldsymbol{c}_{p_i} - \boldsymbol{c}_{\hat{p}_i} \right\|, \\ L_{\text{scale}} &= \frac{\lambda_s}{|P|} \sum_{p_i \in P} \left\| \boldsymbol{s}_{p_i} - \boldsymbol{s}_{\hat{p}_i} \right\|, \end{aligned} \tag{5}$$

where λ_{s1} and λ_{s2} are hyperparameters that control the degree of shape limitation, λ_c for color control, λ_s for scaling control, P denotes the set of optimizing 3DGS points, and each $p_i \in P$ is associated with its corresponding initial point $\hat{p}_i \in \hat{P}$ from which it originated during densification. These regularization terms can be combined with other loss functions—such as the SDS loss for generative tasks or L1 loss for image-guided editing—to encourage the attributes of the optimized points to remain consistent with those of the initial user-specified points. The optimization diagram is shown in Algorithm 1.

ALGORITHM 1: Proposed 3DGS Optimization Pipeline

Input: Set of all 3DGS points P_{all}; selected points to optimize $P \subset P_{\text{all}}$; target images or text prompt; original loss weight w_{ori}; regularization hyperparameters λ_*
Copy P to $\hat{P}$ as user-specified reference;
for $t = 1$ **to** T **do**
  Render 2D images $\mathcal{I}$ from P_{all};
  if *optimizing with target images* **then**
    Compute guidance loss $L_{\text{ori}} = L_1 = \|\mathcal{I} - \mathcal{I}_{\text{target}}\|_1$;
  end
  else if *optimizing with SDS* **then**
    Compute guidance loss $L_{\text{ori}} = L_{\text{SDS}}$ using text prompt;
  end
  Compute regularization terms $L_{\text{shape}}, L_{\text{color}}, L_{\text{scale}}$;
  Compute total loss:
    $L_{\text{total}} = w_{\text{ori}} L_{\text{ori}} + L_{\text{shape}} + L_{\text{color}} + L_{\text{scale}}$;
  Update P by descending ∇L_{total};
  Densify and prune P as needed;
end
Output: Optimized 3DGS points P

4.3 Object Hierarchy

Object hierarchies in 3D software such as Blender and Unity [5, 54] facilitate the management of complex scenes by enabling grouped transformations, cleaner organization, and more intuitive control over related objects. We design the hierarchy system in GaussianShopVR as shown in Figure 5. After manually splitting, the new object becomes a sibling node of the split object, and users can manually adjust the hierarchy relationship using a 2D panel. Each object has its own transformation, including translation, rotation, and scaling. The 3DGS of the parent object can be obtained by: $\Theta_p = \cup\{T_{c1}(\Theta_{c1}), T_{c2}(\Theta_{c2}), ..., T_{cn}(\Theta_{cn})\}$, where $c1, c2, ...cn$ are the children of the parent object, and T is the transformation of each object. Each band of SH coefficients is rotated using matrices introduced by Ivanic and Ruedenberg [22]. Since the transformations are differentiable, the gradients calculated on the ancestors can be backpropagated to the target points of leaf objects.

With this architecture, GaussianShopVR supports more sophisticated 3D authoring tasks beyond simple object placement. Moreover, the hierarchical structure provides valuable contextual information for AI models, enabling users to define editing context for target objects and ensuring a seamless integration into their surrounding environment.

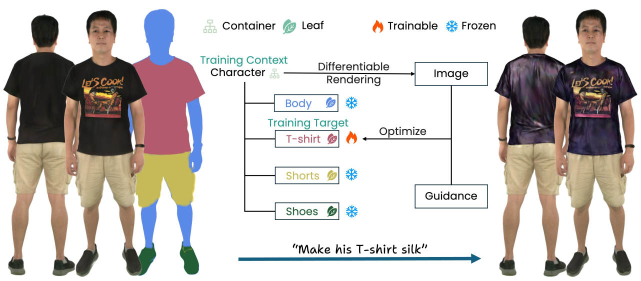

Figure 5: Illustration of the object hierarchy. Only Leaf objects contain real 3DGS points, while container objects contain the view of all children after transformation. Users can select a container object as the context for AI-driven editing.

4.4 Supported Editing Tasks

Our system offers a range of editing tools for 3DGS models, most of which are based on user-specified areas.

4.4.1 Object Splitting. While 3D Gaussian Splatting effectively reconstructs entire scenes from image collections, these reconstructions are typically tangled and inseparable. Users need to split the desired objects from the reconstructed scenes for further authoring. GaussianShopVR addresses this by enabling users to split individual objects after precise area selection. The split object will become the sibling of the existing one in the hierarchy system.

4.4.2 Object Manipulation. To support authoring tasks such as scene composition, users need functions for translating, rotating, and scaling objects. In GaussianShopVR, users can intuitively and immersively manipulate 3DGS objects. After selecting an object either by clicking on it in the hierarchy panel or grabbing it directly in the 3D space, they can translate, rotate, and scale it. To translate, users hold the grip button and move the controller to reposition the object. Rotation is performed by gripping the controllers and twisting their hands, while scaling is achieved by moving the controllers closer together or farther apart while holding the grip buttons, dynamically resizing the objects.

4.4.3 Color Adjustment. Users also need to adjust the appearance of 3DGS objects while preserving photorealism. The appearance of 3DGS points is obtained from the reconstruction modeling by SH coefficients. It is hard to edit manually due to the point-like representation and SH storage [18, 45]. Building on efficient 3DGS point selection, we developed a color adjustment interface using mapping curves for the RGB channels, similar to the curve tool in PhotoShop [1]. This interface enables users to modify the colors of selected points, supporting a wide range of creative applications, as shown in Figure 12.

Specifically, we extract the zero-order spherical harmonic coefficients from all selected 3DGS points and convert them into RGB values. For each RGB channel, we apply a mapping curve, which is initially set as the identity function, to transform the original values into the final results. These curves allow users to adjust the colors of 3DGS points while preserving fine details and texture.

4.4.4 Generation. Object generation enables users to efficiently and intuitively add new elements directly into existing scenes, streamlining workflow, facilitating rapid prototyping, and enhancing creative flexibility. In GaussianShopVR, users first draw point clouds to represent coarse shapes using VR controllers. Users can pick different colors to draw point clouds. After giving a text prompt, GaussianShopVR applies the SDS loss (Equation (4)) and the proposed regularization loss (Equation (5)) to optimize the drawn points. Figure 6 shows the results and the generation process.

Figure 6: Examples of processes of object generation starting from drawn point clouds.

4.4.5 Inpainting. Splitting 3DGS points from an object often results in a visible hole at the intersection, which substantially degrades visual quality. Users can address this problem either by manual editing or AI-driven editing in GaussianShopVR. Figure 7 shows the results of manual inpainting and AI-driven inpainting.

For manual inpainting (shown on the left of Figure 7), users can select a group of nearby points as a patch and copy it to desired locations. This approach is similar to editing with the clone stamp tool of Photoshop [1]. It is particularly effective for surfaces with regular texture patterns.

For AI-driven inpainting, we use the method described in Section 4.2. As shown on the right of Figure 7, users first draw some points to fill the hole and then use the method described in Section 4.2 to collect edited images from different viewpoints.

Figure 7: Examples of manual inpainting and AI-driven inpainting. Manual inpainting: a1) Original scene. b1) Scene with a hole after removing the desk and chairs. c1) Manual-split patch. d1) Scene after inpainting with copied split patches. AI-driven inpainting: a2) Scene with a hole. b2) Drawn point clouds as constraints. c2) 2D images after inpainting as guidance. d2) Scene after inpainting with optimization.

5 Study 1: Efficient 3D Area Selection

5.1 Task Design

Study 1 is designed to evaluate the efficiency of point selection using our VR user interface and the traditional 2D user interface. We use SuperSplat [43] as the 2D baseline, as it is a widely adopted browser-based editor for 3DGS models with a traditional 2D user interface. SuperSplat provides rectangle, brush, and sphere tools for point selection. In this study, participants use both interfaces to complete comparable point selection tasks, enabling a direct assessment of selection efficiency.

We recruited 18 participants and introduced the basic operations of GaussianShopVR and SuperSplat to them. Each participant was given eight minutes per tool to explore and become comfortable with its operations, with guidance provided as needed. Then, participants were informed that they needed to select a petal from a reconstructed flower scene in three minutes using two interfaces, as shown in Figure 8. The target petal was volumetric and closely surrounded by other petals, presenting a realistic 3D selection challenge. The user-selected points were recorded for later analysis.

Figure 8: Illustration of task setup of Study 1. The left shows the original reconstructed flower, and the right shows a pre-edited version with a highlighted petal. Participants were asked to select points on the highlighted petal of the pre-edited flower using GaussianShopVR and SuperSplat [43].

5.2 Results and Discussion

We use the Precision, F1 Score, and Intersection over Union (IoU) as metrics to evaluate point selection in Task 1, and the results are shown in Table 2. A paired t-test was employed to assess the differences in performance between GaussianShopVR and SuperSplat for each of the three metrics at a significance level of 0.05. The analysis of Table 2 investigates whether there are statistically significant differences between the two tools' performance across 18 participants.

During the study, we observed that even after a tutorial and 8-minute hands-on practice, some participants failed to complete the task with the 2D UI. Pa8, Pa10, Pa13, and Pa15 made mistake operations after selecting points, which resulted in the selection being cleared, and they felt very frustrated. Pa2, Pa4, Pa5, and Pa17 failed to clear the mis-selected points selected by raycasting and hidden in the background 3DGS points, even if they were informed of this situation in advance. We excluded these four and performed statistical analysis on the remaining 14 samples.

The paired t-test for precision revealed a significant difference between the two tools, with a t-statistic of 4.03 and a p-value of 0.0014. A similar significant difference was found for the F1 Score, with a t-statistic of 3.51 and a p-value of 0.0038. Likewise, the t-test for IoU produced a t-statistic of 3.49 and a p-value of 0.0040. Cohen's d was computed for the t-test results of each metric, yielding values of 1.60 for Precision, 1.29 for F1, and 1.29 for IoU. Since all these

Table 2: Precision, F1 Score, and IoU with different tools for point selection in Study 1.

		Pa1	Pa2	Pa3	Pa4	Pa5	Pa6	Pa7	Pa8	Pa9	Pa10	Pa11	Pa12	Pa13	Pa14	Pa15	Pa16	Pa17	Pa18	Avg.
GaussianShopVR	Precision	93.1%	94.4%	87.9%	69.1%	95.8%	88.2%	70.8%	81.0%	83.0%	81.9%	80.4%	80.8%	95.8%	81.3%	98.6%	86.5%	69.7%	85.9%	84.6%
	F1	90.4%	79.0%	86.5%	78.9%	86.8%	91.5%	79.8%	87.8%	82.2%	83.5%	82.0%	88.4%	91.3%	85.2%	88.5%	85.7%	74.4%	79.3%	84.5%
	IoU	82.6%	65.3%	76.2%	65.5%	76.7%	84.4%	66.4%	78.3%	69.8%	71.7%	69.5%	79.2%	84.0%	74.3%	79.4%	75.0%	59.2%	65.7%	73.5%
SuperSplat [43]	Precision	44.2%	20.3%	87.8%	31.3%	38.0%	44.8%	71.1%	N/A	60.1%	N/A	86.8%	51.5%	N/A	77.3%	N/A	74.2%	27.3%	78.5%	56.7%
	F1	59.4%	33.0%	90.1%	42.1%	52.6%	59.1%	82.4%	N/A	70.9%	N/A	89.7%	64.6%	N/A	83.4%	N/A	83.6%	42.7%	76.6%	66.4%
	IoU	42.3%	19.7%	82.0%	26.7%	35.6%	42.0%	70.0%	N/A	55.0%	N/A	81.4%	47.7%	N/A	71.6%	N/A	71.9%	27.1%	62.1%	52.5%

Table 3: Votes on the similarity of the generated model to the reference model in terms of shape, pose, and color, along with the overall generation quality. Some users selected "Unable to determine," and these votes were excluded from the counts.

Model	Shape		Poseture		Color		Quality	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
Superman	0	**18**	1	**19**	1	**13**	2	**18**
Pikachu	0	**20**	0	**20**	0	**19**	0	**20**
Spiderman	0	**20**	0	**20**	0	**16**	0	**19**
Rocking Chair	1	**11**	1	**10**	2	**15**	**11**	7
Scissor	0	**20**	0	**20**	0	**17**	1	**19**
Lily	0	**20**	0	**20**	0	**16**	1	**19**
Totoro	0	**20**	0	**20**	0	**18**	1	**19**

values exceed 0.8, they are considered large effects, indicating that the observed differences are not only statistically significant but also practically meaningful. The results show that the VR UI of GaussianShopVR is more efficient for users to select 3DGS points compared with its 2D counterpart and thus can support agile and precise editing area selection.

6 Study 2: Controllable AI Generation

6.1 Task Design

To evaluate whether spatial information provided by GaussianShopVR enhances controllable AI generation, we employed the method outlined in Section 4.4.4 to generate 3D models and assess their shape and posture through a user questionnaire. We designed and distributed a questionnaire with ten questions to evaluate participants' perception of the shape, posture, and quality of the generated models, as well as their attitude toward using point clouds as a new method for controlling AI-driven editing.

Specifically, we selected seven 3D presets from Sketchfab [51] and invited an artist to create corresponding point clouds for each preset in GaussianShopVR. Subsequently, we generated two versions of each model: one using only a text prompt and another combining the text prompt with the drawn point clouds. We adopted SDS-based text-to-3D generation from prior work [66] as the baseline. For controllable generation with spatial input, we used user-drawn point clouds as initialization rather than generations, and added regularization terms in Equation (5) during optimization to better preserve user-indicated structures and appearance.

6.2 Results and Discussion

We collected a total of 20 responses, with participants answering 10 questions for each of the 7 groups of models. As shown in Table 3, the majority of participants across all model groups found that generated results based on text and point cloud prompts were more

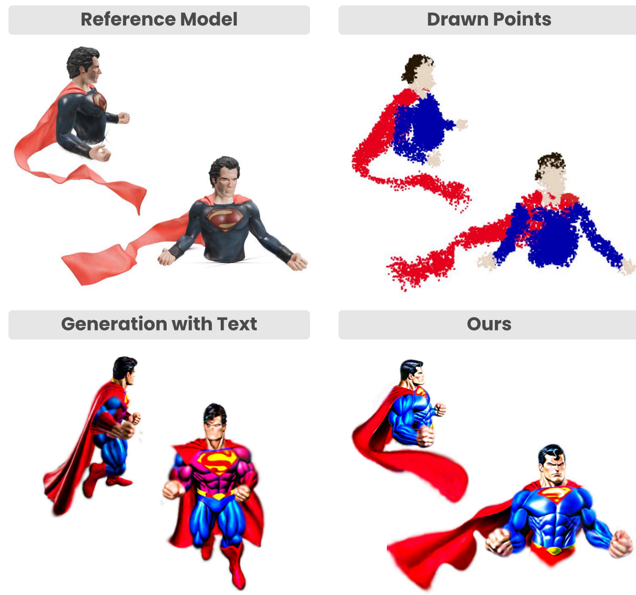

Figure 9: Illustration of a group of models compared in Study 2. Our method leverages text and point cloud prompts to generate objects. We posed ten questions to the participants in the aspects of shape, posture, color, and quality of generated results, as well as their attitude to input modalities.

similar to the reference models compared to generations based on text prompts alone. These findings suggest that GaussianShopVR facilitates more controllable AI generation, thereby enhancing the 3D authoring process.

Additionally, we asked participants about their perception of text and point cloud prompts. On average, 15.7 participants (σ=2.37) agreed that text prompts accurately described the reference model. However, only 6.57 participants (σ=2.19) believed that text prompts could generate models with the same shape and posture as the reference. In contrast, 18.1 participants (σ=1.12) felt that point cloud prompts accurately represented the reference model, and 19.7 participants (σ=1.12) believed that point cloud prompts could successfully generate models with the same shape and posture as the reference.

The results from Study 2 indicate that GaussianShopVR offers more controllable AI generation when using drawn point clouds. Participants also considered point cloud prompts to be a highly effective tool for generating controllable 3D models.

Figure 10: Examples of some 3DGS objects reconstructed and split by participants in Study 3. These objects are then used for their further authoring.

7 Study 3: Open-Ended 3D Authoring

We designed an open-ended 3D authoring task to explore how GaussianShopVR can facilitate 3D authoring. GaussianShopVR enables point-level editing of 3DGS and provides users with an immersive environment. This study aims to investigate how point-editable 3DGS, as a novel creative medium, can enhance 3D authoring, and how immersion contributes to the authoring experience.

7.1 Task Design

We designed Study 3 as an open-ended task in which participants were asked to do 3D authoring using the 3D object reconstructed at the beginning of their session, along with some preset scenes, objects, and humans in Gaussian Splatting. These presets are generated from datasets in 3DGS [27], Mip-NeRF 360 [3], and DEGAS [48]. Users were introduced to the operations of GaussianShopVR by splitting their desired object from the scene reconstructed at the beginning of the user study. Then, they used the split objects along with a preset scene to complete an open-ended 3D authoring task. Finally, we asked each participant to complete a questionnaire and conducted a short semi-structured interview to collect their feedback on the user experience. Study 3 does not include AI-driven editing features as they have been evaluated in Study 2, and current AI models do not yet support real-time editing.

We recruited ten users to participate in our study. Four were male, and six were female. Their average age was 25.2. Half of them had experience with 3D software or relevant design backgrounds, while the remaining participants with no 3D authoring experience were regarded as novice users. On a scale of 1–7, the average self-reported familiarity of the expert group was 5.40 (σ=0.54), while that of the novice group was 2.40 (σ=1.67).

7.2 Results and Discussion

We designed a questionnaire that adapted some questions from the Questionnaire for User Interface Satisfaction [9] and the System Usability Scale [2, 26] questionnaire. The questions and results are shown in Figure 11. In *User Experience Ratings* (Qa), we asked participants to rate how much they agreed with five statements on a scale of 1–7, with 1 being "strongly disagree" and 7 "strongly agree." In *System Features Ratings* (Qb), we asked participants to rate the usefulness of each feature in helping them complete the content creation on a scale of 1–7, with 1 being "not useful at all" and 7 "extremely useful."

Overall, the participants provided positive ratings on their experience using GaussianShopVR (10/10 on Qa1). They found it interesting to do 3D authoring with 3DGS objects in VR.

3DGS as a new creative medium. The main premise of our work is that 3DGS can be a suitable representation for 3D authoring. Previous work like PointShopAR [59] proves that simple point clouds are suitable for environmental prototyping. Compared with the vanilla point clouds, 3DGS can present precise shapes and attractive appearances of reconstructed objects. We believe 3DGS can thus facilitate more generalized creation tasks. Compared with other 3D software or VR authoring tools, participants agreed that GaussianShopVR provides a new way for 3D authoring with an average rating of 6.2 (σ=0.87). Participants provided high ratings on basic VR operations with averages of 6.0 (σ=0.77) on Qb1, 6.5 (σ=0.67) on Qb3, 6.6 (σ=0.49) on Qb4, and 6.2 (σ=0.6) on Qb5.

P8 noted, "*Meshes can also achieve good results, but the operations are not intuitive. You have to practice a lot and spend a lot of time.*" In contrast, 3DGS reconstruction enables quick acquisition of 3D assets and received an average rating of 6.6 from participants. However, content creators are not only interested in capturing real-world assets. They care more about transforming these assets to express their ideas. P8, a new media artist, remarked, "*It is great to obtain 3D assets like taking 3D photos, but I care more about how to transform them into my artwork. Your project seems to provide a way for this novel representation.*" P4 found editing 3DGS points in VR highly intuitive, and P8 believed such operations can support personalized artistic expression. Overall, user feedback suggests that GaussianShopVR facilitates 3DGS as a new creative medium by offering smooth and intuitive editing capabilities.

Immersive 3D authoring. Realistic 3DGS objects combined with intuitive VR interactions significantly enhanced participants' immersive experiences during 3D authoring with GaussianShopVR. All participants explicitly agreed that the immersive spatial context and realistic visual feedback facilitated clear and effective expression of their creative ideas (Qa5). For example, P8 highlighted the system's unique advantage, remarking, "The system enabled immersive prototyping inspired by nature and can help me prototype in situ," emphasizing how immersion directly fosters creative inspiration and situational relevance in the authoring process.

Participants further underscored that the immersive quality of GaussianShopVR notably improved their authoring efficiency. Qualitative feedback consistently highlighted the importance of the user perspective for seamless 3D interaction. Specifically, P4 and P7 explained the limitations of traditional 2D interfaces, which typically adopt an object-centered perspective. They described how this approach frequently results in disorientation, as objects remain static while the user's viewpoint continuously shifts. This constant need for manual reorientation disrupts workflow and hampers creativity. In contrast, participants praised the user-centered perspective of the VR interface in GaussianShopVR. This immersive design naturally adapts to users' movements and actions, maintaining a coherent spatial orientation. As P2 articulated, "By minimizing the need for constant view recalibration and offering an intuitive navigation experience, it supports me in staying focused and engaged." Overall, qualitative feedback strongly supports that the immersive, user-centric environment provided by GaussianShopVR enhances both productivity and creative satisfaction.

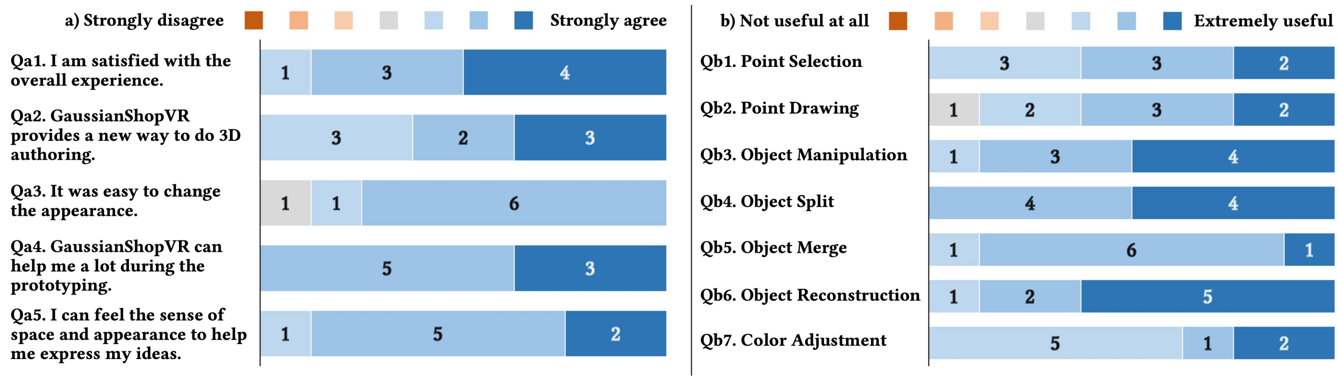

Figure 11: Ratings of a) the overall user experience and b) each feature of GaussianShopVR. The color of a bar represents how much participants agreed with a statement or how useful they found a feature. The number in the bar represents the number of participants who submitted the same rating.

Figure 12: Examples of 3DGS content created using GaussianShopVR, showcasing its capability to enhance creative freedom and editing precision to facilitate 3D authoring.

8 Conclusion

In conclusion, this paper presents GaussianShopVR, an immersive 3D authoring VR system, leveraging the advantages of the 3DGS representation in delivering immersive environments through photorealistic, high-FPS rendering and seamless integration with AI for editing. GaussianShopVR leverages efficient 3D target area identification and spatial control to further facilitate the authoring process, which is evaluated via Study 1 (N=18) and Study 2 (N=20). Study 3 (N=10) demonstrates that GaussianShopVR supports diverse design and creation needs with a suite of enhanced editing tools, providing an intuitive and immersive environment for 3D authoring.

Our system demonstrates the potential of using 3DGS as a new medium for immersive 3D content creation. The photorealism and rendering speed of 3DGS can facilitate immersive 3D authoring in VR. However, there are some limitations and many areas for future work to explore the potential of this new medium:

Relighting. GaussianShopVR can quickly import static photorealistic objects into the 3D editing scene but may encounter inconsistent environmental lighting. Although GaussianShopVR supports RGB curve adjustments for target areas while preserving texture details, the workload is much greater than that of photo editing. Novice users P1 and P2 said they did not know how to use the RGB curves to adjust lighting until they had used this function many times, while expert user P6 complained, "*Adjusting the lighting for the whole environment is hard to achieve.*" There are already some works that enable relighting for 3DGS, such as GS^3 [4] and GS-ID [17]. GS-ID can decompose illumination from reconstructed scenes and relight them. We believe it is possible to integrate such a lighting module into our system.

Resolution. The 3DGS objects are reconstructed from a sequence of photos, so the resolution of the objects is limited to that of the photos. In Study 3, P5 wanted to scale up the reconstructed teddy bear to be a huge monster in a garden scene, but the rendering effect turned out to be very blurry. Although the teddy bear was reconstructed from 4K photos, it only occupied a portion of the images, and the garden scene is much larger. Another observation is that some 3DGS points belonging to different objects are tangled and overlapped, especially with small objects, and it is hard to separate them. This is similar to the blending of pixels in images, where sometimes a pixel is hard to identify as belonging to which object. We found that if we want to reuse reconstructed objects in any scene, we need to solve the problem of the limited resolutions of 3DGS objects, requiring advanced super-resolution techniques.

Acknowledgments

We would like to thank Qianxi Liu for her contributions to the visual refinement of the figures and Linjie Qiu for his assistance with subsequent UI improvements. We also appreciate the reviewers for their valuable comments and helpful suggestions. This work was partially supported by Guangzhou Basic Research Scheme #2024A04J4229 and Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things #2023B1212010007.

References

[1] Adobe. 2024. Photoshop. https://www.adobe.com/products/photoshop.html. Accessed May 25, 2024.

[2] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction* 24, 6 (2008), 574–594.

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *CVPR* (2022).

[4] Zoubin Bi, Yixin Zeng, Chong Zeng, Fan Pei, Xiang Feng, Kun Zhou, and Hongzhi Wu. 2024. GS^3: Efficient Relighting with Triple Gaussian Splatting. In *SIGGRAPH Asia 2024 Conference Papers*.

[5] Blender Foundation. 2024. Blender: A 3D Modeling and Rendering Package. http://www.blender.org. Accessed May 25, 2024.

[6] Chenjie Cao, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. 2025. Mvinpainter: Learning Multi-View Consistent Inpainting to Bridge 2D and 3D Editing. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*.

[7] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. 2023. Segment Any 3D Gaussians. *arXiv preprint arXiv:2312.00860* (2023).

[8] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. 2024. GaussianEditor: Swift and Controllable 3D Editing with Gaussian Splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 21476–21485.

[9] John P Chin, Virginia A Diehl, and Kent L Norman. 1988. Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 213–218.

[10] Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. 2024. Click-Gaussian: Interactive Segmentation to Any 3D Gaussians. *arXiv preprint arXiv:2407.11793* (2024).

[11] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. 2023. LucidDreamer: Domain-free Generation of 3D Gaussian Splatting Scenes. *arXiv preprint arXiv:2311.13384* (2023).

[12] CLARTE-LAB. 2024. GaussianSplattingVRViewerUnity. https://github.com/clarte53/GaussianSplattingVRViewerUnity. Accessed September 15, 2024.

[13] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion Models Beat Gans on Image Synthesis. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*. Article 672, 15 pages.

[14] Wenqi Dong, Bangbang Yang, Lin Ma, Xiao Liu, Liyuan Cui, Hujun Bao, Yuewen Ma, and Zhaopeng Cui. 2024. Coin3D: Controllable and Interactive 3D Assets Generation with Proxy-Guided Conditioning. In *ACM SIGGRAPH 2024 Conference Papers*. 1–10.

[15] Achref Doula, Tobias Güdelhöfer, Andrii Matviienko, Max Mühlhäuser, and Alejandro Sanchez Guinea. 2022. Immersive-Labeler: Immersive Annotation of Large-Scale 3D Point Clouds in Virtual Reality. In *ACM SIGGRAPH 2022 Posters*. Article 27, 2 pages.

[16] Tobias Drey, Jan Gugenheimer, Julian Karlbauer, Maximilian Milo, and Enrico Rukzio. 2020. VRSketchIn: Exploring the Design Space of Pen and Tablet Interaction for 3D Sketching in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[17] Kang Du, Zhihao Liang, and Zeyu Wang. 2024. GS-ID: Illumination Decomposition on Gaussian Splatting via Diffusion Prior and Parametric Light Source Optimization. *arXiv preprint arXiv:2408.08524* (2024).

[18] Xinyu Gao, Ziyi Yang, Bingchen Gong, Xiaoguang Han, Sipeng Yang, and Xiaogang Jin. 2024. Towards Realistic Example-based Modeling via 3D Gaussian Stitching. *arXiv preprint arXiv:2408.15708* (2024).

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Article 574, 12 pages.

[20] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2024. LRM: Large Reconstruction Model for Single Image to 3D. In *The Twelfth International Conference on Learning Representations*.

[21] Jiajun Huang, Shuolin Xu, Hongchuan Yu, and Tong-Yee Lee. 2024. GSDeformer: Direct, Real-time and Extensible Cage-based Deformation for 3D Gaussian Splatting. *arXiv preprint arXiv:2405.15491* (2024).

[22] Joseph Ivanic and Klaus Ruedenberg. 1996. Rotation matrices for real spherical harmonics. Direct determination by recursion. *The Journal of Physical Chemistry* 100, 15 (1996), 6342–6347.

[23] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. 2024. VR-GS: A Physical Dynamics-Aware Interactive Gaussian Splatting System in Virtual Reality. In *ACM SIGGRAPH 2024 Conference Papers*. Article 78, 12 pages.

[24] Ying Jiang, Congyi Zhang, Hongbo Fu, Alberto Cannavò, Fabrizio Lamberti, Henry Y K Lau, and Wenping Wang. 2021. HandPainter - 3D Sketching in VR with Hand-based Physical Proxy. Article 412, 13 pages.

[25] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. 2025. Vace: All-In-One Video Creation and Editing. *arXiv preprint arXiv:2503.07598* (2025).

[26] Patrick W Jordan, Bruce Thomas, Ian Lyall McClelland, and Bernard Weerdmeester. 1996. *Usability Evaluation in Industry*. CRC Press.

[27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4, Article 139 (July 2023), 14 pages.

[28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment Anything. *arXiv preprint arXiv:2304.02643* (2023).

[29] Yushi LAN, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. 2025. GaussianAnything: Interactive Point Cloud Flow Matching for 3D Generation. In *The Thirteenth International Conference on Learning Representations*.

[30] Boyu Li, Linping Yuan, Zhe Yan, Qianxi Liu, Yulin Shen, and Zeyu Wang. 2024. AniCraft: Crafting Everyday Objects as Physical Proxies for Prototyping 3D Character Animation in Mixed Reality. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 14 pages.

[31] Carina Liebers, Marvin Prochazka, Niklas Pfützenreuter, Jonathan Liebers, Jonas Auda, Uwe Gruenefeld, and Stefan Schneegass. 2023. Pointing It Out! Comparing Manual Segmentation of 3D Point Clouds between Desktop, Tablet, and Virtual Reality. *International Journal of Human–Computer Interaction* (2023), 1–15.

[32] Feng-Lin Liu, Hongbo Fu, Yu-Kun Lai, and Lin Gao. 2024. SketchDream: Sketch-based Text-To-3D Generation and Editing. *ACM Transactions on Graphics* 43, 4 (July 2024).

[33] Zhiheng Liu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jie Xiao, Kai Zhu, Nan Xue, Yu Liu, Yujun Shen, and Yang Cao. 2024. InFusion: Inpainting 3D Gaussians via Learning Depth Completion from Diffusion Prior. *arXiv preprint arXiv:2404.11613* (2024).

[34] Zhihao Liu, Fanxing Zhang, and Zhanglin Cheng. 2021. BuildingSketch: Freehand Mid-Air Sketching for Building Modeling. In *2021 IEEE International Symposium on Mixed and Augmented Reality*. 329–338.

[35] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. 2023. Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis. *arXiv preprint arXiv:2308.09713* (2023).

[36] Zhaofeng Luo, Zhitong Cui, Shijian Luo, Mengyu Chu, and Minchen Li. 2024. VR-Doh: Hands-on 3D Modeling in Virtual Reality. *arXiv preprint arXiv:2412.00814* (2024).

[37] Hengyu Meng, Duotun Wang, Zhijing Shao, Ligang Liu, and Zeyu Wang. 2025. Text2VDM: Text to Vector Displacement Maps for Expressive and Interactive 3D Sculpting. *arXiv preprint arXiv:2502.20045* (2025).

[38] 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 *European Conference on Computer Vision*. 17 pages.

[39] Ashkan Mirzaei, Riccardo De Lutio, Seung Wook Kim, David Acuna, Jonathan Kelly, Sanja Fidler, Igor Gilitschenski, and Zan Gojcic. 2024. RefFusion: Reference Adapted Diffusion Models for 3D Scene Inpainting. *arXiv preprint arXiv:2404.10765* (2024).

[40] Nels Numan, Shwetha Rajaram, Balasaravanan Thoravi Kumaravel, Nicolai Marquardt, and Andrew D Wilson. 2024. SpaceBlender: Creating Context-Rich Collaborative Spaces Through Generative 3D Scene Blending. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*.

[41] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[42] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic Image Synthesis with Spatially-Adaptive Normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2337–2346.

[43] PlayCanvas. 2024. SuperSplat v1.2.3. https://playcanvas.com/supersplat/editor. Accessed September 15, 2024.

[44] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh International Conference on Learning Representations*.

[45] Chenqu Ren, Haolei Qiu, Yeheng Shao, Zherui Qiu, and Kaiwen Song. 2024. PaletteGaussian: 3D Photorealistic Color Editing with Gaussian Splatting. In *2024 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 1206–1215.

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.

[47] Artem Sevastopolsky, Philip-William Grassal, Simon Giebenhain, ShahRukh Athar, Luisa Verdoliva, and Matthias Niessner. 2023. HeadCraft: Modeling High-Detail Shape Variations for Animated 3DMMs. *arXiv preprint arXiv:2312.14140* (2023).

[48] Zhijing Shao, Duotun Wang, Qing-Yao Tian, Yao-Dong Yang, Hengyu Meng, Zeyu Cai, Bo Dong, Yu Zhang, Kang Zhang, and Zeyu Wang. 2024. DEGAS: Detailed Expressions on Full-Body Gaussian Avatars. *arXiv preprint arXiv:2408.10588* (2024).

[49] Yulin Shen, Yifei Shen, Jiawen Cheng, Chutian Jiang, Mingming Fan, and Zeyu Wang. 2024. Neural Canvas: Supporting Scenic Design Prototyping by Integrating 3D Sketching and Generative AI. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. Article 1056, 18 pages.

[50] Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. 2024. RealmDreamer: Text-Driven 3D Scene Generation with Inpainting and Depth Diffusion. *arXiv preprint arXiv:2404.07199* (2024).

[51] Sketchfab, Inc. 2025. Sketchfab - The best 3D viewer on the web. https://sketchfab.com/. Accessed April 1, 2025.

[52] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. 2024. LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation. *arXiv preprint arXiv:2402.05054* (2024).

[53] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. 2023. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. *arXiv preprint arXiv:2309.16653* (2023).

[54] Unity Technologies. 2024. Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine. https://unity.com/. Accessed September 18, 2024.

[55] Cyrus Vachha, Yixiao Kang, Zach Dive, Ashwat Chidambaram, Anik Gupta, Eunice Jun, and Björn Hartmann. 2025. Dreamcrafter: Immersive Editing of 3D Radiance Fields Through Flexible, Generative Inputs and Outputs. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. Article 3, 13 pages.

[56] Duotun Wang, Hengyu Meng, Zeyu Cai, Zhijing Shao, Qianxi Liu, Lin Wang, Mingming Fan, Xiaohang Zhan, and Zeyu Wang. 2024. HeadEvolver: Text to Head Avatars via Locally Learnable Mesh Deformation. *arXiv preprint arXiv:2403.09326* (2024).

[57] Yuxin Wang, Qianyi Wu, Guofeng Zhang, and Dan Xu. 2024. Learning 3D Geometry and Feature Consistent Gaussian Splatting for Object Removal. In *European Conference on Computer Vision*. Springer, 1–17.

[58] Zeyu Wang, Cuong Nguyen, Paul Asente, and Julie Dorsey. 2021. DistanciAR: Authoring Site-Specific Augmented Reality Experiences for Remote Environments. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Article 411, 12 pages.

[59] Zeyu Wang, Cuong Nguyen, Paul Asente, and Julie Dorsey. 2023. PointShopAR: Supporting Environmental Design Prototyping Using Point Cloud in Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 15 pages.

[60] Florian Wirth, Jannik Quehl, Jeffrey Ota, and Christoph Stiller. 2019. PointAtMe: Efficient 3D Point Cloud Labeling in Virtual Reality. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. 1693–1698.

[61] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Wang Xinggang. 2023. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. *arXiv preprint arXiv:2310.08528* (2023).

[62] Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. 2024. Recent Advances in 3D Gaussian Splatting. *Computational Visual Media* 10, 4 (Aug. 2024), 613–642. Conference Name: Computational Visual Media.

[63] Bangbang Yang, Wenqi Dong, Lin Ma, Wenbo Hu, Xiao Liu, Zhaopeng Cui, and Yuewen Ma. 2024. Dreamspace: Dreaming Your Room Space with Text-Driven Panoramic Texture Propagation. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces*. IEEE, 650–660.

[64] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. 2023. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction. *arXiv preprint arXiv:2309.13101* (2023).

[65] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. 2024. Gaussian Grouping: Segment and Edit Anything in 3D Scenes. In *European Conference on Computer Vision*.

[66] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. 2024. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6796–6807.

[67] Emilie Yu, Rahul Arora, J Andreas Baerentzen, Karan Singh, and Adrien Bousseau. 2022. Piecewise-Smooth Surface Fitting onto Unstructured 3D Sketches. *ACM Transactions on Graphics* 41, 4 (2022), 1–16.

[68] Emilie Yu, Rahul Arora, Tibor Stanko, J Andreas Bærentzen, Karan Singh, and Adrien Bousseau. 2021. Cassie: Curve and surface sketching in immersive environments. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[69] Ya-Ting Yue, Yong-Liang Yang, Gang Ren, and Wenping Wang. 2017. Scenectrl: Mixed Reality Enhancement via Efficient Scene Editing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 427–436.

[70] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.

[71] Lixiang Zhao, Tobias Isenberg, Fuqi Xie, Hai-Ning Liang, and Lingyun Yu. 2024. MeTACAST: Target- and Context-Aware Spatial Selection in VR. *IEEE Transactions on Visualization and Computer Graphics* 30, 1 (Jan. 2024), 480–494.

[72] Jingyu Zhuang, Di Kang, Yan-Pei Cao, Guanbin Li, Liang Lin, and Ying Shan. 2024. TIP-Editor: An Accurate 3D Editor Following Both Text-Prompts And Image-Prompts. *ACM Transactions on Graphics* 43, 4, Article 121 (July 2024), 12 pages.

[73] Matthias Zwicker, Mark Pauly, Oliver Knoll, and Markus Gross. 2002. Pointshop 3D: An Interactive System for Point-Based Surface Editing. *ACM Transactions on Graphics* 21, 3 (July 2002), 322–329.