Learning Naturally Aggregated Appearance for Efficient 3D Editing

Ka Leong Cheng[1,2], Qiuyu Wang[2], Zifan Shi[1,2], Kecheng Zheng[2], Yinghao Xu[2,3], Hao Ouyang[2], Qifeng Chen[1†], Yujun Shen[2†]

[1]HKUST [2]Ant Group [3]Stanford

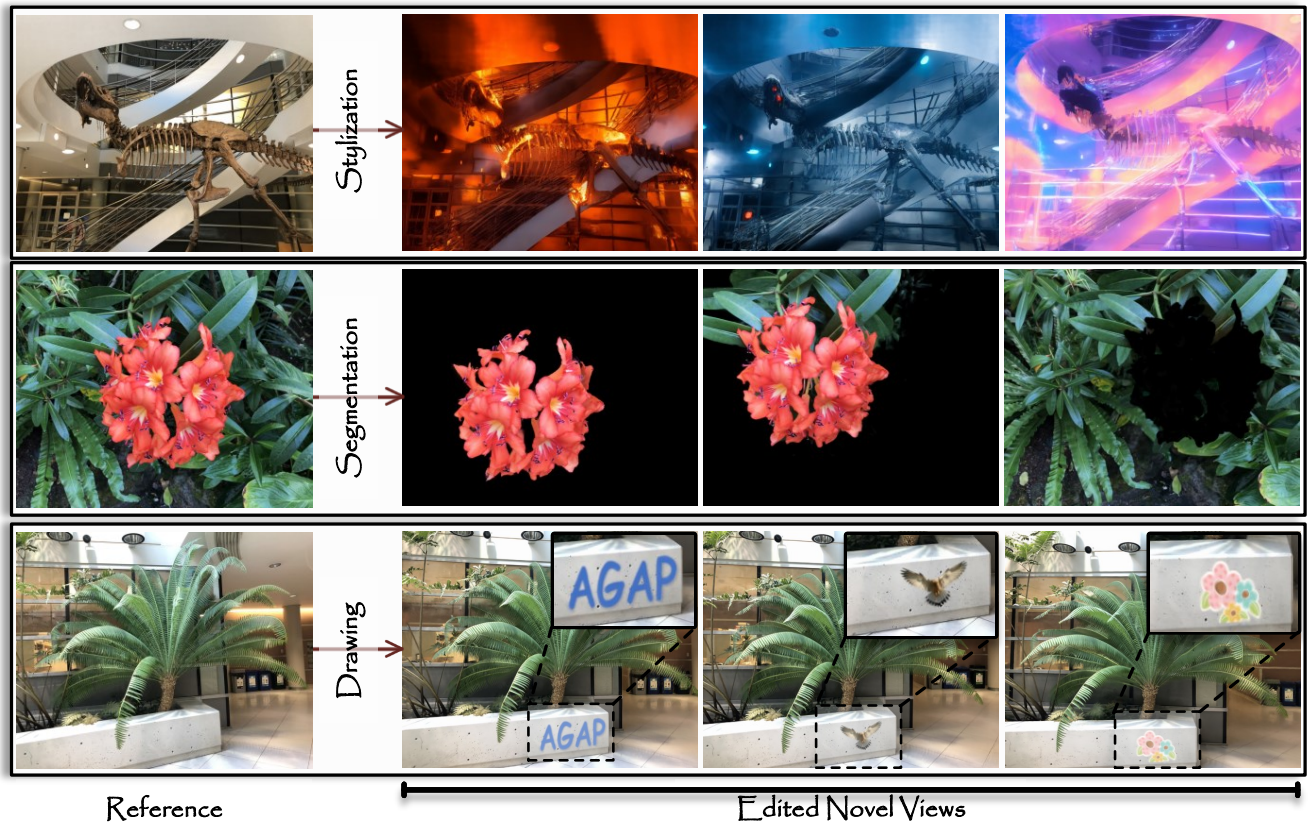


Figure 1. AGAP aggregates 3D appearance as natural 2D canonical images. With image processing tools, AGAP enables various ways of 3D editing without re-optimization, including (a) scene stylization, (b) instance segmentation, and (c) interactive drawing.

Abstract

Neural radiance fields, which represent a 3D scene as a color field and a density field, have demonstrated great progress in novel view synthesis yet are unfavorable for editing due to the implicitness. This work studies the task of efficient 3D editing, where we focus on ***editing speed*** *and* ***user interactivity****. To this end, we propose to learn the color field as an explicit 2D appearance aggregation, also called canonical image, with which users can easily customize their 3D editing via 2D image processing. We complement the canonical image with a projection field that maps 3D points onto 2D pixels for texture query. This field is initialized with a pseudo canonical camera model and optimized with offset regularity to ensure the* ***naturalness*** *of the canonical image. Extensive experiments on different datasets suggest that our representation, dubbed AGAP, well supports various ways of 3D editing (e.g., stylization, instance segmentation, and interactive drawing). Our approach demonstrates remarkable efficiency by being at least 20× faster per edit compared to existing NeRF-based editing methods. Project page is available at https://felixcheng97.github.io/AGAP/.*

Table 1. AGAP supports various editing cases. Our editing time per edit is significantly shorter thanks to the optimization-free editing pipeline. Note that the times listed below do not include the comparable pre-training and rendering times across methods.

	Global stylization	Local stylization	Segmentation	Drawing	Editing time‡
ARF [83]	✓	✗	✗	✗	371s
Ref-NPR [86]	✓	✗	✗	✗	514s
DFFs [35]	✗	✓	✓	✗	516s
IN2N [18]	✓	—*	✗	✗	∼10000s
GaussianEditor [10]	✓	✓	—†	✗	1320s
Ours	✓	✓	✓	✓	20s

1. Introduction

While recent advancements in 3D representations like neural radiance fields (NeRF) [46] have shown impressive reconstruction capabilities for real-world scenes, the need for further progress in 3D editing arises as the desire to recreate and manipulate these scenes. The field of 3D editing has witnessed significant development in recent years. Traditional 3D modeling approaches [30, 61, 62, 76] typically rely on reconstructing scenes using meshes. By combining meshes with texture maps, we can enable appearance editing during the rendering process. However, these methods usually face difficulties in obtaining detailed and regular texture maps, typically in complex scenes, hindering effective editing and user-friendliness.

Recent neural radiance fields offer high-quality scene reconstructions, but manipulating the implicit 3D representation embedded within neural networks is inherently non-straightforward. Existing NeRF-based editing approaches can be mainly divided into two categories: some methods like [9, 11, 70, 78, 82, 85] target geometry editing, usually taking advantage of meshes, while the other [18, 35, 52, 83, 86] focuses on 3D stylization using images or texts as guidance. However, re-optimizing the original NeRF models is necessary to incorporate the desired editing effects into the underlying 3D representation, resulting in time-consuming processes. Consequently, it is crucial to develop a user-friendly framework that can efficiently and effectively support various edits within a single model.

This paper introduces a novel editing-friendly representation **AGAP** with naturally **Ag**gregated **Ap**pearance for efficient 3D editing, consisting of a 3D density grid for geometry estimation and a canonical image plus a projection field for appearance modeling. Our method attempts to link the 3D representation with a natural 2D canonical representation. Concretely, a learnable canonical image is designed as the interface for editing, which aggregates the appearance by projecting the 3D radiance to a natural-looking image by the associated projection field. To ensure the naturalness of the aggregated canonical image with strong representation capacity, the projection field is carefully initialized using a pseudo canonical camera model and complemented by a learned view-dependent offset. The underlying 3D structure is modeled by an explicit 3D density grid.

AGAP supports various ways edits of a 3D scene in a user-friendly and efficient manner by applying different 2D image processing tools on the canonical images without re-optimization. Tab. 1 summarizes the comparison of our method with existing methods in terms of editing functionalities and per-edit efficiency. We evaluate the effectiveness on various datasets, including LLFF [45], DTU [24], NeUVF [41], Replica [17, 65], IN2N [18], NeRF-Synthetic [46], and Mip-NeRF 360 [3] datasets in various editing tasks which are scene stylization, instance segmentation, and texture editing (*i.e.*, drawing). Experimental results show AGAP support various 3D editing tasks with on-par performance but at least 20× faster per edit.

2. Related Work

Implicit 3D representation. 3D modeling [1, 21, 25, 28, 39, 64, 68, 69] is pivotal in computer graphics and computer vision. Traditionally, explicit representations such as voxels and meshes have been employed for 3D shape modeling, but they often face challenges related to detail preservation and limited flexibility in processing. In contrast, implicit 3D representation like NeRF [46], SDF [53, 70, 79], Occupancy Networks [44], describing 3D scenes through continuous implicit functions, excels in capturing detailed geometry with improved fidelity. Many further works aim at improving NeRF in terms of various aspects, such as modeling capacity [2, 3], generative modeling [6, 16, 63, 71], and camera pose estimation [37, 38, 74]. In particular, methods like DVGO [66], Plenoxels [15], InstantNGP [48], TensoRF [7] focus on improving the convergence speed of volume rendering for 3D scenes by modeling the geometry and appearance with explicit grid representations. Our method leverages this technique for our density grid and canonical image as well, enabling efficient and rapid convergence of 3D modeling.

Neural scene editing. Existing research on NeRF editing can be broadly categorized into two: one focuses on editing the geometry [9, 11, 70, 78, 82, 85]; the other, known as

*Possible but highly depends on prompts

†Not inherently support in the official implementation

‡Editing time for stylization evaluated on a single A6000 GPU

style-based editing [18, 35, 52, 83, 86], aims to achieve scene stylization. Our research aligns with the latter category. Many NeRF stylization methods [22, 49, 83] have adopted techniques from 2D image stylization with style loss and content loss on images for NeRF optimization. While these methods can deliver 3D-consistent editing, they are primarily limited to global texture modifications and lack flexibility. Later, CLIP-NeRF [67] incorporates text conditions by regularizing the CLIP embeddings of the global scene with input prompts. Subsequent studies [35] extract 2D features such as DINO [5, 50] for local editing; IN2N [18] proposes an iterative approach to edit the input images using pre-trained diffusion models [4] for underlying NeRF optimization. Despite achieving high-fidelity editing results, most NeRF-based methods [8, 13, 19, 20, 26, 32, 36, 40, 43, 47, 57, 59] necessitate optimization for each text prompt or reference image, which can be inefficient in practical use. More recently, many works [10, 14, 23, 56, 72, 73, 75, 77, 80] start explore 3D editing with Gaussian Splatting [31]. , As we generally focus on NeRF-based editing alternatives, we choose GaussianEditor [10] among them as the representative comparing baseline.

Neural atlases. Our work shares similar insights with the research area of neural atlases [29, 41, 51, 81], which decompose videos into a canonical form with learned deformations, thereby enabling consistent video editing. Approaches such as neural layered atlases [29] employ an implicit network to distinguish foreground and background movement, dividing them into distinct layers. The CoDeF [51] methodology represents 2D videos using content deformation fields by integrating 3D hash tables [48]. However, these approaches lack 3D priors, limiting the effectiveness of 3D viewpoint changes in 3D scene editing.

3. Method

As shown in Tab. 1, existing editing methods [10, 18, 83, 86] necessitate several minutes or even hours per edit to re-optimize their original models. Meanwhile, we argue that their implicit editing procedures through re-optimization reduce the level of user interactivity compared to explicit editing ways. In view of such deficiencies, the core concept of AGAP is to learn a 2D canonical image as the interactive medium and allow users to do efficient and explicit 3D editing by lifting image processing on the 2D canonical image. Under such an idea, the key challenges involve designing a projection field that bridges the appearance of the 3D scene appearance with the 2D canonical image and ensuring the naturalness of the canonical image.

3.1. Preliminary

Volume rendering [27, 46] accumulates colors and densities of the 3D points sampled along the camera rays to render images. For a given camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ denoted by its origin $\mathbf{o} \in \mathbb{R}^3$ and direction $\mathbf{d} \in \mathbb{R}^3$, we sample N points $\{\mathbf{r}(t_i)\}_{i=1}^N$ along the ray defined by a sorted distance vector $\mathbf{t} = [t_1, ..., t_N]^T \in \mathbb{R}^N$.

NeRF [46] models the 3D scene implicitly and leverage MLP networks to decode the density $\sigma_i = \texttt{MLP}(\mathbf{r}(t_i))$ and the view-dependent color $\mathbf{c}_i = \texttt{MLP}(\mathbf{r}(t_i), \mathbf{d})$ of a point located at $\mathbf{r}(t_i)$ on the ray with viewing direction $\mathbf{d}$. To render the image pixel $\hat{C}(\mathbf{r})$, we apply discretized volume rendering by Max [42] along the N sampled ray points with δ_i denoting the distance to the nearby sampled points:

$$\begin{aligned} \hat{C}(\mathbf{r}) &= \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i, \text{where} \\ T_i &= \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j). \end{aligned} \tag{1}$$

Training such MLPs for 3D radiance field modeling requires observed images with known camera poses. Specifically, NeRF model is optimized by minimizing the average $\mathcal{L}_2$ distance between the rendered pixel color $\hat{C}(\mathbf{r})$ and the ground-truth pixel color $C(\mathbf{r})$:

$$\mathcal{L}_{color} = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2. \tag{2}$$

3.2. Model Formulation

Formally, given a set of multi-view training images $\mathcal{I}$, AGAP models the scene appearance by an explicit canonical image ϕ_I plus a corresponding implicit projection field P inspired by [51, 54, 55]; the scene geometry is estimated by an explicit 3D density grid ϕ_G. With such a representation, one can render different views through volume rendering. Our key property is that by explicitly editing the canonical image ϕ_I, it can propagate the edited appearance to the whole scene through the projection field P without any re-optimization. Our overall framework is shown in Fig. 2.

Density. In 3D modeling, textures are applied to the mesh surface to provide visual details such as colors, patterns, and material properties [25, 69]. We opt for an explicit voxel-grid representation [7, 15, 48, 66] instead of an implicit MLP such as NeRF [46] to achieve fast convergence and efficient query. Given a particular query point $\mathbf{p}_{xyz} \triangleq \mathbf{r}(t_i) \in \mathbb{R}^3$, we obtain the corresponding density $\sigma \in \mathbb{R}$ via a trilinear interpolation, followed by a Softplus activation:

$$\sigma = \texttt{Softplus}(\texttt{GridSample}(\mathbf{p}_{xyz}, \phi_G)), \tag{3}$$

where ϕ_G denotes the one-channel voxel grid with learnable parameter ϕ_g at a voxel resolution size of $N_x \times N_y \times N_z$. With a density grid with explicit parameterization, we hope our model can obtain a coarsely accurate density estimation of the 3D geometry at the early training stage, facilitating

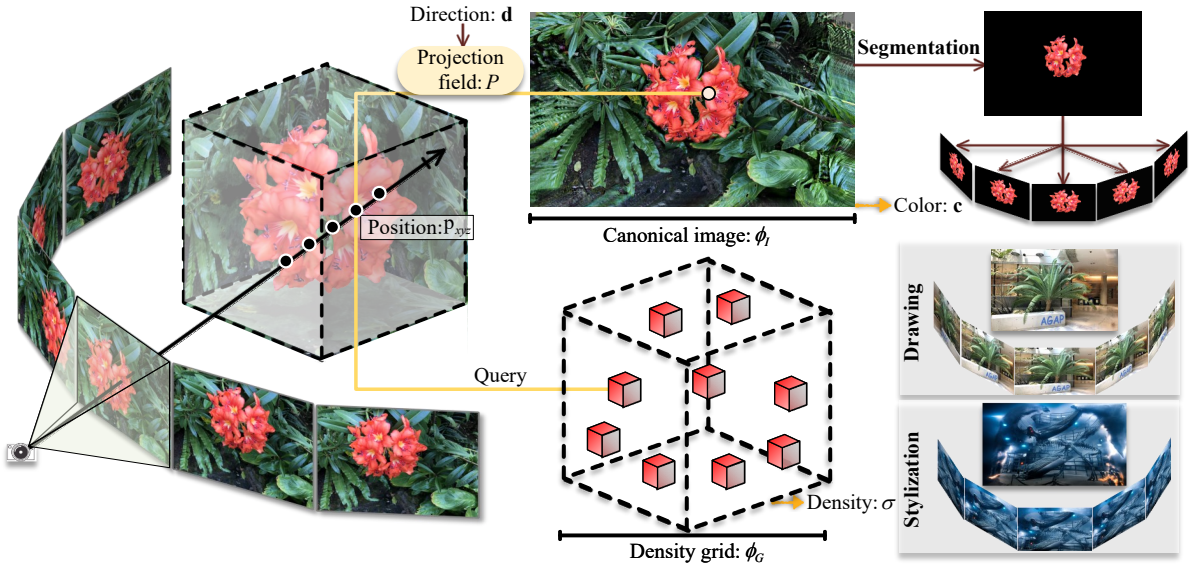


Figure 2. **The overall pipeline**. AGAP consists of two components: (1) an explicit 3D density grid ϕ_G to estimate geometry for density σ; (2) an explicit canonical image ϕ_I with an associated view-dependent projection field P to aggregate appearance for color $\mathbf{c}$. By performing 2D image processing on the canonical image, our method enables various editing (*e.g.*, instance segmentation, interactive drawing, and scene stylization) through volume rendering without the need for re-optimization.

the learning of 2D appearance aggregation. Such a choice is also proven to be crucial by our experiments in Sec. 4.4.

Appearance. In order to empower efficient 3D editing capabilities, we formulate the color appearance by an explicit canonical image $\phi_I \in \mathbb{R}^{H\times W\times 3}$ with an associated view-dependent implicit projection field $P(\cdot,\cdot) : (\mathbb{R}^3, \mathbb{R}^3) \rightarrow \mathbb{R}^2$, where H and W represent the image height and width, respectively. This formulation maps a given query point $\mathbf{p}_{xyz}$ in the 3D field with viewing direction $\mathbf{d}$ to the projected 2D point $\mathbf{p}_{uv}$ on the canonical image ϕ_I. The projection point $\mathbf{p}_{uv}$ is then used to query the RGB color $\mathbf{c} \in \mathbb{R}^3$ from the canonical image ϕ_I via interpolation:

$$\mathbf{c} = \texttt{Sigmoid}(\texttt{GridSample}(\mathbf{p}_{uv}, \phi_I)). \tag{4}$$

3.3. Canonical Projection with Projection Offset

We model the projection field P as a non-learnable canonical projection P_c with a projection offset learning P_o. To model different 3D scenes, we choose suitable foundational projection as the canonical projection P_c derived from different real camera models, which plays a significant role in ensuring naturalness and completeness in the learned canonical. The projection offset P_o aims to address view-dependent effects and handle occlusions in complex scenes.

Specifically, the canonical projection $P_c(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ projects the query 3D point $\mathbf{p}_{xyz}$ to an initial 2D projection $\tilde{\mathbf{p}}_{uv} = P_c(\mathbf{p}_{xyz})$ on the canonical. The projection offset $\Delta\mathbf{p}_{uv}$ is modeled by P_o with parameter weights ϕ_P:

$$\Delta\mathbf{p}_{uv} = P_o(\mathbf{p}_{xyz}, \mathbf{d}; \phi_P). \tag{5}$$

For simplicity, we omit the 3D positional encoding γ_p of $\mathbf{p}_{xyz}$ and the viewing direction encoding γ_d of $\mathbf{d}$ in Eq. (5).

The final projection point $\mathbf{p}_{uv}$ is formulated as follows:

$$\mathbf{p}_{uv} = \tilde{\mathbf{p}}_{uv} + \Delta\mathbf{p}_{uv}. \tag{6}$$

3.4. Optimization and Regularization

Projection regularization. In order to obtain a visually natural canonical image ϕ_I, one important regularization is to avoid the deviation from the perception by the defined pseudo canonical camera. We find the following simple regularization works well and stabilizes the training:

$$\mathcal{L}_{uv} = \|\Delta\mathbf{p}_{uv}\|_2^2 . \tag{7}$$

Total variation regularization. To mitigate floating densities, we incorporate total variation regularization [60] $\mathcal{L}_{tv}$ into the density grid ϕ_G. This regularization term is particularly beneficial during the initial stages of training.

Optimization objective. The final optimization process of our method is formulated as follows:

$$\phi_G^*, \phi_I^*, \phi_P^* = \underset{\phi_G,\phi_I,\phi_P}{\arg\min}\ \mathcal{L}_{color} + \mathcal{L}_{uv} + \mathcal{L}_{tv}. \tag{8}$$

4. Experiments

Recall that the canonical projection P_c can be viewed as positioning a pseudo-canonical camera within the 3D scene. This the crucial step in achieving a natural-looking canonical image ϕ_I by initializing the projection field P, by choosing a suitable foundational projection derived from different real camera models for the canonical projection P_c when modeling different types of 3D data scenes.

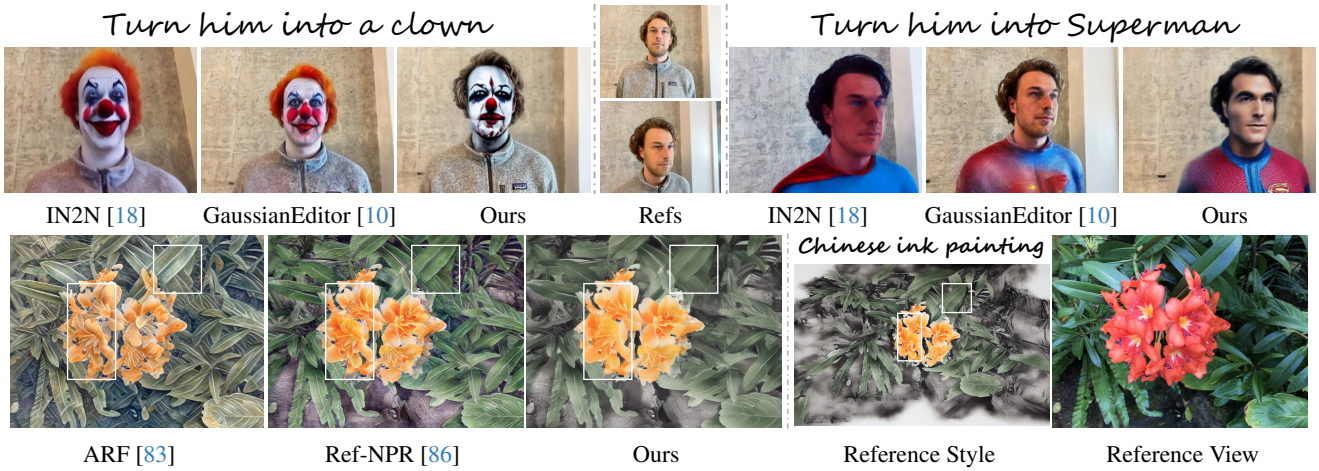


Figure 3. **Visual comparison of novel-view scene stylization results** on the *IN2N* and *LLFF* dataset given different text prompts or image reference. Our method can achieve on-par stylization results with the baselines while requiring no time-consuming re-optimization procedures. As highlighted in row two, our method can better preserve color and textural consistencies aligning with the image reference.

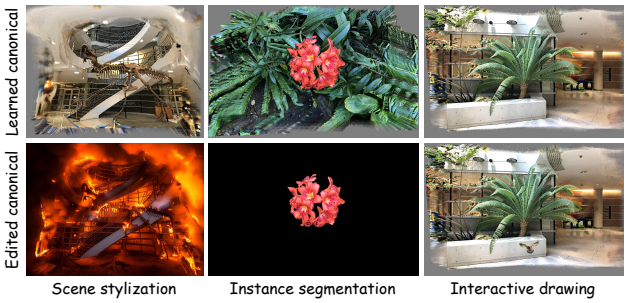


Figure 4. By performing explicit edits on the canonical image ϕ_I, our model propagates the editing effects through the learned projection field P for efficient 3D editing.

4.1. Data Types

Forward-facing data. The *LLFF* [45] dataset comprises real-world forward-facing scenes, where each scene is accompanied by several training images captured by handheld cameras placed in a rough grid pattern nearly on a vertical plane. We evaluate the dataset at a resolution of 756×1008.

We utilize the normalized device coordinate (NDC) space to model the forward-facing captures, where the pseudo canonical camera is defined as the average camera pose at the world origin. The perspective projection f_p can map any given point $\mathbf{p}_{xyz}$ in the world coordinates to its corresponding NDC point $f_p(\mathbf{p}_{xyz}) \triangleq (p_{x'}, p_{y'}, p_{z'})$. The coordinate range of the voxel grid ϕ_G is defined by the bounding box in NDC space. One notable property of NDC is that for a ray $\mathbf{r}$ origin from the camera center, all points along the ray share identical values for $p_{x'}$ and $p_{y'}$. Leveraging this property, we can design an appropriate choice for the canonical projection: $\tilde{\mathbf{p}}_{uv} \triangleq (p_{x'}, p_{y'})$.

Panorama data. The *Replica* [65] dataset is a collection of various high-quality and high-resolution 3D reconstructions of indoor scenes with clean and dense geometry. We evaluate the panorama scenes as processed in SOMSI [17], where each scene is rendered as a grid of equally spaced spherical images at a resolution of 1024×512.

The panorama data captures outward-facing views covering a 360-degree field of view around the global center. We place the pseudo-canonical camera at the global origin and define the Equirectangular projection as the canonical projection. Drawing inspiration from the smooth coordinate transforms in [3, 58], we introduce a new contracted formulation f_c specifically for panorama scenes:

$$f_c(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|}\left(1 - \frac{1}{\|\mathbf{x}\| + 1}\right), \tag{9}$$

where $\mathbf{x} \in \mathbb{R}^3$ is a 3D point in Euclidean space. This design transforms the points in such a way that they are distributed proportionally to the disparity in a unit sphere. Accordingly, the voxel grid ϕ_G is defined as a cube with range $[-1, 1]^3$. Given $f_c(\mathbf{p}_{xyz}) \triangleq (p_{x'}, p_{y'}, p_{z'})$, the canonical projection P_c from a 3D point $\mathbf{p}_{xyz}$ to a 2D canonical image pixel $\tilde{\mathbf{p}}_{uv} \triangleq (\tilde{p}_u, \tilde{p}_v)$ can be formulated as follows:

$$\begin{aligned} \tilde{p}_u &= \tan^{-1}\left(\frac{p_{y'}}{p_{x'}}\right) \in [-\pi, \pi], \\ \tilde{p}_v &= \sin^{-1}(p_{z'}) \in [-\frac{\pi}{2}, \frac{\pi}{2}]. \end{aligned} \tag{10}$$

Object-centric data. The *NeRF-Synthetic* [46] dataset comprises synthetic objects with intricate geometry and realistic materials. Each object in the dataset includes 100 training views and 200 test views, all rendered at a resolution of 800×800. The *DTU* [24] dataset is formulated in a object-level forward-facing setting, where training images are on quarter hemispheres ($\sim \frac{1}{8}$ spheres) with object masks. We evaluate this dataset at a resolution

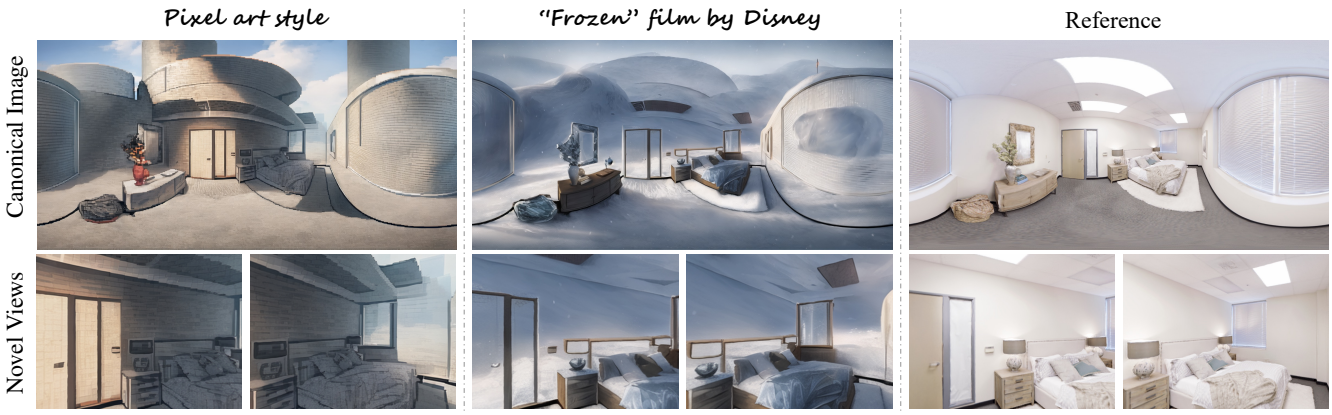


Figure 5. **More visualization of scene stylization results** on the panorama *Replica* dataset given different text prompts.

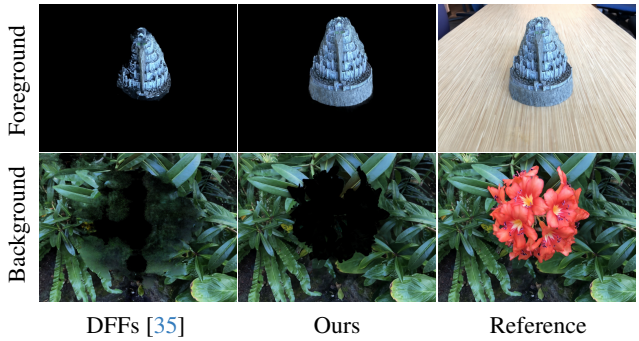


Figure 6. **Visual comparison of foreground and background segmentation results** on the *LLFF* dataset.

of 600×800 after downscaling the images by a factor of 2. The *NeUVF* [41] dataset capture human head videos using 12 calibrated cameras located at a hemisphere with approximately 120°. We use their first frames for static scenes, to specifically to model and edit the scenes that prominently feature human subjects for qualitative evaluation.

Analogous to Earth map, we can place the pseudo canonical camera at the global origin to model object-centric data using a (partial hemispheric) Equirectangular projection. Similar to Eq. (10), the canonical projection P_c from a 3D point $\mathbf{p}_{xyz} \triangleq (p_x, p_y, p_z)$ to a 2D canonical image pixel $\tilde{\mathbf{p}}_{uv} \triangleq (\tilde{p}_u, \tilde{p}_v) = (\tan^{-1}(\frac{p_y}{p_x}), \sin^{-1}(p_z))$.

Unbounded 360-degree data. The *Mip-NeRF 360* [3] and the *IN2N* [18] dataset comprise unbounded outdoor and indoor scenes, which is the most challenging case. Each scene showcases a complex background along with a central object or area, captured at varying high resolutions.

Learning a good 3D-to-2D projection field P for unbounded 360-degree scenes is indeed a non-trivial research challenge [12]. We leverage two canonical images for modeling the foreground central objects using (hemispheric) Equirectangular projection and the unbounded background by the contracted formulation as panorama, respectively.

Table 2. **Reconstructed PSNR** on the *LLFF* and *Replica* datasets.

Methods	*LLFF* dataset	*Replica* dataset
LLFF [45]	24.13	-
NeRF [46]	26.50	-
DVGO [66]	26.34	-
SOMSI [17]	-	39.54
Ours (PE)	24.83	38.42
Ours (Hash)	26.20	38.68

4.2. Implementation

Our pipeline involves a two-step process: 1) training a per-scene reconstruction model; 2) subsequent explicit edits on the canonical image ϕ_I for efficient 3D scene editing.

Training details. By default, we optimize a per-scene model for 60k steps using the Adam optimizer [33] with an initial learning rate of 0.1 for both the explicit density grid ϕ_G and the canonical image ϕ_I, and a learning rate of 0.001 for the implicit projection field P with learnable parameters ϕ_P. All experiments are conducted and tested on a single RTX A6000 GPU. For other implementation details and hyperparameters, please see the supplementary materials.

Editing pipeline. After we obtain the pre-trained model using our novel 3D representation, users can perform explicit edits as shown in Fig. 4 on the canonical image ϕ_I for various efficient 3D scene editing functionalities, such as scene stylization, instance segmentation, and texture editing. Our model can propagating the editing effects through the learned projection field P. Specifically, we utilize the prompt-guided *ControlNet* [84] for scene stylization and *Segment-anything (SAM)* [34] for instance segmentation. As for texture editing, users are free to draw or write on the canonical image.

4.3. Evaluation on Editability

Scene stylization. We conduct a comparative analysis between our method and three state-of-the-art stylization

Edited NVS 1 | Edited NVS 2 | Reference | Edited NVS 1 | Edited NVS 2 | Reference

Figure 7. **Visualization of texture editing (*i.e.*, drawing) results** at different novel viewpoints on the *LLFF*, *NeUVF*, and *DTU* dataset.

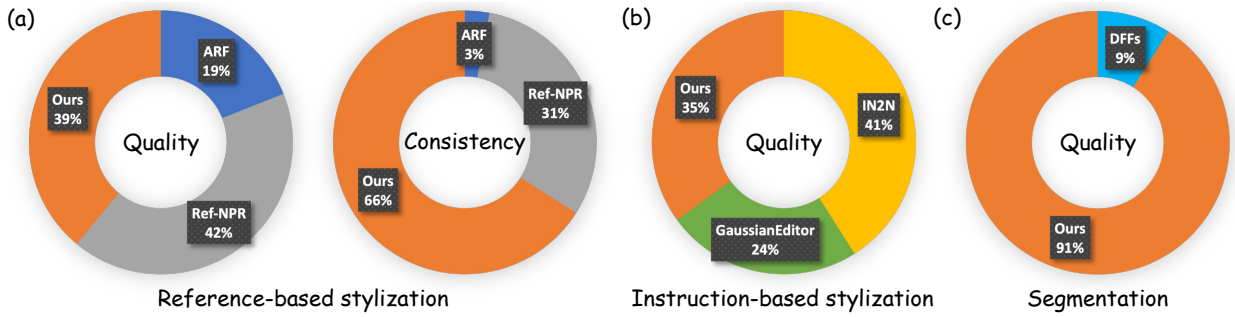


Figure 8. A user study of our method with different alternatives in terms of both perceptual editing quality and consistency.

Table 3. PSNR **ablations** of model components on *trex* scene.

Settings	PE	Hash
I. No canonical projection P_c	23.18	27.56
II. No projection offset P_o	23.81	23.88
III. No viewdir $\mathbf{d}$ in projection offset P_o	25.50	26.76
Full model	25.85	27.24

methods: ARF [83], Ref-NPR [86], and IN2N [18]. Specifically, ARF and Ref-NPR rely on one or multiple stylized reference images, while IN2N utilizes text prompts through Diffusion models [4] for guidance. All these baseline methods require additional optimization processes to achieve stylization given a specific style, while our method is optimization-free. As shown in Tab. 1, our editing speed for stylization is approximately 20× faster than ARF and Ref-NPR and approximately 500× faster than IN2N.

In Fig. 3, we demonstrate some comparing visualizations evaluated on the IN2N dataset and the LLFF dataset. In the first row, we present visual results from the IN2N dataset using different text prompts. While both our methods can effectively edit the scene to the desired style, the IN2N baseline necessitates approximately 3 hours to optimize the underlying NeRF model per edit, whereas our method requires no additional re-optimization. In the second row, we evaluate our method alongside the ARF and Ref-NPR baselines using the LLFF dataset. Both ARF and Ref-NPR produce implicit global stylization that looks like the reference style. However, our method achieves superior color and textural consistencies aligning with the provided reference style image. We present more visual results of stylization in Fig. 5 on the panorama dataset.

Instance segmentation. We evaluate our method with the state-of-the-art DFFs [35] method, which enables NeRFs to decompose a specific object given a text or image-patch query. The evaluation of baseline DFFs is only based on text query according to its official codebase. In Fig. 6, we show some comparing visualization of foreground and background segmentation. Our method allows users to do 3D instance segmentation easily by applying 2D segmentation of the desired objects in the canonical image.

Texture editing. Our method can further do textural appearance editing of the 3D scene by drawing or painting on the canonical image. In Fig. 7, we can observe that our method ensures both appearance and 3D consistency in novel views. For the first sample in row one, we paint an "AGAP" logo on the marble pedestal of the fern plant; for the second sample in row two, we draw two "ladybirds" on the leaves at the right bottom. We further present more visualizations in Fig. 9 and the supplementary materials.

User study. A user study comparing perceptual quality with baseline methods is presented in Fig. 8. We expect our approach to achieve comparable visual performance with

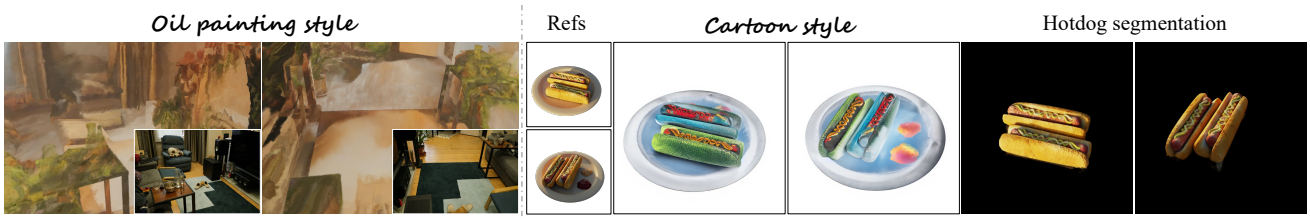
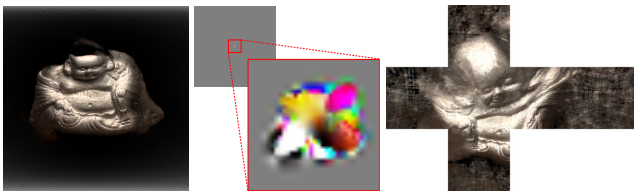


Figure 9. **More visualization of editing results** on the 360° and object-centric datasets.



(a) Ours (*w/* P_c) (b) Ours (*w/o* P_c) (c) UV map [76]

Figure 10. Canonical image of ablating different 2D projections.

Stylization NVS Reference Segmentation NVS

Figure 11. Hash models showcase a moderate level of editability.

existing alternatives, yet be far more efficient (at least 20× faster) and easier (*e.g.*, supporting more types of editing) to use in practice as in Tab. 1. The user study involves 43 participants, each answering 10 comparison questions against baseline methods to select the best option:

- For ***reference-based stylization*** (*i.e.*, ARF and Ref-NPR), participants select the best aligned with the reference image in terms of editing quality and consistency.
- For ***instruction-based stylization*** (*i.e.*, IN2N and GaussianEditor), participants evaluate overall aesthetic appeal based on the provided instruction text prompt.
- For ***segmentation*** comparisons (*i.e.*, DFFs), participants determine whether ours or DFFs provides superior segmentation accuracy and editing performance.

4.4. Ablation and Analysis.

Editability. When designing the projection field, we find that having a good canonical projection is essential for natural and user-friendly editing. As presented by Setting I in Tab. 3, where the canonical projection is removed it while keeping the others unchanged, we can see that the PE model experiences a drop while the hash model shows an increase. However, both models lose the ability to perform editing tasks as the learned canonical images ϕ_I deviate from natural images to latent color maps, where the visual comparison is shown in Figs. 10a and 10b. Apart from the various options for suitable canonical projection when handling different data types in Sec. 4.1, indeed, there

Ours (*w/o* offset P_o) Ours (*w/* offset P_o)

Figure 12. Reconstruction and editing visual patches of ablating the projection offset P_o.

exists some other projection functions like UV maps, but apparently they are hard to edit, as demonstrated in Fig. 10c.

Reconstruction fidelity. AGAP, as a new efficient editing pipeline for 3D modeling, we also report the PSNR results here for completeness to showcase the capacity of faithful reconstruction in Tab. 2. We examine the capacity of our models with PE and Hash designs for the input $\mathbf{p}_{xyz}$ to learn the projection offset P_o. We find that PE models lead to superior editing capacity, whereas hash models prioritize reconstruction quality, meanwhile, maintaining a moderate level of editability as shown in Fig. 11.

Learnable projection offset. In Setting II in Tab. 3, where the entire projection offset P_o is eliminated, a significant drop in performance is observed. In this case, the model cannot successfully handle the occlusion effects as presented in Fig. 12 for both reconstruction and editing. In Setting III in Tab. 3, the removal of view-dependence from the learnable projection offset P_o leads to a minor decrease in terms of statistical reconstruction, as the model no longer considers viewing directions.

5. Discussion and Conclusion

This paper explores the task of efficient 3D scene editing, where we focus on editing efficiency and user interactivity. Specifically, we propose AGAP, an editing-friendly and efficient solution for neural 3D scene editing, by representing a 3D scene using a 2D canonical image equipped with a projection field, such that users can easily perform efficient 3D editing via processing the 2D image. The key challenge and contribution of our method lies in how to regularize the projection field to make the canonical image look natural.

Compared to existing baselines, which require a time-consuming optimization process for one editing style, our approach can perform on-par 3D editing with at least 20× faster speed per edit and be far easier to use in practice. It supports various types of editing, such as scene stylization, instance segmentation, and interactive drawing.

References

[1] Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.J.: Learning representations and generative models for 3d point clouds. In: Proceedings of ICML (2018) 2

[2] Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of ICCV (2021) 2

[3] Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of CVPR (2022) 2, 5, 6

[4] Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of CVPR (2023) 3, 7

[5] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of ICCV (2021) 3

[6] Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3d generative adversarial networks. In: Proceedings of CVPR (2022) 2

[7] Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: Proceedings of ECCV (2022) 2, 3

[8] Chen, H., Loy, C.C., Pan, X.: Mvip-nerf: Multi-view 3d inpainting on nerf scenes via diffusion prior. In: Proceedings of CVPR (2024) 3

[9] Chen, J.K., Lyu, J., Wang, Y.X.: NeuralEditor: Editing neural radiance fields via manipulating point clouds. In: Proceedings of CVPR (2023) 2

[10] Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In: Proceedings of CVPR (2024) 2, 3, 5

[11] Deng, Y., Yang, J., Tong, X.: Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In: Proceedings of CVPR (2021) 2

[12] Desbrun, M., Meyer, M., Alliez, P.: Intrinsic parameterizations of surface meshes. Comput. Graph. Forum **21**(3), 209–218 (2002) 6

[13] Dihlmann, J., Engelhardt, A., Lensch, H.P.A.: Signerf: Scene integrated generation for neural radiance fields. In: Proceedings of CVPR (2024) 3

[14] Fang, J., Wang, J., Zhang, X., Xie, L., Tian, Q.: Gaussianeditor: Editing 3d gaussians delicately with text instructions. In: Proceedings of CVPR (2024) 3

[15] Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of CVPR (2022) 2, 3

[16] Gu, J., Liu, L., Wang, P., Theobalt, C.: Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In: Proceedings of ICLR (2022) 2

[17] Habtegebrial, T., Gava, C.C., Rogge, M., Stricker, D., Jampani, V.: SOMSI: spherical novel view synthesis with soft occlusion multi-sphere images. In: Proceedings of CVPR (2022) 2, 5, 6

[18] Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of ICCV (2023) 2, 3, 5, 6, 7

[19] He, R., Huang, S., Nie, X., Hui, T., Liu, L., Dai, J., Han, J., Li, G., Liu, S.: Customize your nerf: Adaptive source driven 3d scene editing via local-global iterative training. In: Proceedings of CVPR (2024) 3

[20] He, Y., Yuan, W., Zhu, S., Dong, Z., Bo, L., Huang, Q.: Freditor: High-fidelity and transferable nerf editing by frequency decomposition. In: Proceedings of ECCV (2024) 3

[21] Huang, P., Matzen, K., Kopf, J., Ahuja, N., Huang, J.: Deepmvs: Learning multi-view stereopsis. In: Proceedings of CVPR. pp. 2821–2830 (2018) 2

[22] Huang, Y., He, Y., Yuan, Y., Lai, Y., Gao, L.: Stylized-nerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In: Proceedings of CVPR (2022) 3

[23] Jaganathan, V., Huang, H.H., Irshad, M.Z., Jampani, V., Raj, A., Kira, Z.: ICE-G: image conditional editing of 3d gaussian splats. In: Proceedings of CVPR (2024) 3

[24] Jensen, R.R., Dahl, A.L., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Proceedings of CVPR. pp. 406–413 (2014) 2, 5

[25] Ji, M., Gall, J., Zheng, H., Liu, Y., Fang, L.: Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In: Proceedings of ICCV (2017) 2, 3

[26] Jung, H., Nam, S., Sarafianos, N., Yoo, S., Sorkine-Hornung, A., Ranjan, R.: Geometry transfer for stylizing radiance fields. In: Proceedings of CVPR (2024) 3

[27] Kajiya, J.T., Herzen, B.V.: Ray tracing volume densities. In: Proceedings of SIGGRAPH. pp. 165–174 (1984) 3

[28] Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Proceedings of ECCV (2018) 2

[29] Kasten, Y., Ofri, D., Wang, O., Dekel, T.: Layered neural atlases for consistent video editing. TOG **40**(6), 210:1–210:12 (2021) 3
[30] Kazhdan, M.M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of SGP (2006) 2
[31] Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139:1–139:14 (2023) 3
[32] Khalid, U., Iqbal, H., Karim, N., Hua, J., Chen, C.: Latenteditor: Text driven local editing of 3d scenes. In: Proceedings of ECCV (2024) 3
[33] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of ICLR (2015) 6
[34] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W., Dollár, P., Girshick, R.B.: Segment anything. CoRR **abs/2304.02643** (2023) 6
[35] Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation. In: Advances in NeurIPS (2022) 2, 3, 6, 7
[36] Koo, J., Park, C., Sung, M.: Posterior distillation sampling. In: Proceedings of CVPR (2024) 3
[37] Lin, C., Ma, W., Torralba, A., Lucey, S.: BARF: bundle-adjusting neural radiance fields. In: Proceedings of ICCV (2021) 2
[38] Lin, Y., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.: inerf: Inverting neural radiance fields for pose estimation. In: Proceedings of IROS (2021) 2
[39] Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. TPAMI **38**(10), 2024–2039 (2016) 2
[40] Liu, X., Xue, H., Luo, K., Tan, P., Yi, L.: Genn2n: Generative nerf2nerf translation. In: Proceedings of CVPR (2024) 3
[41] Ma, L., Li, X., Liao, J., Wang, X., Zhang, Q., Wang, J., Sander, P.V.: Neural parameterization for dynamic human head editing. ACM Trans. Graph. **41**(6), 236:1–236:15 (2022) 2, 3, 6
[42] Max, N.L.: Optical models for direct volume rendering. TVCG **1**(2), 99–108 (1995) 3
[43] Mazzucchelli, A., Garcia-Garcia, A., Garces, E., Rivas-Manzaneque, F., Moreno-Noguer, F., Penate-Sanchez, A.: Irene: Instant recoloring of neural radiance fields. In: Proceedings of CVPR (2024) 3
[44] Mescheder, L.M., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of CVPR (2019) 2
[45] Mildenhall, B., Srinivasan, P.P., Cayon, R.O., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: practical view synthesis with prescriptive sampling guidelines. TOG **38**(4), 29:1–29:14 (2019) 2, 5, 6
[46] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Proceedings of ECCV. pp. 405–421 (2020) 2, 3, 5, 6
[47] Mirzaei, A., Aumentado-Armstrong, T., Brubaker, M.A., Kelly, J., Levinshtein, A., Derpanis, K.G., Gilitschenski, I.: Watch your steps: Local image and scene editing by text instructions. In: Proceedings of ECCV (2024) 3
[48] Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. TOG **41**(4), 102:1–102:15 (2022) 2, 3
[49] Nguyen-Phuoc, T., Liu, F., Xiao, L.: Snerf: stylized neural implicit representations for 3d scenes. TOG **41**(4), 142:1–142:11 (2022) 3
[50] Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision. CoRR **arXiv/2304.07193** (2023) 3
[51] Ouyang, H., Wang, Q., Xiao, Y., Bai, Q., Zhang, J., Zheng, K., Zhou, X., Chen, Q., Shen, Y.: Codef: Content deformation fields for temporally consistent video processing. CoRR **abs/2308.07926** (2023) 3
[52] Pang, H., Hua, B., Yeung, S.: Locally stylized neural radiance fields. CoRR **abs/2309.10684** (2023) 2, 3
[53] Park, J.J., Florence, P.R., Straub, J., Newcombe, R.A., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of CVPR. pp. 165–174 (2019) 2
[54] Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: Proceedings of ICCV. pp. 5845–5854 (2021) 3
[55] Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. TOG **40**(6), 238:1–238:12 (2021) 3
[56] Qiu, R., Yang, G., Zeng, W., Wang, X.: Feature splatting: Language-driven physics-based scene synthesis and editing. In: Proceedings of ECCV (2024) 3
[57] Radl, L., Steiner, M., Kurz, A., Steinberger, M.: Laenerf: Local appearance editing for neural radiance fields. In: Proceedings of CVPR (2024) 3
[58] Reiser, C., Szeliski, R., Verbin, D., Srinivasan, P.P., Mildenhall, B., Geiger, A., Barron, J.T., Hedman,

P.: MERF: memory-efficient radiance fields for real-time view synthesis in unbounded scenes. TOG **42**(4), 89:1–89:12 (2023) 5

[59] Rojas, S., Philip, J., Zhang, K., Bi, S., Luan, F., Ghanem, B., Sunkavalli, K.: Datenerf: Depth-aware text-based editing of nerfs. In: Proceedings of ECCV (2024) 3

[60] Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60**(1), 259–268 (1992) 4

[61] Schönberger, J.L., Frahm, J.: Structure-from-motion revisited. In: Proceedings of CVPR (2016) 2

[62] Schönberger, J.L., Zheng, E., Frahm, J., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Proceedings of ECCV (2016) 2

[63] Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: GRAF: generative radiance fields for 3d-aware image synthesis. In: Advances in NeurIPS (2020) 2

[64] Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhöfer, M.: Deepvoxels: Learning persistent 3d feature embeddings. In: Proceedings of CVPR (2019) 2

[65] Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C.Y., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.A.: The replica dataset: A digital replica of indoor spaces. CoRR **abs/1906.05797** (2019) 2, 5

[66] Sun, C., Sun, M., Chen, H.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proceedings of CVPR. pp. 5449–5459 (2022) 2, 3, 6

[67] Wang, C., Chai, M., He, M., Chen, D., Liao, J.: Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In: Proceedings of CVPR (2022) 3

[68] Wang, J., Sun, B., Lu, Y.: Mvpnet: Multi-view point regression networks for 3d object reconstruction from A single image. In: Proceedings of AAAI (2019) 2

[69] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.: Pixel2mesh: Generating 3d mesh models from single RGB images. In: Proceedings of ECCV (2018) 2, 3

[70] Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: Advances in NeurIPS (2021) 2

[71] Wang, Q., Shi, Z., Zheng, K., Xu, Y., Peng, S., Shen, Y.: Benchmarking and analyzing 3d-aware image synthesis with a modularized codebase. In: Advances in NeurIPS (2023) 2

[72] Wang, Y., Wu, Q., Zhang, G., Xu, D.: Gscream: Learning 3d geometry and feature consistent gaussian splatting for object removal. In: Proceedings of ECCV (2024) 3

[73] Wang, Y., Yi, X., Wu, Z., Zhao, N., Chen, L., Zhang, H.: View-consistent 3d editing with gaussian splatting. In: Proceedings of ECCV (2024) 3

[74] Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf-: Neural radiance fields without known camera parameters. CoRR **abs/2102.07064** (2021) 2

[75] Wu, J., Bian, J., Li, X., Wang, G., Reid, I.D., Torr, P., Prisacariu, V.A.: Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In: Proceedings of ECCV (2024) 3

[76] Xiang, F., Xu, Z., Hasan, M., Hold-Geoffroy, Y., Sunkavalli, K., Su, H.: Neutex: Neural texture mapping for volumetric neural rendering. In: Proceedings of CVPR (2021) 2, 8

[77] Xu, T., Hu, W., Lai, Y., Shan, Y., Zhang, S.: Texture-gs: Disentangling the geometry and texture for 3d gaussian splatting editing. In: Proceedings of ECCV (2024) 3

[78] Yang, B., Bao, C., Zeng, J., Bao, H., Zhang, Y., Cui, Z., Zhang, G.: Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In: Proceedings of ECCV (2022) 2

[79] Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: Advances in NeurIPS (2021) 2

[80] Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. In: Proceedings of ECCV (2024) 3

[81] Ye, V., Li, Z., Tucker, R., Kanazawa, A., Snavely, N.: Deformable sprites for unsupervised video decomposition. In: Proceedings of CVPR. IEEE (2022) 3

[82] Yuan, Y., Sun, Y., Lai, Y., Ma, Y., Jia, R., Gao, L.: Nerf-editing: Geometry editing of neural radiance fields. In: Proceedings of CVPR (2022) 2

[83] Zhang, K., Kolkin, N.I., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: ARF: artistic radiance fields. In: Proceedings of ECCV (2022) 2, 3, 5, 7

[84] Zhang, L., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of ICCV (2023) 6

[85] Zhang, Y., Peng, S., Moazeni, A., Li, K.: PAPR: proximity attention point rendering. In: Advances in NeurIPS (2023) 2

[86] Zhang, Y., He, Z., Xing, J., Yao, X., Jia, J.: Ref-npr: Reference-based non-photorealistic radiance fields for controllable scene stylization. In: Proceedings of CVPR (2023) 2, 3, 5, 7