

Appendix for DRAFT: Direct Radiance Fields Editing with Composable Operations

Zhihan Cai *¹
cai-zh21@mails.tsinghua.edu.cn

¹ Tsinghua University

² Xi'an Jiaotong University

Kailu Wu *¹
wkl22@mails.tsinghua.edu.cn

Dapeng Cao²
dapengcao@stu.xjtu.edu.cn

Feng Chen¹
chenf20@mails.tsinghua.edu.cn

Kaisheng Ma †¹
kaisheng@mails.tsinghua.edu.cn

A Supplemental Demo Video

Because 3D consistency and visualization are important for the quality of NeRF models, both of which are difficult to demonstrate in a limited number of views. To truly appreciate the nuanced improvements and intricate details achieved through our methodology, **we encourage you to engage with our demo video**. This dynamic visual presentation not only provides a more immersive understanding of the editing results but also showcases the robustness of our approach in handling intricate 3D scenes.

B More Implementation Details

B.1 Implementation Details and Inference Speed

Representations. We employ DVGO [14] for neural-based approaches and Plenoxels [13] for non-neural-based approaches. We use DVGO by default. Since Plenoxels use sparse voxel grids, we convert sparse voxel grids to dense voxel grids to facilitate editing. For scenes from LLFF, they are trained with $192 \times 192 \times 128$ voxels. For scenes from Mip-360, they are trained with 200^3 voxels. For other scenes, they are trained with 160^3 voxels. The original DVGO [14] can not accommodate forward-facing and unbounded inward-facing scenes. In response, we use their enhanced work, DVGOv2 [14], which addresses these limitations. The training resolutions for DVGOv2 is described in the paper. The training resolution for Plenoxels is 256^3 .

Editing Methods. The interpolation operation for all editing methods in all scenes uses tri-linear interpolation. In the segmentation-based selection method, we use a pre-trained

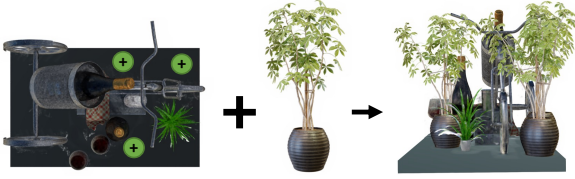


Figure 1: Editing results on cross-scene editing. We insert the Ficus from the NeRF dataset into the Wineholder from the NSVF dataset.

instance segmentation network from PointRend [5] which is trained on COCO [17] using ResNet101 [18] as the backbone. Among the masks that exhibit reasonably accurate segmentation, we discard the 25% of points situated farthest from the camera in experiments. For seam carving, we choose the parallel push-relabel algorithm to solve the maximum flow problem. Seam carving is the most time-consuming operation, 50 times of seam carving on 160^3 voxels takes around 12 minutes on a 2-core Intel(R) Xeon(R) Gold 6145 machine.

B.2 Details for Equ.4

To optimize equation in Equ.4 in the main paper, we use Adam with $\text{lr} = 1$, $\text{betas} = (0.9, 0.999)$, $\text{weight_decay} = 0$ as the optimizer. The optimization process has 500 steps, and the learning rate is decayed by 0.1 after the 300 steps. The optimization process takes approximately 18 minutes on a single NVIDIA P100. The value of \mathbf{v} is initialized with $\text{INTRPL}(\mathbf{x}_{pre}, \mathbf{V}^{CLR})$ in the equation.

B.3 Details for Non-rigid Deformation

We demonstrate non-rigid deformation using cage-based deformation as an editing operation. The cage-based deformation is implemented based on the released code from Deforming-NeRF [19]. The settings for cage interpolation is exactly the same as Deforming-NeRF. Check the supplementary video for a detailed comparison.

C Radiance Fields

NeRFs[8] use Multi-Layer Perceptron (MLP) network $F_{\Theta}(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma)$ to represent a 3D scene, where \mathbf{x} represent a spatial location, \mathbf{d} represent a viewing direction, \mathbf{c} is the emitted color, and σ is the volume density. Each pixel in an image can be cast into a ray in world coordinates, and we use a discrete set of samples N on ray \mathbf{r} to render the corresponding color $\hat{\mathbf{C}}(\mathbf{r})$ and depth $\hat{D}(\mathbf{r})$ with volume rendering[4]:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i, \quad (1)$$

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) d_i, \quad (2)$$

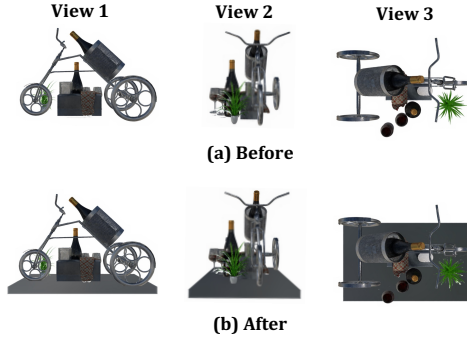


Figure 2: Applications of *Copy-and-Paste* to add a base for Wineholder from NSVF.

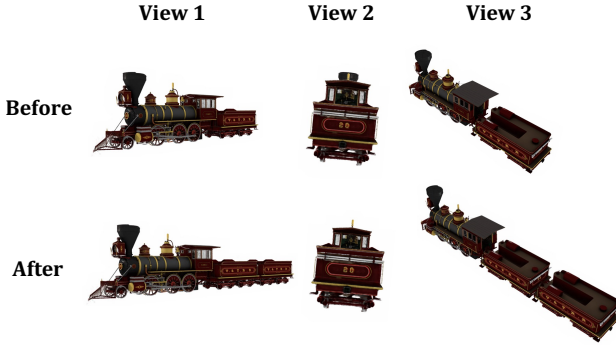


Figure 3: More results of object *Copy-and-Paste* on the Steamtrain from NSVF dataset. We duplicate the carriage to form a longer steam train.

where

$$T_i = \exp \left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right). \quad (3)$$

For sample i , T_i denotes the accumulated transmittance along the ray, d_i is the distance from the camera, \mathbf{c}_i is the color, σ_i is the density, and δ_i is the distance between adjacent samples.

D Additional Experimental Results

D.1 Cross-scene Editing

We validate the ability to edit across scenes. Here, the scenes are reconstructed by Plenoxels because the same hyper-pixel provides the same visualization result in all scenes based on Plenoxels. In fact, representations based on neural networks, such as DVGO, can also support cross-scene editing by training the same neural network on different scenes at the same time [6]. In Fig. 1 we place several ficusses in the Wineholder scene. Note that this is the result of rendering directly on the edited feature grids, rather than rendering on separate scenes and then combining them.

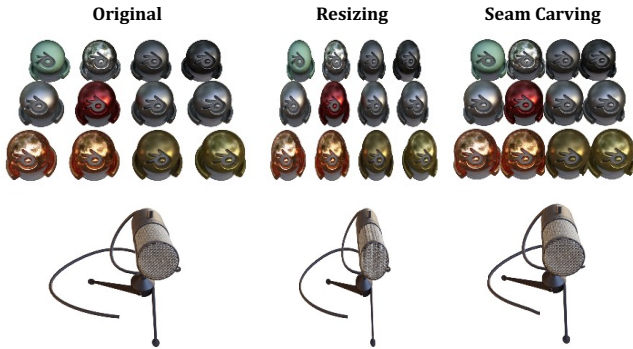


Figure 4: More seam carving results on bounded scenes from NeRF dataset.

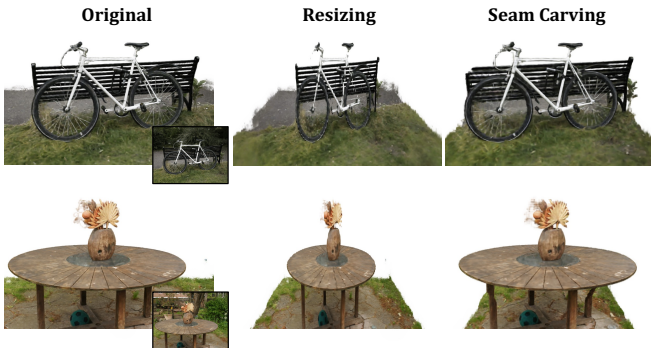


Figure 5: More seam carving results on unbounded 360 scenes from Mip-360 dataset. Note that only the foreground is carved.

D.2 Common Operations

The subsequent figures present additional instances of our proposed common operations, illustrating the efficacy of our approach. As shown in Fig. 2, we apply the *Copy-and-Paste* technique to the Wineholder scene from NSVF, adding a base to it. Expanding on the versatility of our method, Fig. 3 demonstrates the outcomes of *Copy-and-Paste* operations applied to the Steamtrain scene. We selectively isolate a train carriage using *Hexahedron Selection* and subsequently duplicate it at the rear, effectively extending the length of the train. Notably, the extended carriage retains the high-quality visual attributes of the original, underscoring the proficiency of our approach in seamlessly replicating and extending complex structures. These results showcase the effectiveness of our method in preserving visual details during complex manipulations.

D.3 Seam Carving

In this section, we demonstrate additional application of our seam carving algorithm to both bounded scenes from the NeRF dataset (as shown in Fig. 4) and unbounded 360 scenes from



Figure 6: Ablation on the size of voxel grids using Lego from NeRF Synthetic.

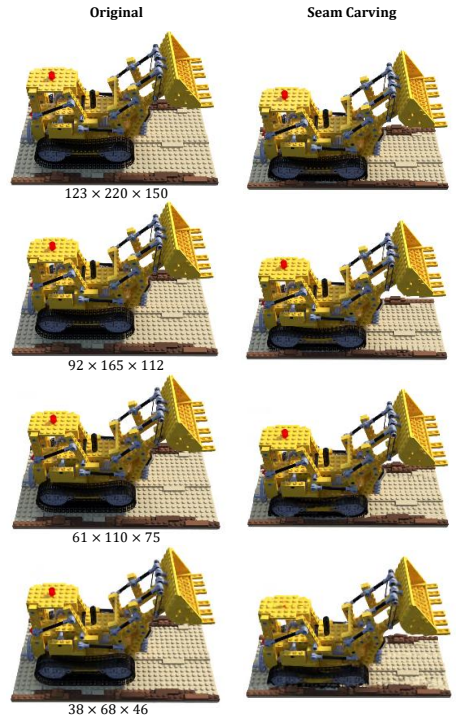


Figure 7: Ablation on the size of voxel grids using Flower from LLFF.

the Mip-360 dataset [24] (presented in Fig. 5). Notably, because of the distinct nature of the background and foreground components within the radiance field representation of the Mip-360 dataset, our seam carving procedure focuses exclusively on the foreground elements. Subsequently, to ensure visual coherence and direct the viewer’s focus to the enhanced foreground, we replace the background with a plain white canvas.

E Additional Ablation Study

The ablation study focusing on the size of Voxel Grids is presented in Fig. 6 and Fig. 7. As depicted in these figures, the visual results of seam carving maintain striking similarities even when applied to voxel grids of different sizes. Moreover, these visualizations showcase the robustness of our seam carving approach, revealing consistent outcomes across diverse datasets. This uniformity in visual quality underscores the stability and effectiveness of our method across varying grid configurations.

F Future Work

The quality of the editing results in our proposed editing system is intricately tied to the quality of the radiance field. In instances where the original radiance field quality is suboptimal,

the selection of specific objects becomes challenging, leading to an increased likelihood of artifacts during the editing operation. Similarly, the quality of seam carving of the radiance fields depends on the scene itself.

Despite the flexibility of our editing system to use radiance fields based on sparse voxels, it's essential to note that the editing operations are conducted in dense voxels. This may result in substantial memory consumption and lead to slow rendering time when dealing with large scenes.

In the future, we will explore the following aspects:

- Implement more power editing methods in this system, such as patch matching [10] and poisson editing [9].
- Extend the system to point-cloud-based and mesh-based hybrid representations.
- Introduce 3D generative models to assist in editing operations in order to repair artifacts caused by editing operations.
- Translate the post-edit scene to sparse voxel grids which can provide much faster rendering speed and consume much smaller memories.

References

- [1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. Patch-match: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. doi: 10.1145/1531326.1531330. URL <https://doi.org/10.1145/1531326.1531330>.
- [2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CoRR*, abs/2111.12077, 2021. URL <https://arxiv.org/abs/2111.12077>.
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL <https://doi.org/10.1109/CVPR.2016.90>.
- [4] James T. Kajiya and Brian Von Herzen. Ray tracing volume densities. In Hank Christiansen, editor, *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1984, Minneapolis, Minnesota, USA, July 23-27, 1984*, pages 165–174. ACM, 1984. doi: 10.1145/800031.808594. URL <https://doi.org/10.1145/800031.808594>.
- [5] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross B. Girshick. Pointrend: Image segmentation as rendering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9796–9805. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00982. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Kirillov_PointRend_Image_Segmentation_As_Rendering_CVPR_2020_paper.html.

- [6] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4340–4350, January 2023.
- [7] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.
- [8] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, 2022. doi: 10.1145/3503250. URL <https://doi.org/10.1145/3503250>.
- [9] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003. doi: 10.1145/882262.882269. URL <https://doi.org/10.1145/882262.882269>.
- [10] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Superfast convergence for radiance fields reconstruction. *CoRR*, abs/2111.11215, 2021. URL <https://arxiv.org/abs/2111.11215>.
- [11] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved direct voxel grid optimization for radiance fields reconstruction. *CoRR*, abs/2206.05085, 2022. doi: 10.48550/arXiv.2206.05085. URL <https://doi.org/10.48550/arXiv.2206.05085>.
- [12] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. *CoRR*, abs/2207.12298, 2022. doi: 10.48550/arXiv.2207.12298. URL <https://doi.org/10.48550/arXiv.2207.12298>.
- [13] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *CoRR*, abs/2112.05131, 2021. URL <https://arxiv.org/abs/2112.05131>.