

Supplementary Materials: Learning Naturally Aggregated Appearance for Efficient 3D Editing

Ka Leong Cheng^{1,2}, Qiuyu Wang², Zifan Shi^{1,2}, Kecheng Zheng², Yinghao Xu^{2,3},
Hao Ouyang², Qifeng Chen^{1†}, Yujun Shen^{2†}

¹HKUST ²Ant Group ³Stanford

A. Supplementary Video

To offer a more comprehensive demonstration of our visual results, we have included a supplementary video showcasing three editing cases (scene stylization, instance segmentation, and texture editing) on diverse 3D scenes. Please check “demo.mp4” for details.

B. Failure Cases and Limitation

Recall that in this paper, we come up with an editing-friendly representation, AGAP, which permits explicit 3D editing with the help of a natural 2D canonical image. In this section, we present some failure cases and discuss the limitation.

Our method supports texture editing by directly painting onto the canonical image. However, such painting might be distorted when the novel viewpoint exhibits occlusions on the edited regions. As shown in Fig. S1, we can easily paint an “AGAP” logo onto the marble pedestal of the fern plant in the canonical image, allowing us to directly obtain the edited NVS from different novel viewpoints through neural rendering. However, the logo appears distorted in the regions that are occluded by the fern plant.

Our pipeline includes a projection offset P_o to handle moderate levels of occlusion, which implicitly projects and clusters the 3D points to nearby pixels. However, we acknowledge that our method has limitations in handling extensive occlusions in the 3D scenes. Suppose a 3D scene is extremely complex and contains numerous extensive occlusions. Projecting such a scene onto a 2D plane (like a UV map) is possible, but creating a 2D projection that naturally and fully displays the scene for easy interactivity is very challenging and nearly impossible. Hence, such cases are beyond the scope of our current study, and we mainly focus on 3D scenes with moderate levels of occlusion.

Table S1. **Hyperparameters** for training various scenes in different datasets.

Data Types	Image Size	Weight Factor	
		λ_{uv}	λ_{tv}
Forward-facing [3, 7]	(768, *)	10^{-5}	10^{-5}
Object-centric [4, 6, 8]	(768, *) / (*, 768)	10^{-1}	10^{-4}
Panorama [1–3, 11]	(768, 1536)	10^{-1}	10^{-4}

C. Training Details

Our 3D editing pipeline involves a two-step process: (1) we first train a per-scene reconstruction model using the proposed AGAP representation, which includes an explicit density grid ϕ_G , an explicit canonical image ϕ_I , and an associated projection field ϕ_P ; (2) we can then perform explicit 2D edits on the canonical image ϕ_I for 3D scene editing, including scene stylization, instance segmentation, and texture editing. All experiments, including training on various scenes from different datasets, are conducted and tested on a single RTX A6000 GPU, with specific hyperparameter details outlined in Tab. S1.

Optimization. In the first stage, we employ the Adam optimizer [5] to optimize a per-scene model for 60k steps with an initial learning rate of 0.1 for both the explicit 3D density grid ϕ_G and 2D canonical image ϕ_I , and a learning rate of 0.001 for the implicit projection field P with learnable parameter ϕ_P . The optimization of the entire model involves an objective function comprising three main components: (1) an average \mathcal{L}_2 photometric loss \mathcal{L}_{color} between the rendered pixel color $\hat{C}(\mathbf{r})$ and the ground-truth color $C(\mathbf{r})$; (2) a projection regularization \mathcal{L}_{uv} aimed at minimizing the projection offset $\Delta \mathbf{p}_{uv}$; and (3) a total variation regularization applied to the density grid ϕ_G .

Weight factor. As stated in the main paper, the final optimization process of our method to model the scene for



Figure S1. Limitation of texture editing on occluded regions.

efficient editing can be formulated as follows:

$$\phi_G^*, \phi_I^*, \phi_P^* = \arg \min_{\phi_G, \phi_I, \phi_P} \mathcal{L}_{color} + \mathcal{L}_{uv} + \mathcal{L}_{tv}, \quad (S1)$$

where the second and the third terms are controlled by their corresponding weight factors λ_{uv} and λ_{tv} , respectively. To be specific, the weight factor λ_{uv} is set as 10^{-5} for forward-facing scene and larger value of 10^{-1} or 10^{-2} for panorama and inward-facing 360° scenes; the weight factor λ_{tv} is set as 10^{-4} for panorama scene and 10^{-5} for other scenes. Note that for panorama and inward-facing 360° data, the total variation term is disabled after 20000 steps to learn depths in detail.

Progressive training. Similar to [1, 8, 12], we apply progressive scaling for our voxel grid ϕ_G and canonical image ϕ_I for a coarse-to-fine learning process. By gradually refining the resolution of both representations, we enable a more detailed and comprehensive learning process.

At specific scaling-up milestone steps, we increase the ϕ_G voxel count by a factor of 2 and the ϕ_I pixel count by a factor of 4. For the forward-facing and object-centric data scenes, the voxel grid ϕ_G is scaled up at $\{2000, 4000, 6000, 8000\}$ training steps and the canonical image ϕ_I is scaled up at $\{8000, 16000\}$ training steps. For the panorama data types, the voxel grid ϕ_G is scaled up at $\{2000, 4000, 6000, 8000, 10000, 12000, 14000, 16000\}$ training steps, and the canonical image ϕ_I is scaled up at $\{4000, 8000, 12000, 16000\}$ training steps.

Size of voxel grid ϕ_G and canonical image ϕ_I . After the progressive scaling up, The final resolution of the voxel grid ϕ_G is set as $384 \times 384 \times 256$ for forward-facing scenes and $320 \times 320 \times 320$ for other scenes.

For the NDC canonical camera of forward-facing scenes, we set the height H_I of the learnable explicit canonical image ϕ_I as 768, and the canonical image width W_I is adaptively calculated according to the width-height aspect ratio of the training images and the computed bounding box of the scene in NDC space. Denoting the bounding box in NDC space as (x'_{min}, x'_{max}) in x' dimension, (y'_{min}, y'_{max}) in y' dimension, and $(z'_{min}, z'_{max}) = (-1, 1)$ in z' dimension and the aspect ratio as r_I , we can then

calculate the canonical image width as:

$$W_I = H_I \times r_I \times \frac{x'_{max} - x'_{min}}{y'_{max} - y'_{min}}. \quad (S2)$$

For the canonical camera of panorama scenes, the canonical image height is set to be 768 and the width W_I is set to be $2 \times 768 = 1536$ according to the definition of Equirectangular projection.

For the canonical camera of object-centric scenes, the canonical image width and height are adaptive, where the canonical image width-height aspect ratio $r_I = \frac{W_I}{H_I}$ is calculated according to the uv range:

$$r_I = \frac{u_{max} - u_{min}}{v_{max} - v_{min}}, \quad (S3)$$

where $u \in [-\pi, \pi]$ and $v \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. The shorter dimension, whether width or height, is set to 768.

Annealed positional and hash encoding. The projection offset employs Fourier positional encoding [13] or multi-resolution hash encoding [9] to capture high-frequency information. Given an input vector $\mathbf{x} \in \mathbb{R}^3$, the corresponding encoding can be defined as follows:

- The positional encoding is defined as $\gamma_{pe}(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times (1+2K)}$ to encode 3-dimensional vector \mathbf{x} up to K frequencies as $\gamma_{pe}(\mathbf{x}) = [\mathbf{x}, F_1(\mathbf{x}), \dots, F_K(\mathbf{x})]$. For the k -th frequency of positional encoding, we have the encoding function $F_k(\mathbf{x}) = [\sin(2^k \mathbf{x}), \cos(2^k \mathbf{x})] \in \mathbb{R}^{2 \times 3}$.
- The hash encoding is defined as $\gamma_h(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^{3+DK}$ to encode the vector \mathbf{x} by a K -resolution hash grid with D -dimensional feature per layer as $\gamma_h(\mathbf{x}) = [\mathbf{x}, H_1(\mathbf{x}), \dots, H_K(\mathbf{x})]$. For the k -th resolution hash grid with D -dimensional feature at each layer, we have the encoding function $H_k(\mathbf{x}) \in \mathbb{R}^D$.

Motivated by Nerfies [10], the positional or hash encoding can incorporate an optional annealing learning strategy. Specifically, we introduce a weight factor $w_k^n = \frac{1}{2}(1 - \cos(\alpha_k^n \pi))$ for some encoded frequency F_k^n or H_k^n at some training step n , such that we have $F_k^n(\cdot) = w_k^n F_k(\cdot)$ or $H_k^n(\cdot) = w_k^n H_k(\cdot)$ and

$$\alpha_k^n = \min(\max(\frac{n - N_s}{N_e - N_s} K - k, 0.0), 1.0), \quad (S4)$$

where N_s and N_e denote the start and end steps for anneal encoding, respectively. The strategy aims to facilitate the learning of low-frequency details and gradually incorporate high-frequency bands as the training progresses.

For all the experiments, the encoding γ_d of direction \mathbf{d} specifically employs positional encoding γ_{pe} , where we set $K = 4$ with the optional annealing learning strategy off. Concerning the encoding γ_p of position \mathbf{p}_{xyz} , we choose to use positional encoding γ_{pe} for PE models and hash encoding γ_h for hash models, where we set $K = 8$ with the annealed learning starting at training step $N_s = 4000$ and ending at $N_e = 8000$ for PE models, and we set $D = 2$ and $K = 16$ without the optional annealed learning strategy for hash models.

References

- [1] 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) [1](#), [2](#)
- [2] 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)
- [3] Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of ICCV (2023) [1](#)
- [4] 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) [1](#)
- [5] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of ICLR (2015) [1](#)
- [6] 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) [1](#)
- [7] 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) [1](#)
- [8] 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) [1](#), [2](#)
- [9] 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](#)
- [10] 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) [2](#)
- [11] 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) [1](#)
- [12] 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](#)
- [13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in NeurIPS. pp. 5998–6008 (2017) [2](#)