

# Supplementary Material: Seal-3D: Interactive Pixel-Level Editing for Neural Radiance Fields

## 1. Details of Editing Proxy Functions

### 1.1. Anchor Tool

The anchor tool stretches a control point  $\mathbf{x}^c$  along a translation vector  $\mathbf{t}$ , with its surrounding region. The translation function is defined as  $\text{stretch}(\cdot; \mathbf{x}^c, \mathbf{t})$ . We describe the mapping function for the anchor tool as follows in the main paper:

$$\mathbf{x}^s = \text{stretch}^{-1}(\mathbf{x}^t; \mathbf{x}^c, \mathbf{t})$$

$$F^m := (\mathbf{x}^t, \mathbf{d}^t) \mapsto (\mathbf{x}^s, \mathbf{d}^t)$$

As Fig. 1 shows, the mapping function  $\text{stretch}^{-1}$  is realized by projecting a point  $\mathbf{x}^t$  in the target conic space  $\mathcal{T}$  (blue) to the source space  $\mathcal{S}$  (green). The detailed steps are as follows.

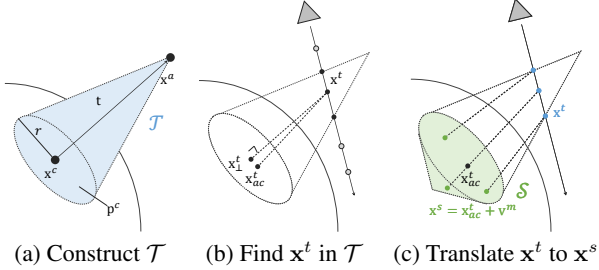


Figure 1: The details of the anchor tool’s mapping function.  $\mathcal{T}$  (blue) is mapped to  $\mathcal{S}$  (green) to make sure  $\mathbf{x}^t$  on the surface of the cone are mapped to the surface of the object, while  $\mathbf{x}^t$  inside the cone are mapped to the inner space of the object.

1. Construct the target conic space  $\mathcal{T}$ . The base of the cone is a circle with radius  $r$  defined by user input on a plane  $\mathbf{p}^c$  where the control point  $\mathbf{x}^c$  is on. We search the surface points  $\mathbf{x}^p$  near  $\mathbf{x}^c$  and fit the  $\mathbf{p}^c$  from  $\mathbf{x}^p$  using SVD decomposition and least-squares fitting. The point  $\mathbf{x}^a = \mathbf{x}^c + \mathbf{t}$  is the apex of the cone.
2. Find  $\mathbf{x}^t$  that are inside  $\mathcal{T}$ . Here we define the projected point from  $\mathbf{x}^t$  to  $\mathbf{p}^c$  as  $\mathbf{x}_{\perp}^t$ , the projected point from  $\mathbf{x}^a$  to  $\mathbf{p}^c$  as  $\mathbf{x}_{\perp}^a$ , and the projected point from  $\mathbf{x}^t$  to

$\mathbf{p}^c$  along direction of the vector  $\overrightarrow{\mathbf{x}^a \mathbf{x}^c}$  as  $\mathbf{x}_{ac}^t = \mathbf{x}^t + \frac{|\overrightarrow{\mathbf{x}^t \mathbf{x}_{\perp}^t}|}{|\overrightarrow{\mathbf{x}^a \mathbf{x}_{\perp}^a}|} \cdot \overrightarrow{\mathbf{x}^a \mathbf{x}^c}$ . The conditions are as follows:

$$\overrightarrow{\mathbf{x}_{\perp}^t \mathbf{x}^t} \cdot \overrightarrow{\mathbf{x}_{\perp}^a \mathbf{x}^a} > 0 \quad (1)$$

$$|\overrightarrow{\mathbf{x}^c \mathbf{x}_{ac}^t}| < r \quad (2)$$

$$|\overrightarrow{\mathbf{x}^t \mathbf{x}_{\perp}^t}| < |\overrightarrow{\mathbf{x}^a \mathbf{x}_{\perp}^a}| \quad (3)$$

$$\frac{|\overrightarrow{\mathbf{x}^t \mathbf{x}_{ac}^t}|}{|\overrightarrow{\mathbf{x}^c \mathbf{x}_{ac}^t}|} < \frac{|\overrightarrow{\mathbf{x}^a \mathbf{x}^c}|}{r} \quad (4)$$

3. Translate  $\mathbf{x}^t$  to  $\mathbf{x}^s$ . The mapping function is defined as follows, where  $w^m$  is the weight of the  $\mathbf{x}^a \mathbf{x}^c$  direction offset vector  $\mathbf{v}^m$ , which is designed to monitor the surface thickness. In our implementation, we set  $w^m$  to 0.1.

$$\mathbf{v}^m = -w^m \cdot \frac{|\overrightarrow{\mathbf{x}^a \mathbf{x}^c}| - |\overrightarrow{\mathbf{x}^t \mathbf{x}_{ac}^t}|}{|\overrightarrow{\mathbf{x}^a \mathbf{x}^c}|} \cdot \mathbf{x}^a \mathbf{x}^c$$

$$\text{stretch}^{-1}(\mathbf{x}^t; \mathbf{x}^c, \mathbf{t}) = \begin{cases} \mathbf{x}_{ac}^t + \mathbf{v}^m & , \text{if Eqs. (1) to (4)} \\ \mathbf{x}^t & , \text{otherwise} \end{cases}$$

### 1.2. Color Tool

The color tool transfers the color of the target space. To preserve shading details, we first convert the color in RGB space to HSL or HSV color space, modify the lightness (L of HSL) or value (V of HSV), and convert the modified color back to RGB space. Take the HSL color space as an example, we keep the value of hue (H) and saturation (S) from the modification color  $c^m$  and offset the lightness (L) of  $c^m$  using the lightness of the original color  $c^o$ . Here we define the function RGB2HSL as a conversion from RGB space to HSL space and the function HSL2RGB as a conversion from HSL space to RGB space. The mapped color  $c^t$  is computed as follows:

$$\begin{aligned}
[h^m, s^m, l^m] &= \text{RGB2HSL}(c^m) \\
[h^o, s^o, l^o] &= \text{RGB2HSL}(c^o) \\
c^t &= \text{HSL2RGB}([h^m, s^m, l^m + l^o - \text{mean}(l^o)]) \quad (5)
\end{aligned}$$

Color modification with HSV color space is similar to HSL, while the lightness (L) is replaced with the value (V).

## 2. Teacher Model Output v.s Local Pertaining in Fast Preview

Local pretraining is proposed for the fast preview. Using the teacher model output for preview could be a bottleneck of speed during preview, giving a rendering speed of only 1-2 FPS when the editing space is very large and the sampling-computing-mapping operations of the proxy function become extremely complicated, much slower than the NGP rendering speed at 20-30 FPS, making it not practical for interactive preview.

## 3. Additional Results

Additional results are shown in Figs. 2 to 5.

## References

- [1] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 2
- [2] 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. In *ECCV*, 2020. 2



Figure 2: Example of color and brush editing on the real-world scene: paint a pink heart on the truck (Tanks and Temples[1]).

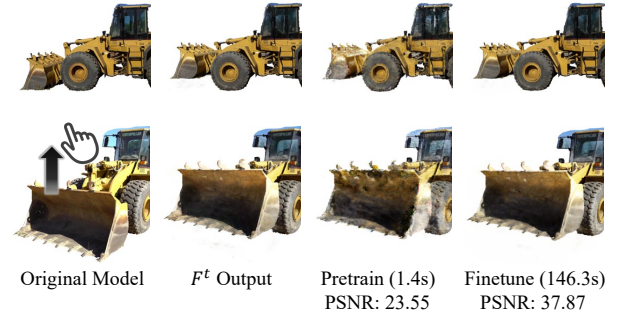


Figure 3: Example of bounding shape editing on the real-world scene: lift the bucket of forklift (Tanks and Temples[1]).

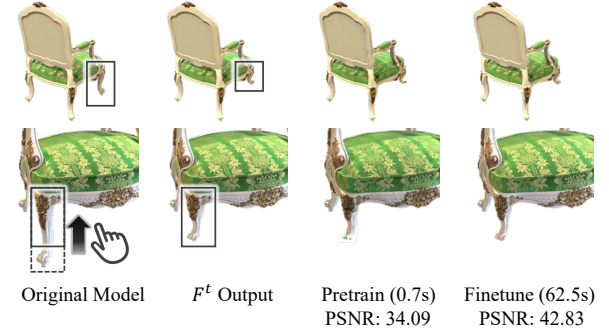


Figure 4: Example of bounding shape editing: shrink the chair leg (NeRF Blender[2]).

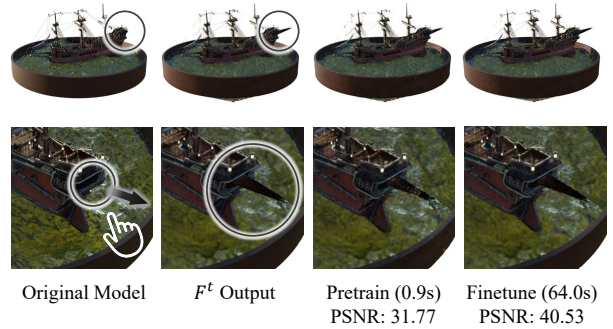


Figure 5: Example of anchor editing: fake bowsprit of the ship (NeRF Blender[2]).