Supplementary Material

8 Details in Stage I

8.1 Multi-View Consistent Initialization

We utilize the multi-view depth-conditioned albedo diffusion model from Rich-Dreamer [33] for the initialization. The rendered depth maps are concatenated with the latent features to serve as input for the UNet denoiser [34]. We only sample 8 images around the underlying 3D model instead of an overcomplete set, and we unproject all images to get an averaged RGB UV map as initialization. The invisible regions on the UV map are extracted by pixel detection and in-painted [47] by the region neighborhood. The inpainted UV map is then decoded to latent space and noised to the specified diffusion step. Although the sampled images have fewer details and are averaged in UV space, this is sufficient for initialization [28].

8.2 Defects in Latent UV Mapping

Figure 10 illustrates why details are dropped during the aggregation step when using latent UV map. Each pixel of the latent image represents a patch (e.g., 8×8) of the original image. Latent pixels warping means patches warping on RGB space, which inevitably leads to blurring or jagged lines after decoding.

Fig. 10: The left image is the original RGB image. The second column is the warping image caused by latent UV. The third column is the warping image by RGB UV. Using RGB UV warping, everything is in good order. Using latent UV warping, the straight horizontal line becomes a set of horizontal line segments, and the details are blurred.

9 Texture Stitching details

In this section, we provide details on the formulations of $\phi_f(z_f)$ and $\phi_{ff'}(z_f, z_{f'})$. **Formulation of $\phi_f(z_f)$.** With $\mathcal{C}_{f,i} = \{\boldsymbol{c}\}$ we denote all colors of pixels of image I_i associated with face f. Let $\mathcal{C}_f = \cup_{1 \le i \le k}$ collect all the colors of the pixels associated with the face f. We perform mean sift clustering among $\mathcal{C}_f$.

With $\boldsymbol{c}_f$ and σ_f we denote the center of the cluster and the associated variance, respectively. We then define

$$\phi_f(z_f) = \begin{cases} +\infty & \mathcal{C}_{f,z_f} = \emptyset \\ \frac{1}{\mathcal{C}_{f,z_f}} \sum\limits_{\boldsymbol{c}\in\mathcal{C}_{f,z_f}} \frac{\|\boldsymbol{c}-\boldsymbol{c}_f\|^2}{2\sigma_f^2} & \text{otherwise} \end{cases}. \tag{8}$$

Formulation of $\phi_{ff'}(z_f, z_{f'})$. Denote $\mathcal{P}_{f,i}$ as the set of pixels in I_i that belong to face f. If $\mathcal{P}_{f,i} \neq \emptyset$, we compute $\boldsymbol{f}_{f,i} = (\overline{\boldsymbol{c}}_{f,i}, \mu\overline{\boldsymbol{d}}_{f,i}^{\text{SIFT}}$, where $\overline{\boldsymbol{c}}_{f,i}$ is the average pixel color among $\mathcal{P}_{f,i}$ and $\overline{\boldsymbol{d}}_{f,i}^{\text{SIFT}}$) is the average SIFT pixel descriptor among $\mathcal{P}_{f,i}$. We set $\mu = 1$ in this paper. Note that instead of merely using color differences, we observe that incorporating SIFT helps place cuts among textureless regions.

Our definition of $\phi_{ff'}(z_f, z_{f'})$ differs from that of [49] in the sense that we take advantage of the consistency of the images, in contrast to simply using color differences. Specifically,

– When $\mathcal{P}_{f,z_f} = \emptyset$ or $\mathcal{P}_{f',z_{f'}} = \emptyset$,

$$\phi_{ff'}(z_f, z_{f'}) = +\infty. \tag{9}$$

– When $\mathcal{P}_{f,z_{f'}} = \emptyset$ and $\mathcal{P}_{f',z_f} = \emptyset$,

$$\phi_{ff'}(z_f, z_{f'}) = \|\boldsymbol{f}_{f,z_f} - \boldsymbol{f}_{f',z_{f'}}\|^2. \tag{10}$$

– When $\mathcal{P}_{f,z_{f'}} \neq \emptyset$ and $\mathcal{P}_{f',z_f} = \emptyset$,

$$\phi_{ff'}(z_f, z_{f'}) = \|\boldsymbol{f}_{f,z_f} - \boldsymbol{f}_{f,z_{f'}}\|^2. \tag{11}$$

– When $\mathcal{P}_{f,z_{f'}} = \emptyset$ and $\mathcal{P}_{f',z_f} \neq \emptyset$,

$$\phi_{ff'}(z_f, z_{f'}) = \|\boldsymbol{f}_{f,z_f} - \boldsymbol{f}_{f',z_f}\|^2. \tag{12}$$

– When $\mathcal{P}_{f,z_{f'}} \neq \emptyset$ and $\mathcal{P}_{f',z_f} \neq \emptyset$,

$$\phi_{ff'}(z_f, z_{f'}) = \frac{1}{2}\Big(\|\boldsymbol{f}_{f,z_f} - \boldsymbol{f}_{f',z_f}\|^2 + \|\boldsymbol{f}_{f,z_f} - \boldsymbol{f}_{f,z_{f'}}\|^2\Big). \tag{13}$$

Note that we only compute the color difference in Eq. (10) when z_f and $z_{f'}$ are not available in either f or f'. Otherwise, we employ the color consistency in Eqs. (11) to (13) which better reveals the appearance continuity after stitching.

10 Visualization of Ablation Study

Figure 11 is a visualization of ablation study. The green box in "No alternating optimization" indicates that the inconsistency issues were partly resolved during view selection and alignment stages.

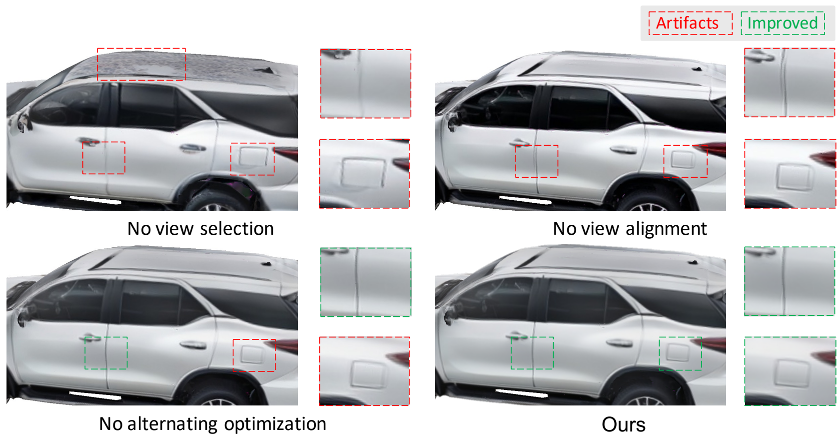

Fig. 11: Visualization of Ablation Study

11 Additional Results

11.1 Additional Baselines

The latest non-open source methods TexFusion [1] and Decorate3D [12] are compared with our method in Fig. 12 and Fig. 13, from where we see that our results are more natural in color and have clearer details.

Fig. 12: Qualitative comparison with TexFusion: clearer details.

Fig. 13: Qualitative comparison with Decorate3D: more natural in color.

11.2 More Visual Results

Figure 14 shows the results of different prompts on the same mesh. We present more visual results on 25 objects from different categories in GObjaverse dataset [33] in Figs. 15 to 17. Each textured mesh is rendered to 8 views. A full list of object indexes and prompts can be found in Tab. 2.

Fig. 14: Different prompts for the same mesh.

Fig. 15: Gallery of textured meshes.

Fig. 16: Gallery of textured meshes.

Fig. 17: Gallery of textured meshes.

Table 2: The object indexes and the corresponding text prompts.

Object Index	Prompt
0002c6eafa154e8bb08ebafb715a8d46	Birthday cake
000f88bb21164319ae797d315be6bc0e	Watch Model With Stand
0023717f4f564cc99f4ded70db04f590	Rusty kerosene lamp
0023b3edbc114be188ca9d8f729dfaaf	a vintage car
0025c5e2333949feb1db259d4ff08dbe	A wooden bird house
00286954e2d54db8bc7832cc8682b6ff	Furniture Bed
002e02c30121465c8a01bcb83b584ea5	A pair of blue jeans
003199cc6ff2410cb2d8e6f8a9cbb163	a Portugal Fire Hydrant
0032696f5871429fbd0549d9628f812c	A Christmas present
0033322379a24798a6875a5cb2de54f5	A raspberry
0046f208ef8d4988ba7bb9d297f29ec7	An old Dutch windmill building
004fb4d72f6c4e55a15b9025a868d1a3	a roast chicken
0056880681c044cb9fe815a9eed0425d	A rhino
00602ef508784e5384665aacaaf1f3a0	A handle saw
0064add4992b426cb2f862e5875ebf6d	A pink donut
0083fa5f10a442408e0f3f88df19c8ad	An ancient shield
0087dc01648d4cc792a7d1e49848b825	Stone horse head sculpture
00978a128283411582590096643ec101	A pale yellow seahorse
00b267b43669422cbb4ec3a4e9b1c16e	A pair of sunglasses
00d56831f9bc49f9a668f418c1af7558	A dancing brown dog in a white shirt, wearing dark sunglasses
010b9ece8a3a49e3b73be0b3cd02c720	A brown vase with red roses
011f2cd821e94596863378daa134cf0e	An Apple computer monitor
013c3a1d945a4336a87f889c3d4c25b1	Precision Sniper Rifle, CSGO AWP DRAGON LORE
015777939fc3429ba4b5343be9d51ffa	A kid's bike
017fe235577b4083ab32c2b7949ba022	A platinum ring studded with precious stones