

Supplementary Material

A. Implementation Details

A.1. Text Prompt of Stage1

In the first stage of texture generation, we use different text prompts when sampling the image corresponding to each view. To determine the text prompt for a specific view, we calculate the proportion of pixels rendered for each instance in the view relative to the total number of pixels. If this proportion exceeds a set threshold σ , we include the text corresponding to that instance in the global prompt. For example, taking “A Chinese style bedroom” as the holistic prompt, if the proportions of instances: “single bed”, “wardrobe” and “chair” exceed σ , the text prompt used for sampling this view becomes “A Chinese style bedroom with single bed, wardrobe and chair.” In our implementation, the threshold σ is set to 0.01.

A.2. Text Prompt of Stage2

During the texture generation phase for each instance, we derive the style prompt from the holistic prompt. For example, in “A Chinese style bedroom”, the style prompt is “Chinese”. When applying MVRS to a room without furniture, we use the same prompt for all viewpoints, such as “A Chinese style bedroom, without furniture.” For individual furniture instances, we use view-specific prompts. For example, for the instance “single bed”, the prompt would be “A Chinese-style single bed, [DIR] view” where [DIR] represents the relative position of the viewpoint. The [DIR] value changes based on the azimuth angle of the viewpoint and may include directions such as ‘front’, ‘front side’, ‘rear’, ‘side’, or ‘top-down’.

A.3. Time steps for MVIS

During the MVIS sampling process, we adopt different strategies based on the time step t . When $t \in \{T \dots 0.9T\}$, we use the standard diffusion sampling method. For $t \in \{0.9T \dots 0.5T\}$, the MVIS is applied. During $t \in \{0.5T \dots 0.3T\}$, we alternate between diffusion sampling and the MVIS. Finally, for $t \in \{0.3T \dots 0\}$, we revert to the standard diffusion sampling method. This multi-stage sampling strategy accelerates the overall sampling process while maintaining high texture quality.

A.4. Time steps for MVRS

Similar to the multi-stage sampling strategy in MVIS, we also adopt a multi-stage approach in MVRS to accelerate the process and improve texture quality. When $t \in \{T \dots 0.8T\}$, we use the standard RePaint sampling method without projecting images to texture space. For $t \in \{0.8T \dots 0.5T\}$, we apply the standard MVRS method. During $t \in \{0.5T \dots 0.3T\}$, we alternate between diffusion

| Method | PF \uparrow | VQ \uparrow | TG \uparrow |
|----------------|---------------|---------------|---------------|
| Text2Tex-H [6] | 3.06 | 2.72 | 2.43 |
| Text2Tex-C [6] | 3.22 | 3.10 | 3.15 |
| SceneTex [7] | 3.47 | 3.28 | 3.42 |
| Ours | 4.00 | 4.13 | 4.31 |

Table 2. **User study result.** We report the prompt fidelity (PF), visual quality (VQ) and texture-geometry alignment (TG) results for user study. We show that our method produces high quality textures that coherent with input text prompt.

sampling and the MVIS. Finally, for $t \in \{0.3T \dots 0\}$, we switch back to the standard diffusion sampling method.

A.5. Camera selection of both stage

Our two-phase camera configuration:

Global Phase: N cameras (N=6) are positioned to look at the center of the room, with a radius equal to half the shortest axis of the room.

Instance Phase: N cameras (N=12 for room-frame repainting, N=9 for furniture repainting) are positioned around each instance, looking at its center. The camera distance is set to 0.95 times of the diagonal length of the furniture’s bounding box when repainting furniture. For room-frame repainting, the same camera parameters as global phase are used, but with a field of view (FOV) of 80 degrees.

A.6. Parameters

Throughout all sampling processes, we set the inference steps to 50. The classifier-free guidance scale is linearly reduced from 10.0 to 7.0 over the sampling time steps. The ControlNet conditioning scale is fixed at 1. Additionally, the exp parameter in *dynamic_merge* is linearly increased from 1.0 to 6.0 over the sampling time steps. The T value is consistent with the default setting in Stable Diffusion, which is 1000.

B. A more intuitive explanation of the second stage.

The detailed implementation of MVRS is presented in Alg.2. After completing the first stage of texture generation for the entire room, an initial texture \mathcal{T}_{MVIS} is obtained. However, due to occlusion issues, the initial texture contains numerous untextured black regions. In MVRS, for the n-th viewpoint, the regions with an initial texture are denoted as P^n , and the corresponding rendered image is represented as \mathcal{I}_{MVIS}^n . To address the occlusion problem, we perform separate MVRS operations for each instance in the room. After completing the MVRS operations for all instances, we take the union of the texture maps corresponding to each instance to produce the final texture map

for the entire scene. This final texture represents the output generated by our proposed framework.

C. Ablation study on the scenond stage

As shown in the Fig.7, using only Stage 1 results in large black areas in the texture caused by occlusion, where textures cannot be generated. By incorporating the second stage, the occlusion issue is resolved while preserving the global style consistency established in the first stage, and the overall texture quality is significantly improved.

D. User Study Details

We conducted the user study using a web-based questionnaire system to compare our method with three baselines from the human perspective. Fig.6 illustrates the interface of our questionnaire system. From the generated results of our method and the baselines, we randomly selected 6 textured scenes created for the same indoor scene using the same textual prompt. In the interface, we first present the textual prompt used for texture generation, followed by rendered images of the textured scene from four different viewpoints. Participants were then asked to evaluate the generated textures across three dimensions: Prompt Fidelity(PF), Texture and Geometry Alignment(TG) and Visual Quality(VQ), with scores ranging from 1 (low) to 5 (high). In total, we collected 450 ratings from 25 participants and computed the average score for each method. The result is presented in Tab.2

E. More visual result

E.1. Different style for same room

We present more indoor scene texturing results of our method for one room from 3D-Front in Fig. 8.

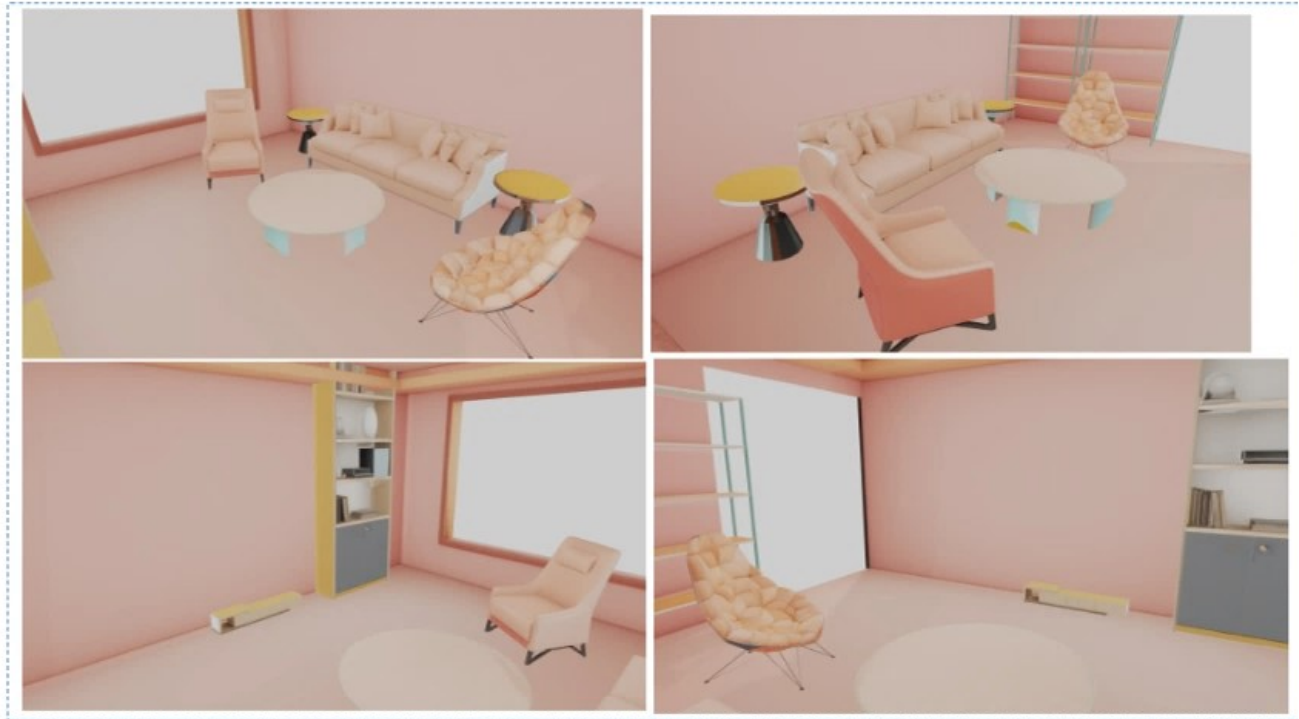
E.2. Visual result on ScanNet++ dataset

We are providing multi-view visual results of texturing scenes that from ScanNet++ dataset[47]. It is worth noting that ScanNet++ features more complex layouts compared to 3D-FRONT, as it is a real-world scan dataset. Visual results are shown in Fig.9

E.3. Example of the intermediate denoising process

During the MVIS process of generating multiple consistent images, intermediate results emerge from the multi-view sampling process, as shown in the Fig.10. During MVIS process, the $x_{0,t}$ images across different viewpoints at different steps are shown in Fig.10

* **01** A Memphis style livingroom



bad

good

The degree of alignment between generated textures and the textual description.

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

The degree of alignment between texture and geometry.

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

The visual quality of the generated texture.

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

Figure 6. **The interface of the questionnaire system used in user study.** We present 4 rendered views from 6 different texturing results to each participant and ask them to rate the scenes across three dimensions.

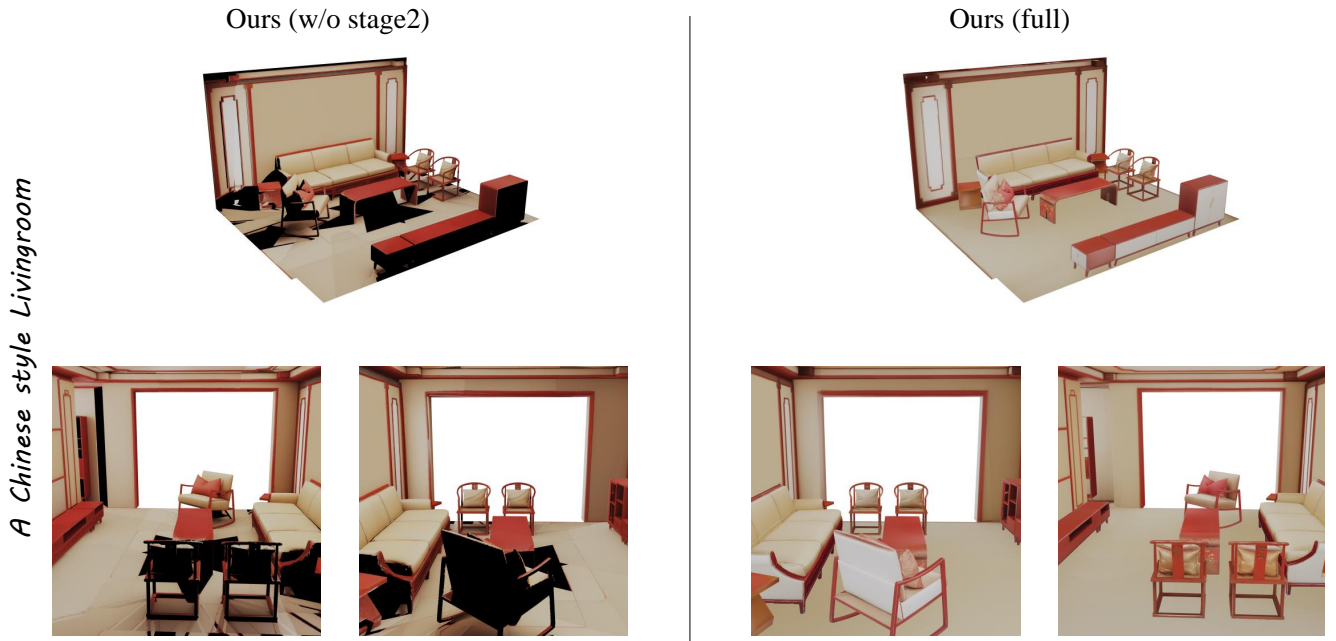
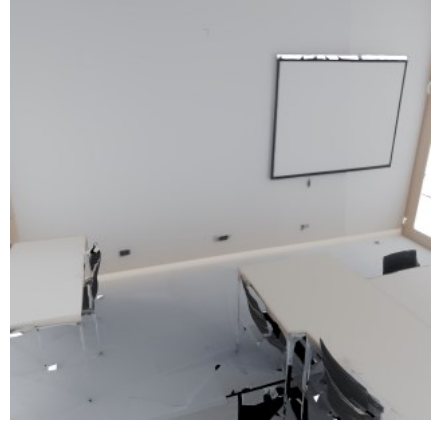


Figure 7. **Ablation studies on the stage2.** Using only Stage 1 results in large black areas in the texture caused by occlusion, where texture cannot be generated. By incorporating Stage 2, the occlusion issue is resolved while preserving the global style consistency established in Stage 1, and the overall texture quality is significantly improved.



Figure 8. **Synthesized texture for 3D-FRONT scenes** Our method generates various texture for the same input scene by utilizing the prompt template: 'a <STYLE> living room' with 6 distinct styles for texture creation.

A Modern style office

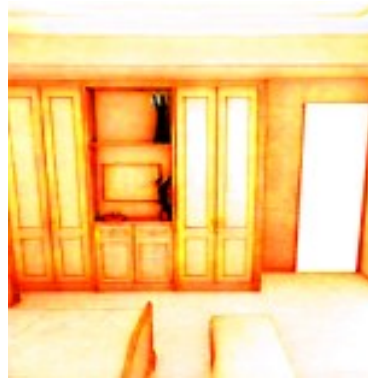
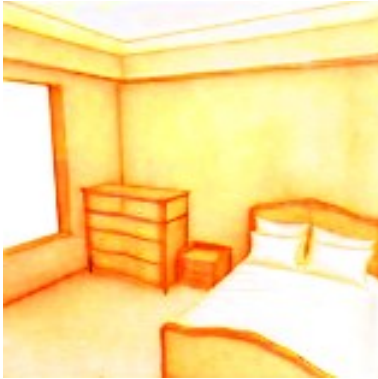


A Modern style computer room

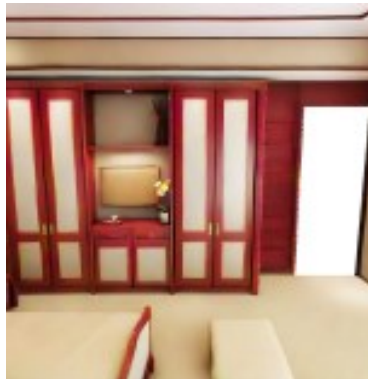


Figure 9. **Synthesized texture for ScanNet++ scenes** Our method generates similar style texture for different input scene by utilizing the prompt template: ‘a modern style <TYPE>’ with 2 distinct room types for texture creation.

t=980



t=480



t=0

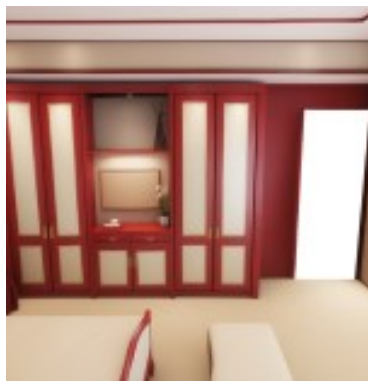
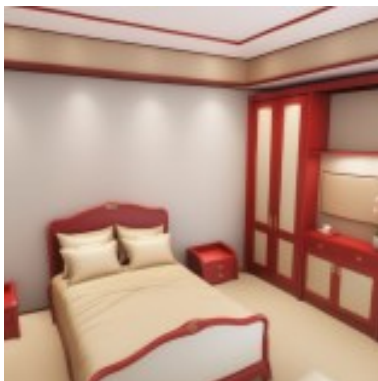
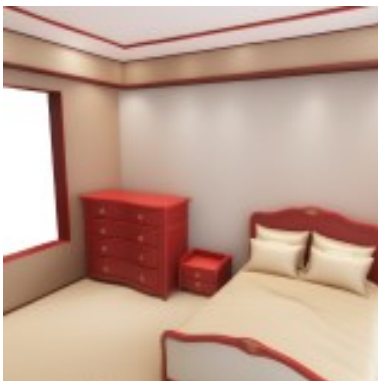


Figure 10. **Example of MVIS intermedia result** An example of the intermediate of Multi-View Integrated Sampling process, showcasing the images $x_{0,t}$ across different viewpoints at different steps.

Algorithm 2 Multi-view Integrated Repaint Sampling of One Instance

Input: Mesh \mathcal{M} , Text y , Cameras $\{C^1, \dots, C^N\}$, Textured mask $\{P^1, \dots, P^N\}$, Stage1 generated texture \mathcal{T}_{MVIS}

Parameters: DDPM noise schedule $\{\sigma_t\}_{t=T}^0$

Initialization:

$\mathcal{T} = 0$

$\{x_T^n\}_{n=1}^N \sim \{\mathcal{N}(0, I)\}$ # Unpainted noisy latents

$\{\mathcal{I}_{MVIS}^n \leftarrow \mathcal{R}(\mathcal{T}_{MVIS}, \mathcal{M}, c^n)\}_{n=1}^N$ # Painted area is colorful, unpainted area is black

$\{x_{MVIS}^n \leftarrow \mathcal{E}(\mathcal{I}_{MVIS}^n)\}_{n=1}^N$ # Painted noise-free latents

▷ **Early steps (conditioning on painted area):**

for $t \in \{T \dots T_{end1}\}$ **do**

for $n \in \{1 \dots N\}$ **do**

$\epsilon_n \sim \mathcal{N}(0, I)$

$\epsilon_t^n \leftarrow \epsilon_\theta(x_t^n, y, d_n, t)$

$x_{0,t}^n \leftarrow \frac{x_t^n - \sqrt{1 - \bar{\alpha}_t} \epsilon_t^n}{\sqrt{\bar{\alpha}_t}}$

$\mu_{t-1}^n \leftarrow \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} x_{0,t}^n + \frac{\sqrt{\bar{\alpha}_t(1 - \bar{\alpha}_{t-1})}}{1 - \bar{\alpha}_t} x_t^n$

$x_{t-1}^n \leftarrow \mu_{t-1}^n + \sigma_t \epsilon_n$ # Unpainted latent with t-1 level noise

$\tilde{x}_{t-1}^n \leftarrow \sqrt{\bar{\alpha}_t} x_{MVIS}^n + \sqrt{1 - \bar{\alpha}_t} \epsilon_n$ # Painted latent with t-1 level noise

$x_{t-1}^n \leftarrow \tilde{x}_{t-1}^n \odot P^n + x_{t-1}^n \odot (1 - P^n)$ # Mask combine

end for

end for

▷ **Middle steps (conditioning on painted area):**

for $t \in \{T_{end1} \dots T_{end2}\}$ **do**

for $n \in \{1 \dots N\}$ **do**

$\epsilon_n \sim \mathcal{N}(0, I)$

$\epsilon_t^n \leftarrow \epsilon_\theta(x_t^n, y, d_n, t)$

$x_{0,t}^n \leftarrow \frac{x_t^n - \sqrt{1 - \bar{\alpha}_t} \epsilon_t^n}{\sqrt{\bar{\alpha}_t}}$ # Predicted noise-free image at timestep t

$\mathcal{I}_t^n \leftarrow \mathcal{D}(x_{0,t}^n)$

$\mathcal{T}^n \leftarrow \mathcal{R}^{-1}(\mathcal{I}_t^n, \mathcal{M}, C^n)$ # Project view-specific noise-free images to view-specific texture maps

end for

$\mathcal{T} = \text{dynamic_merge}(\{\mathcal{T}^n\}_{n=1}^N)$ # Merge view-specific texture maps to a global-consistent texture

for $n \in \{1 \dots N\}$ **do**

$\tilde{x}_{0,t}^n \leftarrow \mathcal{E}(\mathcal{R}(\mathcal{T}, \mathcal{M}, C^n))$ # Global-consistent noise-free unpainted area latent at timestep t

$\mu_{t-1}^n \leftarrow \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} \tilde{x}_{0,t}^n + \frac{\sqrt{\bar{\alpha}_t(1 - \bar{\alpha}_{t-1})}}{1 - \bar{\alpha}_t} x_t^n$

$x_{t-1}^n \leftarrow \mu_{t-1}^n + \sigma_t \epsilon_n$ # Unpainted latent with t-1 level noise

$\tilde{x}_{t-1}^n \leftarrow \sqrt{\bar{\alpha}_t} x_{MVIS}^n + \sqrt{1 - \bar{\alpha}_t} \epsilon_n$ # Painted latent with t-1 level noise

$x_{t-1}^n \leftarrow \tilde{x}_{t-1}^n \odot P^n + x_{t-1}^n \odot (1 - P^n)$ # Mask combine

end for

end for

▷ **Latter steps (not conditioning on painted area):**

for $t \in \{T_{end2} \dots 0\}$ **do**

$\mathcal{T} \leftarrow \text{MVIS}(\mathcal{M}, y, \{c^n\}_{n=1}^N, \{x_t^n\}_{n=1}^N)$

end for

$\mathcal{T}_{MVIS} \leftarrow \mathcal{T}$

return Texture map \mathcal{T}_{MVIS}
