DreamSpace: Dreaming Your Room Space with Text-Driven Panoramic Texture Propagation

Bangbang Yang[1] Wenqi Dong[2] Lin Ma[1] Wenbo Hu[1]
Xiao Liu[1] Zhaopeng Cui[2] Yuewen Ma[1*]
[1]PICO, ByteDance [2]State Key Lab of CAD&CG, Zhejiang University

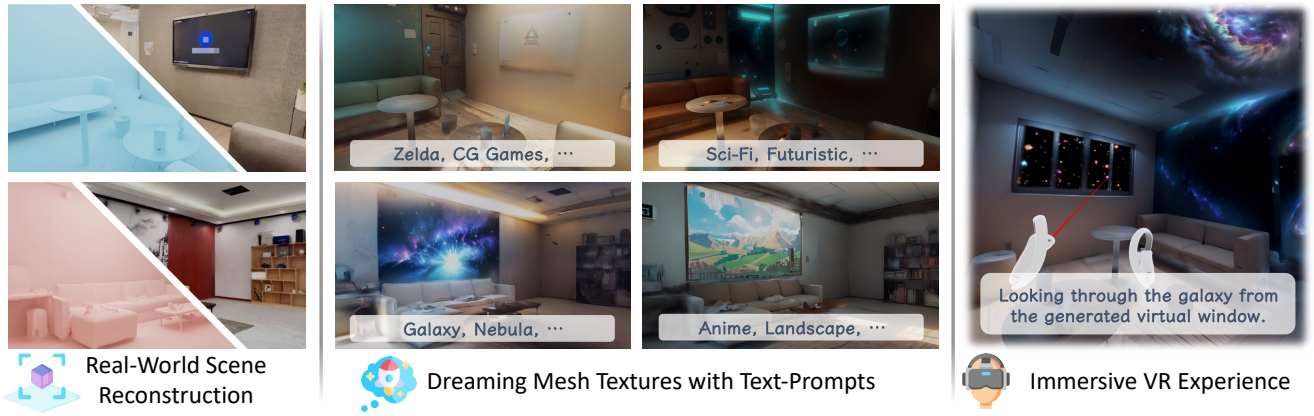


Figure 1. DreamSpace allows users to personalize their own spaces' appearances with text prompts and delivers immersive VR experiences on HMD devices. Specifically, given a real-world captured room, we generate enchanting and holistic mesh textures based on the user's textual inputs, while ensuring semantic consistency and spatial coherence (*e.g.*, the sofa still retain its recognizable form as a sofa, but in fantasy styles).

Abstract

Diffusion-based methods have achieved prominent success in generating 2D media. However, accomplishing similar proficiencies for scene-level mesh texturing in 3D spatial applications, e.g.*, XR/VR, remains constrained, primarily due to the intricate nature of 3D geometry and the necessity for immersive free-viewpoint rendering. In this paper, we propose a novel indoor scene texturing framework, which delivers text-driven texture generation with enchanting details and authentic spatial coherence. The key insight is to first imagine a stylized 360° panoramic texture from the central viewpoint of the scene, and then propagate it to the rest areas with inpainting and imitating techniques. To ensure meaningful and aligned textures to the scene, we develop a novel coarse-to-fine panoramic texture generation approach with dual texture alignment, which both considers the geometry and texture cues of the captured scenes. To survive from cluttered geometries during texture propagation, we design a separated strategy, which conducts texture inpainting in confidential regions and then learns an implicit imitating network to synthesize textures in occluded and tiny structural areas. Extensive experiments and the immersive VR application on real-world indoor scenes demonstrate the high quality of the generated textures and the engaging experience on VR headsets. Project webpage:* https://ybbbbt.com/publication/dreamspace.

1. Introduction

In our childhood, we might have imagined the world we live in with fantasy looking that follows real-world shapes but beyond reality, such as starry skies on the rooftops, beds with fancy adventurous decorations, or even virtual windows through which to gaze upon the galaxy. Nowadays, with the advancements of HMD devices, we have the ability to visually immerse ourselves in virtual scenes with 6-DoF rendering, which opens up the possibility of experiencing scene assets with various stylized textures. Consequently, a following question is: can we realize the dream of generating fully-immersive scenes with fantasy styles from reality,

*Corresponding author.

i.e., by giving text prompts, and automatically transferring textures of our living room with enchanting and meaningful details?

Over the past few years, enormous efforts have been paid in the field of scene stylization (or texture synthesis) [2, 5, 16, 18, 22, 40, 56]. However, existing methods either only transfer low-level styles without semantically meaningful textures (*e.g.*, imitating Van Gogh's paintings instead of generating recognizable visual elements [22,56]), or focus on texture editing [2, 18] on 3D objects with NeRF representation [31] but struggle to generate high-fidelity textures for the whole space and achieve real-time rendering on HMD devices. Very recently, with the advancements of diffusion-based generative methods (*e.g.*, Stable Diffusion [41]), it has become feasible to synthesize images based on text prompts with pleasant looking while maintaining the same scene structure by adding depth/edge conditions [33, 57]. Nevertheless, since perspective image views only convey a partial appearance of the entire 3D scene, it's non-trivial to automatically project it to 3D scene geometries. As a result, it usually requires skillful artists to run multiple generations and laboriously perform texture painting with 3D modeling software (*e.g.*, Dream-Texture for Blender Addon [25]).

In this paper, we propose a novel text-driven indoor scene texturing framework, which allows to generate meaningful and appealing mesh textures of real-world scenes based on text prompts, while preserving semantic consistency and spatial coherence (*e.g.*, the furniture still looks like its own types but in different fashions, as shown in Fig. 1). Unlike the object texturing task [5, 40] that synthesize textures from multiple perspective views towards the object, for scene-level tasks, we should consider the panoramic semantics and consistency in a unified process to ensure a seamless texturing result (see Sec. 4.2). To this end, we propose to texture scene meshes in a top-down manner, where we first generate an initial panoramic texture at the central viewpoint in a panoramic diffusion process and then propagate the panoramic texture to the rest of the regions. Meanwhile, both the initial and the propagated textures will be baked into the resulting meshes through UV maps, which can be uploaded into a commodity-level HMD device for immersive VR applications (see the supplementary video for more details).

However, it is nontrivial to design such a scene-level mesh texturing framework in a top-down panoramic manner, since there are several challenges when texturing on unstructured and cluttered real-world scenes. **1)** To display sharp and visually comfortable content on HMD devices, the desired panoramic texture should be high-resolution, free of tiling seams to avoid the sense of spatial fragmentation, and spatially coherent following equirectangular projection (*e.g.*, all the furniture and room structure such as floor and ceiling should be recognizable and not distorted).

To fulfill all the above demands, we employ a coarse-to-fine panoramic texture generation strategy, where we first generate a low-resolution panorama with a panoramic diffusion model to ensure proper panoramic scene structure, and then upscale it following equirectangular seam fixing to achieve seamless and high-resolution textures. **2)** Even with depth or edges as conditioning input [33, 57], existing diffusion models cannot ensure adequate alignment between geometry and textures, and such misalignment would inevitably introduce noticeable texture projection artifacts (see Sec. 4.4 and Fig. 10). To address this issue, we propose a novel dual texture alignment strategy, where the style-first textures and the alignment-first textures would be both generated and blended according to viewpoint depth changes. In this way, we effectively mitigate the geometry-texture misalignment while preserving visually appealing generated styles. **3)** Real-world reconstructed scenes often have intricate occlusions when observing from perspective views (*e.g.*, narrow spaces such as the gap between the wall-mounted TV and the wall, or floor areas under the sofa, or thin structures like plant leaves or legs of furniture), making it challenging for viewpoint-based texture painting to effectively cover every aspect of the scene. To this end, we design a holistic texture propagation pipeline. Specifically, for regions free of occlusion from the new viewpoint, we employ diffusion-based [41,57] confidential texture inpainting. Then, we leverage a coordinate-based implicit texture imitating network, which learns style mapping from real-world colors to stylized colors, and imitates textures for the rest of uncovered regions. By cooperating inpainting and imitating techniques, our method smoothly propagates initial panoramic textures to the whole space while preserving spatial coherence.

We summarize the technical contribution as follows. First, we propose a novel scene-level mesh texturing framework in a top-down panoramic manner, which allows users to generate engaging UV textures of real-world scene reconstructions based on text prompts. Second, we develop a coarse-to-fine texture generation strategy to ensure the correct perspective and high resolution, and a dual texture alignment mechanism to alleviate geometrical misalignment without compromising style quality. Moreover, to cope with the cluttered real-world geometries, we design a holistic texture propagation paradigm with inpainting and implicit imitating techniques, which smoothly paints the entire space with coherent textures. Finally, extensive experiments on real-world datasets demonstrate that our method achieves significantly better scene-level mesh texturing quality than existing methods, which also brings immersive and impressive VR experiences when visualized on HMD devices.

2. Related Works

Scene-Level Stylization. In the field of computer vision and graphics, neural network-based stylization has been studied for years. Starting from Gatys *et al.*'s work [14], early literature [8, 15, 24] mainly requires a style image as a reference, and optimize a perceptual loss or use a model to perform style transfer in 2D image domain [24, 26, 28, 52]. With the quick development of neural rendering techniques [31], such style transfer pipeline has soon be deployed into 3D space domain [6,7,11,23,56], which mainly inherit the perceptual loss paradigm to optimize the appearance of the view-dependent color field while freezing the density field. To obtain meaningful stylization results, recent works also use larger-scale external data-driven priors (*e.g.*, CLIP model [39]) for style transfer (or editing) [2,18], which achieves stylized results that also follow human language prompts, but these works mainly cannot be scaled to large indoor scenes that allow immersive room touring. However, during the rendering stage, NeRF-based methods typically require extensive computation due to network inference, which is not computational-friendly for all-in-one HMD devices. Hence, another line of works tries to directly stylize upon the scene meshes by hand-crafted annotation [12, 19] or upon the point cloud [4]. For example, Text2Scene [49] optimizes scene-level mesh textures with differentiable local fields to satisfy users' prompts, but requires structured CAD scenes, which is not applicable for real-world scene reconstructions. StyleMesh [22] proposes to operate neural style transfer on the parameterization of UV textures, which produces stylized mesh that can be feasibly rendered on standard graphics pipeline, but only transfer appearance up to global styles without strong semantic meaning (*e.g.*, mimicking artists' stroke), which cannot ensure sufficient visual comfort when displayed in HMD devices. Therefore, existing works for scene-level stylization either are not applicable for immersive indoor scene-scale scenarios with affordable computation on HMD devices [2,18], cannot support semantic meaningful style generation [6, 7, 11, 22, 23, 56], or require well-structured CAD model instead of real-world reconstruction [49].

Diffusion-based Mesh Texture Generation. Very recently, due to the emerging usage of large vision-language model in vision tasks, the generative methods [1, 10, 13, 20, 30, 32, 34, 42, 54] have gained tremendous develop in the past few months. Among them, diffusion-based generative models have attracted lots of attention in various modalities, such as high-resolution image generation [37, 41], human voice generation [27], or even 3D model generation [21,38]. Notably, the open-source of Stable Diffusion also sparks a trend of AI-assisted creation throughout the whole community, which also derives a lot of following modules upon its pre-trained weights, such as injecting various controlling conditions [33, 57], video generation [17], high-fidelity image inpainting [48] or even object texturing or mesh generations [5,29,40]. For example, Text2Room [21] uses Stable Diffusion to generate indoor 2D views, and lifted into 3D spaces with depth prediction and consecutive image inpainting, which enables to build up a novel indoor scene based on users' text prompts, but it struggles to produce clean textures or processes on a pre-captured scene reconstruction. Therefore, for the mesh texture generation task with given targeting meshes, there are mainly two different pathways. One is to use Score Distillation Sampling losses (SDS loss) from DreamFusion [38], which trains a generative NeRF by extracting supervisory signals from the denoising process of diffusion model upon the NeRF rendered views. Inspired by DreamFusion, LatentNeRF [29] proposes to use SDS loss to paint textures on the exact mesh with the unwrapped UV texture map. While the application of SDS over the mesh texturing task is technically plausible, it cannot unlock the full generative ability of the diffusion model, which results in much blurry rendering when compared with 2D domain image synthesis [40, 41, 45]. Hence, another possible route is to first generate 2D textures [5, 25, 40] that align with 3D geometry using depth-aware conditioning techniques [33,57], and then project it into UV textures. For example, the popular Blender addon Dream-Texture [25] uses customized geometry node to render depth from interactive modeling views, and then projects the textures through the view frustum. Nevertheless, since a single 2D viewpoint only reflects partial textures of a complete 3D model, Dream-Texture cannot correctly justify where to paint and simply projects textures through the entire mesh (*i.e.*, back face with the same textures as the front face), which results in incorrect textures when viewing from 360° viewpoints. To tackle the challenge of 2D-to-3D texturing ambiguity, TEXTure [40] and Text2Tex [5] propose to synthesize multi-view textures from orbiting viewpoints aiming at the object center, and use depth-aware texture inpainting to fill the new unpainted areas while preserving consistent texture from the partially painted area. However, such multi-view texturing pipeline assume the object can be fully observed without tiny / far-away structures or complex occlusions, which cannot be satisfied in real-world cluttered scenes. Therefore, recent work MVDiffusion proposes to leverage 3D correspondence in an attention mechanism during the multi-view diffusing process, which achieves multi-view consistency to a certain degree but still cannot achieve satisfactory mesh texturing results (see Sec. 4.2). Another concurrent work RoomDreamer tries to generate textures in cubemap format and also uses inpainting to fill the rest areas, but it still cannot ensure sufficient spatial coherence and also lacks proper ways to handle the unobserved regions (*e.g.*, gap between the desk and the floor). On the contrary, we propose to generate 360^{circ} textures in the panoramic space with a coarse-to-fine panoramic diffusion process,

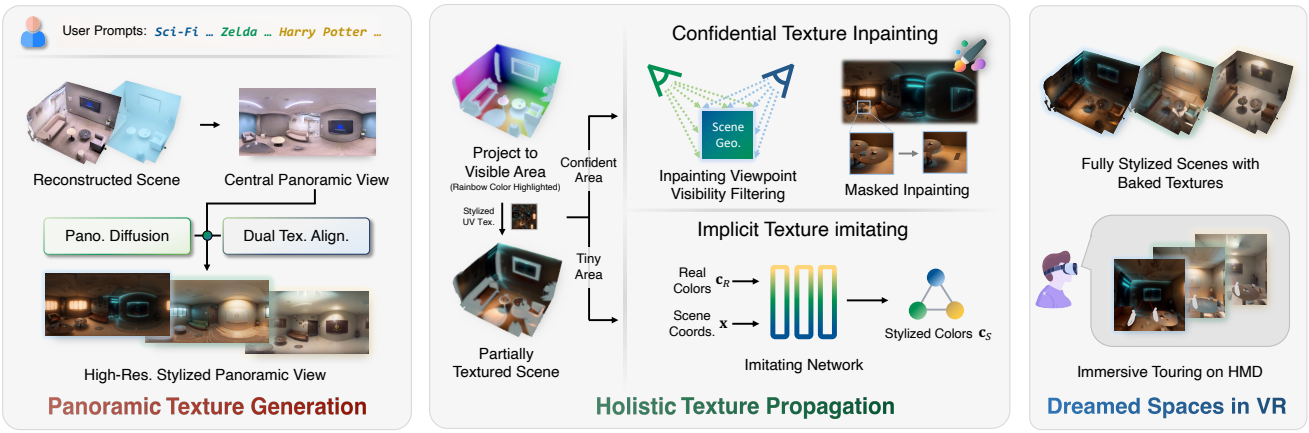


Figure 2. **Framework of DreamSpace.** Given a reconstructed real-world scene and users' text prompts, we first generate a high-resolution and geometrically aligned panoramic texture at the central viewpoint. Then, we propagate the textures into the rest regions with holistic texture propagation, where the confidential texture inpainting fills textures at the large confident areas and the implicit texture imitating predicts colors at the tiny areas. The resulting scene meshes with baked stylized UV textures can be uploaded into HMD devices for immersive VR touring.

and then propagate it into the rest region with inpainting and imitating, which both achieves texture synthesis with strong semantic meaning and takes into account the occlusion and tiny structures in real-world scene reconstruction.

3. Method

We introduce DreamSpace, a novel text-driven framework for generating semantically meaningful and spatial coherence scene textures for real-world indoor scenes. As demonstrated in Fig. 2, we texture the scene in the panoramic space with a top-down fashion, where we first generate a stylized 360° view from the central viewpoint, and then propagate it to the entire scene. To generate the high-resolution panoramic view with appropriate structure relationship and consistent semantic meaning, we design a coarse-to-fine panoramic texture generation process conditioned on reconstructed geometry and texture cues (Sec. 3.1), and a dual texture alignment strategy to alleviate texture misalignment to the geometry (Sec. 3.2). Once the initial stylized panoramic view is generated, we project textures to the visible area through UV maps, and then propagate it with confidential texture inpainting for visible areas at new viewpoints and implicit texture imitating for tiny areas, so as to obtain a fully stylized scene mesh. Note that our method does not rely on volumetric rendering with any geometry approximation [31]. Therefore, the baked resulting mesh is exactly what you see during the generation, and is compatible with standard rendering pipelines, which then can be easily uploaded and experienced in all-in-one HMD devices without PC streaming.

3.1. Panoramic Texture Generation

Generating in panoramic space. Different from previous object mesh texturing methods [5, 21, 40] that repeatedly generates multiple perspective views towards object centers, we urge that the scene-level texture generating task should consider the full 360° view of the scene as a whole, *i.e.*, generating in panoramic texture space (a.k.a. through equirectangular projection), rather than using multiple perspectives [5, 40] or cubemaps [45] with perplexing viewpoint specific prompts (*e.g.*, "floor/ceiling in a single color" when looking at the floor [21]). To this end, given a user prompts P and the reconstructed real-world scene (*i.e.*, a textured scene mesh), our first attempt is to generate a vivid and high-resolution stylized panoramic view that observes the scene from a central viewpoint. While it is plausible to use a depth-aware latent diffusion model (LDM) [33,41,57] to generate textures that fit to the observed scene depth, we find it still faced with several challenges. First, existing generic or LoRA-fine-tuned LDMs cannot ensure accurate equirectangular projection, which results in distorted texture when projecting back to the mesh. Second, the desired panoramic texture should be high-resolution (*e.g.*, 2K resolution or more) and free of tiling seams to guarantee acceptable visual quality in immersive VR applications, which is also not directly feasible for texture generation methods.

Coarse-to-fine conditioned generation. To handle the challenges above, we design a coarse-to-fine conditioned generation paradigm, where we first generate a low-resolution panoramic view with proper spatial structure, and then upscale it to the high resolution. Specifically, we first train a panoramic diffusion model by fine-tuning generic LDM [41] with carefully filtered equirectangular

projected images (see the supplementary material for more details). Next, for an input textured scene mesh, we render the panoramic colored image I_{P} with distance map D (*i.e.*, distance from camera center $\mathbf{c}$ to mesh surface) at the scene center, and feed them together with user's prompts P to the fine-tuned LDM with multi-condition controls [57] to obtain stylized image $\hat{I}_S$, as:

$$\hat{I}_S = F_{\text{c}}(P; D, \mathcal{E}(I_{\text{P}})) \tag{1}$$

where F_{c} is the LDM with multi-conditioning, $\mathcal{E}(I_{\text{P}})$ is the soft edgemap extracted with Su *et al.*'s work [47]. During the inference, we adapt the asymmetric tiling strategy [51] by hijacking all the 2D convolutions of the UNet with horizontal circular padding for the last 60% timestamps, so as to make sure the left and right side of the equirectangular image can be continuous (*e.g.*, maintaining the wall and the furniture to keep the same tone and continuous patterns on both sides). Then, we utilize tiled diffusion [3] with a generic LDM to upscale the $\hat{I}_S$ into $\hat{I}_{SL}$, which produces 3 times larger panoramic images with extra rich details.

Equirectangular seam fixing. During the upscaling stage, we find that the tiled upscaling strategy would inevitably break the equirectangular traits of the images (*i.e.*, patterns become no longer tiling along the horizontal direction, and the top and lower part of the panoramic are not the correct stretching follows equirectangular projection), primary due to the reason that each processed tile is agnostic to the whole perspective knowledge. Therefore, we also conduct inpainting on the top/down polar and left-right tiling side of the image. Specifically, for the top/down polar, we unwrap the panorama to the upward and downward perspective view and inpaint the central disk area, and then warp it back. For the horizontal tiling seam, we roll the half image along the x-axis and inpaint the middle part that covers both left-right sides of the panorama. So far, we can obtain a high-resolution stylized panoramic image that satisfies equirectangular projection and also maintains semantic coherence.

3.2. Dual Texture Alignment

Dilemma of stylization and alignment. Although using depth or hedges as conditional control can effectively direct the LDM to produce somewhat consistent textures to the target mesh [25, 33, 57], we find that in scene-level texturing tasks, such alignment is not sufficient since the geometry of the real-world scenes is generally much more complicated than single objects. One plausible workaround might be directly denoising with moderate or small noises upon real image views (a.k.a. LDM's image-to-image mode with lower denoising strength). However, due to the incomplete denoising process, such a method would generally result in blurry images or unsatisfactory styles. Therefore, we are faced with a dilemma that the visually appealing viewpoint stylization and perfect geometric alignment cannot be achieved together at one time.

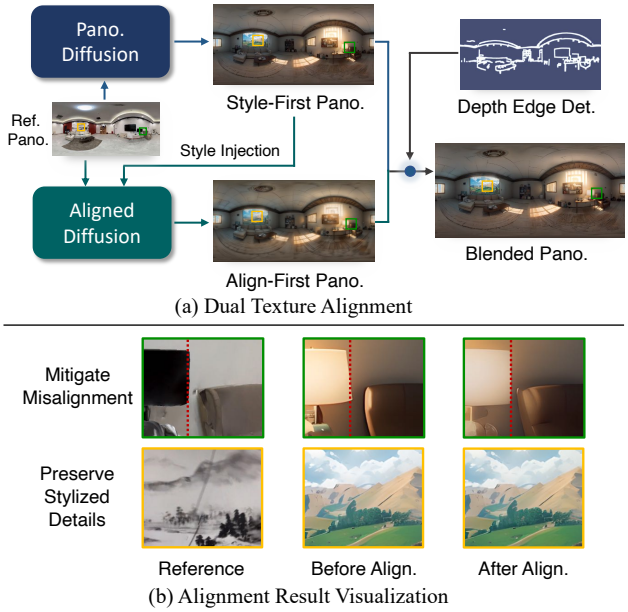


Figure 3. **Overview of dual texture alignment.** To mitigate geometry-texture misalignment, we first synthesize style-first panorama and align-first panorama, and then blend these dual textures according to depth edge detection, which brings aligned panoramic textures while preserving visually appealing stylized details.

Alignment with dual texture blending. To solve the dilemma, we propose to break the stylized panoramic texture generation in a dual process, and then fuse the dual textures in a geometry-aware manner, as demonstrated in Fig. 3. For brevity, we named these dual textures style-first panorama and align-first panorama (see the middle part of Fig. 3 (a)), where the first one is synthesized in a way as introduced in Sec. 3.1 which ensures high-quality styles, and the second one is synthesized with a customized aligned diffusion process that tends to align the original scene more strictly while maintaining a similar style. Specifically, for generating align-first panorama $\hat{I}_A$, we start by denoising on the real-world reference panorama but utilize multi-control techniques [57] , as:

$$\hat{I}_A = F_{\text{c}}(P; \mathcal{C}(I_{\text{P}}), \mathcal{T}(I_S)), \tag{2}$$

where $\mathcal{C}(I_{\text{P}})$ is the canny edge control that enforces alignment, and $\mathcal{T}(I_S)$ is the tile control [57] that injects styles from the style-first panorama. To make the same size as $\hat{I}_{SL}$, we upscale the $\hat{I}_A$ into $\hat{I}_{AL}$ with Wang's work [53], which empirically would not introduce noticeable tiling seams. Note that we do not need this panorama to be perfectly stylized (which in practice is noticeably blurry than the style-first one, as shown in Fig. 3 (a)). Then, we deter-

mine the pixel areas for blending the align-first panorama with the style-first panorama. We observe that the misalignment issue generally happens where the scene depth changes evidently. Hence, we simply generate the blending mask by detecting depth edges from the panoramic depth map following the dilation and blurring operations, and then blend these dual textures with masked Poisson image editing [36] (a.k.a. seamless cloning with the align-first panorama as the source and style-first panorama as the target). In this way, we can successfully mitigate the geometry-texture misalignment while maintaining the desired stylized details untouched (see Fig. 3(b), where the edge of the black monitor and sofa are much better aligned, while the stylized posters on the wall keep unchanged).

3.3. Holistic Texture Propagation

Panoramic texture projection through UV maps. Once the initial stylized panoramic view is synthesized, we project it to the visible areas through UV maps in the panoramic space, as illustrated in Fig. 2 (the left column of the holistic texture projection). In practice, we first obtain scene coordinates $\mathbf{x}$ (3D position) for valid pixels $\mathbf{p}$ in the corresponding UV map, as:

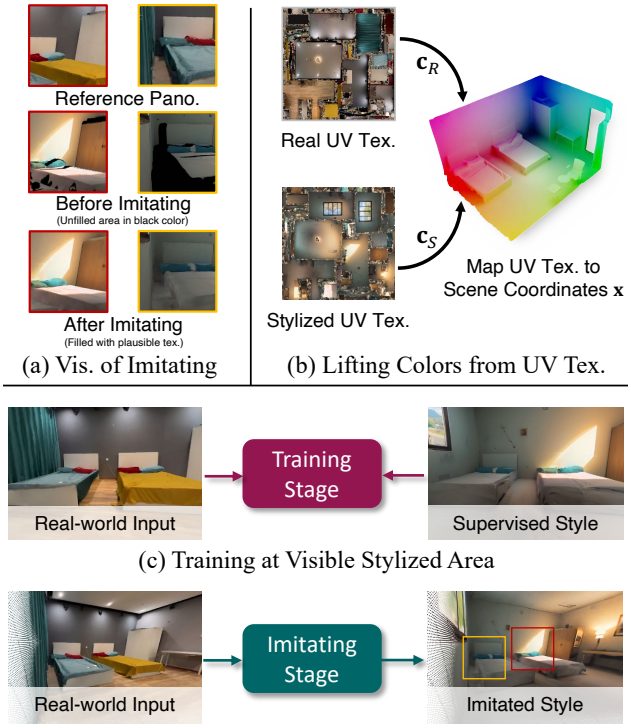
$$\mathbf{x} = \text{Interp}(\text{MapTex}(\text{TexCoord}(\mathbf{p}), \{T\})), \tag{3}$$

where $\text{TexCoord}(\mathbf{p})$ is the texture coordinate of each $\mathbf{p}$, $\{T\}$ is the mesh triangles, $\text{MapTex}(\cdot)$ maps the texture coordinate into triangle vertices with barycentric weights, and each $\mathbf{x}$ is barycentric interpolated from the triangles' vertices. Next, for each $\mathbf{x}$, we compute ray directions from the observing camera center $\mathbf{c}$, and map the direction $\mathbf{d} = \mathbf{c} - \mathbf{x}/\|\mathbf{c} - \mathbf{x}\|$ to the panoramic space through equirectangular projection. Then, for each $\mathbf{x}$, we compare its observing distance to the rendered scene depth and determine if the corresponding UV pixel $\mathbf{p}$ is visible from the viewpoint with a distance threshold $\epsilon = 0.01$. We go through all the UV pixels with the visibility test and form an initial visibility mask $M_{\text{init\_vis}}$ on the UV space, as:

$$M_{\text{init\_vis}}(\mathbf{p}) = \begin{cases} 1, & \text{if } \|\mathbf{p} - \mathbf{x}\| < \epsilon \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Finally, we assign stylized panoramic colors to the UV spaces according to the initial visibility mask $M_{\text{init\_vis}}$ and corresponding ray directions $\mathbf{d}$, which produces the partially textured scene (see the middle part of Fig. 2).

Separated strategies for confidential and tiny areas. By projecting initial textures to the scene, the main impression of the styled space has been already shaped, while there are still some uncovered areas that need to be filled (*e.g.*, the gray region at the partially textured mesh in Fig. 2). Previous methods that use LDM for object mesh texturing [5, 40] mainly rely on inpainting with various area selection and masking methods (*e.g.*, maintaining a trimap by



(a) Vis. of Imitating (b) Lifting Colors from UV Tex.

(c) Training at Visible Stylized Area

(d) Imitating at Unseen Area

Figure 4. **Overview of implicit texture imitating.** We first lift colors from UV textures according to UV pixels' scene coordinates. Then, during the training stage, we train an implicit texture imitating network from visible stylized areas using lifted real-world/stylized colors and coordinates. During the imitating stage, we feed the real-world color and coordinates into the network to imitate plausible textures in unseen areas.

TEXTure [40]), which aim to cover the entire mesh surface as complete as possible. However, for real-world scene texturing with cluttered geometries, solely relies on automatic inpainting cannot ensure proper texturing for thin structures (*e.g.*, leaves and furniture legs) or severely occluded areas (*e.g.*, floor under the sofa or gaps between wall-mounted TV and the wall) that cannot be observed from normal camera positions. Besides, duplicated inpainting on the same area of the mesh surface would also result in blurry appearance or artifacts due to the inconsistency nature of LDM's inpainting result (as demonstrated in Sec. 4.2). Therefore, we propose separate strategies for areas with different visibility. Instead of conducting inpainting multiple times, we only inpaint at the confidential areas (*i.e.*, areas that is definitely free of occlusion) in very few viewpoints (*e.g.*, only two in our experiments) and then leverage a novel implicit texture imitating network to smoothly fill the rest of areas with plausible appearance.

Confidential Texture Inpainting. Given a partially textured mesh, we first perform confidential texture inpainting

in the panoramic space as demonstrated in the middle part of Fig. 2. During this procedure, we do not aim to fill every aspect of the space, but only cover the confidential areas that are totally free of occlusion when observing from new viewpoints, where the viewpoint can be selected by SfM poses with farthest point sampling or interactive user selection. To begin with, for each viewpoint, we first determine the panoramic inpainting mask M_{inp} from the new camera poses. Practically, we reuse the UV-space initial visibility mask $M_{\text{init\_vis}}$ by regarding it as the UV texture, and render the panoramic image on the current viewpoint, and then perform dilation and blurring to the image to obtain the M_{inp}. We then leverage depth-aware inpainting LDM [41,57] F_{inp} to synthesis masked areas, as:

$$\hat{I}_{\text{inp}} = F_{\text{inp}}(P, \hat{I}_{\text{M}}; D, M_{\text{inp}}), \tag{5}$$

where $\hat{I}_{\text{M}}$ is the rendered panoramic image with partially textured mesh, $\hat{I}_{\text{inp}}$ is the inpainting output image. Note that the inpainting results $\hat{I}_{\text{inp}}$ will not be fully projected into the stylized UV texture, but only retain confidential areas by UV space masked filtering. More specifically, we design three UV-space mask filters that ensure a confidential texture projection. First, we filter inpainting areas with abrupt depth changes using a depth edge filtering mask $M_{\text{dep\_edge}}$, which can be constructed by assigning the UV mask with panoramic depth edge detection as introduced in Sec. 3.2. Second, we consider the surface normal and distances by rejecting small grazing viewing angles (10°) or too far surface points (distance larger than 2.5 meters) to form a safe viewing mask $M_{\text{safe\_view}}$, which is constructed by calculating barycentric interpolated normal vectors from vertex normal for each valid UV pixel along with the scene coordinates. Third, we perform visibility test on the inpainting views with a similar formulation as Eq. (4), which constructs the inpainting visibility mask $M_{\text{inp\_vis}}$. We combine all the above masks to achieve a confidential texture projecting areas in UV space, as:

$$M_{\text{conf}} = M_{\text{dep\_edge}} \cap M_{\text{safe\_view}} \cap M_{\text{inp\_vis}}, \tag{6}$$

where M_{conf} is the combined confidential mask. Note that all the masks are constructed in UV space instead of a certain camera perspective or panoramic view, which avoids the influence of viewpoint-specific occlusion. We assign inpainting panoramic texture into the stylized UV texture with the mask M_{conf}, which further fills the partially stylized scenes with more textures.

Implicit Texture Imitating. To complement the unobserved or unpainted areas for scene-level mesh texturing, we design a novel implicit texture imitating mechanism. As demonstrated in Fig. 4, the goal of the texture imitating is to learn the style mapping from the partially stylized scenes, and then smoothly predict plausible texture for unseen areas. In practice, we first lift real-world colors $\mathbf{C}_R$ and stylized colors $\mathbf{C}_S$ from the corresponding UV textures into the scene coordinates $\mathbf{x}$ (see Eq. (3) and Fig. 4 (b)). During the training stage (see Fig. 4 (c)), we learn an implicit imitating network F_{Imit} (*i.e.*, a coordinate-based MLP), which gives the input as scene coordinate $\mathbf{x}$ and real-world colors $\mathbf{C}_R$ from the partially textured scenes, and is supervised by existing visible stylized colors $\mathbf{C}_S$ with L2 loss, as:

$$\mathcal{L}_{\text{imit}} = \|\hat{\mathbf{C}}_S - \mathbf{C}_S\|_2 \text{ , where } \mathbf{C}_S = F_{\text{Imit}}(\gamma(\mathbf{x}), \mathbf{C}_R), \tag{7}$$

where $\gamma(\cdot)$ is the positional encoding [31], and $\hat{\mathbf{C}}_S$ is the predicted imitating color. Then, during the imitating stage (see Fig. 4 (d)), we feed the network with all the valid UV pixels' scene coordinates $\mathbf{x}$ and real-world colors $\mathbf{C}_R$ to predict the imitated colors $\hat{\mathbf{C}}_S$. As visualized in Fig. 4 (a), the uncovered areas in the stylized scene can be smoothly filled after the imitating while also preserving spatial coherence (*e.g.*, the pillows and the bedsheet are faithfully predicted as blue and white textures). Finally, we fuse the imitated colors into the partially textured meshes through the accumulated visibility mask M_{accu} (by combing $M_{\text{init\_vis}}$ and all the $M_{\text{inp\_vis}}$), which produces the fully stylized scenes with baked textures, as demonstrated in the right part of Fig. 2.

4. Experiments

In this section, we first compare our framework with existing methods on the generative scene-level mesh texturing task (Sec. 4.2) on real-world indoor scene datasets. Next, we analyze the necessity of panoramic space texture synthesis by comparing it with the cubemap space (Sec. 4.3). Then, we perform ablation studies on the design of our texturing framework (Sec. 4.4). Finally, we build up an immersive VR application by uploading fully textured scenes into the HMD devices (Sec. 4.5).

4.1. Datasets

DreamSpot Dataset. To demonstrate the applicability in real-world indoor scenes, we create a new dataset named DreamSpot, which contains three scenes that cover several typical scenarios in daily lives (*i.e.*, meeting room, living room, and bedroom, where the first two are used for comparison). Specifically, we use an iPhone to capture RGB images of the room and then use out-of-box SfM [43] with MonoSDF [55] for geometric reconstruction, and utilize texture mapping [35] to obtain scene meshes with real-world UV textures.

Replica Dataset. We also use three real-world scenes from the Replica dataset [46] to evaluate our method, *i.e.*, Room 0, Room 1, and Office 0. Since the original Replica dataset uses a customized shader for HDR rendering, which is not directly compatible with textured mesh-based pipelines

Figure 5. We compare our scene-level mesh texturing with StyleMesh [22], MVDiffusion [50] and TEXTure [40] on our captured DreamSpot dataset, where the figures include the overview of textured scene meshes and the corresponding rendered views.

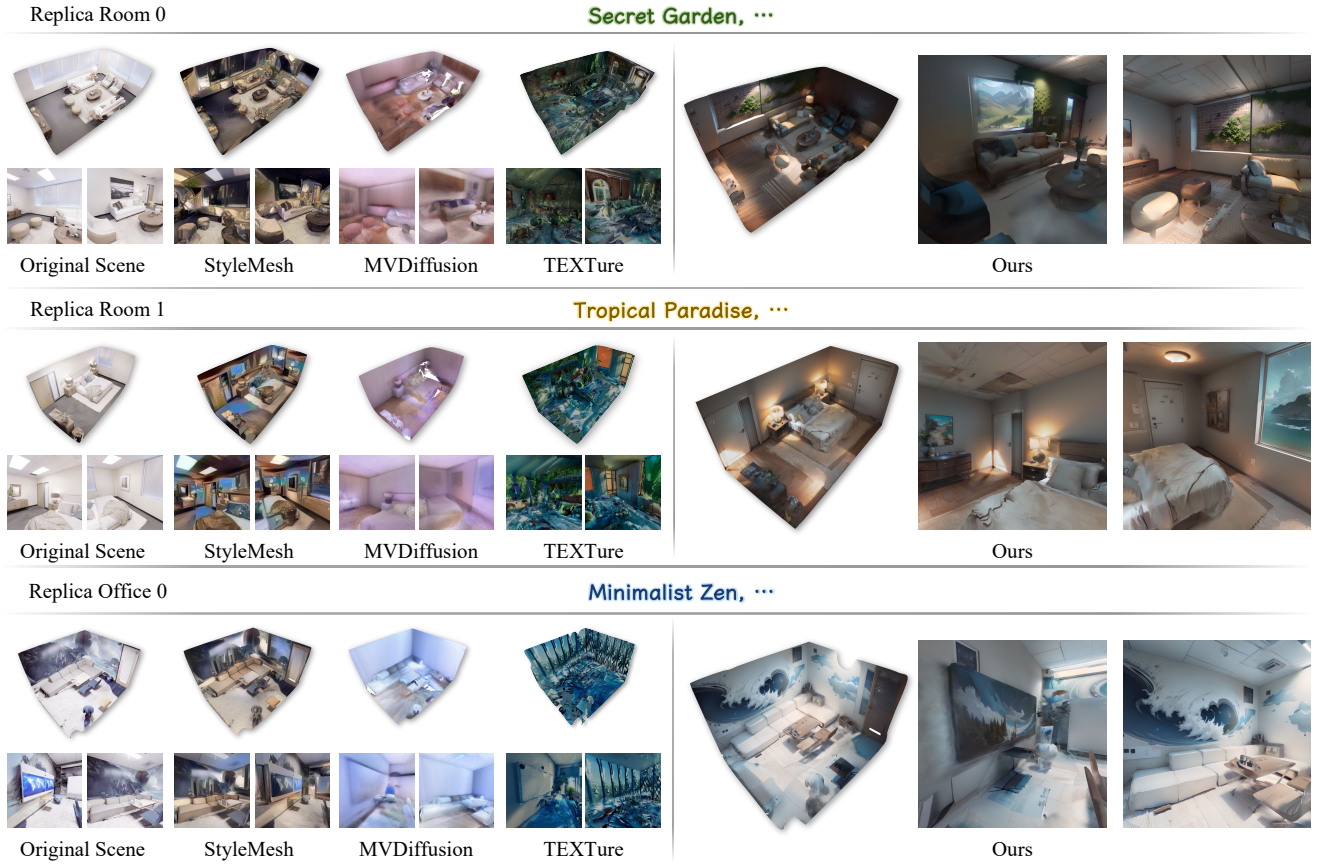


Figure 6. We compare our scene-level mesh texturing with StyleMesh [22], MVDiffusion [50] and TEXTure [40] on the Replica dataset, where the figures include the overview of textured scene meshes and the corresponding rendered views.

such as our method and StyleMesh [22]. Hence, we first pre-process these scenes by baking the appearance into unwrapped UV textures with Blender.

4.2. Comparison on Generative Mesh Texturing

Experiment setting. We first evaluate our method by comparing it with SOTA mesh texturing (or stylization) works

Methods	Quantitative Metrics		User Study	
	CLIP Score ↑	Aesthetic ↑	Correctness ↑	Quality ↑
StyleMesh [22]	0.184	4.812	2.68	2.76
MVDiffusion [50]	0.174	4.263	1.37	1.49
TEXTure [40]	0.187	5.265	2.57	2.20
Ours	**0.214**	**5.771**	**3.38**	**3.55**

Table 1. We perform quantitative evaluation and user studies on the rendered views of textured mesh for StyleMesh [22], MVDiffusion [50], TEXTure [40] and our method.

on the scene-level meshes both quantitatively and qualitatively. Specifically, given a reconstructed textured scene mesh and user-defined text prompts (*e.g.*, "galaxy themes", or "secret garden"), our task is to synthesize textures that fit the scene geometry while following the semantic meaning of the prompts. We choose the UV texture stylization method (StyleMesh [22]), multi-view consistent 2D diffusion model (MVDiffusion [50]), and LDM-based depth-aware mesh texturing method (TEXTure [40]) as competitors. Note that not all methods can directly process on meshes or leverage existing textures, *i.e.*, StyleMesh and our method use real-world textures and geometry as input, while TEXTure and MVDiffusion can only use pure geometry or 3D correspondence as guidance, and MVDiffusion also uses TSDF fusion to fuse generated images into colored meshes. For StyleMesh, since it uses perceptual loss for style transfer and requires a reference style image, we additionally use LDM [41] with text prompts to generate a style image as its input. During the texturing process, all the other methods perform optimization or generation in perspective views, while our method uses panoramic views. Therefore, to make a fair comparison, we manually designed a perspective camera scanning trajectory for each scene with the best effort to cover the whole space while avoiding being too close to the mesh surface. Once the mesh texturing is finished, we render the textured mesh into multiple perspective views with OpenGL, which will be used for metric comparisons and user study.

Quantitative comparison. For quantitative comparison, we use CLIP Score [39] to measure the matching degree between rendered views and the given text prompts. Besides, we also use aesthetic scoring introduced by LAION [44] to measure the aesthetic quality of the generated images, since it has been proven to be more authentic than FID for recent diffusion-based generative methods [37]. As presented in Fig. 1, our method consistently achieves the highest scores in both metrics, which demonstrates that our synthesized texture follows the given text prompts better and also maintains high quality when rendered from perspective views.

Qualitative comparison. We visualize the qualitative comparison results in Fig. 5 and Fig. 6, where we both exhibit the overview of the fully textured meshes and

(a) Scene Reconstruction of DreamSpot Bedroom

(b) Texture Synthesis in Different Spaces

(c) Rendered View of Stylized Meshes

Figure 7. We compare mesh texturing with textures generated from different spaces (*i.e.*, panoramic texture or cubemap texture).

the corresponding perspective mesh rendering views. For StyleMesh, since it utilizes VGG perceptual loss [24] for UV texture style transfer without high-level semantic priors such as CLIP [39], it generally cannot synthesize novel and meaningful textures and behaves more like mimicking strokes and color tones of the given style image. For example, in the "galaxy theme" of the meeting room (see Fig. 5), StyleMesh mainly turns the environment into dark galaxy tones while failing to generate rich galaxy textures. For MVDiffusion, though it leverages corresponding attention module to preserve multi-view consistency by extracting 3D correspondence from camera poses and scene depths, we find the resulting synthesized images cannot fulfill the requirement of scene-level texturing task due to the insufficient consistency, which results in blurry appearance in most of the cases (*e.g.*, for both cases in Fig. 5, the boundary of stylized television is much blurrier than ours). For TEXTure, because its repetitive inpainting strategy is mainly designed for object meshes, we find it struggles to generate satisfactory textures when conducting on scene-

Figure 8. We perform ablation studies of the coarse-to-fine strategy during the panoramic texture generation, including the coarse-to-fine upscaling and equirectangular seam fixing.

level meshes (*e.g.*, in Fig. 6 Replica Office 0, it produces repetitive artifacts on the walls) and also fails to project textures into scenes with cluttered geometry (*e.g.*, pieces of unpainted areas in Fig. 6 Reolica Room 0). To avoid potential visual discomfort, we have slightly dimmed the results of TEXTure in Fig. 5 and Fig. 6. From the analysis above, we believe that relying on perspective view for generating indoor scene textures is fairly difficult to obtain spatial coherent and consistent results, and also struggles to cover every visible area of real-world complex scenes. By contrast, our method uses panoramic scene texturing, which not only preserves semantic meaning (*e.g.*, furniture still looks like furniture, but in fantasy styles, and the generated floor texture is free of excessive details or severe artifacts), but also creates novel and enchanting textures by faithfully projecting generated textures into the meshes (*e.g.*, galaxy on the floor in Fig. 5 meeting room galaxy theme, vibrant grass decorations in Fig. 6 Room 0 "secret garden", and the impressive landscape poster in Fig. 6 Room 1 "tropical paradise"), while also properly fills unseen spaces (*e.g.*, areas under the chair in Fig. 6 Office 0 "minimalist zen") thanks to texture propagating techniques.

User study. We also conducted a user study to compare our method with others on the generated mesh textures of the DreamSpot and Replica datasets. Specifically, we ask 20 users to sort the rendered views from textured meshes generated by methods in two aspects, *i.e.*, image-text matching correctness and the perceptual quality, and assign the scores by their ranking (*i.e.*, with a score of 4 for the ordered best one and a score of 1 for the last one). As reported in Fig. 1, we achieve the most preferences among all the methods by a large margin, which highlights the impressive visual quality and image-text matching degree of our method.

4.3. Panoramic Texture vs. Cubemap Texture

We suggest that, to pursue global consistency and spatial coherent for the scene-level mesh texturing task with the LDM diffusion process, the texture should be first synthesized in a panoramic space with equirectangular projection, rather than using multi-view fashion (*e.g.*, as shown in Sec. 4.2) or cubemap spaces (*e.g.*, RoomDreamer [45]). To prove this, we also compare our panoramic texturing pipeline with a cubemap-based pipeline, where the cubemap is directly generated by depth-aware LDM following Song *et al.*'s work [45]. As demonstrated in Fig. 7, due to the discontinuity and unclear spatial semantic meaning, cubemap textures tend to produce excessive details on top faces, and also fail to make a smooth content transition on disconnected edges (see Fig. 7 (b)), which results in spurious textures on the rooftop and mixed textures on the chair (see Fig. 7 (c)). By contrast, generating textures in panoramic space like ours not only achieves better spatial structural meaning (*i.e.*, let the fine-tuned LDM know that the upper image area is the ceiling and the bottom area is the floor), but also ensures spatial continuity and coherence (*e.g.*, semantic meaningful galaxy ceiling and white chairs with clean textures in Fig. 7).

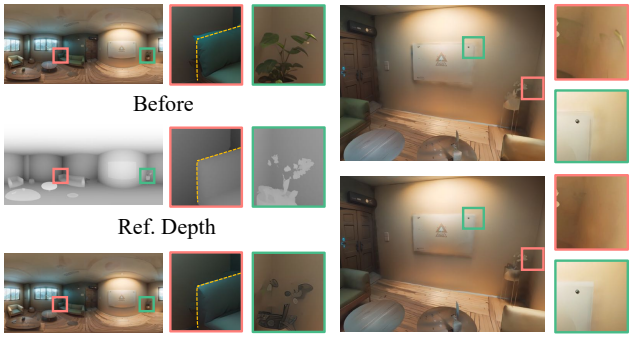


Figure 9. We inspect the efficacy of dual texture alignment on the initial panoramic space and rendered mesh.

4.4. Ablation Studies

Coarse-to-fine generation. We first analyze the coarse-to-fine strategy in panoramic texture generation (Sec. 3.1). Specifically, we ablate the coarse-to-fine upscaling and

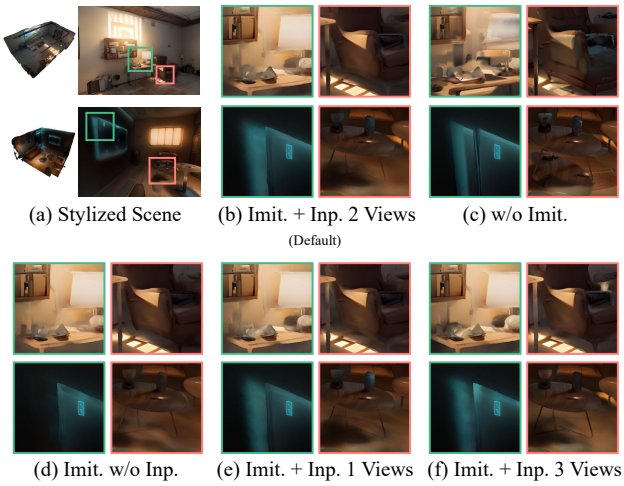


Figure 10. We analyze the effectiveness of imitating and inpainting in holistic texture propagation.

equirectangular seam fixing for the initial panoramic texture generation. As shown in Fig. 8 (b) and (c), by enabling the coarse-to-fine upscaling technique, we can obtain textures with richer details (*e.g.*, much cleaner galaxy-style poster, seeing clearer winding landscape path from the window), which is essential for satisfactory immersive VR experience as it amplifies the details of the scene. By employing equirectangular seam fixing (see Fig. 8 (d) and (e)), we can significantly remove tiling seams on the projected mesh textures (*e.g.*, seams on the window and the roof are gently removed), which ensures the spatial consistency for the synthesized panoramic texture.

Dual texture alignment. We then study the necessity of the dual texture alignment strategy (Sec. 3.2). To clearly demonstrate the efficacy, we both visualize the panoramic space alignment and the resulting meshes in Fig. 9. It is clear that LDM tends to produce textures where the boundary of the object cannot be aligned to the real-world geometry (*e.g.*, the highlighted contour of the green sofa, and the leaves of a potted plant in Fig. 9 (a)), while dual texture alignment would mitigate such misalignment at the panoramic space. After projecting textures to meshes following Sec. 3.3 with carefully visibility test, we still observe the artifacts by misalignment (*e.g.*, dirty textured walls caused by erroneously projecting leaves' textures on the wall in the first row of Fig. 9). By introducing dual texture alignment for panoramic textures, we further alleviate the misaligned artifacts caused by texture projection (*e.g.*, clean textured walls in the second row of Fig. 9).

Texture propagation with inpainting and imitating. We also inspect the necessity of the texture inpainting and imitating techniques (Sec. 3.3) for panoramic texture projection in Fig. 10. By default, we enable texture imitating with two viewpoint inpainting (see Fig. 10 (b)). To ablate the texture imitating, we use a see-through texture projecting similar to Dream-Texture [25] to avoid texturing vacancy, where all the valid UV pixels would be assigned to a color through equirectangular projection. As shown in Fig. 10 (c), the texture projection without imitating would inevitably introduce erroneous texturing results, *e.g.*, much chaotic appearance of the desk and duplicated round table on the floor in the first row of Fig. 10 (c). When ablating texture inpainting techniques, the framework loses knowledge of what the occluded area should look like and only guesses the occluded appearance with texture imitating. As shown in Fig. 10 (d), our method still achieves plausible texturing results without noticeable artifacts, but might lose some semantic meaningful content such as the blue glow at the back of the monitor (the last row of Fig. 10 (d)). By enabling the inpainting and imitating together, we can achieve texturing results with both clean textures at cluttered geometry (*e.g.*, the first row of Fig. 10 (e)) and novel content at inpainted areas (*e.g.*, the fancy blue glow of the monitor at the last row of Fig. 10 (d)).

Number of inpainting viewpoints. We finally analysis on the number of inpainting viewpoints in Fig. 10. Different from previous works that use repetitive inpainting on perspective views to cover all the visible surfaces of the mesh, our method follows the principle that generates an informative panoramic texture and then propagates it through inpainting and imitating techniques. Therefore, we don't rely on too many inpainting views, since inpainting itself cannot always produce reasonable images especially when observing occluded areas from small grazing angles (*e.g.*, small gaps between the sofa and the floor). As shown in Fig. 10, we don't observe significant improvement when increasing the number of inpainting views, as the first panoramic texture already endues sufficient appearance and overall impression of the indoor scenes.

4.5. Immersive VR Application

Once the stylized texture has been generated for the given scene mesh, we can directly place it into game engines such as Unity and upload it to the HMD devices for virtual touring. To further improve the immersive experience, as shown in Fig. 11, we also make transparent windows on the user-defined region by assigning transparent alpha values on the baked UV images, where the UV space alpha mask is generated in a way similar to inpainting masks (Sec. 3.3). Then, we pack the scene with an additional generated panoramic skybox by an unconstrained version of the panoramic diffusion model (*i.e.*, the LDM in Sec. 3.1 that trained on broaden equirectangular projection images). During the rendering, we use the generated panoramic skybox as the background and open the virtual window with transparent UV textures. In this way, we can build up a fan-

(a) Input: stylized DreamSpot bedroom with transparent window

(b) Input: panoramic Galaxy by unconstrainted pano. diffusion

(c) Virtual touring in the dreamed bedroom while seeing the galaxy

Figure 11. We build up a VR application by uploading textured scene assets with transparent windows and generated skyboxes into the HMD devices, which delivers an enchanting and immersive VR experience by allowing 6-DoF free-viewpoint touring with teleportation (red dot on the ground) in the fully stylized spaces.

tasy VR application, which allows users to enjoy the stylized space with their familiar scene structure but totally different appearance, *i.e.*, seeing the nebula from the virtual window on a galaxy-theme bedroom. Please refer to the supplementary video for the video recording of the immersive VR application.

5. Conclusion

We have proposed a novel text-driven indoor scene texturing framework, which enables to generate high-resolution and semantic meaningful UV textures for real-world scenes based on text prompts. The key insight of our work is to first synthesize a stylized panoramic view of the scene that already conveys a global consistent appearance, and then propagate it to the rest regions. For texture propagation, we design novel confidential inpainting and implicit imitating techniques, which properly handle cluttered real-world geometry and maintain spatial coherence for occluded areas or thin structures. The resulting stylized textured mesh can be feasibly uploaded into HMD devices, which delivers immersive VR experiences.

Limitations and future works. Despite the novel scene-texturing capability provided by our method, it still has some limitations. First, the panoramic texture synthesized by our method already bakes the scene lighting effects, which cannot support custom lighting or dynamic shadows in the rendering pipeline. Second, to ensure high-quality texturing and a completely immersive VR experience, our method requires the input reconstruction to include real-world textures, and also relies on the quality of the scene reconstruction (*e.g.*, incomplete scanned scenes without a roof such as ScanNet [9] is not preferred). Third, our method does not support extra large rooms (*e.g.*, theater, church) or outdoor spaces, as such scenarios might need multiple partitioned stylized panoramas to fill the entire scene. In the future, we plan to support PBR texturing by fine-tuning LDM with PBR-based equirectangular projections, which would be more compatible with modern physically based rendering pipelines. Besides, we can also incorporate our scene texturing pipeline with a visual positioning system, so as to align the stylized scene with the physical real world on HMD devices, which could deliver appealing MR experiences.

Acknowledgements. We thank Freepik for icons in the figures.

References

[1] Naofumi Akimoto, Yuhi Matsuo, and Yoshimitsu Aoki. Diverse plausible 360-degree image outpainting for efficient 3dcg background creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11441–11450, 2022. 3

[2] Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20919–20929, 2023. 2, 3

[3] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023. 5

[4] Xu Cao, Weimin Wang, Katashi Nagao, and Ryosuke Nakamura. Psnet: A style transfer network for point cloud stylization on geometry and color. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer vision*, pages 3337–3345, 2020. 3

[5] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023. 2, 3, 4, 6

[6] Yaosen Chen, Qi Yuan, Zhiqiang Li, Yuegen Liu, Wei Wang, Chaoping Xie, Xuming Wen, and Qien Yu. UPST-NeRF: Universal photorealistic style transfer of neural radiance fields for 3d scene. In *arXiv preprint arXiv:2208.07059*, 2022. 3

[7] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Stylizing 3d scene via implicit representation and hypernetwork. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1475–1484, 2022. 3

[8] Tai-Yin Chiu and Danna Gurari. Iterative feature transformation for fast and versatile universal style transfer. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 169–184. Springer, 2020. 3

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 12

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3

[11] Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Unified implicit neural stylization. *arXiv preprint arXiv:2204.01943*, 2022. 3

[12] J Fišer, O Jamriška, et al. Styleblit: Fast example-based stylization with local guidance. *ACM Transactions on Graphics*, 37(4), 2018. 3

[13] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022. 3

[14] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 3

[15] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3985–3993, 2017. 3

[16] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 2

[17] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 3

[18] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023. 2, 3

[19] Filip Hauptfleisch, Ondrej Texler, Aneta Texler, Jaroslav Krivánek, and Daniel Sỳkora. Styleprop: Real-time example-based stylization of 3d models. In *Computer Graphics Forum*, volume 39, pages 575–586. Wiley Online Library, 2020. 3

[20] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 3

[21] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. *arXiv preprint arXiv:2303.11989*, 2023. 3, 4

[22] Lukas Höllein, Justin Johnson, and Matthias Nießner. Stylemesh: Style transfer for indoor 3d scene reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6198–6208, 2022. 2, 3, 8, 9

[23] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18342–18352, 2022. 3

[24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 3, 9

[25] Carson Katri. Dream-texture. https://github.com/carson-katri/dream-textures, 2023. Accessed: 2023-10-03. 2, 3, 5, 11

[26] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10051–10060, 2019. 3

[27] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11020–11028, 2022. 3

[28] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *Proceedings of the European conference on computer vision (ECCV)*, pages 768–783, 2018. 3

[29] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023. 3

[30] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 3

[31] 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, 3, 4, 7

[32] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 conference papers*, pages 1–8, 2022. 3

[33] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 2, 3, 4, 5

[34] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International*

Conference on Machine Learning, pages 8162–8171. PMLR, 2021. 3

[35] OpenMVS OpenMVS. open multi-view stereo reconstruction library. *GitHub Repos*, 2020. 7

[36] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 577–582. 2023. 6

[37] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 3, 9

[38] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 3

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 9

[40] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023. 2, 3, 4, 6, 8, 9

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3, 4, 7, 9

[42] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022. 3

[43] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 7

[44] Christoph Schuhmann. Clip+mlp aesthetic score predictor. https://github.com/christophschuhmann/improved-aesthetic-predictor, 2023. Accessed: 2023-10-03. 9

[45] Liangchen Song, Liangliang Cao, Hongyu Xu, Kai Kang, Feng Tang, Junsong Yuan, and Yang Zhao. Roomdreamer: Text-driven 3d indoor scene synthesis with coherent geometry and texture. *arXiv preprint arXiv:2305.11337*, 2023. 3, 4, 10

[46] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 7

[47] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5117–5127, 2021. 5

[48] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 3

[49] Fuwen Tan, Song Feng, and Vicente Ordonez. Text2scene: Generating compositional scenes from textual descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6710–6719, 2019. 3

[50] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv preprint arXiv:2307.01097*, 2023. 8, 9

[51] tjm35. Asymmetric tiling for stable-diffusion-webui. https://github.com/tjm35/asymmetric-tiling-sd-webui, 2023. Accessed: 2023-10-03. 5

[52] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016. 3

[53] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)*. 5

[54] Tianhao Wu, Chuanxia Zheng, and Tat-Jen Cham. Ipo-ldm: Depth-aided 360-degree indoor rgb panorama outpainting via latent diffusion model. *arXiv preprint arXiv:2307.03177*, 2023. 3

[55] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022. 7

[56] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, pages 717–733. Springer, 2022. 2, 3

[57] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2, 3, 4, 5, 7