InstanceTex: Instance-level Controllable Texture Synthesis for 3D Scenes via Diffusion Priors

MINGXIN YANG, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China
JIANWEI GUO*, MAIS, Institute of Automation, Chinese Academy of Sciences, China
YUZHI CHEN, School of Artificial Intelligence, University of Chinese Academy of Sciences, China
LAN CHEN, MAIS, Institute of Automation, Chinese Academy of Sciences, China
PU LI, MAIS, Institute of Automation, Chinese Academy of Sciences, China
ZHANGLIN CHENG*, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China
XIAOPENG ZHANG, MAIS, Institute of Automation, Chinese Academy of Sciences, China
HUI HUANG, Shenzhen University, China

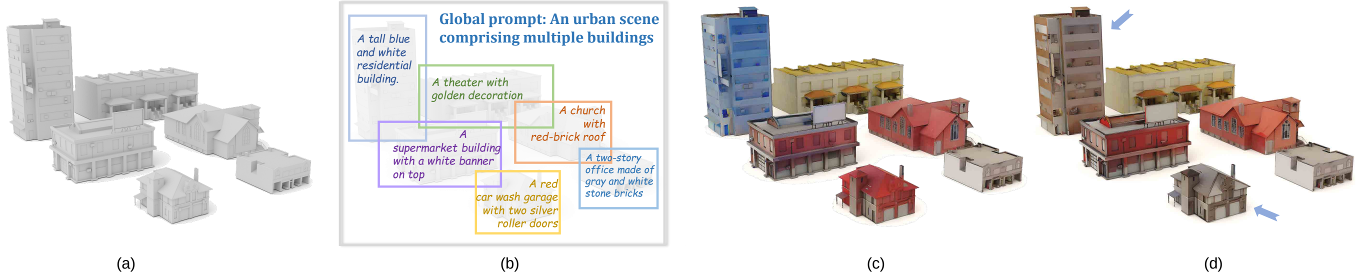

Global prompt: An urban scene comprising multiple buildings
A tall blue and white residential building.
A theater with golden decoration
A church with red-brick roof
A supermarket building with a white banner on top
A red car wash garage with two silver roller doors
A two-story office made of gray and white stone bricks
(a)
(b)
(c)
(d)

Fig. 1. We introduce *InstanceTex*, an automatic method for controllable texture synthesis for 3D scenes. Give an untextured scene composed of multiple objects (a), *InstanceTex* enables users to specify instance-level text prompts (b), facilitating the generation of high-fidelity textures while maintaining stylistic coherence (c). Compared to previous texture synthesis methods, our approach provides precise control over individual instances, *i.e.,* enabling selective modification of target objects, such as specific buildings indicated by arrows in (d).

Automatically generating high-quality textures for complex scenes remains a significant challenge in computer graphics. Recent advances in text-to-texture synthesis using 2D diffusion models have yielded impressive results for individual objects but struggle to maintain style consistency and semantic alignment when applied to larger scenes. These methods often require extensive optimization time and substantial memory resources. To address these challenges, we present *InstanceTex*, a novel approach to creating realistic and style-consistent textures for large scenes containing multiple objects. The core idea of InstanceTex lies in the instance-level controllable texture synthesis, which utilizes an instance layout representation to allow precise semantic control over individual instances while maintaining overall style consistency. We also introduce a local synchronized multi-view diffusion strategy to improve local texture consistency by sharing the latent denoised content among neighboring views in a mini-batch. Additionally, we introduce *Neural MipTexture*, inspired by Mipmaps, specifically designed for scene texture mapping to minimize aliasing effects. Extensive texturing experiments on both indoor and outdoor scenes demonstrate that InstanceTex can produce high-quality and consistent textures that outperform existing texture generation methods in terms of quality and consistency.

CCS Concepts: • **Computing methodologies** → **Image processing**.

Additional Key Words and Phrases: Texture Generation, Diffusion Models

ACM Reference Format:
Mingxin Yang, Jianwei Guo, Yuzhi Chen, Lan Chen, Pu Li, Zhanglin Cheng, Xiaopeng Zhang, and Hui Huang. 2024. InstanceTex: Instance-level Controllable Texture Synthesis for 3D Scenes via Diffusion Priors. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3680528.3687633

1 INTRODUCTION

Automatic 3D content creation is a fundamental task in computer graphics, and recent years have witnessed remarkable progress in 3D generation [Li et al. 2023; Shi et al. 2022]. One crucial goal of this task is to provide realistic 3D models that can serve as essential assets in a variety of downstream applications, such as games and films, augmented reality, digital twins, etc. For many of these applications, texture generation is essential for authoring photo-realistic 3D objects without increasing their geometric complexity [Hasselgren et al. 2021; Knodt et al. 2023; Yuksel et al. 2019].

Creating high-quality textures remains a daunting and time-consuming task, often requiring domain-specific expertise and laborious manual efforts. Recent advances in mesh-conditioned texture synthesis have made significant progress in producing realistic and

[*] Corresponding authors: Jianwei Guo (jianwei.guo@nlpr.ia.ac.cn), Zhanglin Cheng (zl.cheng@siat.ac.cn)

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1131-2/24/12…$15.00
https://doi.org/10.1145/3680528.3687633

diverse textures, largely due to the power of diffusion and large language models in image generation via text prompting [Ramesh et al. 2022; Rombach et al. 2022; Saharia et al. 2022]. Seminal project-and-inpaint methods [Chen et al. 2023b; Richardson et al. 2023] apply an iterative scheme to generate partial textures with pre-trained depth-aware diffusion models, which are then stitched together and projected back to mesh vertices or UV atlas. To reduce the seam and inconsistent artifacts caused by synthesizing partial texture independently, holistic generation methods based on multi-view diffusion (*e.g.,* SyncMVD [Liu et al. 2023b] and TexFusion [Cao et al. 2023]) allow diffusion processes from different views to generate the entire output simultaneously. However, performance drops significantly as the number of views increases, particularly for complex objects.

Moreover, existing text-to-texture methods typically focus on texturing single objects with rather small-scale and simple mesh geometry. When these methods are scaled up to generate textures for larger scenes, issues such as texture seams and accumulated artifacts are exacerbated. Specifically, occlusion among multiple objects and the absence of instance information make texture generation prone to erroneous results due to inherent ambiguity. On the contrary, texturing each object individually and then integrating them into a coherent scene poses a significant challenge in terms of introducing stylistic inconsistencies. For example, in an urban street scene, maintaining style consistency between adjacent structures and their surroundings (*e.g.,* buildings, street furniture, and sidewalks) is essential. A recent approach, SceneTex [Chen et al. 2024], has improved style and geometry consistency in indoor scenes through multi-resolution texture field optimization, but the method requires considerable time (up to 20 hours) to converge for a single scene.

In this paper, we present *InstanceTex*, a novel framework for effectively creating high-quality textures for a wide range of scene meshes, ranging from indoor to larger-scale urban scenes. At the core of our method is an instance-level controllable texture synthesis approach. We assume the input scene is instance-segmented and build an instance layout representation that specifies every instance's location (*i.e.,* 3D bounding box) and appearance style via text prompts. As naively texturing every object individually can undermine scene style consistency, we first propose instance-conditioned image generation based on the flexible inpainting framework [Chen et al. 2023b]. To address discontinuities typically encountered during texture generation, we divide the image inpainting process into an instance-level inpainting stage and a scene-level inpainting stage. The former allows precise control over each instance, while the latter ensures global style consistency. We further enhance our approach by incorporating a texture refinement strategy that innovatively integrates a local multi-view diffusion (MVD) into the inpainting pipeline. This process unifies the diffusion process among neighboring views to jointly denoise them, thereby improving local consistency. Finally, to map the generated multi-view images back to the UV atlas, we propose *Neural MipTexture*, a neural multiscale texture mapping algorithm for creating high-fidelity texture maps. Our Neural MipTexture is specially designed to address the problem of aliasing artifacts that commonly occur when texturing large scenes.

We evaluate our approach and find it effective in maintaining viewpoint consistency, where textures are well aligned with instance-level text prompts and scene geometry. In summary, our contributions include:

- *InstanceTex*, a fully automatic method for generating high-quality and style-consistent textures on large scene geometries, offering precise instance-level control.
- An instance-conditioned diffusion model that guides the inpainting process based on an instance layout representation, enabling the generation of multi-view textures with correct semantic alignment.
- A novel local multi-view diffusion approach that enhances local style consistency in texture generation.
- *Neural MipTexture*, a neural multiscale texture mapping algorithm that reduces the aliasing artifacts, resulting in high-fidelity texture maps.

2 RELATED WORK

Our work diverges from traditional texture mapping approaches for realistic 3D scene reconstruction [Bi et al. 2017; Fu et al. 2018; Gal et al. 2010; Waechter et al. 2014; Xiong et al. 2023; Zhou and Koltun 2014], which rely on a collection of captured photographs and calibrated cameras. Instead, we focus on texture synthesis for 3D meshes based on given text prompts.

Text-driven mesh texturing. Many previous methods [Bokhovkin et al. 2023; Chen et al. 2023a; Dundar et al. 2023; Gao et al. 2022, 2021; Oechsle et al. 2019; Siddiqui et al. 2022] leverage categorical information as a prior and train generative models upon a specific dataset (*e.g.,* ShapeNet [Chang et al. 2015], urban meshes [Georgiou et al. 2021; Kelly et al. 2018]) to synthesize textures or textured 3D shapes. Although these methods can achieve plausible results, they suffer from significant limitations: reliance on specific training data restricts their generalizability to objects in other categories and limits the diversity of generated textures.

More recently, diffusion models have emerged as robust zero-shot generation approaches and have spawned many text-driven texture generation methods. Text2mesh [Michel et al. 2022] learns per-vertex color with local displacements guided by text, while CLIP-Mesh [Mohammad Khalid et al. 2022] optimizes textured geometry by deforming an initial sphere. Score distillation loss harnesses the capability of the more advanced Stable Diffusion [Rombach et al. 2022], and has been used in DreamFusion [Poole et al. 2023], Magic3D [Lin et al. 2023], DreamGaussian [Tang et al. 2023], and LatentPaint [Metzer et al. 2023]. However, they often result in over-saturated colors, over-smoothed textures, and long convergence times, hindering these optimization-driven approaches from being used in practical applications.

Methods closely related to our work are optimization-free approaches. TEXTure [Richardson et al. 2023] alternates between a pre-trained depth-conditioned diffusion model and an inpainting diffusion model, partitioning the texture into three distinct regions ("keep", "refine", and "generate") and employing different strategies for each segment to maintain local consistency within partial synthesis. Text2Tex [Chen et al. 2023b] uses the same idea and proposes a dynamic viewpoint selection to search the optimal viewpoints

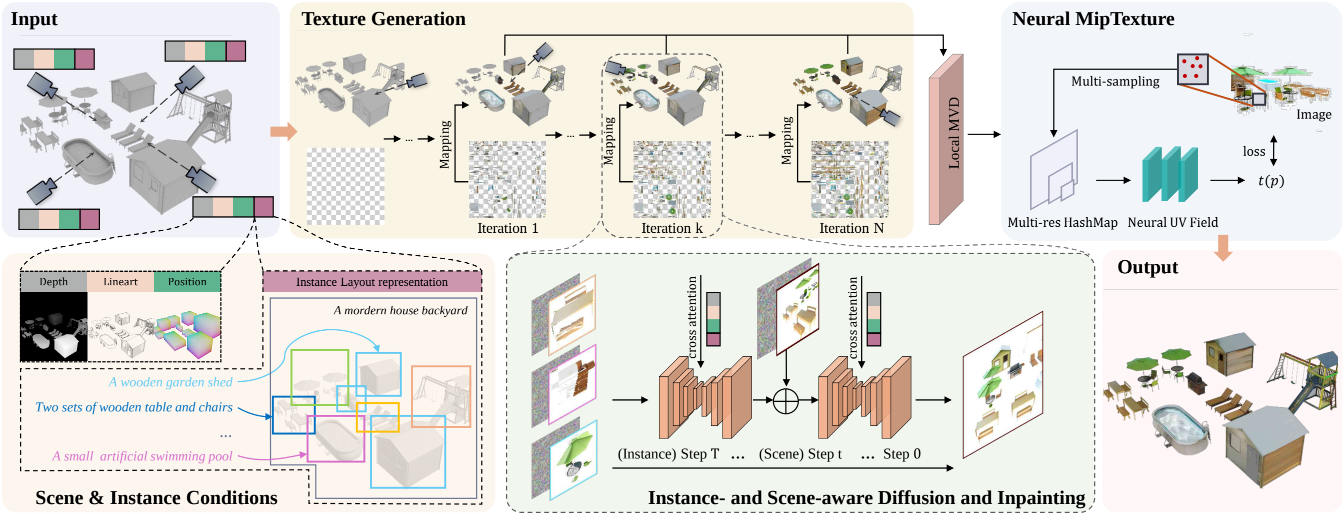


Fig. 2. **Overview of InstanceTex**. Starting from an untextured scene geometry, we build an instance layout representation, which can be fed into a diffusion model with other scene conditions (*i.e.*, depth, lineart and position maps). To generate textures, we adopt an inpainting scheme, where our core contribution lies in the instance- and scene-aware diffusion and inpainting to generate multi-view images. A local multi-view diffusion (MVD) strategy is further proposed to ensure texture consistency. Finally, we develop Neural MipTexture, a new texture mapping approach for large scene texture reconstruction.

based on dynamic partitioning iteratively. However, due to asynchronous stochastic diffusion from different viewpoints, they suffer from apparent seams, over-fragmentation, and long-range texture inconsistency. As a concurrent work, TexFusion [Cao et al. 2023] extends single-image diffusion and reformulates view-dependent texture synthesis as a holistic auto-regressive model to mitigate the inconsistent generation. Point-UV [Yu et al. 2023] leverages 3D point diffusion to generate a global coarse texture but is less effective for scenes with more complex geometry due to the scarcity of large-scale 3D datasets and the enormous amount of points required to describe the intricate geometry of such scenes.

3D scene-level texture synthesis. The aforementioned works focus primarily on texturing individual objects with small-scale geometry. For larger urban scenes, FrankenGAN [Kelly et al. 2018] introduces a cascade GAN model to adopt distinct texture generation for each part of a coarse building, while PUT [Georgiou et al. 2021] concentrates on re-targeting texture from panoramic images to novel urban area meshes employing a contrastive and adversarial model. These approaches, however, heavily rely on the generative model trained on specific datasets, resulting in limited texture diversity in their results. RoomDreamer [Song et al. 2023] employs 2D diffusion models to generate 3D scene geometry and textures based on text prompts but modifies the input mesh. SceneTex [Chen et al. 2024] generates textures for indoor scenes using a multiresolution texture field and incorporates a cross-attention decoder to ensure style consistency. However, SceneTex's texture optimization via variational score distillation is memory-intensive and requires around 20 hours to converge, making it unsuitable for practical use. In contrast, we adopt a flexible inpainting framework to texture larger scenes efficiently and integrate instance-conditioned diffusion and a local MVD approach to resolve style inconsistency.

Text-to-image generation. Diffusion models have shown impressive capabilities in image synthesis [Chang et al. 2023; Dhariwal and Nichol 2021], using text conditions to control synthesis. Approaches like Stable Diffusion [Rombach et al. 2022], Imagen [Saharia et al. 2022], GLIDE [Nichol et al. 2022] follow this paradigm and develop various architectures for text-based image generation and editing.

Recent work has incorporated layout as a conditional representation for image generation [Fu et al. 2021; Jia et al. 2024; Taghipour et al. 2024; Zheng et al. 2023], which provides semantics with specific spatial positions for more controllable inference. Furthermore, layout with instance prompts [Wang et al. 2024] has extended traditional methods by incorporating additional decorative descriptions for individual instances, increasing the precision of style and content control. While some approaches generate entire 3D scenes with layout priors [Lu et al. 2024; Zhai et al. 2024], few have integrated layout conditions into 3D-aware texture generation to improve instance-level control. Our work fills this gap by leveraging 3D bounding boxes as an intermediate layout representation and using them alongside instance prompts to texture complex scene meshes with detailed control over individual instances.

3 OVERVIEW

Our goal is to synthesize highly detailed and semantically aligned textures for large-scale scene geometries, ensuring continuity both locally and globally. Our framework takes three inputs: an untextured 3D scene composed of multiple objects $\{O_i\}_{i=1}^{n}$, a scene-level text prompt y describing the global style, and instance-level text prompts $\{y_i\}$ describing the desired appearance of each object.

The pipeline of our scene texturing is shown in Fig. 2. We assume that the input scene is unwrapped, where UV coordinates map each vertex of the mesh to a texel in a texture map. If not, we parameterize the mesh into a texture map using Blender's Smart UV mapping tool*. The key to our framework is an instance layout representation that specifies the location and text prompt for every instance, enabling precise yet flexible instance-level control. First, we define a set of viewpoints $\{v_j\}_{j=1}^{m}$, from each of which we render a depth map and a lineart map of the scene, as well as a pose-aware position map of the observed instances. These three maps and the rendered 2D instance layout are fed into a diffusion model to generate instance-conditioned images, allowing flexible appearance specification for multiple objects (Sect. 4.1). Next, we improve the inpainting scheme [Chen et al. 2023b] to generate multi-view images. A novel instance- and scene-aware inpainting module is proposed to reduce instance information leakage and enhance style coherence (Sect. 4.2). Furthermore, we develop a local multi-view diffusion refinement to enhance local texture consistency (Sec. 4.3). Finally, with the generated multi-view images, we propose Neural MipTexture, a neural multiscale texture mapping algorithm, to obtain the complete high-fidelity texture maps (Sec. 5).

4 INSTANCE-CONDITIONED MULTI-VIEW TEXTURE GENERATION

Our texture generation process follows a project-and-inpaint framework [Chen et al. 2023b], which is more flexible than holistic generation methods [Cao et al. 2023; Liu et al. 2023b] for texturing large scenes usually involving a complex setting of camera viewpoints. Different from [Chen et al. 2023b], our work is based on an instance layout representation, providing strong control over both the global style and each detailed object appearance. Specifically, given a sequence of pre-defined viewpoints, we iteratively synthesize and update the 3D texture. We first generate a view using an instance-conditioned diffusion model and project this view onto the scene geometry. Then we render the partially textured scene from the next viewpoint and fill in the missing texture regions using an instance-aware inpainting scheme.

4.1 Single-view Diffusion with Instance Layout

Instance layout and input conditions. We define a scene's instance layout $\mathcal{L}$ as a set of 3D bounding box $\mathcal{B}_i$ and a textual description y_i of every object instance. Our key insight is that this 3D representation $\mathcal{L}$, when rendered to 2D $\mathcal{L}_{2D}$, incorporates semantic information associated with locations, enabling control over powerful 2D diffusion models to generate textures that align with the scene layout. To implement this level of instance-specific control during the diffusion process, we adopt InstanceDiffusion introduced by [Wang et al. 2024] as our base diffusion model.

Beyond the semantic alignment provided by the instance layout, we also ensure that the generated textures accurately reflect the scene's geometry. To achieve this, we render a *depth map* and a *lineart map* from a given viewpoint as geometric cues for the 2D diffusion model.

Furthermore, directional prompts are crucial in texturing individual objects [Chen et al. 2023b; Yu et al. 2023], as they help eliminate semantic misalignment by providing relative pose information. However, generating suitable directional prompts for a large scene is challenging due to the varying poses of different objects within the same scene. Fortunately, benefiting from our instance layout $\mathcal{L}$, we can define a pose-aware *position map* that represents the relative pose between the current camera and the object instances. Specifically, for each object, we rescale its 3D bounding box into the $[0, 1]$ range and project it into 2D space according to the viewpoint, as defined by the following equation:

$$c(p, \mathcal{B}_i) = \frac{p - min(p_{\mathcal{B}_i})}{max(p_{\mathcal{B}_i}) - min(p_{\mathcal{B}_i})}. \tag{1}$$

Here, the value at any point p on the position map corresponds to its relative position within the instance bounding box $\mathcal{B}_i$. Therefore, for each scene view, a position map containing multiple objects is generated as an additional controllable condition.

The conditions of the depth map, lineart map, and position map are usually incorporated into the diffusion model using ControlNet [Zhang et al. 2023]. However, the public ControlNet, being pre-trained on the architecture of Stable Diffusion [Rombach et al. 2022], can introduce noticeable artifacts when directly applied to InstanceDiffusion. To facilitate geometry alignment, we fine-tune ControlNet by training three adaptors with InstanceDiffusion for the depth map, lineart map, and position map, respectively. Additional details are provided in our supplementary material.

Modified diffusion process. The above conditions are fed to InstanceDiffusion for instance-conditioned texture generation. Specifically, InstanceDiffusion, a modified version of Stable Diffusion, can accept instance layouts as input conditions. We pre-encode the instance layout with a specified UniFusion block [Wang et al. 2024] and divide the denoising process into two stages: instance-level and scene-level denoising. Given the instant layout, instance-level denoising is first performed individually for each instance. Then, the scene latent x_s^τ is produced by averaging the intermediate instance latent x_i^τ with the expression: $x_s^\tau = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^\tau$, where the parameter τ is the time step that the merging operation is inserted. Scene-level denoising is then conducted by feeding the merged latent to the subsequent denoising steps.

4.2 Instance and Scene Aware Texture Inpainting

After generating textures for a single view, the next step is to synthesize multi-view texture images with consistent styles. We leverage a depth-based inpainting scheme, as introduced in Text2tex [Chen et al. 2023b] and TEXTure [Richardson et al. 2023], to maintain consistency between multi-view images. We start from a random viewpoint and synthesize the corresponding image using the single-view Diffusion model, conditioned on the instance layout, depth map, lineart, and positional map. Then, we take an incremental inpainting process for other viewpoints. For a specific viewpoint, we render the mesh textured with an RGB UV texture representation unwrapped from all previously synthesized images. Subsequently, in each denoising step, we blend a noised latent $\hat{x^t}$ of the rendered image with the denoised latent x^t of the current viewpoint using an inpainting mask:

$$x^t = x^t \odot M_{inpaint} + \hat{x^t} \odot (1 - M_{inpaint}) \tag{2}$$

*https://docs.blender.org/manual/en/latest/modeling/meshes/editing/uv.html

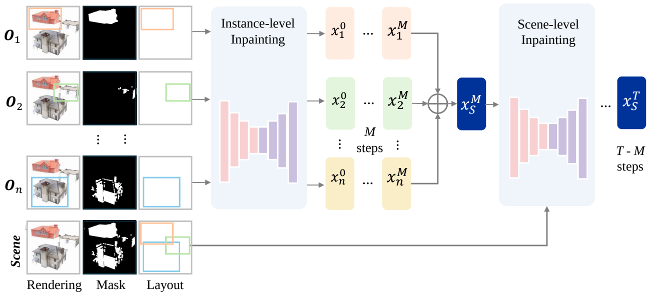


Fig. 3. Illustration of instance and scene-aware texture inpainting.

where the inpainting mask $M_{inpaint}$ is derived by rendering the existing UV mask into the current viewpoint.

However, directly applying the above blending approach throughout the entire denoising process can lead to severe artifacts due to information leakage, *i.e.,* one instance's prompt affects the appearance of another. This issue arises from the merging operation in the InstanceDiffusion model, which disrupts the standard denoising process by dividing it into instance-level and scene-level denoising. This disruption hinders the correct derivation of the noised latent introduced in the blending, as we can not infer the noised latent of the instances before the merging operation. To address this, We introduce another UV texture, referred to as latent UV texture, which captures the latent before the merging operation. This allows us to divide the multi-object inpainting process into two stages: instance-level and scene-level inpainting, as illustrated in Fig. 3. During instance-level inpainting (before merging), since the denoising process is conducted individually for each instance, we adapt the inpainting method to be instance-aware:

$$x_i^t = x_i^t \odot M_{inpaint}^i + \hat{x}_i^t \odot \left(1 - M_{inpaint}^i\right), t \in \tau \dots T \tag{3}$$

where x_i^t represents the i^{th} instance latent in timestep t, $M_{inpaint}^i$ is the corresponding individual inpainting mask, and $\hat{x}_i^t$ is derived by adding noise towards the rendered latent image. During scene-level inpainting (after merging), it generally follows the blending approach introduced in Eq. 2:

$$x_s^t = x_s^t \odot M_{inpaint}^{scene} + \hat{x}_s^t \odot \left(1 - M_{inpaint}^{scene}\right), t \in 0 \dots \tau \tag{4}$$

where x_s^t is the scene-level latent in time step t, $M_{inpaint}^{scene}$ represents the inpainting mask for the entire scene, and $\hat{x}_s^t$ is obtained by adding noise towards the rendered RGB image. This two-stage inpainting process enables InstanceTex to generate multi-view images with consistent object textures. It is worth noting that a scene often contains many instances, making occlusions and small objects texturing two common challenges. Our supplementary material provides a detailed explanation of how we address these issues.

4.3 Texture Refinement via Local Multi-View Diffusion

For large-scale scene texture generation, inpainting-based methods often struggle to maintain geometric and semantic consistency over long sequences. To improve the multi-view consistency, we introduce a local multi-view diffusion (LMVD) module to refine the images generated in previous steps.

In detail, the LMVD module can be integrated into the above inpainting process. After generating a first batch containing k images $\mathcal{I}_{b1} = \{I_1, I_2, \dots, I_k\}$ based on the inpainting framework, we feed these images $\mathcal{I}_{b1}$ to the LMVD module and apply the resampling operation of a latent diffusion model [Rombach et al. 2022] to refine local details of each image while maintaining its overall structure. Each image I_i is first encoded into a latent space, and noise is gradually added to obtain a noisy latent image x_i^t at each intermediate time step t. Note that in our case, x_i^T at $t = T$ is not pure Gaussian noise to preserve the original image's details. Next, starting from x_i^T, we perform the denoising process using latent-level multi-view blending, which enables information exchange between different views.

Inspired by synchronized MVD [Liu et al. 2023b], we synchronize the diffusion process by sharing latent information among different views. Specifically, at each denoising step, all latent images x_i^t are projected onto the shared UV texture space. Information is then shared across the overlapping regions of these views within the UV domain. The latent values from different views are blended in these overlapping regions in a weighted manner, where the weight w_i of an image I_i is computed as the cosine value between the normal vectors of the visible faces in the 3D mesh and the view direction. Finally, this texture is mapped back to the corresponding views to yield an updated x_i^t that is used for denoising.

By sharing denoised content among local views, we can achieve better local consistency. Furthermore, to ensure global consistency as much as possible, we set m overlapping views between two consecutive batches.

5 TEXTURE MAPPING VIA NEURAL MIPTEXTURE

Once consistent multi-view images $\{I_j\}_{j=1}^m$ have been generated, the next step is to unwrap a texture map $\mathcal{T}$ from them. Since our approach focuses on scene texturing, both near and distant objects are always rendered within the same image. Unlike single-object texturing, directly unwrapping the generated images onto a conventional UV map can lead to severe aliasing artifacts, as most pixels are allocated to the UV space of nearby objects, leaving only a few for distant ones. Therefore, a single UV map resolution cannot effectively capture scene details, *e.g.,* causing blurring with low-resolution UV maps and noisy patterns with high-resolution ones.

To address these aliasing artifacts caused by varying camera-object distances, we propose Neural MipTexture, a neural pre-filtering approach inspired by mipmapping and its neural extension Zip-NeRF [Barron et al. 2023]. While Zip-NeRF represents a scene as a 3D neural hash field and approximates the pixel color through multi-sampling, we extend it to UV textures using a multi-resolution 2D hash map with a tiny MLP. Practically, for a given pixel p in the image space, we initially sample L points $\{q_i\}_{i=1}^L$ uniformly within the pixel ($L = 6$ by default). These sampled points are then projected onto the UV space to obtain the corresponding UV coordinates $\{u_i\}_{i=1}^L$. We then query the corresponding features $\{f_i\}_{i=1}^L$ from the neural UV field, which is parameterized by a multi-resolution 2D hash map. These features are averaged to get a feature

$\bar{f} = AVG(\{f_i\}_{i=1}^{L})$ to represent the pixel. Then $\bar{f}$ is fed into a tiny multi-layer perceptron (MLP) $h(\bar{f}; \Phi)$ to derive the final pixel color c_p.

The neural UV field is optimized using an L2 loss between the ground truth color p and the predicted color c_p:

$$L(p) = \left\| h\left(AVG\left(\{f(q_i;p)\}_{i=1}^{L} \right); \Phi \right) - \hat{c_p} \right\|_2 \tag{5}$$

Benefiting from the multi-resolution Hash map, the whole optimization converges quickly (within a few minutes) with stable gradients. Note that while SceneTex [Chen et al. 2024] uses a similar multi-resolution 2D hash map to smooth back-propagated gradients, InstanceTex differs in its unique goal of resolving aliasing in textures and its specific implementation of multi-sampling within a pixel.

6 RESULTS AND EVALUATION

We demonstrate the effectiveness of our method through high-fidelity texturing results on different categories. We then conduct qualitative and quantitative comparisons with competing methods and validate our design choices through ablation studies.

6.1 Experimental Setup

Dataset. For performance evaluation and comparisons, we carry out experiments on 11 large scenes, including 2 indoor scenes from 3D-FRONT [Fu et al. 2021] used in SceneTex, 2 indoor scenes generated by EchoScene [Zhai et al. 2024], 3 manually created scenes featuring furniture and tea sets, and 4 outdoor scenes come from public datasets (Block-2 [Kelly et al. 2018]) or collected from Sketchfab [Ske 2022][†].

Implementation details. For each scene in our evaluation dataset containing multiple instances, we manually select 15-40 surrounding cameras to fully cover all instances. We manually generate the 3D-oriented bounding boxes for all instances within the scene mesh and derive 2D layouts by projecting these 3D bounding boxes into preset viewpoints. Our diffusion pipeline is developed based on InstanceDiffusion and ControlNet module v1-1 of the Huggingface Diffusers library, with projection functions implemented using Pytorch3D. Since edges deliver vital cues for elements like windows and cages in urban scenes, we externally integrate rendered lineart maps as additional conditions to enhance the fidelity of synthesized texture with mesh geometry details. We render these edge maps using Blender's "Freestyle" rendering feature[‡]. The default batch size for the local MVD is set to 8. For the Neural MipTexture module, we set the learning rate as 1e-3 to optimize the neural texture field, with the entire optimization using all synthesized images and converging after approximately 2000 iterations. To facilitate smoother texture, we additionally apply a Laplacian smoothing loss alongside the re-rendered RGB loss during optimization. The overall synthesis process takes around 30 minutes on average to converge on an NVIDIA® RTX 4090.

6.2 Qualitative Evaluation

We compare our approach against state-of-the-art texture generation methods, including Text2tex [Chen et al. 2023b], TEXTure [Richardson et al. 2023], SyncMVD [Liu et al. 2023b], and SceneTex [Chen et al. 2024], and Meshy [Meshy 2023], a commercial generative tool known for consistent 3D object texturing. All methods were tested using the same input prompts and viewpoints for a fair comparison.

Fig. 4 shows the qualitative comparisons for outdoor and indoor scenes, respectively. Although TEXTure generates high-quality textures for individual objects, it still suffers from over-fragmentation and hallucinates scene components when texturing both outdoor and indoor scenes. Text2Tex easily generates salt-and-pepper noise and obvious seams, and also struggles to keep global style consistency across objects. SyncMVD and Meshy offer consistent style schemes but tend to generate over-saturated colors and misinterpret prompts. While SceneTex, designed specially for texting indoor scenes, can synthesize high-quality textures with overall coherent styles, it fails to accurately match the input prompts due to its inability to precisely control the appearance of target objects, despite considering instance texture features. In contrast, guided by instance layout representation, our InstanceTex demonstrates higher quality, seamless, and instance-aware texturing results with accurate semantic alignment to the input prompts.

Comparison to individual texturing baselines. We also compare InstanceTex against baselines on individual object texturing, where the input scene is first separated into distinct instances, and textures are generated for each object according to specified instance text prompts. For a fair comparison, we combine scene-level prompts with instance-level prompts to condition the texture generation of each object. We compare InstanceTex with two representative single-object texturing approaches: Text2tex [Chen et al. 2023b] and SyncMVD [Liu et al. 2023b]. As shown in Fig. 5, the individual texturing results of Text2Tex and SyncMVD generally lack global consistency and result in disharmonious textures, particularly in indoor scenes with uniformly styled furniture. In contrast, the renderings of InstanceTex align with specified instance prompts and maintain global style consistency across objects, as illustrated by the consistent blue design on the teacups and the uniform texture pattern between the leather sofa and chairs.

Evaluation on complex scenes. We further conduct a stress test on several challenging scenes, including noisy meshes with irregular geometry (produced by a layout-based 3D scene generation approach EchoScene [Zhai et al. 2024]), and complex scenes with repetitive objects and complicated text prompts. As shown in Fig. 6, the texturing results on EchoScene validate InstanceTex's robustness. Besides, InstanceTex still achieves consistent texturing on more intricate scenes with complex patterns that cause high occlusions. More results are available in the supplementary material.

6.3 Quantitative Evaluation

For quantitative evaluation, we utilize three common metrics for texture generation assessment. We first utilize CLIP-Score [Hessel et al. 2021] to assess how well the generated textures align with the provided text prompts. Then we calculate the Fréchet Inception

[†] We carefully select meshes where artists had not restricted use in generative AI models at the time of download.

[‡] https://docs.blender.org/manual/en/latest/render/freestyle/introduction.html

Table 1. Quantitative comparison on the performance of texturing results on two indoor and four outdoor scenes. The first- and second-place performances are highlighted using bold and italic fonts, respectively.

Methods	**Indoor Scenes**			**Outdoor Scenes**		
	FID↓	KID↓	CLIP ↑	FID↓	KID↓	CLIP ↑
TEXTure	103.21	8.19	19.43	122.75	9.90	18.60
Text2tex	96.23	9.73	22.63	111.21	9.54	19.75
SyncMVD	*92.44*	9.54	22.48	109.52	9.73	20.66
Meshy	96.03	7.61	21.98	*89.3*	*7.56*	*21.28*
SceneTex	91.56	*6.56*	*23.69*	/	/	/
InstanceTex	**83.62**	**6.19**	**27.35**	**82.91**	**5.96**	**27.90**

Distance (FID) [Heusel et al. 2017] and Kernel Inception Distance (KID) [Bińkowski et al. 2018], which measure the difference between the output distribution of the generated images of Instance Diffusion with ControlNet and our textured objects under specified viewpoints. (we used 40 views for evaluation in our experiments). The generated images of Instance Diffusion serve as the ground truth for all comparison methods because Stable Diffusion struggles to synthesize well-aligned images with object-level prompts.

Table 1 reports the quantitative evaluation results, demonstrating that our method significantly outperforms prior methods across all metrics. Notably, InstanceTex achieves nearly 13% and 31% improvements in CLIP-Score on indoor and outdoor scenes respectively, indicating our model's superiority in semantic-aligned texture generation. Moreover, the improvements in FID and KID demonstrate the superior capability of InstanceTex in generating realistic and high-fidelity textures across diverse scenes with different categories and styles. Detailed quantitative comparisons for each scene in our dataset are available in the supplementary material.

6.4 Ablation Study

Instance-level conditions with instance layout. Leveraging instance-level conditions enhances the generation of accurate texture. As illustrated in the first column in Fig. 7, without instance layout as a condition, the synthesized texture lacks fidelity toward the text prompts (*e.g.,* the umbrella is not green as specified in the given text prompt in Fig. 4). The CLIP-Score in Table 2 further verifies that the instance layout is vital for correct semantic alignment.

Instance-aware inpainting scheme. Given that our denoising procedure is divided into two stages, namely instance-level and scene-level generation, directly conducting inpainting for the scene content inherited from previous frames induces scene-level context into instance-level denoising. Such an implementation would disrupt the independent instance generation inherent in instance-level denoising, potentially resulting in instance-level information leakage. As shown in the second column in Fig. 7, the green pattern from the prompt "a green sunshade" is leaked into the "wooden garden shed". Moreover, only leveraging scene-level inpainting increases the difficulty for the diffusion model to produce harmonized results, thereby disturbing the texture consistency (see the second row in Fig. 7). Our introduced instance- and scene-aware inpainting scheme largely resolves the information leakage and texture incoherence, as demonstrated in all the metrics in Table 2.

Table 2. Ablation study of InstanceTex using Garden scene.

Methods	FID↓	KID↓	CLIP ↑
w/o Instance layout	94.50	7.91	18.67
w/o Instance-level inpainting	117.21	9.65	20.85
w/o Position map	98.19	8.23	23.32
w/o Local MVD	89.35	6.58	26.41
w/o Neural MipTexture	95.34	8.35	27.45
Full model	**88.18**	**6.47**	**28.15**

Position map condition. Previous studies [Chen et al. 2023b; Liu et al. 2023a,b] have shown the significance of directional prompts in maintaining coherence in 3D content generation. However, it is nontrivial to represent the large scene containing multiple instances by text prompts. This motivates us to propose the pose-aware position map as directional guidance. As demonstrated in the third column in Fig. 7, without the position map, the generation result lacks consistency and produces sharp edges during inpainting.

Local MVD refinement. The multi-view diffusion model has showcased superior performance in 3D content synthesis. Therefore, we reformulate this strategy as an image-to-image refiner in our pipeline. The fourth column in Fig. 7 shows that our local MVD refinement resolves the visual inconsistency by the multi-view denoising process. The numerical values in Table 2 also demonstrate the improved coherence of the generated textures.

Neural MipTexture. To resolve the aliasing effects caused by varying camera-object distances, we leverage a neural Mip UV texture representation to unwrap the texture from synthesized images. As shown in the fifth column in Fig. 7, the absence of Neural MipTexture results in the textures exhibiting local noisy artifacts, thus significantly decreasing the fidelity of the rendered image.

7 CONCLUSION AND FUTURE WORK

We presented *InstanceTex*, a novel method for text-driven texture generation for 3D scenes. Our major contribution lies in incorporating instance layout in the diffusion model, enabling precise control over individual objects while maintaining high-quality results. Specifically, we proposed an instance-aware inpainting scheme and a local multi-view diffusion strategy to ensure texture consistency. We also developed a new texture mapping approach tailored for large scene texture reconstruction. We evaluated *InstanceTex* on several large 3D scenes and demonstrated its advantages over state-of-the-art methods. We believe our work represents a significant advance in enabling non-experts to create large-scale 3D assets.

Limitations and future work. Due to the image resolution limitation (*i.e.,* 512 × 512) of InstanceDiffusion, the generated textures still lack sufficient high-definition details. Second, pre-sampling a well-distributed sequence of viewpoints with complete coverage over the scene surface is crucial for generating satisfactory results, particularly in large-scale scenes. Inspired by the *Next-Best-View* planning [Maldonado et al. 2016; Smith et al. 2018] used in geometry reconstruction, we plan to develop an automatic view planning algorithm to improve the texturing efficiency and quality, while minimizing the number of views needed. Furthermore, we observed that shading effects are sometimes inevitably integrated into the generated images, leading to unwanted shadows and highlights. This issue could potentially be addressed by incorporating a material generation model to replace the base model. Finally, our current focus is on object-level conditioning. In the future, we aim to incorporate more detailed structured representations commonly found in scene models and implement texture reuse (such as facade/door/window) to enhance high-definition control over the structured elements.

ACKNOWLEDGMENTS

We thank the reviewers for their constructive comments. This work was supported in part by the National Key R&D Program of China (2022ZD0160801), Guangdong Basic and Applied Basic Research Foundation (2023B1515120026, 2023B0303000016), NSFC (62172416, U22B2034, U21A20515), Shenzhen Science and Technology Program (GJHZ20210705141402008), and Scientific Development Funds from Shenzhen University.

REFERENCES

2022. *SketchFab. The best 3d viewer on the web.* https://sketchfab.com/

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 19697–19705.

Sai Bi, Nima Khademi Kalantari, and Ravi Ramamoorthi. 2017. Patch-based optimization for image-based texture mapping. *ACM Trans. Graph.* 36, 4 (2017), 106:1–106:11.

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. In *International Conference on Learning Representations*.

Alexey Bokhovkin, Shubham Tulsiani, and Angela Dai. 2023. Mesh2Tex: Generating Mesh Textures from Image Queries. In *IEEE International Conference on Computer Vision (ICCV)*.

Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. 2023. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*. 4169–4181.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).

Ziyi Chang, George A Koulieris, and Hubert PH Shum. 2023. On the Design Fundamentals of Diffusion Models: A Survey. *arXiv preprint arXiv:2306.04542* (2023).

Dave Zhenyu Chen, Haoxuan Li, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2024. SceneTex: High-Quality Texture Synthesis for Indoor Scenes via Diffusion Priors. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2023b. Text2tex: Text-driven texture synthesis via diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*.

Qimin Chen, Zhiqin Chen, Hang Zhou, and Hao Zhang. 2023a. ShaDDR: Interactive Example-Based Geometry and Texture Generation via 3D Shape Detailization and Differentiable Rendering. In *SIGGRAPH Asia 2023 Conference Papers*. Article 58, 11 pages.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794.

Aysegul Dundar, Jun Gao, Andrew Tao, and Bryan Catanzaro. 2023. Fine detailed texture learning for 3d meshes with generative models. *IEEE Trans. Pattern Anal. Mach. Intell.* (2023).

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 2021. 3d-front: 3d furnished rooms with layouts and semantics. In *IEEE International Conference on Computer Vision (ICCV)*. 10933–10942.

Yanping Fu, Qingan Yan, Long Yang, Jie Liao, and Chunxia Xiao. 2018. Texture mapping for 3d reconstruction with RGB-D sensor. In *IEEE Computer Vision and Pattern Recognition (CVPR)*. 4645–4653.

Ran Gal, Yonatan Wexler, Eyal Ofek, Hugues Hoppe, and Daniel Cohen-Or. 2010. Seamless montage for texturing models. *Comp. Graph. Forum* 29, 2 (2010), 479–486.

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

Lin Gao, Tong Wu, Yu-Jie Yuan, Ming-Xian Lin, Yu-Kun Lai, and Hao Zhang. 2021. Tm-net: Deep generative networks for textured meshes. *ACM Trans. Graph.* 40, 6 (2021), 1–15.

Yiangos Georgiou, Melinos Averkiou, Tom Kelly, and Evangelos Kalogerakis. 2021. Projective Urban Texturing. In *International Conference on 3D Vision (3DV)*. IEEE, 1034–1043.

Jon Hasselgren, Jacob Munkberg, Jaakko Lehtinen, Miika Aittala, and Samuli Laine. 2021. Appearance-Driven Automatic 3D Model Simplification. In *Eurographics Symposium on Rendering*. 85–97.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In *EMNLP*.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems* 30 (2017).

Chengyou Jia, Minnan Luo, Zhuohang Dang, Guang Dai, Xiaojun Chang, Mengmeng Wang, and Jingdong Wang. 2024. Ssmg: Spatial-semantic map guided diffusion model for free-form layout-to-image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 2480–2488.

Tom Kelly, Paul Guerrero, Anthony Steed, Peter Wonka, and Niloy J Mitra. 2018. FrankenGAN: guided detail synthesis for building mass-models using style-synchonized GANs. *ACM Trans. Graph.* 37, 6 (2018), 14 pages.

Julian Knodt, Zherong Pan, Kui Wu, and Xifeng Gao. 2023. Joint UV Optimization and Texture Baking. *ACM Trans. Graph.* 43, 1, Article 2 (2023), 20 pages.

Chenghao Li, Chaoning Zhang, Atish Waghwase, Lik-Hang Lee, Francois Rameau, Yang Yang, Sung-Ho Bae, and Choong Seon Hong. 2023. Generative AI meets 3D: A Survey on Text-to-3D in AIGC Era. *arXiv preprint arXiv:2305.06131* (2023).

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3D: High-Resolution Text-to-3D Content Creation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. 2023a. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9298–9309.

Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. 2023b. Text-Guided Texturing by Synchronized Multi-View Diffusion. *arXiv preprint arXiv:2311.12891* (2023).

Fan Lu, Kwan-Yee Lin, Yan Xu, Hongsheng Li, Guang Chen, and Changjun Jiang. 2024. Urban Architect: Steerable 3D Urban Scene Generation with Layout Prior. *arXiv preprint arXiv:2404.06780* (2024).

Oscar Alejandro Mendez Maldonado, Simon Hadfield, Nicolas Pugeault, and R. Bowden. 2016. Next-Best Stereo: Extending Next-Best View Optimisation For Collaborative Sensors. *Proc. BMVC* (2016), 1–12.

Meshy. 2023. Meshy – 3d ai generator. https://www.meshy.ai/

Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-nerf for shape-guided generation of 3d shapes and textures. In *IEEE Computer Vision and Pattern Recognition (CVPR)*. 12663–12673.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2022. Text2mesh: Text-driven neural stylization for meshes. In *IEEE Computer Vision and Pattern Recognition (CVPR)*. 13492–13502.

Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. 2022. CLIP-Mesh: Generating Textured Meshes from Text Using Pretrained Image-Text Models. In *SIGGRAPH Asia 2022 Conference Papers*. Article 25, 8 pages.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 16784–16804.

Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. 2019. Texture fields: Learning texture representations in function space. In *IEEE International Conference on Computer Vision (ICCV)*. 4531–4540.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh International Conference on Learning Representations*.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. TEXTure: Text-Guided Texturing of 3D Shapes. In *ACM SIGGRAPH 2023 Conference Proceedings*. Article 54, 11 pages.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* 35 (2022), 36479–36494.

Zifan Shi, Sida Peng, Yinghao Xu, Yiyi Liao, and Yujun Shen. 2022. Deep generative models on 3d representations: A survey. *arXiv preprint arXiv:2210.15663* (2022).

Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. 2022. Texturify: Generating textures on 3d shape surfaces. In *European Conference on Computer Vision (ECCV)*. Springer, 72–88.

Neil Smith, Nils Moehrle, Michael Goesele, and Wolfgang Heidrich. 2018. Aerial Path Planning for Urban Scene Reconstruction: A Continuous Optimization Method and Benchmark. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37, 6 (2018), 183:1–183:15.

Liangchen Song, Liangliang Cao, Hongyu Xu, Kai Kang, Feng Tang, Junsong Yuan, and Zhao Yang. 2023. RoomDreamer: Text-Driven 3D Indoor Scene Synthesis with Coherent Geometry and Texture. In *ACM International Conference on Multimedia*. 6898–6906.

Ashkan Taghipour, Morteza Ghahremani, Mohammed Bennamoun, Aref Miri Rekavandi, Hamid Laga, and Farid Boussaid. 2024. Box It to Bind It: Unified Layout Control and Attribute Binding in T2I Diffusion Models. *arXiv preprint arXiv:2402.17910* (2024).

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. 2023. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653* (2023).

Michael Waechter, Nils Moehrle, and Michael Goesele. 2014. Let there be color! Large-scale texturing of 3D reconstructions. In *European Conference on Computer Vision (ECCV)*. 836–850.

Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. 2024. InstanceDiffusion: Instance-level Control for Image Generation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Weidan Xiong, Hongqian Zhang, Botao Peng, Ziyu Hu, Yongli Wu, Jianwei Guo, and Hui Huang. 2023. TwinTex: Geometry-Aware Texture Generation for Abstracted 3D Architectural Models. *ACM Trans. Graph.* 42, 6 (2023), 1–14.

Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengzhe Liu, and Xiaojuan Qi. 2023. Texture Generation on 3D Meshes with Point-UV Diffusion. In *IEEE International Conference on Computer Vision (ICCV)*. 4206–4216.

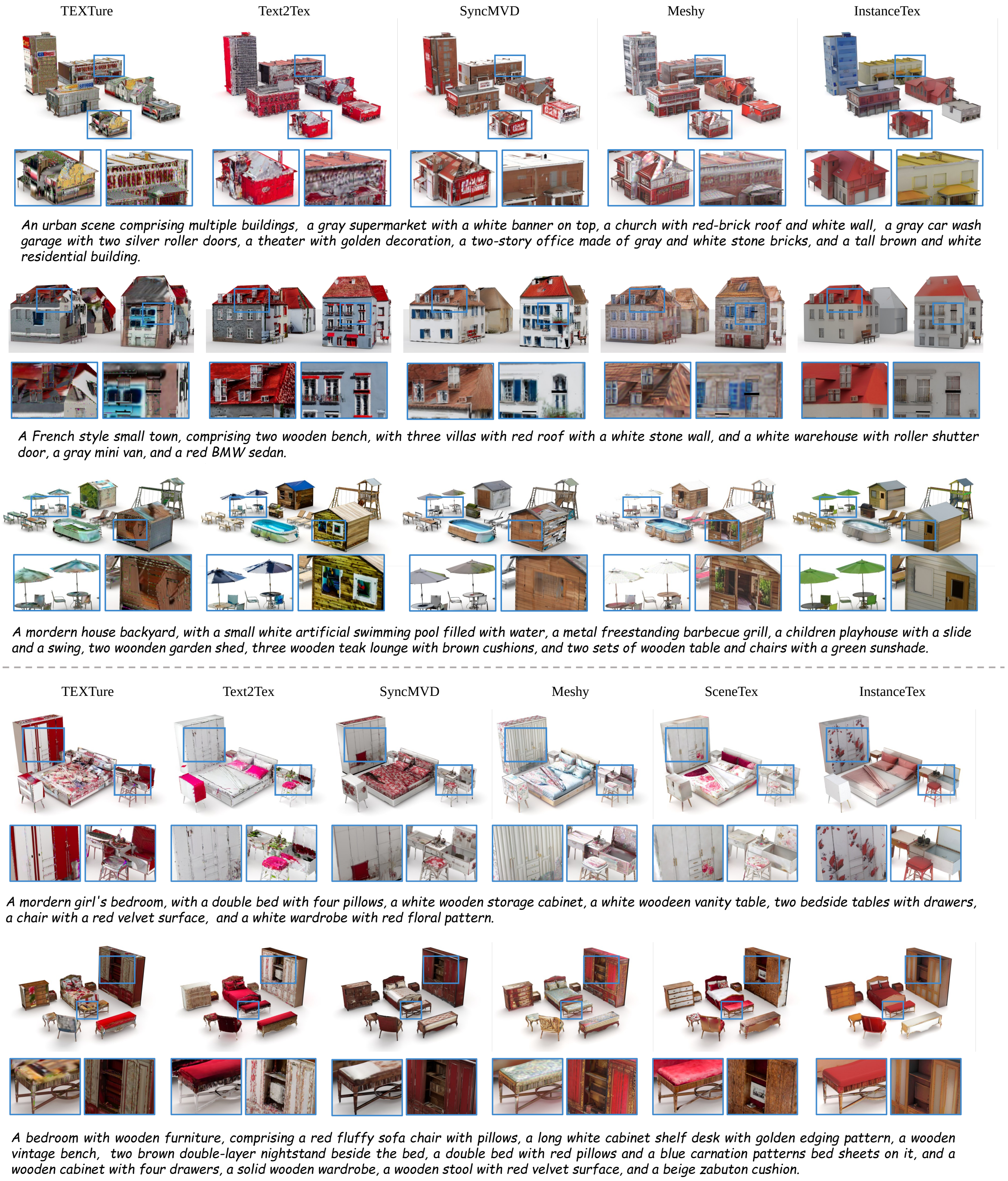
Cem Yuksel, Sylvain Lefebvre, and Marco Tarini. 2019. Rethinking texture mapping. In *Comp. Graph. Forum*, Vol. 38. Wiley Online Library, 535–551.

Guangyao Zhai, Evin Pınar Örnek, Dave Zhenyu Chen, Ruotong Liao, Yan Di, Nassir Navab, Federico Tombari, and Benjamin Busam. 2024. EchoScene: Indoor Scene Generation via Information Echo over Scene Graph Diffusion. *ECCV* (2024).

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models.

Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. 2023. LayoutDiffusion: Controllable Diffusion Model for Layout-to-Image Generation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*. 22490–22499.

Qian-Yi Zhou and Vladlen Koltun. 2014. Color map optimization for 3D reconstruction with consumer depth cameras. *ACM Trans. Graph.* 33, 4 (2014), 155:1–155:10.



An urban scene comprising multiple buildings, a gray supermarket with a white banner on top, a church with red-brick roof and white wall, a gray car wash garage with two silver roller doors, a theater with golden decoration, a two-story office made of gray and white stone bricks, and a tall brown and white residential building.

A French style small town, comprising two wooden bench, with three villas with red roof with a white stone wall, and a white warehouse with roller shutter door, a gray mini van, and a red BMW sedan.

A mordern house backyard, with a small white artificial swimming pool filled with water, a metal freestanding barbecue grill, a children playhouse with a slide and a swing, two woonden garden shed, three wooden teak lounge with brown cushions, and two sets of wooden table and chairs with a green sunshade.

A mordern girl's bedroom, with a double bed with four pillows, a white wooden storage cabinet, a white woodeen vanity table, two bedside tables with drawers, a chair with a red velvet surface, and a white wardrobe with red floral pattern.

A bedroom with wooden furniture, comprising a red fluffy sofa chair with pillows, a long white cabinet shelf desk with golden edging pattern, a wooden vintage bench, two brown double-layer nightstand beside the bed, a double bed with red pillows and a blue carnation patterns bed sheets on it, and a wooden cabinet with four drawers, a solid wooden wardrobe, a wooden stool with red velvet surface, and a beige zabuton cushion.

Fig. 4. Visual comparison of text-guided texture generation on three outdoor scenes (Block-1, Block-2, Garden) and two indoor scenes (Room-3 and Room-6). The detailed comparisons are shown in the zoomed-in insets.

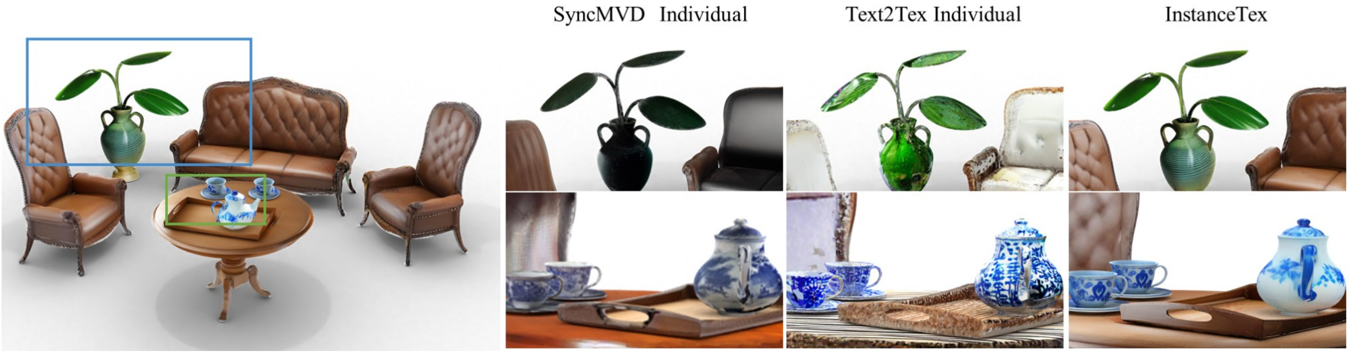


Fig. 5. Visual comparison against two individual texturing baselines, SyncMVD and Text2Tex, on an indoor scene (Furniture-1). Due to the page limit, we only show the overall view of our textured model, which is shown on the left to the zoomed-in insets. Please refer to the supplementary for a full comparison.

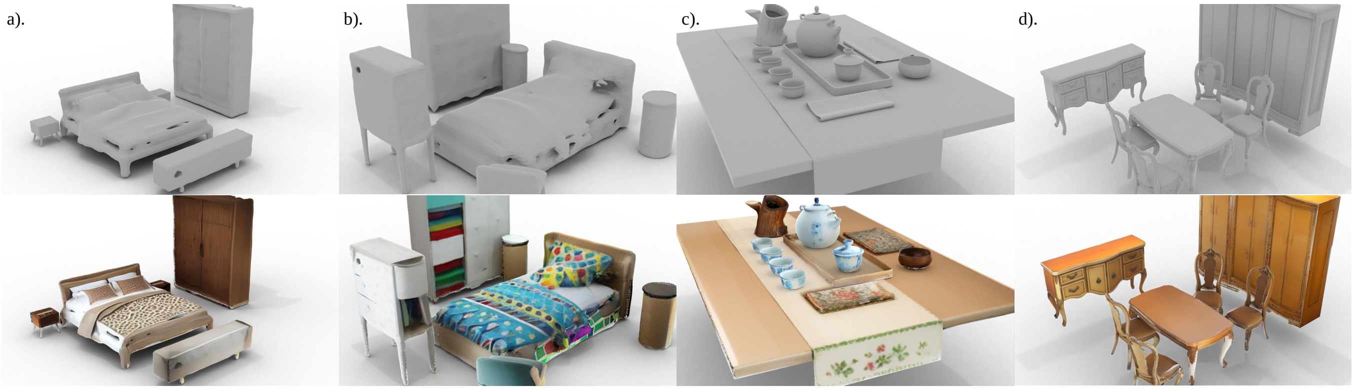


Fig. 6. Evaluation on the texture generation for challenging complex cases, where the scenes in (a) and (b) are generated by EchoScene [Zhai et al. 2024], while (c) and (d) are randomly selected scenes with either intricate geometry or complex text prompts.

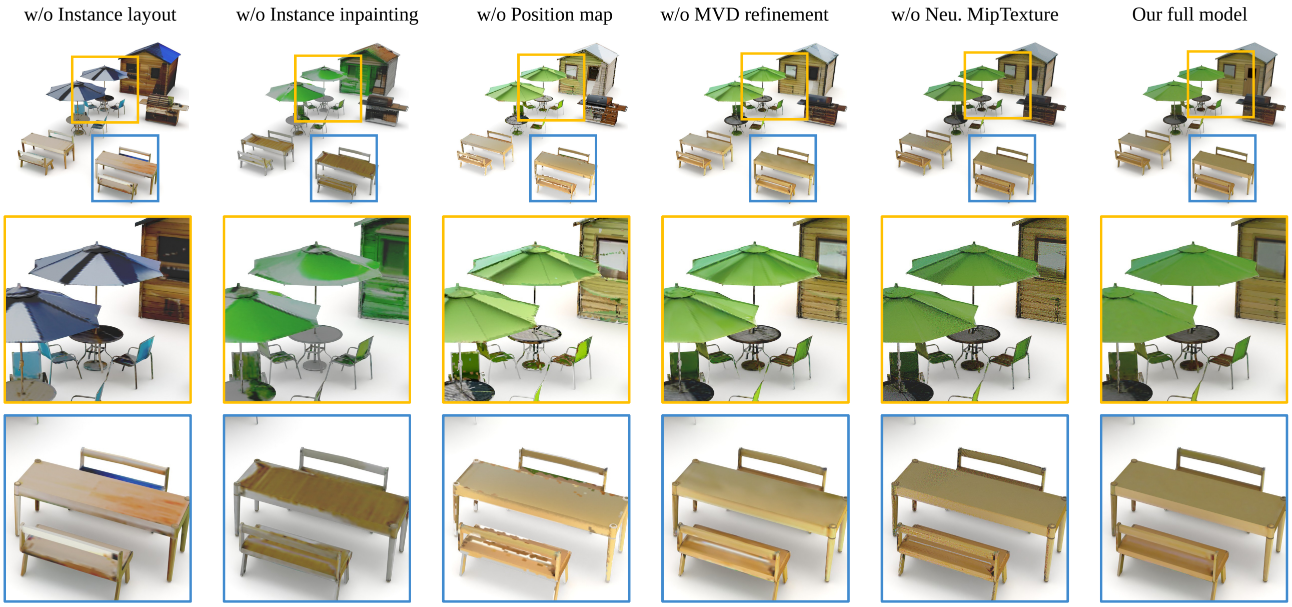


Fig. 7. Ablation Study on the key components of InstanceTex using the Garden scene.